



# Sisense V7.4

---

Technical Documentation

# Table of Contents

- New to Sisense .....7**
- Minimum Requirements and Supported Platforms..... 10
- Supported Web Browsers..... 10
- Supported Operating Systems ..... 11
- Capacity and Hardware Requirements..... 11
- Supported Locales..... 15
- Sisense Basic Concepts and Terminology ..... 16
- Sisense User Roles ..... 33
- Sisense Architecture ..... 37
- Downloading and Installing Sisense..... 44
- Setting Up Your Production Environment..... 52
- Customizing the Installation ..... 56
- Performing a Silent Installation..... 59
- Installing Sisense Offline..... 62
- Backing Up Sisense..... 67
- Researching Your Upgrade ..... 75
- Backing Up Sisense ..... 76
- Performing an Upgrade ..... 83
- Testing Your Upgrade..... 90
- Upgrading Your Production Environment..... 92
- Licensing Terminology ..... 92
- Launching Sisense the Sisense Web Application ..... 95
- Tutorial: Getting Started with Sisense..... 96
- Manage Data.....120

Data Models .....	120
Navigating the Data Page.....	123
ElastiCubes.....	125
Creating ElastiCubes .....	133
Navigating the Model Editor.....	134
Building ElastiCubes .....	276
ElastiCube Build Settings .....	279
Scheduling Data Loads .....	293
Changing an ElastiCube .....	294
Live Connect.....	295
Creating Live Models .....	297
Adding Live Connections .....	298
Working with Live Models .....	299
Publishing Live Models.....	303
Working with Live Widgets .....	303
Deleting ElastiCubes and Live Models.....	309
Sisense Server Console.....	311
Dashboards.....	313
Creating Dashboards .....	317
Widget Designer .....	390
Managing Widgets.....	411
Field Suggestions.....	415
Customizing the Dashboard Layout.....	736
Changing the Dashboard's Color Palette.....	744
Changing a Dashboard's Data Source .....	747
Exploring Dashboards.....	747
Interacting with Filters as a Viewer.....	750
Changing Sisense's Language.....	752

Drilling Down in a Widget .....	754
Making Selections in a Widget.....	755
Reporting in Sisense.....	758
Exporting Widgets .....	759
Exporting and Sharing Dashboards .....	760
Creating PDF Reports .....	760
Customizing PDF Reports.....	761
Managing Dashboards.....	764
Viewing Dashboards .....	765
Locating Dashboards .....	769
Duplicating Dashboards .....	770
Organizing Dashboards .....	771
Deleting Dashboards.....	771
Exporting and Importing Dashboards .....	773
IntroductionSisense Pulse .....	777
Sisense Alerts .....	778
Creating Data Alerts .....	781
Creating Build Alerts.....	783
Sisense Pulse .....	785
Managing Tiles in Sisense Pulse.....	787
Advanced Settings.....	793
Notifying Additional Users.....	794
Setting Notification Channels.....	796
Configuring Additional Options .....	807
Customizing Notification Messages.....	809
Administration .....	811
Managing Sisense Users .....	812
Managing Dashboard Users .....	813

Managing User Groups.....	819
User Sessions.....	825
Customizing User Roles.....	829
Integrating Active Directory.....	843
Managing Your Users Dashboards.....	871
Data Management.....	873
Adding and Removing ElastiCube Servers.....	873
Sharing ElastiCube Models.....	876
Managing Drill Hierarchies.....	878
System Settings.....	881
System Configuration.....	897
License Utilization.....	898
Configuration Manager.....	900
Sisense Migration.....	904
Migrating ElastiCubes.....	906
Migrating Dashboards in Sisense.....	914
Embedded Analytics.....	922
OEM Architecture.....	926
Security at Sisense.....	984
Security Bulletin.....	984
Sisense Security Overview.....	986
Data Security.....	993
Setting Up SSL.....	1002
Optional Security Hardening for Sisense Web Pages.....	1008
Multi-Node Deployments.....	1011
Installing the Multi-Node Deployment Wizard.....	1027
Distributing ElastiCube Builds to Query Nodes.....	1043
Setting Up ElastiCube Sets.....	1070

Routing Queries in ElastiCube Sets .....	1094
Securing the Message Broker's Communication .....	1095
Overview .....	1097
Activating Usage Analytics .....	1100
Dashboard Analytics .....	1102
Domain Analytics .....	1104
General Analytics.....	1105
Groups Analytics.....	1107
User Analytics .....	1108
Restoring Usage Analytics .....	1110
Managing Plug-ins .....	1111
Setting Up a Custom Email Server .....	1113
Activating a Sisense AMI from the AWS Marketplace.....	1116
<b>Viewing Dashboards on Mobile Devices.....</b>	<b>1119</b>

# **New to Sisense**

This section describes how Sisense works, how to set Sisense up, and provides a tutorial for building your first dashboard from scratch.

# Learn how Sisense Works

Sisense's end-to-end, single stack solution simplifies complex data in 3 short steps:

1. **Model:** Blend massive amounts of data from a variety of sources into the Sisense ElastiCube.
2. **Analyze:** Once your data is inside the ElastiCube, interact with it in a visual way to get deeper understanding, or zoom out to understand the holistic data landscape.
3. **Impact:** Create dashboards that make it easy to get insights and take action.

The topics below describe in more detail how Sisense works.

- [Sisense Basic Concepts and Terminology](#)
- [Sisense Architecture Technical Overview](#)

# Set Up Sisense

Before you start building ElastiCubes and designing dashboards, you need to set Sisense up, the topics below explain how:

- [Downloading and Installing Sisense](#)
- [Setting Up Your Production Environment](#)
- [Customizing the Installation](#)
- [Performing a Silent Installation](#)
- [Migrating Sisense](#)
- [Upgrading Sisense](#)
- [Launching Sisense](#)

# Give Sisense a Try

Now that you've set Sisense up, it's time to play. See the tutorial below to learn how to import data and build your first dashboard.

- [Tutorial: Getting Started](#)

See [Minimum Requirements and Supported Platforms](#) to see what you need to support Sisense.

## Minimum Requirements and Supported Platforms

The following prerequisites and supported platforms are required for working with Sisense.

### Supported Web Browsers

The Sisense Web Application runs in the following HTML5 supported browsers:

- Internet Explorer 11
- Google Chrome
- Firefox
- Safari version 7 and higher

**Note:** When embedding iFrames, Sisense supports Safari 10 and higher

The Sisense Web Application also works in mobile phone and tablet browsers that support HTML5. Click [here](#) to learn more about mobile compatibility.

Microsoft Edge is not currently supported.

## Supported Operating Systems

ElastiCube Server and ElastiCube Manager can be installed on the following 64-bit operating systems:

- Windows 7
- Windows Server 2008 R2 through Windows Server 2019  
If you are installing Sisense in Windows Server 2019, see [Disabling Windows Defender Real-Time Protection](#).

**Note:** While Sisense supports Windows 7, it is highly recommended that production environments use Windows Server 2008 R2 and later.

## Capacity and Hardware Requirements

Sisense easily scales up to billions of records with typical query response times of split seconds.

This section suggests system requirements for various performance capacities of the ElastiCube Server when connecting to data sources with live connections versus importing data into an ElastiCube. Actual capacity requirements are provided after consultation with a Sisense technical representative at [support@Sisense.com](mailto:support@Sisense.com). Extreme scenarios may require additional resources.

Note the following:

- For Sisense Viewers, only an HTML5 compliant web browser is required.
- For cloud deployments, regardless if you are using AWS/Azure/Rackspace/etc., you must choose a machine that meets the recommended hardware configuration described in the table below.

## Live Models

When connecting to a [Sisense Live](#) data source, Sisense recommends 32GB memory with 16 cores. For use cases with 100s of users concurrently connecting to your live data source, contact your CSM for more information about Sisense's minimum requirements. You can create up to 400 live models.

## Data Models

When importing your data into an ElastiCube, the table below describes the minimum requirements for your **production ElastiCube Server**. The number of rows column refers to the total number of rows for all the ElastiCube that support your dashboards. For trial users, 8GB typically allows you to test up to 1M rows when there are 10 or less users.

# of rows	< 10 users		10s of users		100s of users	
	RAM (GB)	Logical Cores	RAM (GB)	Logical Cores	RAM (GB)	Logical Cores
Up to 100K	16	8	16	8	16	8
up to 500K	16	8	16	8	16	8
up to 1M	16	8	16	8	32	8
up to 10M	16	8	16	8	32	8
up to 50M	16	8	32	8	64	16
up to 100M	25	8	32	8	64	16
up to 200M	32	8	32	8	64	16
up to 300M	32	8	64	16	64	16
up to 500M	64	16	128	16	256	32
up to 600M	128	16	128	32	256	32
up to 700M	128	16	128	32	512	32
up to 800M	128	16	256	32	512	32
over 800M	256	16	256	32	512	32

\* Users are defined as concurrent users of the system regardless of licenses.

## Disk Space

The Sisense installation file is typically around 1.5GB in size. Once Sisense is installed, Sisense uses about 20GB of space. In general, when deciding how much disk space to reserve for Sisense, keep in mind the size of the amount of data to be imported into your ElastiCube models. You should keep additional space to support your ElastiCube models as these are duplicated during the build process. The duplicate is removed at the end of the process.

**Note:** Sisense should be installed on SSD drives.

If you need to store your ElastiCube models in another location other than the default C drive, see [Change The Location Of The ElastiCube Data Folders](#).

# Sisense Recommendations

Sisense recommends that your Sisense server meets or exceeds the minimum requirements listed above. The actual requirements of your Sisense server may vary depending on the number of concurrent users, builds running in parallel, ElastiCubes hosted on a server, and additional factors specific to your server, for example, non-Sisense applications running on the same server.

For optimal performance, Sisense recommends:

- No more than 40 ElastiCubes on a single Sisense server
- No more than 4 concurrent builds at a time
- No more than 150 concurrent users

For optimal performance, Sisense highly recommends that your servers have a processor that supports AVX (Advanced Vector Extensions), which is leveraged by Sisense for improved query performance and user concurrency.

## Supported Locales

The formats for dates, times and numbers in your dashboards are based on your computer's operating system or browser's locale settings (depending on your browser).

Locales from the following countries are supported:

- Unites States
- United Kingdom
- Israel
- Canada
- South Africa
- Australia
- Netherlands
- Germany
- Ireland
- Mexico
- France

- China
- Brazil

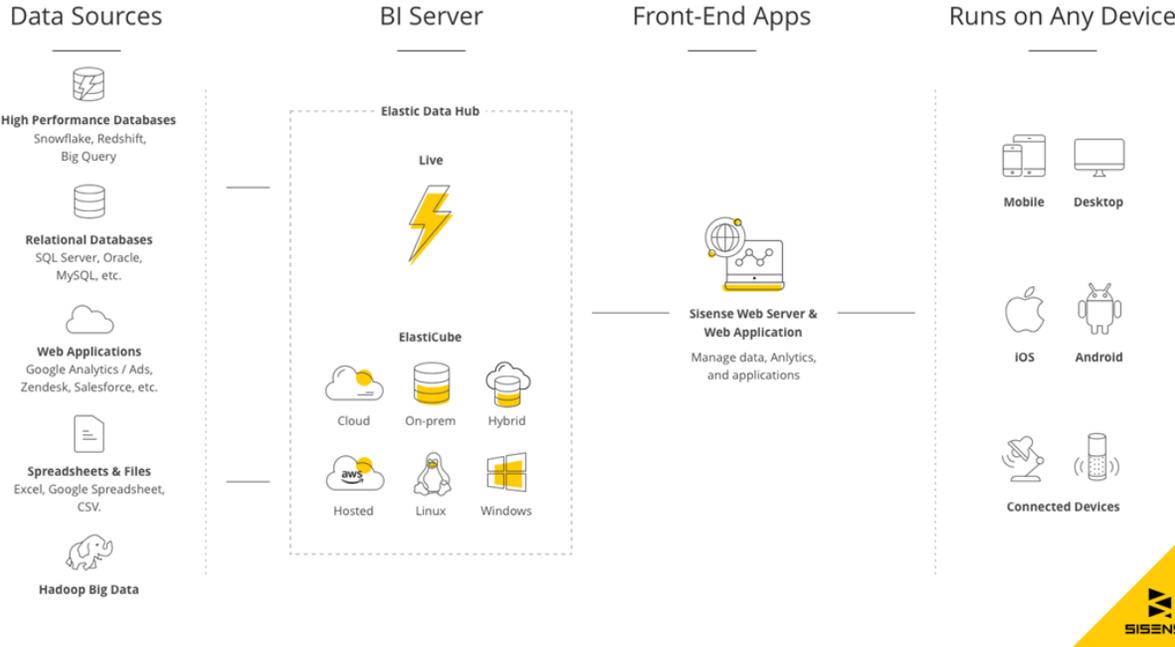
### **Notes**

- You can override the default locale settings for all users by updating a parameter via the Rest API. [Click here to learn how to override default locale settings.](#)
- To use a locale file that is not included in the above list of countries, you can manually add a locale file in the following location: `C:\Program Files\Sisense\PrismWeb\client\resources\base\localization`. [Click here to access locale files and view their codes.](#)

## **Sisense Basic Concepts and Terminology**

# Sisense Architecture

The Sisense system is comprised of the following components.



## ElastiCube Manager

Sisense is a visual environment in which you create ElastiCubes. Sisense is available online or locally on the desktop where Sisense is installed. Sisense enables you to structure and import multiple data sources, create relationships between data, and perform ETL (Extract, Transform, and Load) processes to prepare data for analysis and visualization. For more information, see [ElastiCube Manager](#).

## ElastiCube Server

The ElastiCube Server is installed locally on your computer and enables access to ElastiCubes. Both the Sisense Web Application and Sisense query the ElastiCube Server and receive results.

## Sisense Server Console

The Sisense Server Console provides administration functions for managing ElastiCubes on the ElastiCube Server.

## Sisense Web Application

The Sisense Web application is an interactive web application that provides the user interface in which users can design, share, view and explore dashboards.

The Sisense Web application runs in popular web browsers and enables access to the Sisense server. Dashboards can also be viewed on mobile devices.

## Sisense Application Server

The Sisense Application Server is installed locally on your computer and hosts the Sisense Web Application.

# Data Models

Data models are abstract entities that organize your data and determine how your tables relate to one another. Sisense has two types of models, Live models and ElastiCube models. Which model you use depends on how you connect to your data source. If you are connecting to a live data source, you will use a Live model. If you are importing your data into Sisense, you will use an ElastiCube model.

Data models enable you to mash up terabytes of data from a variety of sources, for example:

- Traditional relational databases, such as SQL Server, MySQL or Oracle
- File-based data sources, such as spreadsheets (Excel) and CSV files
- High Performance databases such as Snowflake, Redshift, Big Query
- Online web services, such as Salesforce.com, Google AdWords, Google Analytics, Zendesk and more

Live and ElastiCubes models are created and managed from the **Data** page. Both are described in more detail below.

## Live Models

Live models are a type of data model used to manage the schema over your Live data source. Live connections are useful for getting near real-time updates, with changes in your data reflected in your dashboard. For more information, see [Sisense Live](#).

## ElastiCube Models

ElastiCube are Sisense's proprietary super-fast data stores. ElastiCubes are specifically designed to withstand the extensive querying typically required by business intelligence applications.

ElastiCubes are based on Sisense's proprietary In-Chip technology, which leverages the Sisense engine. Read more about the technology behind ElastiCube in Sisense's [technology section](#).

# ElastiCube Sets

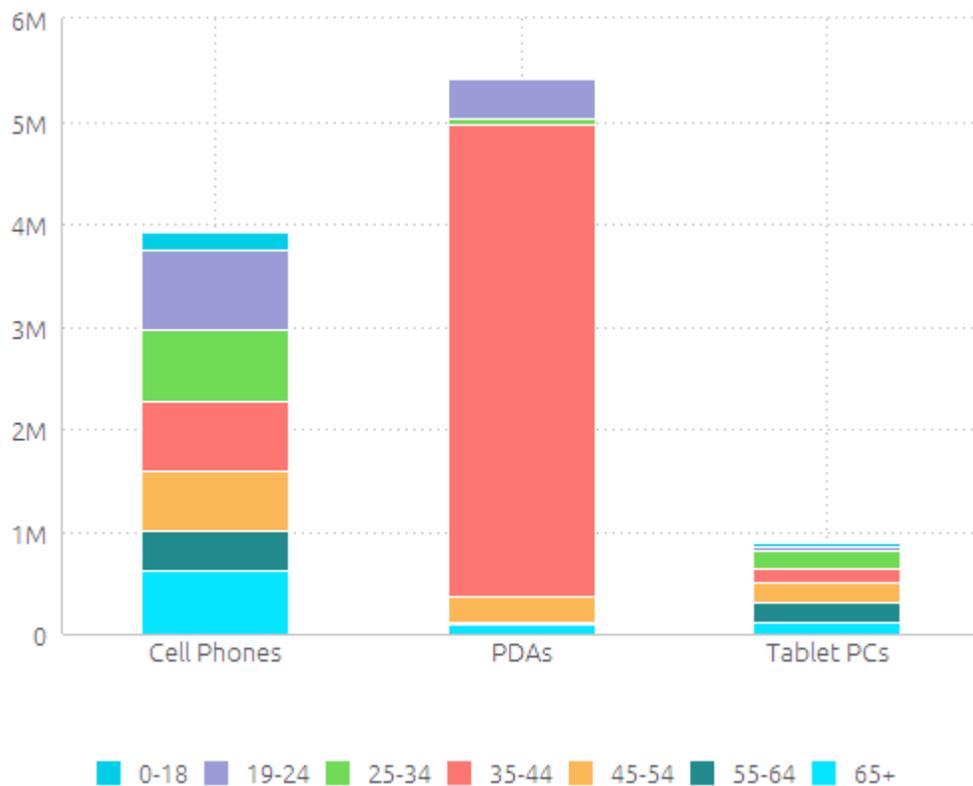
Sisense ElastiCube Sets are collections of ElastiCubes with identical schemas that support high availability by allowing you to query running ElastiCubes within the ElastiCube Set while other ElastiCubes are building.

For more information see [ElastiCube Sets](#).

# Widgets

Each widget is a dynamic visualization of data. You pick the type of data to appear in a widget and you pick the type of visualization (chart type). A few examples of widgets are displayed below:

## Column Chart Widget



# Indicator Widget

GROWTH OF VISITORS

13.2%

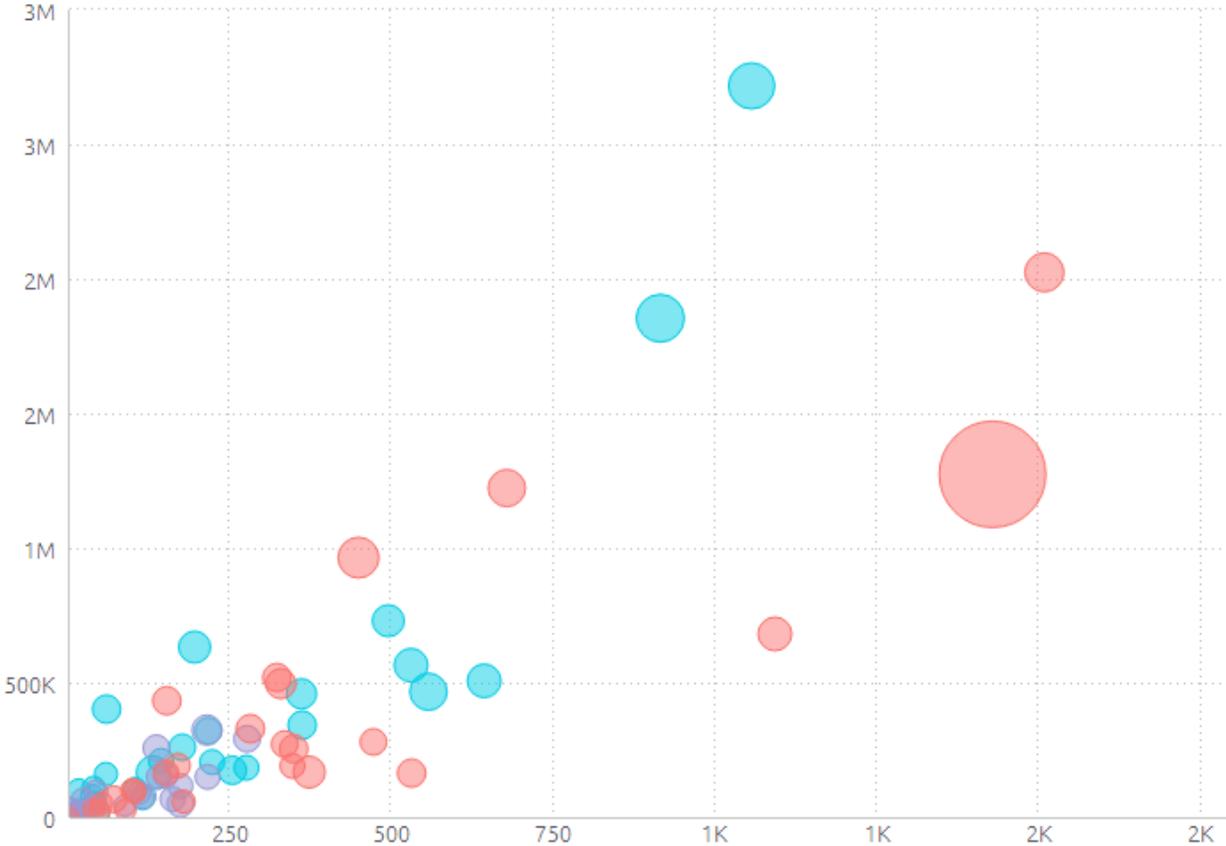
---

# of Visitors 220K

## Pivot Widget

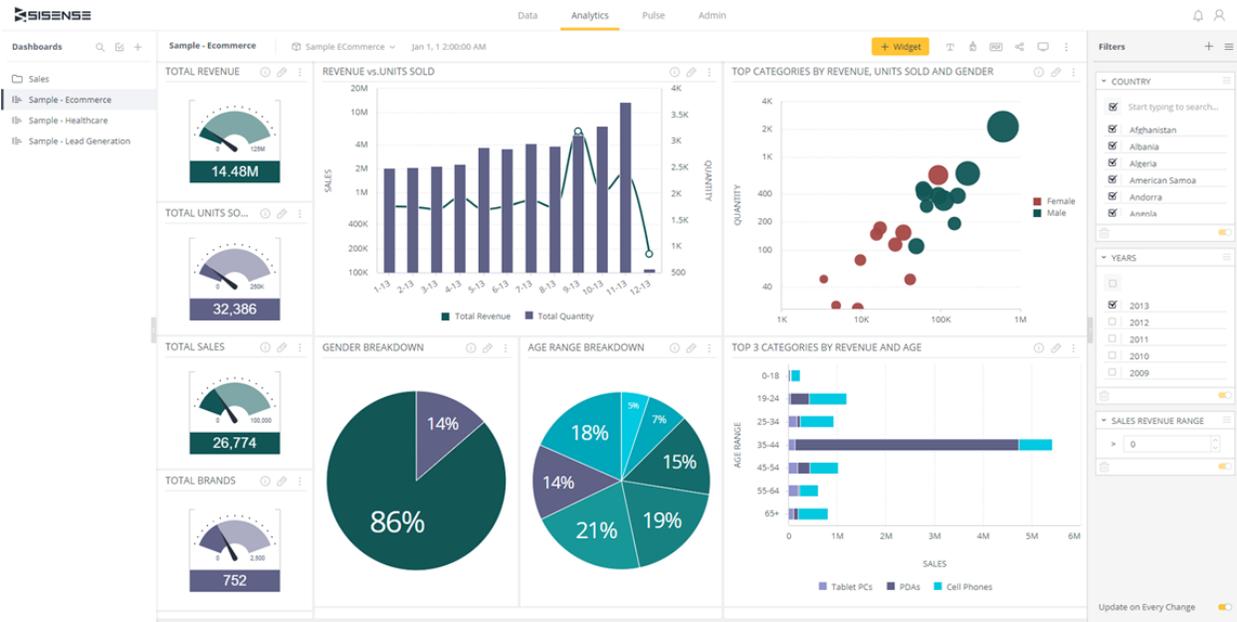
Category	Female		Male	
	Total Revenue	Total Quantity	Total Revenue	Total Quantity
Apple Mac Desktops	1,151	4	26,071	47
Apple Mac Laptops	5,310	18	50,717	110
Calculators	609	15	3,147	38
Camcorders	2,790	23	26,890	124
Camera Flashes	1,168	12	16,140	115
Car Amplifiers	734	6	4,619	32
Car Speakers and Subwoofers	612	9	6,968	90
Cell Phones	94,323	638	605,011	2,122
Digital Cameras	34,045	153	216,707	669
DVD Players	1,488	23	5,309	85

# Scatter Chart Widget



# Dashboards

A dashboard is a collection of one or more widgets that visualize the data that you select and design.



You define which widgets appear in the dashboard, their design, how they are organized and the filtering of the data that appears.

You can access your own dashboards and dashboards others shared with you.

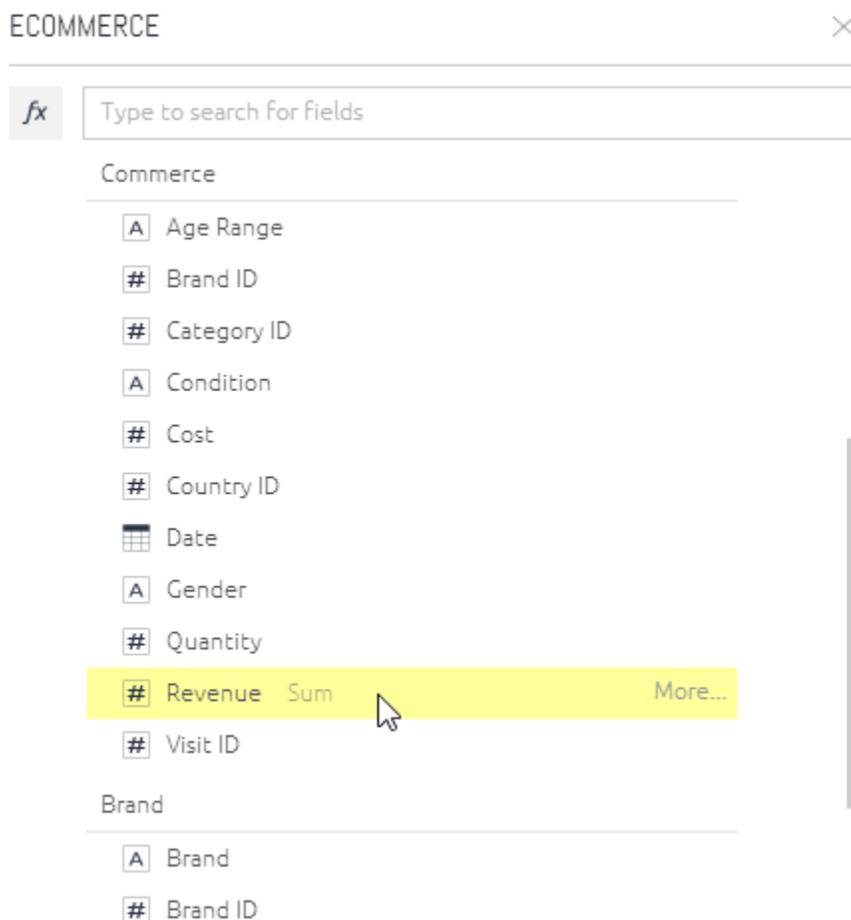
Sisense allows you to create as many dashboards as you need. No limitations!

When you create a dashboard, you are the dashboard's owner. Owners can share dashboards that they have created with other Sisense users.

Sisense provides a variety of built-in automatic dashboard sharing features. You can easily share a dashboard with others in the Sisense environment or have a dashboard automatically delivered to your chosen recipients by email (on a scheduled basis or upon each data update).

# Fields

Widgets are composed of fields that represent the data in your data sources. You create widgets by simply selecting from the fields displayed in a Data Browser, which appears in various places across the product. Each field represents a column of data in the ElastiCube.



When designing widgets, fields can be categorized into three groups:

- **Numeric Fields:** Numbers (quantified data), such as salaries, sales, scores, number of clicks and so on. This is data that you may want to aggregate or calculate. For example, the sum of sales or the average of costs.
- **Date Fields:** Dates can describe both date and time values. Dates can be used to organize your data into hierarchies according to year, quarter, and

month or into buckets of time such as by hour or 15 minute intervals. For more information, see [Working with Time](#).

- **Descriptive Fields:** Items used to label and categorize, such as Products, Locations, Categories and so on.

Generally, widgets combine both these types of fields.

For example, to show the sales of a product over time, Sales is the numeric data and both Products and Time are descriptive data.

Descriptive data can be considered as numeric when it is simply a count of the number of items. For example, Activities can be considered numeric when it is a count of the number of Activities. A number can be considered as descriptive when it is an actual entity. For example, an identification ID or phone number.

For information regarding Sisense internals and a more advanced explanation of the Sisense architecture, click [Sisense Architecture Internals](#).

## Sisense User Roles

Sisense licenses three categories of roles:

### Administrator

- **Admin:** Administrators can access the **Admin** page of the Sisense Web Application from where they can manage all system configuration, data sources, ElastiCubes, users, user groups and more. Sisense supports multiple Administrators.

Note: Sys.Admin is a special type of Administrator who installs Sisense on your system. There is only one Sys.Admin per account.

- **Data Admin:** Data Administrators can manage all data sources in the system. They have access to the **Admin** page of the Sisense Web Application where they can only manage ElastiCubes and live connections. Data Administrators can also add and manage servers. This is useful for migrating Sisense across environments. Data Administrators do not have access to User Management and System Configuration.

### Designer

- **Data Designer:** Data Designers can create and edit ElastiCubes they created or were shared with them. Data Designers can access the **Admin** page of the Sisense Web Application where they can manage the

ElastiCubes and live Connections they have access to in the Data Sources page. In addition, Data Designers can see servers, but can not add new servers. Data Designers do not have access to User Management and System Configuration.

- **Designer:** Designers can create, design, edit and share dashboards. Sisense Designers determine whether the user with who they share a dashboard has editing rights (is a Designer) or only viewing rights (Viewer).

### Viewer

- **Viewer:** Viewers can view, explore, drill down, make selections and filter the dashboards that Designers share with them. Viewers only need a standard web browser; no plugins or downloads. They cannot create new dashboards or edit existing ones.

**Note:** This roles can be customized through the Sisense REST API. For more information, see [Customizing User Roles](#).

	Administrator Roles		Designer Roles		Viewer Roles
	Admin	Data Admin	Data Designer	Designer	Viewer
<b>System Management</b>					
Access Admin Page	✓	✓	✓		
Edit System Configuration	✓				
Configure SSO	✓				
<b>User Management</b>					
Create/Delete Users	✓				
Create/Delete Groups	✓				
Add Active Directory Users	✓				
<b>ElastiCube Management</b>					

	Administrator Roles		Designer Roles		Viewer Roles
	Admin	Data Admin	Data Designer	Designer	Viewer
Add/Delete Servers	✓	✓			
View Servers	✓	✓	✓		
Add Live Connections	✓	✓	✓		
Create/Delete ElastiCubes	✓	✓	✓		
Edit ElastiCubes	✓	✓	✓		
Share ElastiCubes	✓	✓	✓		
Edit Servers	✓	✓	✓		
Create ElastiCube Sets	✓	✓			
Delete ElastiCube Sets	✓	✓			
Manage Hierarchies	✓	✓	✓		
Configure Data Access Rights	✓	✓	✓		
Stop/Restart Servers	✓	✓			
Stop/Restart ElastiCubes	✓	✓	✓		
Set Up Data Security	✓	✓	✓		
Access Data Page	✓	✓	✓		
<b>Dashboard Design</b>					
Create/Edit/Delete Dashboards	✓	✓	✓	✓	

	Administrator Roles		Designer Roles		Viewer Roles
	Admin	Data Admin	Data Designer	Designer	Viewer
Copy Dashboard to Another Server	✓	✓	✓	✓	
Restore Dashboards	✓	✓	✓	✓	
Change Color Palettes	✓				
Create/Edit/Delete Widgets	✓	✓	✓	✓	
Rename Widgets	✓	✓	✓	✓	
Duplicate Widgets	✓	✓	✓	✓	
Change Widget Type	✓	✓	✓	✓	
Drill into Widget	✓	✓	✓	✓	✓
Reorder Widgets	✓	✓	✓	✓	
Create/Delete Dashboard Filters	✓	✓	✓	✓	
Toggle Dashboard Filters On/Off	✓	✓	✓	✓	✓
Create/Delete Widget Filters	✓	✓	✓	✓	
Toggle Widget Filters On/Off	✓	✓	✓	✓	
<b>Sharing</b>					

	Administrator Roles		Designer Roles		Viewer Roles
	Admin	Data Admin	Data Designer	Designer	Viewer
Export Dashboards to PDF, Img, CSV	✓	✓	✓	✓	✓
Share Dashboards	✓	✓	✓	✓*	
Migrate Dashboards	✓	✓	✓	✓*	
<b>Pulse</b>					
Access Pulse	✓	✓	✓	✓	✓
Create Data Alert	✓	✓	✓	✓	✓
Create Build Alert	✓	✓			
Delete Alerts *	✓	✓	✓	✓	✓
Edit Scripts	✓	✓	✓	✓	

\* The user must be the owner of the dashboard, ElastiCube, or alert.

## Sisense Architecture

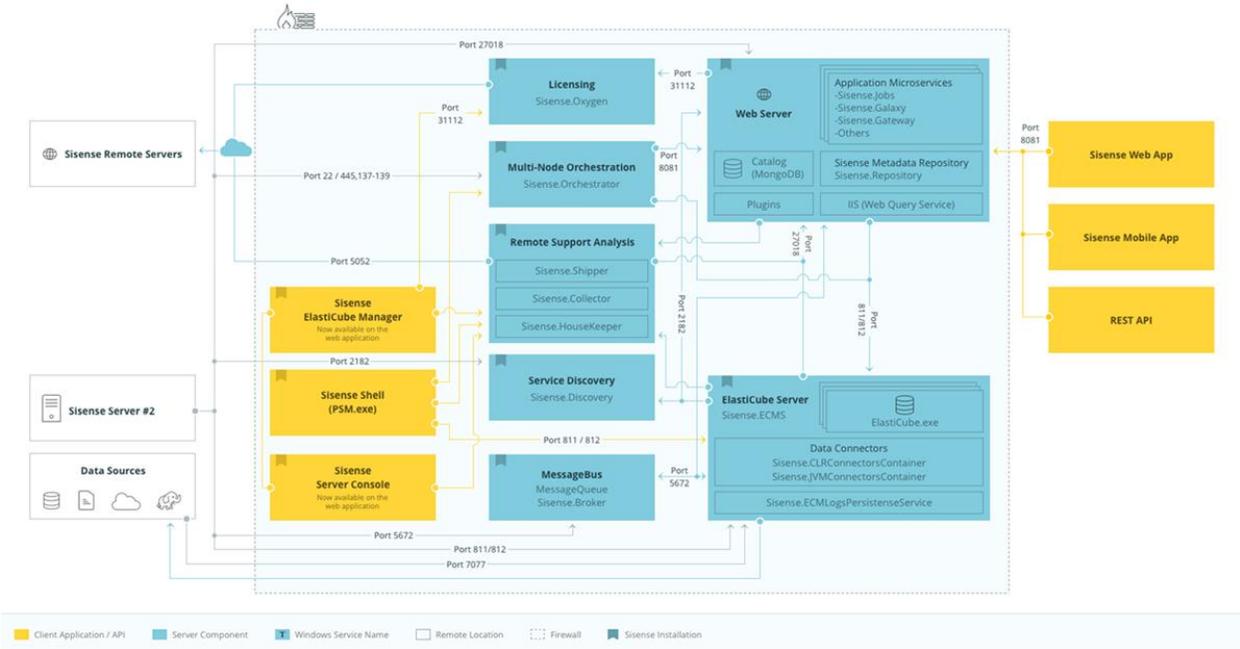
This topic provides an advanced description of Sisense's architecture for IT managers and Administrators responsible for supporting Sisense in their organization.

For a high-level overview of the Sisense architecture, click [Sisense Architecture](#).

When working with Sisense, there are two possible types of deployments. The first type is a **Single Server Deployment** in which Sisense is installed on a single server. Single Server Deployments can be leveraged to support single integrations or as proof of concept integrations for larger deployments.

The second type is a **Distributed Deployment** where Sisense is installed on multiple machines, but only part of the components are active on each one. This type of configuration is used to support heavier traffic loads and [high availability](#). The Sisense full-stack solution is provided in a single installation process. After installing Sisense on a server, you will have access to all of the Sisense functionality.

The diagram below describes the main components of a Sisense installation and how Sisense supports a full-stack solution:



In this diagram, the yellow components specify Sisense client applications or APIs. The blue components specify components of the Sisense server. The Windows service and application names that you can see in the Windows Task Manager under the **Services** or **Processes** tabs appear in a dark blue font. When Sisense is installed on a single server, the dotted line indicates the boundaries of the server console firewall. The lines between the components indicate communication between them, and its direction. When the Sisense Server listens on a specific port for inbound traffic, the relevant ports must be open in the firewall for the communication to succeed.

The diagram above describes the ports used by the services, for more information about communication ports, click [here](#).

# Sisense Architecture

Sisense supports a full stack solution from data, such as transaction details, user information, and machine-generated data, to dashboards accessible by viewers. The Sisense Server is responsible for managing data. Before you can begin to manage data however, Sisense communicates with remote servers for importing data sources, licensing information, tracking and similar activities.

The Sisense Server manages data and supports the Sisense Web Server, which hosts the Sisense Web Application. In a Single deployment, the Sisense Server resides locally on your machine. In a Clustered deployment, the Sisense Server can reside on one of your machines or across many machines depending on your needs. For example, if you want to support a high availability environment, you may install the ElastiCube Server on one machine, the Application Server on another, and the Multi-Node Orchestration server on another machine.

# Sisense Server

The Sisense Server is comprised of the following components:

1. **Web Server:** The Sisense Web Server is installed locally on your computer and hosts the Sisense Web application. It provides access to Sisense dashboards and Sisense Admin screens for user management. Each dashboard created is maintained within the Web Server catalog and application database, which contains metadata regarding users, groups, dashboards, data models, and system settings. The Web Server is comprised of many additional services required by the Sisense front-end, including a Proxy service for routing application requests, query services that transform JAQL to SQL, plug-in management, and more.

There are three main client applications that communicate with the Application Server:

1. **Sisense Web App:** This Sisense Web Application is the client application for dashboard creation and access, and system administration.
  2. **Sisense Mobile App:** This Sisense Mobile Application is an Android and Apple-device compatible application for viewing dashboards across devices.
  3. **REST API:** The Sisense REST API provides programmatic access to the Sisense Application Server functionality including managing ElastiCubes, user and group management, white labeling, user roles, reporting, and security.
2. **ElastiCube Server:** The ElastiCube Server supports the management of ElastiCubes and the data they contain. The Sisense Server also supports the Application Server, which queries the ElastiCube Server where the ElastiCubes are hosted. Within the ElastiCube Server are the following main components:
    1. **ElastiCube Catalog:** The catalog maintains a list of ElastiCubes contained within the ElastiCube server.
    2. **Data Connectors:** The Data Connectors service is responsible for managing both native and 3rd party connectors.
  3. **Client Applications:**
    1. **Sisense ElastiCube Manager:** A Windows client application for managing data sources. The ElastiCube Manager provides the GUI for creating ElastiCubes, including importing data sources, preparing them, and building the ElastiCubes.

2. **Sisense Shell (PSM.exe):** A command line interface for managing ElastiCubes, such as building an ElastiCube, attaching and detaching ElastiCubes from a server, updating data sources, etc. The Sisense Shell can also be controlled programmatically.
3. **Sisense Server Console:** A Windows application that controls Sisense servers, including attaching ElastiCube folders to Sisense servers and starting and stopping ElastiCubes.
4. **Licensing:** This service checks and validates licensing information.
5. **Multi-Node Orchestration:** The Sisense Orchestrator Service is an automated service that can be configured to synchronize builds across the ElastiCube Set.
6. **Remote Support Analysis:** This service is responsible for controlling logging, monitoring and automatic software updates.
7. **Message Bus:** The Message bus is a module responsible for transporting events across Sisense components.

## Sisense Remote Servers

Sisense remote servers are used for license management, monitoring and software updates.

## Data Sources

Data sources may be files or databases located on servers on a local network, or remote locations and web services such as SalesForce and Google AdWords.

Sisense supports a variety of data sources through native connectors, customer REST connectors, and ODBC drivers. These connectors are used to import data into the ElastiCube Server.

## Downloading and Installing Sisense

This topic describes how to download Sisense and the installation process for getting Sisense up and running.

# Downloading Sisense

If you want to try Sisense for the first time, sign up for the free trial at [Sisense.com](https://sisense.com).

Otherwise, you can download the latest version of Sisense [here](#).

See [Sisense Release Versions](#) if you need to download a previous version of Sisense.

# Installing Sisense

The entire Sisense solution can be installed from one file, either locally or in a central place in your organization within a couple of minutes.

All Sisense components are installed with a default configuration, as follows:

- Sisense Web Application
- ElastiCube Manager
- Sisense Server Console
- ElastiCube Server

**Note:** To install Sisense, you must have administrator privileges on the machine where you are installing Sisense and your environment must meet the [Minimum Requirements and Supported Platforms](#). In addition, Internet Explorer's enhanced security option should be disabled.

**Note:** If you are installing Sisense in Windows Server 2019, see [Disabling Windows Defender Real-Time Protection](#).

## To install Sisense:

1. Open the downloaded executable file to run the installation.  
**Note:** If your Window Firewall is active, click Allow access to continue installing Sisense.
2. If you are not logged into Sisense, enter your username and password to confirm licensing and begin the installation process. If you are logged in

already, the activation screen is not displayed.



## Sisense Activation

Sign In [or create a new account](#)

**Check your inbox** - your password was sent there when you registered

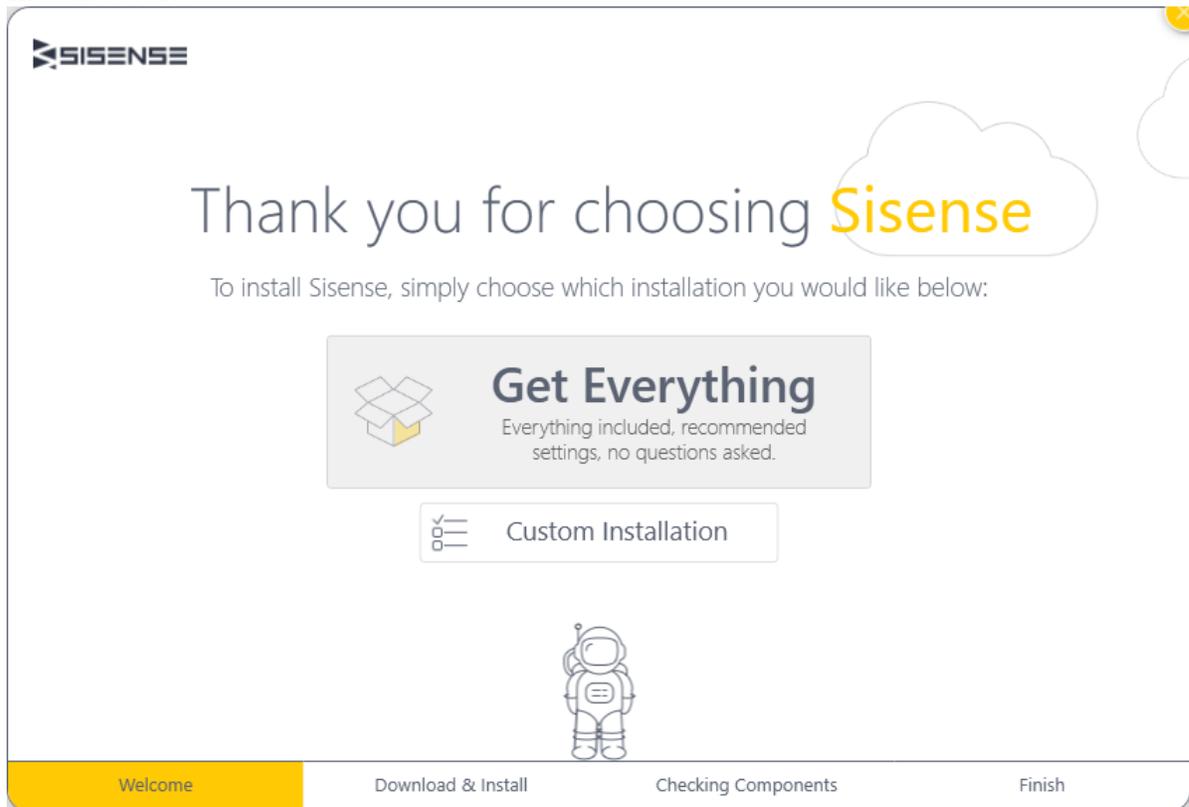
[Forgot Password?](#)

[Behind a Firewall?](#)

[Got a Serial Number?](#)

 Sign In

3. After entering your login details, select **Get Everything** to run the default installation:



If you are upgrading your version, click **Upgrade**.



Gideon,  
Welcome Back!



**Continue**

Use your current account  
and continue with Upgrade



My Account

By clicking the button above you agree to our [Terms and Conditions](#)

# Installing Sisense behind a Firewall

The standard installation requires internet access to download the necessary components. For machines behind a firewall, or without internet access, you can download and install the full self-contained installation file.

For detailed instructions, see [Installing Sisense Offline](#).

# Recommended Antivirus Exclusions for Sisense

Some folders, processes, and services may have to be excluded from anti-virus scanning when you use anti-virus software. If these are not excluded, you may see unexpected behavior such as problems accessing or installing Sisense. See Sisense's [Recommended Exclusions](#) for the complete list of services, processes, and folders that need to be excluded.

## Related Topics:

If you need to customize your installation, see [Customizing the Installation](#). You can customize your installation during the first installation or after an installation, but not during an upgrade.

If you want to install Sisense on a remote machine, see [Performing a Silent Installation](#).

If you are migrating your installation to a different machine, see [Backing Up and Migrating a Sisense Installation](#).

If you have any problems installing Sisense, see the [Installation And Upgrade FAQ](#).

## Setting Up Your Production Environment

This topic describes how to set up a production environment for hosting your Sisense implementation.

Sisense recommends installing Sisense in a staging environment, where you can check all of your ElastiCubes and dashboards, as well as test different configurations and customization options. When everything is working as expected, deploy Sisense in your production environment. To learn more about migrating your installation, see [Migrating Sisense](#).

**Note:** Port 8081, or 443 for secure connections, should be reserved for the Sisense application only. No other applications should run on these ports on your server.

# Providing Remote Access to the Sisense Web Environment

To maximize the Sisense experience, you will want to make your dashboards accessible to external users outside the organization's network. This requires the following configurations:

- Open TCP port (default is 8081) to access the Sisense Web Application internally.
- Ensure that your Sisense machine has a public IP address or domain name associated with a public IP for external access.
- Open port 443 if you are using SSL. Sisense recommends using SSL for secure connections or a VPN solution. For more information about implementing SSL, see [Setting Up SSL](#).

External users who were granted access can view dashboards by entering the machine's IP and the port number, for example: xx.xxx.xx.xx:8081 or my.company.com:8081

Click [here](#) to learn more and see troubleshooting tips.

# Providing Remote Access to Sisense

In some cases, you may want to provide external access to Sisense for designing and modeling your ElastiCubes. In such cases it is necessary to consult your Sisense Success Manager to make sure your licensing terms allow this option. In addition, make sure that ports 811, 812 on your Sisense machine are open.

**Note:** To avoid exposing your data, do not open ports 811/812 without the proper security measures.

# Sisense Email Server

Sisense provides you with an email server for sending reports and notifications to your users from Sisense, to be used in POC and testing environments. When you migrate your deployment to a production environment, you should configure your own custom email server, so you can manage your emails according to your company's policies.

For more information, see [Setting Up a Custom Email Server](#).

# Base URLs

Base URLs are the consistent part of a web address for a site or web application. In Sisense, the default address is localhost:8081. Sisense allows you to specify a Sisense hostname as a subfolder of a domain, for example, baseurl.sisense.com/reporting.

For more information, see [Configuring Base URLs](#).

# Customizing the Installation

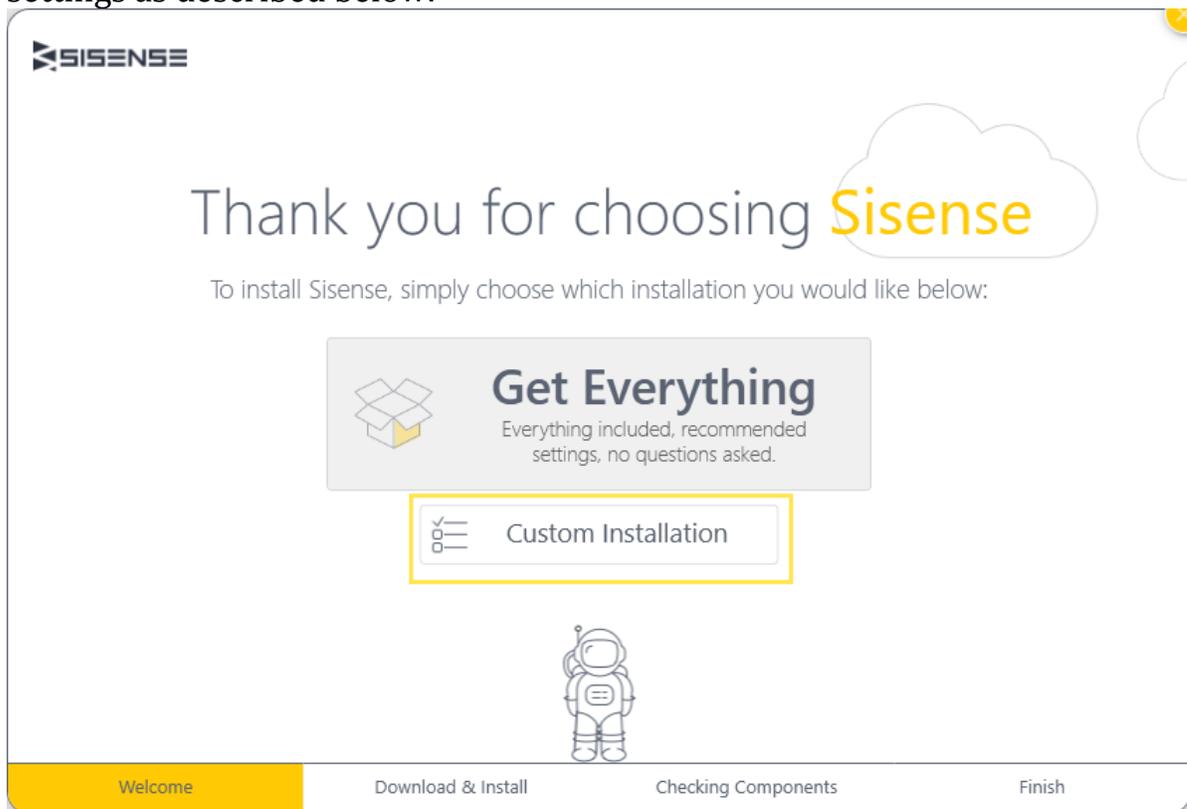
This topic describes how you can customize the settings for your web server application during the installation process or after Sisense is installed.

For example, you can configure your server to use the Long index edition of Sisense if your ElastiCubes are expected to contain a table with over 300 million rows.

**Note:** You cannot customize the settings during an upgrade process. If you upgraded Sisense and want to customize settings, refer to the post-installation option described below.

**To access the customization options during the installation process:**

- While [installing Sisense](#), click **Customize Installation**, and define the settings as described below.

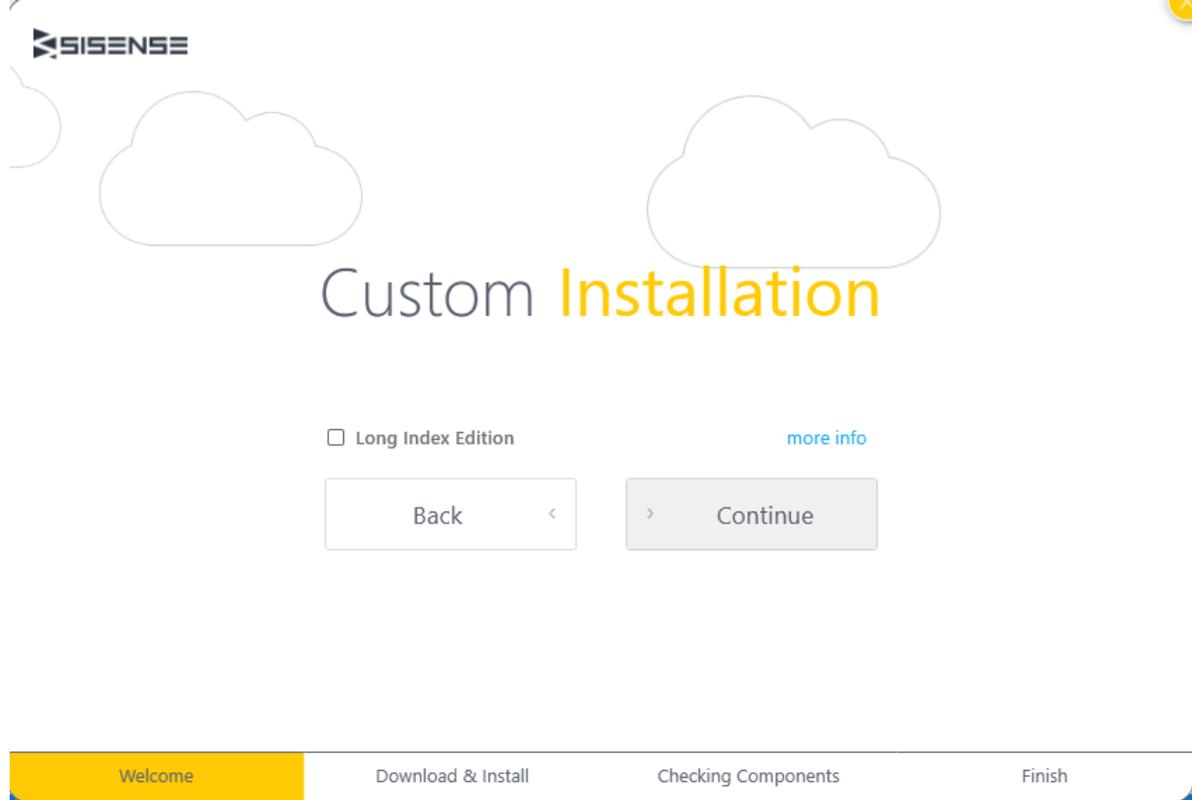


## To access the customization options after you have already installed

Sisense:

1. Open the Control Panel, and go to **Programs > Programs and Features**. Right-click **Sisense**, and click **Change**.

2. Click **Continue** > **Change Features**, and define the settings as described below.



# Settings

## ElastiCube Server

Select the [Long Index](#) edition if your ElastiCubes are expected to contain a table with over 300 million rows.

It is not recommended to use this configuration setting for ElastiCubes in which the largest table does not include at least 300 million rows.

## Performing a Silent Installation

This topic describes how to install Sisense silently.

You can install Sisense using the command-line interpreter, for example, when you need to run an installation on a remote machine.

You can also customize the installation using variables, for example, install Sisense without the sample dashboards.

### To perform a silent installation:

1. Download the [latest sisense version](#).
2. Open the command line interpreter and type in the installation commands as follows:
  - Full installation (without previous user activation)  
`<Location of downloaded executable file>\SisenseLatestFull.exe -q -username="<>" -password="<>"`
  - Full installation (for user already activated by Sisense)  
`<Location of downloaded executable file>\SisenseLatestFull.exe -q`
  - Full offline installation (requires a license key from Sisense)  
**Note:** To get an offline license, go to [My Account](#) and click **GET KEY**, or contact your Sisense Success Manager.  
After attaining your key, copy and paste it after `-offlinelicense=` (replacing `<offline_license_key>` in the example below).  
`<Location of downloaded executable`

```
file>\SisenseLatestFull.exe -q -  
offlinelicense=<offline_license_key>
```

- **Installation without Sample ElastiCubes and dashboards**

<Location of downloaded executable

```
file>\SisenseLatestFull.exe -nosamples -q
```

- **Choosing Sisense Website name and port**

<Location of downloaded executable

```
file>\SisenseLatestFull.exe -q -
```

```
webname=<website_name> -webport=<website_port>
```

# Silent Customized Installations

There are two options available for customizing silent installations:

- **bigdata:** Add if you want to install the Long Index version of Sisense. Long Index determines how Sisense transforms your data when loading it in to an ElastiCube. Sisense builds your ElastiCube using pointers that represent your data. These pointers are used to find the actual values so no information is lost. To reduce memory consumption and improve performance, Sisense builds ElastiCubes with 32-bit pointers reducing the size of the pointers. You may experience “Out of memory” errors when working with big data such as: 300 million rows of numeric fields 2GB of string fields For example, let’s assume that your data contains strings with a length of 1024 characters and 4,194,304 unique rows such that  $1024 * 4,194,304 > 4\text{GB}$  in content after indexing, you should install the Long Index version of Sisense. By installing the Long Index edition, Sisense will use 64-bit pointers. While this is fine for big data, you should use the default installation of Sisense when working data smaller than the figures mentioned above for faster performance. edition if your ElastiCubes are expected to contain a table with over 300 million rows. It is not recommended to use this configuration setting for ElastiCubes in which the largest table does not include at least 300 million rows.
- **webport:** Add to customize what port Sisense runs on.

For example:

```
SisenseLatestFull.exe -q -bigdata -webport=80
```

After Sisense is installed, the only option to change the port is through the **Port** field in the Configuration Manager.

## To uninstall Sisense using the command-line interpreter:

In the command-line interpreter, enter:

```
<Location of downloaded executable  
file>\SisenseLatestFull.exe /q /uninstall
```

# Installing Sisense Offline

The standard installation of Sisense requires internet access to download additional necessary components. To support machines behind firewalls, or without internet access, you need to download and install the full Sisense [installation file](#). After you have installed Sisense, you need to provide an offline activation key.

The steps below describe how to install Sisense offline and activate it.

# Prerequisites

Before installing Sisense offline, make sure you have the following installed:

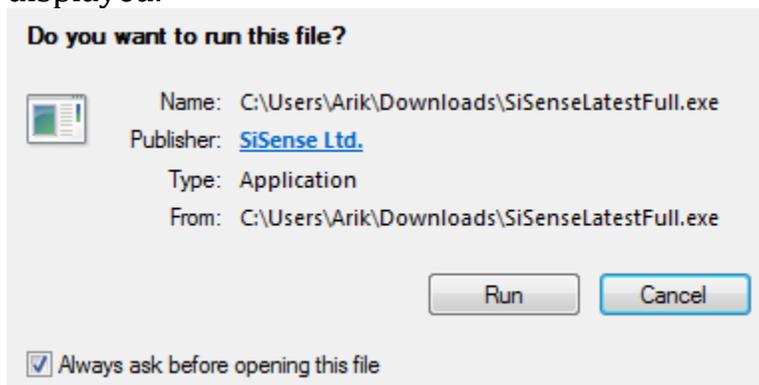
1. [Microsoft .NET 4.6.1 Framework](#)
2. **IIS - Internet information Services** (Microsoft's Web server), see the following instructions per your operating system  
[Windows 7 or 8](#)  
[Windows Server 2008](#)  
[Windows Server 2012](#)

# Installing Sisense

**Note:** If you are installing Sisense in Windows Server 2019, see [Disabling Windows Defender Real-Time Protection](#).

## To install Sisense offline:

1. Download the [latest full version of Sisense](#).
2. Open the installation file and click **Run**. The Welcome to Sisense window is displayed.



3. In the Welcome to Sisense window, click **Activate**.

## Welcome to Sisense



By clicking the button above you agree to our [Terms and Conditions](#)

4. In the Sisense Activation window, click **Behind a Firewall**. The product ID is displayed.

Sisense Activation

Sign In or create a new account

Email

Password

Check your inbox - your password was sent there when you registered

Behind a Firewall?

Got a Serial Number?

5. Copy the product ID.

Sisense Activation

Sign In behind a Firewall

Paste Key here...

Don't have a key?

1. Copy the following ID `535892DC57B526FA0532B4E62796B6C6`
2. Log into your account at <http://www.sisense.com/my-account/>
3. Use the ID in the 'Get License Key' section to generate the license key.
4. Paste it into the box on the left.

Back <

→ Sign In

6. On an online machine, open <http://www.sisense.com/my-account/> and log in with your Sisense credentials.

7. In the My Account page, from the left menu, select **GET KEY**.
8. Paste your product ID from the Sisense installation into the field and click **GET KEY**. Your Off License key is displayed.

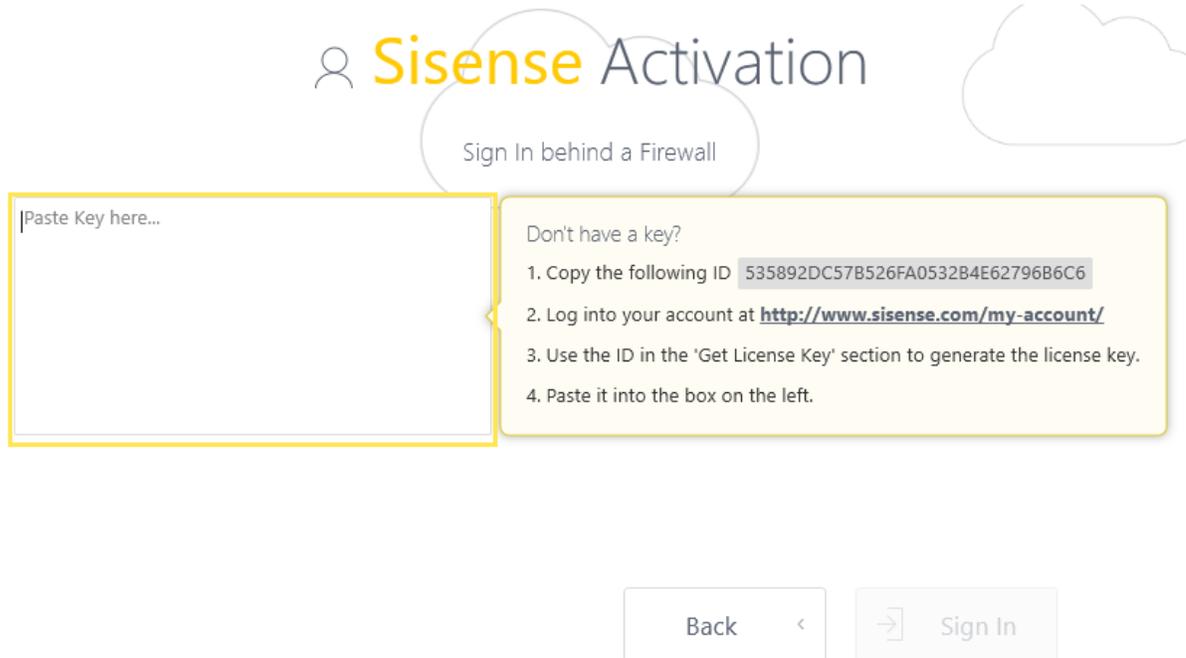
#### GET OFFLINE LICENSE KEY

SiSense Prism occasionally validates its license status via an Internet connection. For situations where an Internet connection is unavailable (or blocked by a firewall), an offline license key can be generated from this page instead.

```
nMaUr4KLZGTXLwTwleaqYD0uqjFUQZ3Mpvt2DTFcx2eF4EWlafg7fDT85hI2
Nx47we6lk3XR94RGBXs5V9k7FgkNs9TPITiJQgPqkhET5tAmh+QJR9WCU52
1Hcz                                     300DC
St8e                                     75Lng
JjMd                                     19gfaI0
/gsqk                                    OEgjS
cwb6                                     3rYOIG
UoTI                                     xlcBcV
Wzyl                                     jdejbD
maE                                       iSzgY
Ve/9                                     SbDyp
515h                                     Vfmjq3
FDh`                                     3WmB
oJyr                                     jz2AB
sE5+                                     Ln+Bk
uub6                                     iarYhr
XrCV                                     :ldwrM
CaXU                                     1qJ5u
zQWN69hvr0IDTbQwx7vgh9r+niK+1jU+P6fN38s9Q6hYsdVTILylkWIFTDbljR4/
1o99zfnPYwxBzEAgwZ+elwZ2mxcHmuMjLTXfhAEIjXymegYZhC8b3JN1BWdj/
7L
```

**GET KEY**

9. Copy the key and paste it into the key area in the Sisense Activation window.



10. Click **Install**. Sisense is activated and you can begin to install Sisense.

## Backing Up Sisense

The backup process requires backing up your ElastiCubes and the Sisense Web Application.

In addition, if you have any custom plugins located in the directory `C:\Program Files\Sisense\PrismWeb\plugins`, this folder should be backed up as well.

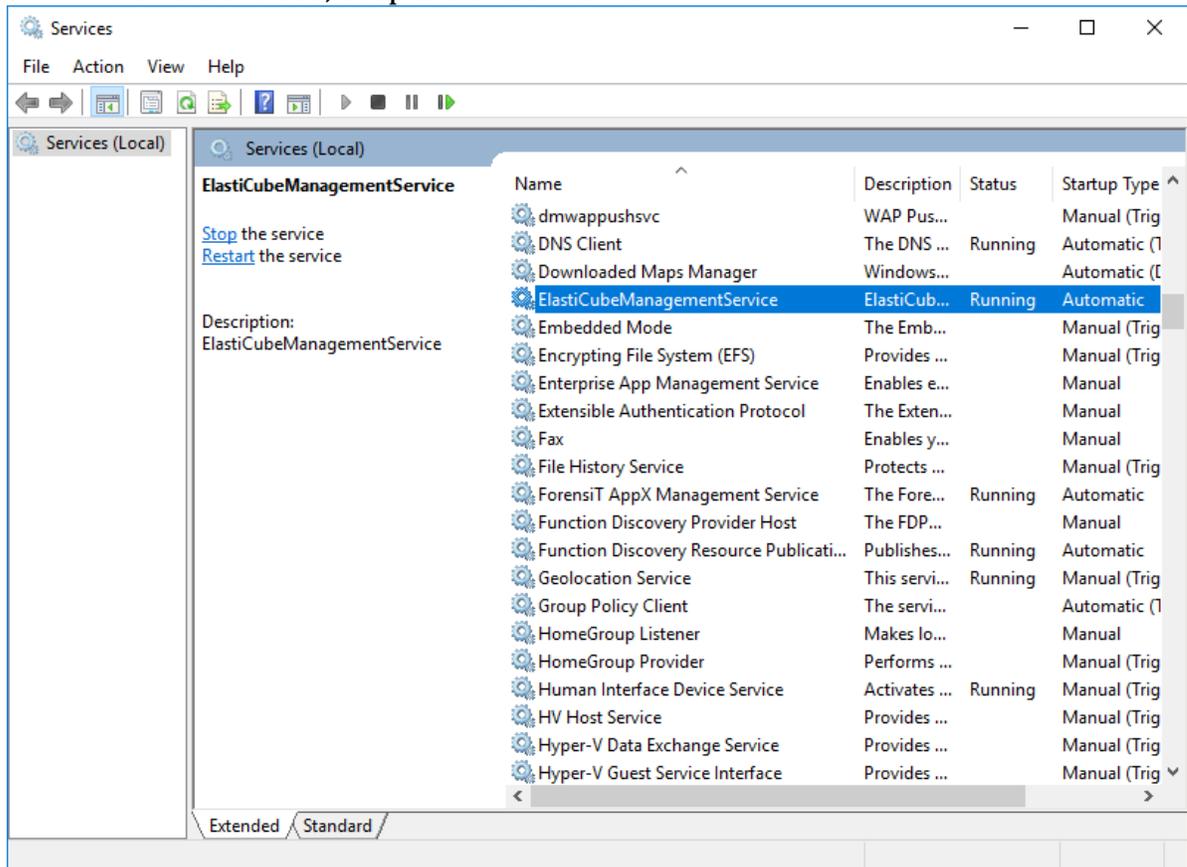
The procedures below describe how to back up your ElastiCubes, the Sisense Web Application, and Sisense in high availability environments.

# Backing Up Your ElastiCubes

Your ElastiCube Server contains information about your ElastiCubes, including the schema and the data that was imported when the ElastiCube was built. This information is saved in a collection of folders stored in the ElastiCube Data folder. The files and folders in the ElastiCube Data must be saved and their structured maintained to ensure your implementation will work as expected on the migrated machine.

## To back your ElastiCubes:

1. In Windows Services, stop **Sisense.ECMS**.

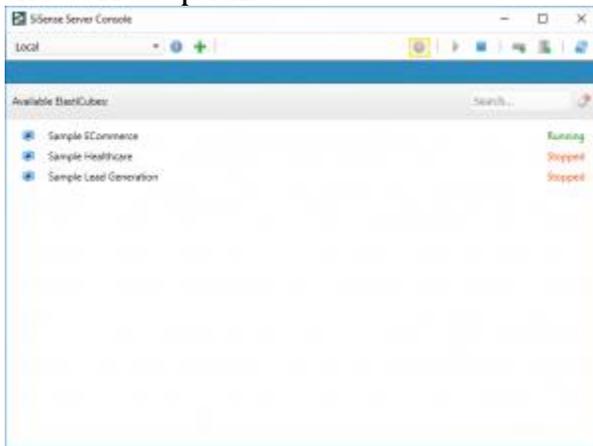


2. Open the directory:

`%ProgramData%\Sisense\PrismServer\ElastiCubeData`

**Note:** This is the default path. To check which path is being used, open the Sisense Server Console and click on the Server Preferences icon  to verify

the correct path.



3. Copy the data to your backup location.

# Backing Up the Sisense Web Application

The Sisense Web Application uses information stored in an instance on the Sisense application database where the Sisense Web Application is installed. This instance stores data about users, dashboards, widgets, and other information. This information must be stored to ensure your Sisense Web Application runs properly.

## To back up the Sisense Web Application:

1. In Windows Services, stop the Sisense.Repository and Sisense.Discovery services.
2. Open the directory:  
`%ProgramData%\Sisense\PrismWeb\Repository\DB`
3. Copy the data to your backup location maintaining the same file structure.
4. Open the directory:  
`%ProgramData%\Sisense\Infra\Discovery`
5. Copy the data to your backup location maintaining the same file structure.
6. (Optional) Back up your custom plugins by copying the directory:  
`C:\Program Files\Sisense\app\plugins`

For information on how to automate the backup process, see [Automatically Backup Sisense Web Data](#).

# Backing Up Sisense in a High Availability Environment

If you have implemented a high availability environment for Sisense and want to back up your installation, the files described below should be backed up in addition to the files described above in [Backing Up Your ElastiCubes](#) and [Backing Up Your Sisense Web Application](#):

## Build Node(s):

- C:\Program Files\Sisense\Sisense.Orchestration\config\config.json

## Web Node(s):

- C:\Program Files\Sisense\Infra\MongoDB\keyfile. See [Creating a KeyFile](#) for more information.
- C:\Program Files\Sisense\Infra\MongoDB\mongodbconfig.conf
- C:\Program Files\Sisense\Infra\Discovery\conf\zoo.cfg

# Upgrade Guide

Sisense releases several major versions a year, and a few minor versions in between. Each version includes new features and enhancements as well as other improvements.

Sisense allows you to decide when you want to upgrade to the latest version.

There are no automatic updates for Sisense. If you decide to upgrade, follow the steps below to ensure a smooth upgrade. These steps include best practices, instructions for upgrading in single server versus Multi-Node environments, and suggestions for avoiding potential problems that can occur.

**Important:** Which version you are upgrading from matters. If you are upgrading from a version earlier than Sisense V7.2, see the [Sisense V7.2 Upgrade guide](#). If you are upgrading from Sisense V7.2 and later, there are no special requirements.

The following are a list of steps you should review or perform when upgrading Sisense.

1. [Researching Your Upgrade](#)
2. [Backing Up Sisense](#)
3. [Performing an Upgrade](#)
4. [Testing Your Upgrade](#)
5. [Upgrading Your Production Environment](#)

## Next Steps

- [Researching Your Upgrade](#)

# Upgrade Guide

Sisense releases several major versions a year, and a few minor versions in between. Each version includes new features and enhancements as well as other improvements.

Sisense allows you to decide when you want to upgrade to the latest version. There are no automatic updates for Sisense. If you decide to upgrade, follow the steps below to ensure a smooth upgrade. These steps include best practices, instructions for upgrading in single server versus Multi-Node environments, and suggestions for avoiding potential problems that can occur.

**Important:** Which version you are upgrading from matters. If you are upgrading from a version earlier than Sisense V7.2, see the [Sisense V7.2 Upgrade guide](#). If you are upgrading from Sisense V7.2 and later, there are no special requirements.

The following are a list of steps you should review or perform when upgrading Sisense.

1. [Researching Your Upgrade](#)
2. [Backing Up Sisense](#)
3. [Performing an Upgrade](#)
4. [Testing Your Upgrade](#)
5. [Upgrading Your Production Environment](#)

## Next Steps

- [Researching Your Upgrade](#)

## Researching Your Upgrade

Before you upgrade, you should learn about the new features and enhancements in the version you are upgrading to, and check if there are any breaking changes. See the [Release Notes](#) for more information. Also, check Sisense's [Minimum Requirements](#) to make sure your environment meets Sisense's recommendations, and that all the necessary [communication ports](#) are open. If you are upgrading from a version earlier than Sisense V7.2, review the Sisense V7.2 Upgrade guide first for some important changes.

**Best Practice:** Sisense recommends that you never install a beta version of Sisense in your production environment. You should also never restore your production environment using a backup of a beta version.

Sisense recommends that you review the following:

- [Release Notes](#)
- [Minimum Requirements](#)
- [Communication Ports](#)

If you are upgrading from a version of Sisense earlier than Sisense V7.2, see the [Sisense V7.2 Upgrade Guide](#). If you are upgrading from version Sisense V7.2 or later, there are no special requirements.

## Next Steps

- [Backing Up Sisense](#)

## Backing Up Sisense

The backup process for Sisense requires backing up your ElastiCubes and the Sisense Web Application. In addition, if you have customized your environment, you may want to back up your customizations as well.

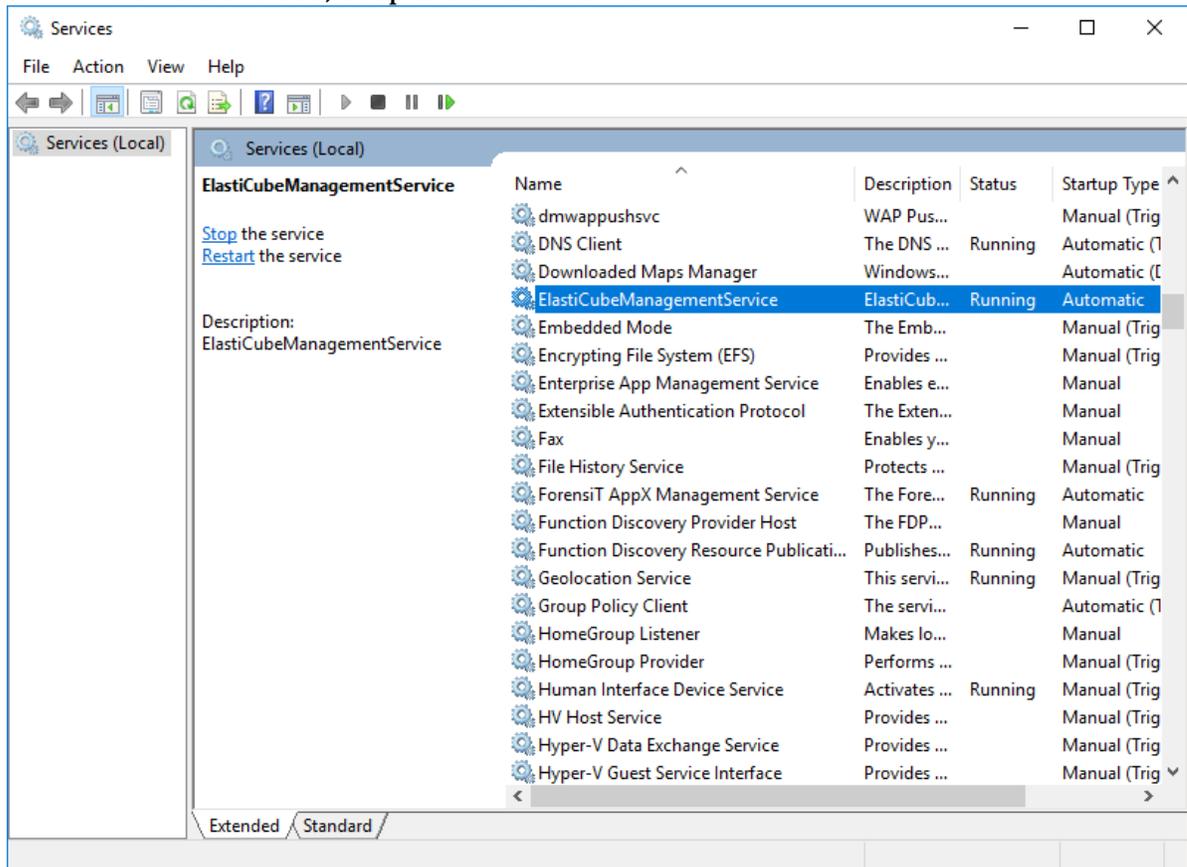
This topic describes how to back up your ElastiCubes, the Sisense Web Application, and any customizations you may have performed.

# Backing Up Your ElastiCubes

Your ElastiCube Server contains information about your ElastiCubes, including the schema and the data that was imported when the ElastiCube was built. This information is saved in a collection of folders stored in the ElastiCube Data folder. The files and folders in the ElastiCube Data must be saved and their structured maintained to ensure your implementation will work as expected on the migrated machine.

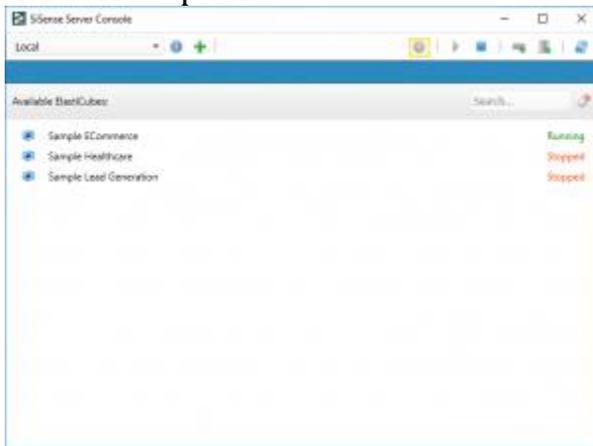
## To back your ElastiCubes:

1. In Windows Services, stop **Sisense.ECMS**.



2. Open the directory:  
`%ProgramData%\Sisense\PrismServer\ElastiCubeData`  
**Note:** This is the default path. To check which path is being used, open the Sisense Server Console and click on the Server Preferences icon  to verify

the correct path.



3. Copy the data to your backup location.

# Backing Up the Sisense Web Application

The Sisense Web Application uses information stored in an instance on the Sisense application database where the Sisense Web Application is installed. This instance stores data about users, dashboards, widgets, and other information. This information must be stored to ensure your Sisense Web Application runs properly.

You can automate the back up process for the Sisense Web Application. For more information, see [Automatically Backup Sisense Web Data](#).

## To back up the Sisense Web Application:

1. In Windows Services, stop the Sisense.Repository and Sisense.Discovery services.
2. Open the directory:  
%ProgramData%\Sisense\PrismWeb\Repository\DB
3. Copy the data to your backup location maintaining the same file structure.
4. Open the directory:  
%ProgramData%\Sisense\Infra\Discovery
5. Copy the data to your backup location maintaining the same file structure.
6. (Optional) Back up your custom plugins by copying the directory:  
C:\Program Files\Sisense\PrismWeb\plugins

# Backing Up Sisense in a High Availability Environment

If you have implemented a high availability environment for Sisense and want to back up your installation, the files described below should be backed up in addition to the files described above in [Backing Up Your ElastiCubes](#) and [Backing Up Your Sisense Web Application](#):

## Build Node(s):

- C:\Program Files\Sisense\Sisense.Orchestration\config\config.json

## Web Node(s):

- C:\Program Files\Sisense\Infra\MongoDB\keyfile. See [Creating a KeyFile](#) for more information.
- C:\Program Files\Sisense\Infra\MongoDB\mongodbconfig.conf
- C:\Program Files\Sisense\Infra\Discovery\conf\zoo.cfg

# Save Custom Configurations

While Sisense preserves your configuration between upgrades, it is a best practice to backup configuration files for any custom configurations you may have in your environment.

The list below includes possible custom configurations that you should back up if you've implemented:

**Note:** Save copies of these files in a safe location.

- **SSL:** If you have configured secure connections to Sisense, you should back up your SSL certificates. See [Setting Up SSL](#) for more information.
- **SSO:** If your users sign in through SSO, you should back up your SSO configuration. See [Single Sign On](#) for more information.
- **Plug-ins/Add-ons:** If you have implemented any custom plugins or add-ons, you should back these up. See Backing Up the Sisense Web Application Step 6 above for more information.
- **Metadata Translations:** If you have translated Sisense metadata, you should back up your translations. See [Translating Sisense Metadata](#) for more information.
- **Rebranded Email Templates:** If you have customized Sisense's automated emails, you should back up your templates. See [Rebranding Sisense Automated Emails](#) for more information.
- **Message Broker:** If you have implemented high availability for the message broker, you should back up the configuration. See [Replicating the Message Broker](#) for more information.
- **FIPS:** If your Sisense implementation is FIPS-compliant, you should back up the MongoDB configuration file. See [FIPS](#) for more information.

## Virtual Machines

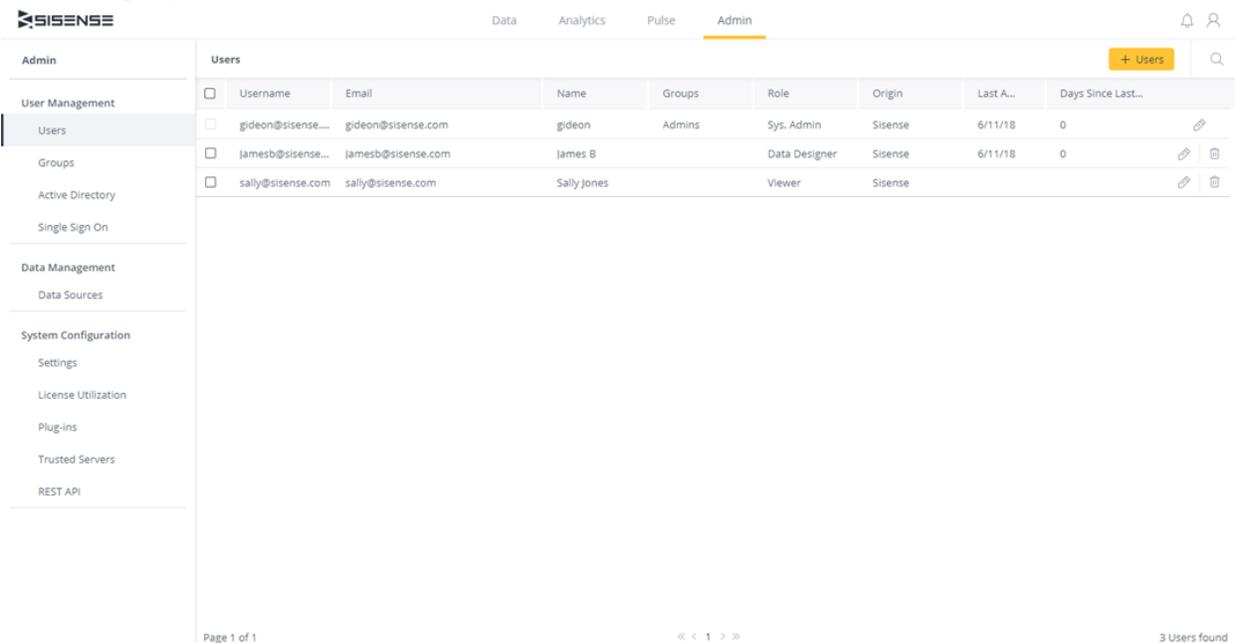
If your Sisense environment is hosted on a virtual machine, you should save an image of your environment and save it in a safe location. If you create an image, you do not need to back up the other custom configuration files described above.

# Save Sisense Assets

In addition to your ElastiCubes, dashboards, and any custom configuration files you have backed up, you should also collect information about your environment so you can verify that your system is fully functional after upgrading Sisense.

It is recommended that you note the number of dashboards, ElastiCubes, users, and groups you have. Having this information will help you later on when you verify the success of your upgrade.

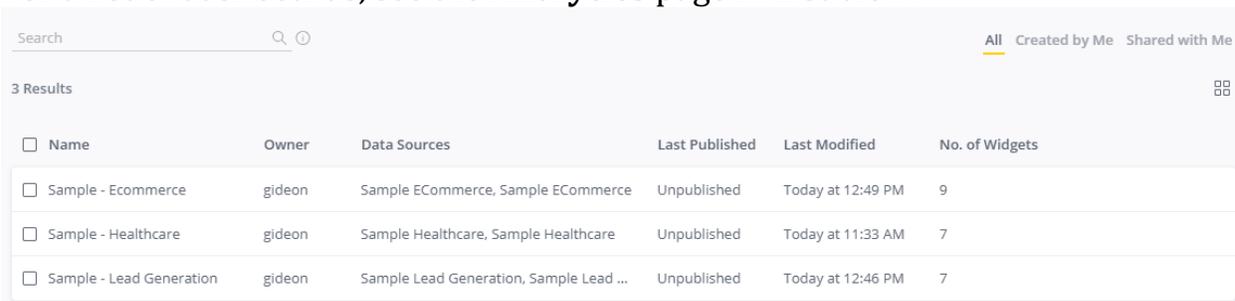
You can see the complete list of your Users, Groups, and Data Sources in the **Admin** page.



The screenshot shows the Sisense Admin interface. The top navigation bar includes 'Data', 'Analytics', 'Pulse', and 'Admin' (which is highlighted). A sidebar on the left contains various management options like 'User Management', 'Groups', 'Active Directory', etc. The main content area displays a table of users.

Username	Email	Name	Groups	Role	Origin	Last A...	Days Since Last...
gideon@sisense...	gideon@sisense.com	gideon	Admins	Sys. Admin	Sisense	6/11/18	0
Jamesb@sisense...	Jamesb@sisense.com	James B		Data Designer	Sisense	6/11/18	0
sally@sisense.com	sally@sisense.com	Sally Jones		Viewer	Sisense		

For a list of dashboards, see the **Analytics** page in List view.



The screenshot shows the Sisense Analytics page in List view. It features a search bar, filter tabs for 'All', 'Created by Me', and 'Shared with Me', and a table of dashboard results.

Name	Owner	Data Sources	Last Published	Last Modified	No. of Widgets
Sample - E-commerce	gideon	Sample ECommerce, Sample ECommerce	Unpublished	Today at 12:49 PM	9
Sample - Healthcare	gideon	Sample Healthcare, Sample Healthcare	Unpublished	Today at 11:33 AM	7
Sample - Lead Generation	gideon	Sample Lead Generation, Sample Lead ...	Unpublished	Today at 12:46 PM	7

## Next Steps

- [Performing Your Upgrade](#)

## Performing an Upgrade

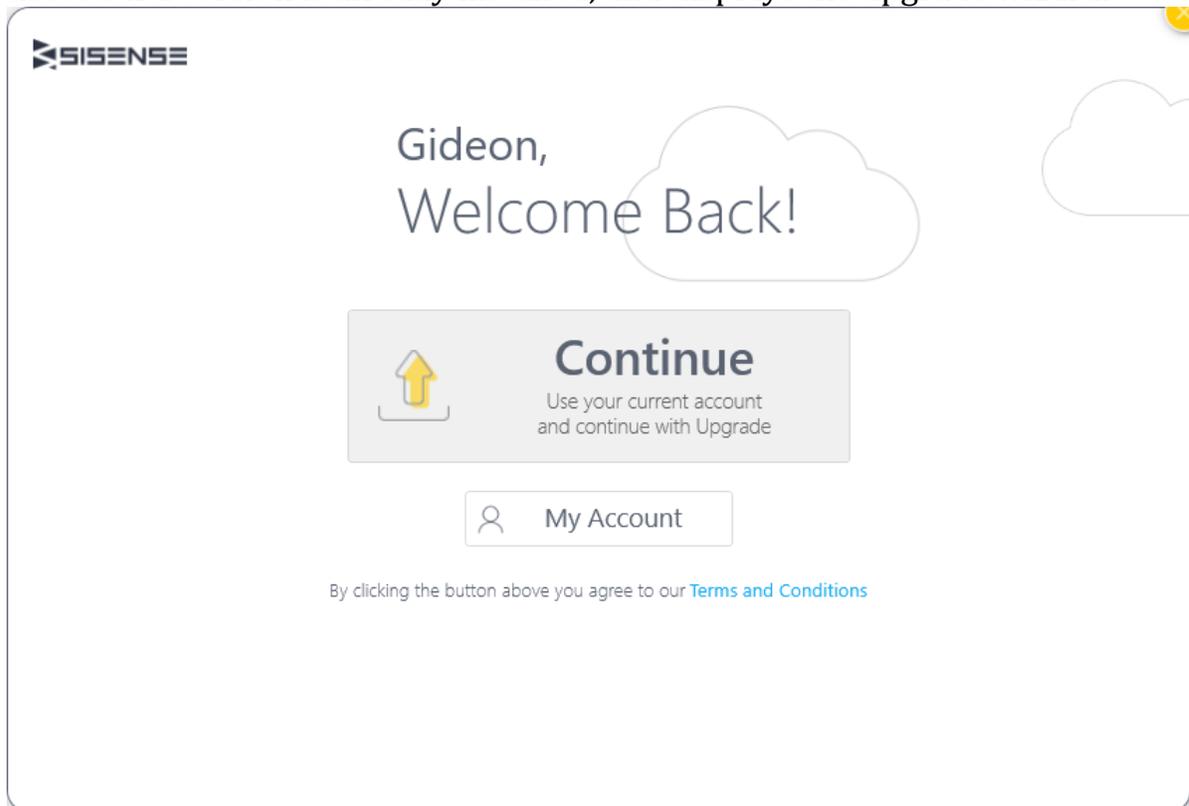
After backing up your Sisense files and configuration, the next step is to upgrade Sisense. If you are hosting a single Sisense server, then you can upgrade Sisense using the Sisense Installer. If you are hosting multiple Sisense servers, you can use the Multi-Node Deployment tool to silently upgrade each of your servers remotely.

**Best Practice:** Upgrade a staging environment so you can know how the upgrade will affect your users and your configuration before upgrading your production environment.

# Upgrading a Single Server Environment

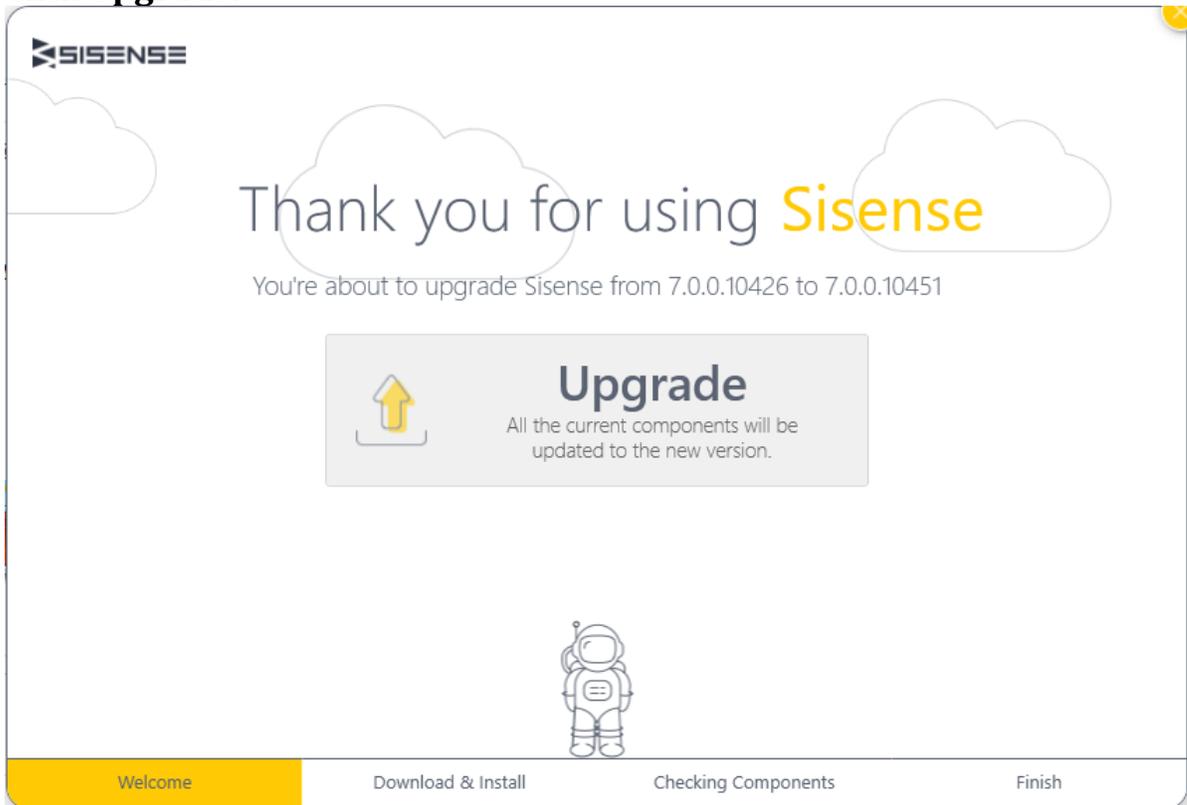
## To install the latest version:

1. Download the latest version of Sisense. For more information, see [Downloading and Installing Sisense](#).
2. Run the installation file as a Windows Administrator. Sisense automatically detects if a version is already installed, and displays the upgrade wizard.



3. Click **Continue**.

4. Click **Upgrade**.



The screenshot shows a Sisense upgrade dialog box. At the top left is the Sisense logo. The main text reads "Thank you for using Sisense" with "Sisense" in yellow. Below this, it says "You're about to upgrade Sisense from 7.0.0.10426 to 7.0.0.10451". A central grey button contains an upward arrow icon and the text "Upgrade" in bold, followed by "All the current components will be updated to the new version." At the bottom center is a small astronaut icon. A progress bar at the bottom shows four steps: "Welcome" (highlighted in yellow), "Download & Install", "Checking Components", and "Finish".

**SISENSE**

Thank you for using **Sisense**

You're about to upgrade Sisense from 7.0.0.10426 to 7.0.0.10451

 **Upgrade**  
All the current components will be updated to the new version.



Welcome      Download & Install      Checking Components      Finish

# Upgrading a Multi-Node Deployment

To upgrade a Multi-Node deployment, you can use the Multi-Node deployment tool. This tool supports Sisense V7.2 and later. The Multi-Node deployment tool remotely installs or upgrades Sisense on your server to the version you select in the tool.

Before proceeding with the installation, ensure that the relevant [communication ports](#) are available on each of the servers. See [Communication Ports for Multi-Node Deployments](#) for more information.

## To upgrade a Multi-Node deployment:

1. Download the Deployment Wizard. See [Installing the Sisense Multi-Node Deployment Wizard](#) if you have not already downloaded and installed it.
2. After installing the Deployment Wizard, the screen below should be displayed. If you have closed the Deployment Wizard, in your Sisense High Availability Configuration directory, run **SisenseHAWizard.exe**.



## Sisense Deployment

Welcome!  
Enter your Sisense credentials to begin

Sign In

[Behind a firewall?](#)

[Contact Support](#)

Sign In

3. Enter your Sisense login credentials and click **Sign In**.



## Sisense Deployment

Select the version to install:

[Contact Support](#)

Back

Next

4. Select the version of Sisense you are upgrading to.
5. Click **Download Agent Installation** to install the Sisense Agent and click **Next**.
6. In the Deployment Wizard, enter your Windows server credentials. These credentials will be used to connect to each of the servers in your environment, so the user name and password should be the same for each

server and the user must have Admin privileges.

The screenshot shows the Sisense configuration interface. At the top left is the Sisense logo. Below it, there is a section titled 'Enter your Windows server credentials' with a note 'The user must have administrator privileges'. This section contains two input fields: 'User Name' and 'Password'. Below this is the 'Server Configuration' section with a note 'You must insert at least 3 servers'. It features a table with columns: 'Server', 'Role', 'Components', and 'Status'. The first row has an 'Add Server' button in the 'Server' column, 'Application Node' in the 'Role' column, 'Web and Query' in the 'Components' column, and a trash icon in the 'Status' column. Below the table is a '+ Add Server' button. At the bottom left is a 'Contact Support' link, and at the bottom right are 'Back' and 'Next' buttons.

7. In the **Server Configuration** area, enter the address of your server in the **Add Server** field.
8. In **Role**, select the role of the server. There are two possible roles, **Application Node** and **Build Node**. If your server is going to support Sisense's web services or act as a query node, select **Application Node**. If your server is going to be the build node where ElastiCubes are built and then distributed to other nodes, select **Build Node**.
9. If you selected **Application Node** as your server's role, you need to define what components it will support. There are three options you can select:
  - **Web**: Supports the web services of Sisense
  - **Query**: Responsible for running queries and hosting ElastiCubes
  - **Web and Query**: The server will support the web application and act as a query node
10. Click **Add Server** and repeat Steps 6-8 for each server you wish to add.
11. After each of your servers has been configured, click **Next**.
12. (Optional) If you have a load balancer, enter its IP address in the **Load Balancer** field. Your load balancer routes requests from your application nodes to your active query nodes to maximize performance and capacity utilization. This ensures communication between your load balancer,

## Sisense, and the nodes in your Multi-Node environment.



### Load Balancer

Add server host

### MongoDB Configuration

MongoDB ReplicaSet

Primary Preferred

Nearest

Select Servers

10.

10.

10.

MongoDB Connection String

Enter the connection string, for example:  
mongodb://username:password@host1:port1,host2:port2,host3:port3/?  
replicaSet=replset1&readPreference=primaryPreferred

Contact Support

Back

Next

13. In the **MongoDB Configuration** area, create a replica set for supporting your application database. This ensures that each application database has the same metadata about ElastiCubes, dashboards, and filters. For more information, see [Creating Replica Sets](#).
14. Click **Next**. Sisense will automatically complete the setup of your Multi-Node environment. Close the Deployment Wizard once the setup is complete. After you have created your environment, the next step is to define how builds are distributed across all your servers. See [Distributing ElastiCube Builds to Query Nodes](#).

## Next Steps

- [Testing Your Upgrade](#)

## Testing Your Upgrade

After you have upgraded your staging environment or your production environment, Sisense recommends you test the following to verify that your upgrade was a success:

1. If you have implemented SSL, access the Sisense server from an external network, using SSL, and ensure the dashboards load as expected.
2. If you have implemented SSO, log in to Sisense using SSO, and make sure that a user can see all of their dashboards.
3. If you use any plugins or add-ons, load dashboards using plugins, and make sure it's loaded correctly.
4. If you have implemented active directory, log in to Sisense using an account from active directory and make sure your user can log in and see all of their dashboards.
5. Create a new ElastiCube and import a data source (Excel or CSV file, or any other)
6. If you have dashboards or widgets using custom scripts, load the dashboard or the widget to make sure they're loaded correctly.
7. Run a build of an existing ElastiCube successfully.
8. Create a new dashboard, and add a Table or Pivot widget. Ensure the widget loads data.
9. If you collected information about your environment, including how many ElastiCubes, dashboards, users, and groups you have, verify that the number of assets in the upgraded deployment is correct.
10. Export a widget to Excel. Ensure the file is downloaded and has the relevant data.
11. Export a dashboard to PDF. Ensure the PDF is created and opened successfully.
12. Send a PDF report of a dashboard through Sisense and ensure it's received.

If you have a multi-node deployment, **in addition** to the list above, test the following as well:

- Shut down one of the query nodes and verify that dashboards return the correct result.
- Verify that ElastiCube distribution works and that the **Last Build** time in the Data Source section of the **Admin** page changes.

## Next Steps

- [Upgrading Your Production Environment](#)

## Upgrading Your Production Environment

If you have tested Sisense in a staging environment successfully, you are ready to upgrade your production deployment. Follow the same steps described in Step 3 to upgrade your production environment. Then, follow the steps described in Step 4 to test your production environment.

- [Performing an Upgrade](#)
- [Testing Your Upgrade](#)

## Licensing Terminology

The following is a list of some of the terms defining license scope or product options that will be reflected on the Sales Order defining the license granted in the Software.

**“Administrator(s)”** means the Authorized User(s) who are Personnel of Licensee who have the authorization and access for server, user and data management.

**“Advanced IT Package”** means a set of features that are bundled in a license option for Authorized Servers, that is licensed hereunder if specified on the applicable Sales Order and that includes:

1. **SSO** is defined at this link: </documentation/configuring-single-sign-ss/>
2. **API** is defined at the following links:
  - **REST API** – <https://documentation.sisense.com/sisense-rest-api/>
  - **SQL API** – <https://documentation.sisense.com/sql-api/>
3. **SDK** means the Software Development Kit, a component of the Software, which allows the development or configuration of the Software.

**“Authorized Users”** means individual users granted access to use the Software on a named basis. Each Authorized User shall receive a personal login and password which shall be maintained securely by Licensee from unauthorized use. The number of Authorized Users licensed hereunder is specified on the Sales Order(s) executed hereunder.

**“Authorized Servers”** means the number of Licensee’s servers on which the Software may be installed as specified in the applicable Sales Order. For purposes of this Agreement, each Authorized Server shall be licensed for a specific number of Rows or Cores as specified on the applicable Sales Order.

**“Backup/Development Server”** means a Software instance that is to be used for the sole purpose of development, backup, staging and other non-production uses and may not be used in any way for production use.

**“Central Monitoring Service”** means a set of features permitting monitoring by Licensee of certain technical aspects of the operation of the Software. The features are bundled in a license option for Authorized Servers, that is licensed hereunder if specified on the applicable Sales Order. The features currently consist of:

- **Sisense Monitor:** a service that collects data about query, build and Authorized Server performance and displays this information in a dashboard to assist Licensee in monitoring and troubleshooting potential issues.
- **Note:** Licensing of Authorized Servers with the Advanced IT Package is a prerequisite for licensing the Central Monitoring Service. The Advanced IT Package includes a license to permit use of the Central Monitoring Service on up to two (2) Authorized Servers. Use of the Central Monitoring Service on additional Authorized Servers must be purchased and reflected on the applicable Sales Order.

**“Core”** shall be a collection of one or more processor threads and a set of shared execution resources. A processor thread is the architectural state within a processor that tracks execution of a software program thread/task.

Hyperthreading and other current and future technologies that materially

expand the processing capacity of a Core shall not be permitted to increase the licensed processing capacity of the Core-based licenses granted under this Agreement unless otherwise agreed by the parties.

**“Dashboards”** means the graphic representation of data extracted from the data source(s) ingested into the Software as designed and created by Designers and/or Administrators using the Software in accordance with this Agreement.

**“Designer(s)”** means the Authorized User(s) who are Personnel of Licensee that are authorized to create, edit and share ElastiCubes and Dashboards.

**“Natural Language Package”** means a set of natural language generation features available for license as an add-on to the Authorized Server licenses as described in the Documentation.

**“Personnel”** shall mean employees and contractors of Licensee where such personnel’s access to the Software is controlled by Licensee.

**“Row”** shall be a single record in a table of a Sisense ElastiCube. For licensing purposes, only Rows in tables that were added to the ElastiCube from external sources are counted (i.e., Base table). If a base table is used by several ElastiCubes, it is counted only once.

**“Viewer(s)”** means the Authorized User(s) who are Personnel of Licensee that are authorized to view and filter the Dashboards that Designers share with them via a standard web browser. An Unlimited Viewer license allows unlimited Viewers to be authorized to access the specified Authorized Server. Please note that while there is no legal limit to the number of Viewers, the technical capacity of the server may limit the number of Viewers that may actually access the Authorized Server.

# Launching the Sisense Web Application

This topic describes how to launch the Sisense Web Application and ElastiCube Manager.

## To launch the Sisense Web Application:

Use one of the following options to launch Sisense:

- If you have Sisense installed on your machine, select Sisense from the **Start** menu.
- From a standard browser, go to the URL of the Sisense environment provided to you by your System Administrator.

## To launch Sisense:

Use one of the following options:

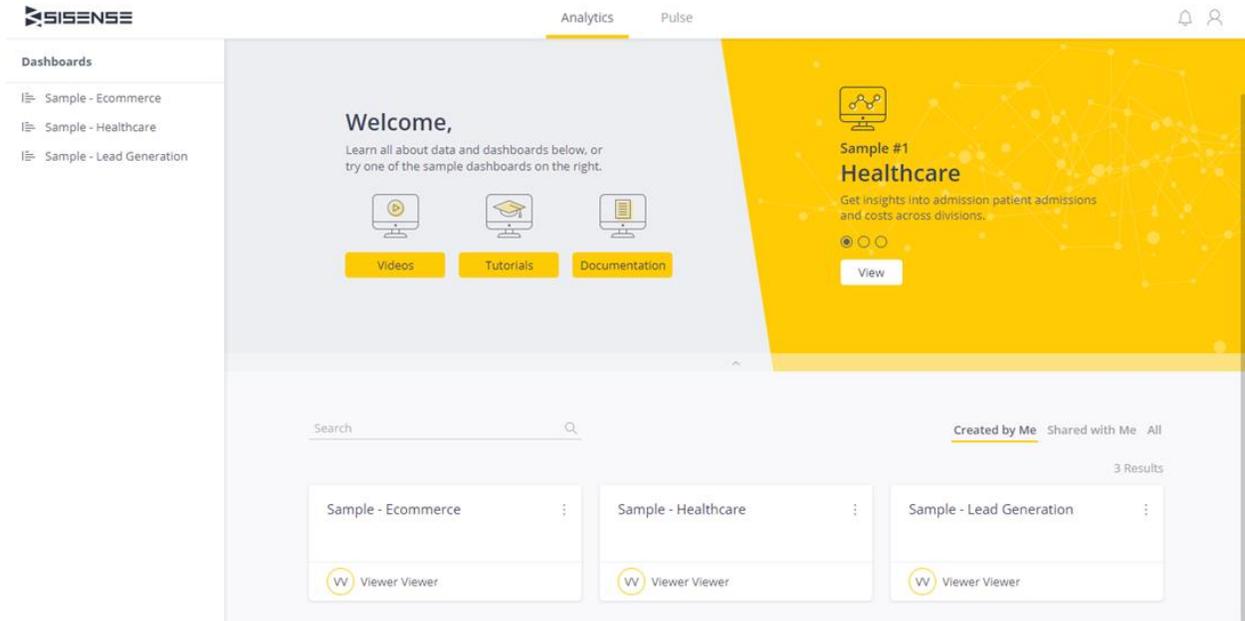
**From Windows:** Open the Windows **Start** menu and select Sisense ElastiCube Manager. This opens the desktop version of Sisense.

**From the Sisense Web Application:** Click **Data** in the top menu. This opens the **Data** page where a list of your ElastiCubes is displayed.

## To log into Sisense:

- If you installed Sisense, you can use the same credentials to log into Sisense.
- If an account has been created for you, you should receive a password activation email. Click on the link to activate your account.

After logging into the Sisense Web Application, the **Analytics** page is displayed.



From the **Analytics** page, you can see dashboards that have been shared with you or some sample dashboards provided by Sisense.

## **Tutorial: Getting Started with Sisense**

# Welcome!

In this basic tutorial, you will learn how to connect to data, and how to build your first dashboard.

The demo data in this tutorial is based on sample E-commerce data, and a common market analysis scenario will be used.

If you have already connected to data sources, you can skip this step and go directly to Step 2 – Your First Dashboard.

## ***Step 1 Connect to Data***

# Introduction

To work with data in Sisense you need ElastiCubes. An ElastiCube model is Sisense's proprietary analytical database, which enables you to connect multiple data sources and run complex queries in split seconds.

You can connect to databases (like SQL Server, MySQL, Oracle, etc.), files (text, csv, Microsoft Access, etc.) and online web services (Google AdWords/Analytics, Salesforce, Zendesk, etc.).

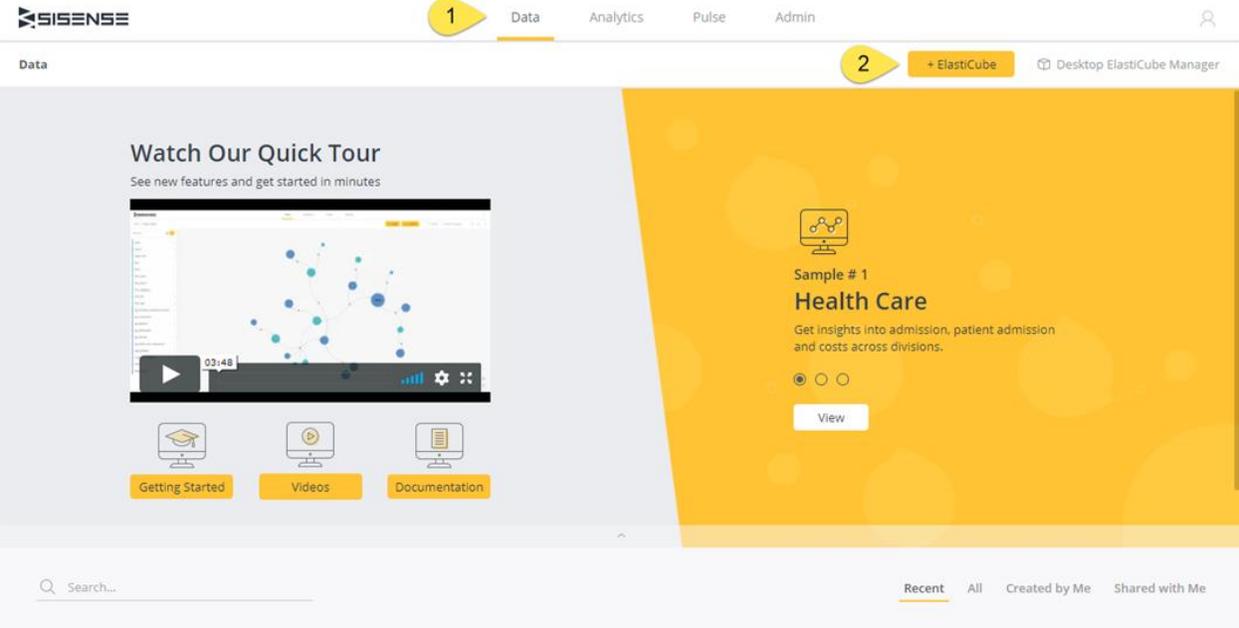
In this tutorial you will create your first ElastiCube model using some sample ECommerce data.

Please download the following two sample files:

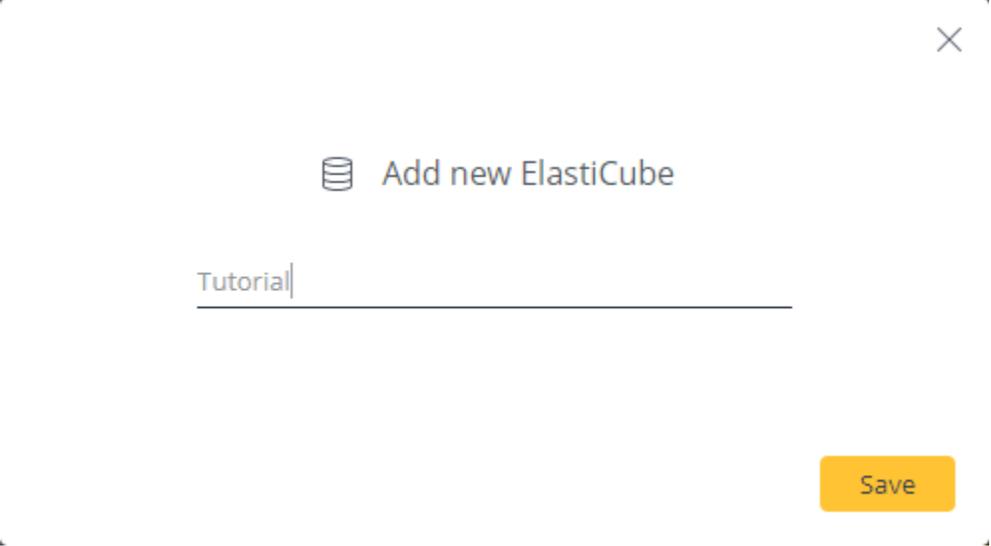
1. [GettingStarted ECommerce.csv](#) – A CSV file with a few hundred thousand entries.
2. [GettingStarted Brands.xlsx](#) – An Excel file with data on brands sold in our ECommerce sample.

# 1. Open the Data Page

Look for **Sisense** in your Windows Start Menu and open it. From the menu bar, select **Data > + ElastiCube**. This creates an empty ElastiCube model. You can also choose to create a Live model where you connect directly to your data source, but for this tutorial, let's stick with an ElastiCube.



Now, give your ElastiCube model the name "Tutorial":



You just created an empty ElastiCube. The next step is to add some data to it.

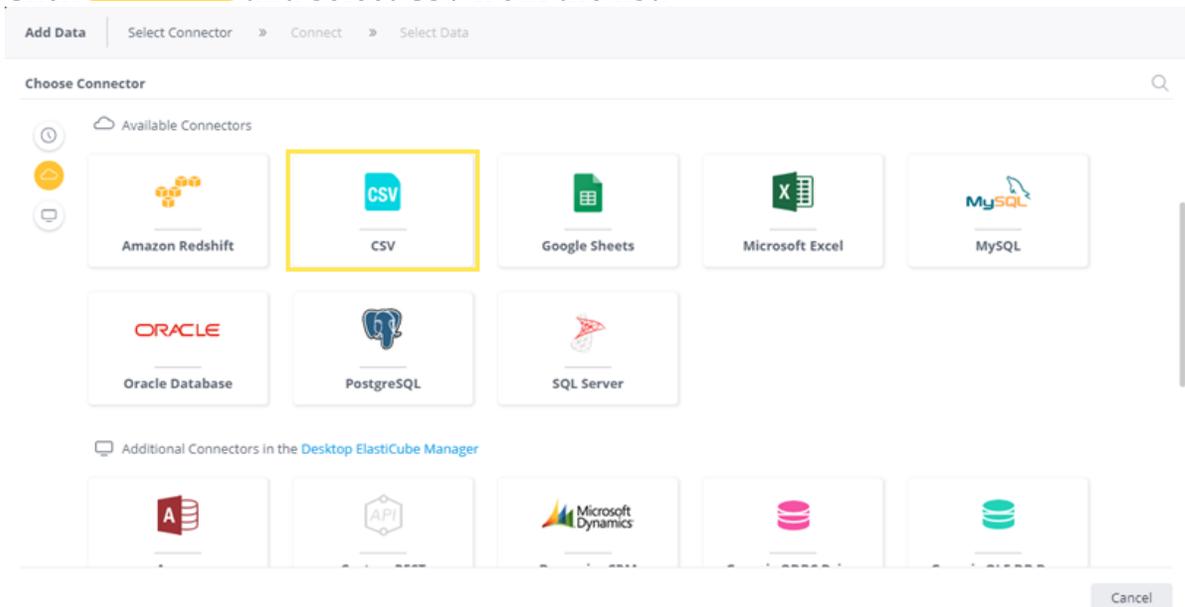
## 2. Add Data Sources

In Sisense, you can import data into the ElastiCube, or connect directly to your data source with a Live connection. With our CSV and Excel files, we will import these into the ElastiCube.

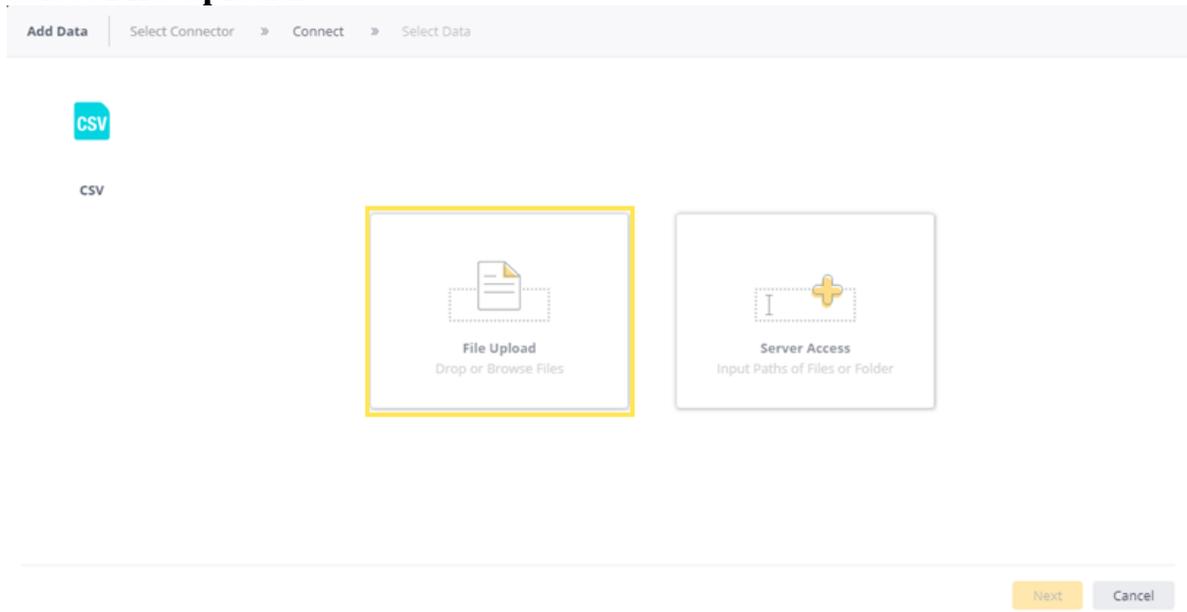
### Table #1 – CSV File

Import the base table, which is in the form of a CSV file. This table contains info about what kind of products have been purchased, and how much they cost.

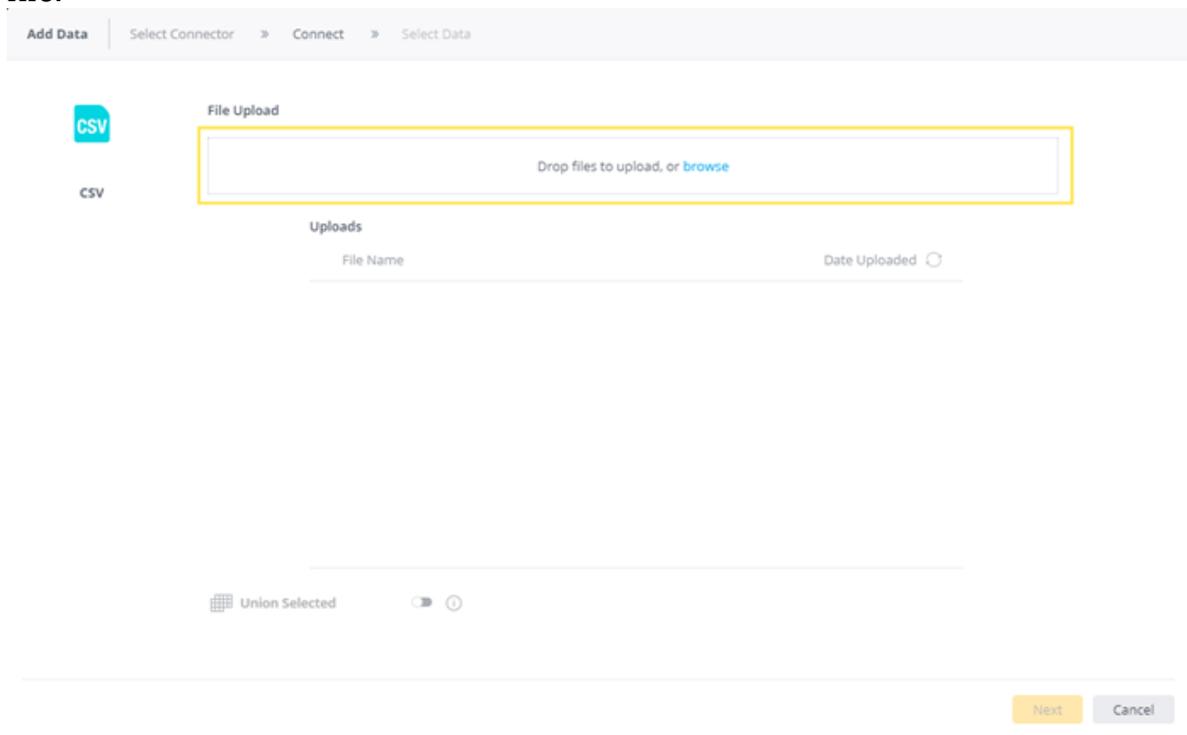
1. Click **+ Data** and select CSV from the list:



2. You will be prompted to choose how you want to upload your CSV file. Select **File Upload**.



3. Drag the GettingStarted.CSV file to the Upload area or click browse and navigate to the file.



4. After you upload the CSV file, select it from the Uploads list and click **Next**:

The screenshot shows a web interface for uploading a CSV file. At the top, there is a breadcrumb trail: "Add Data" > "Select Connector" > "Connect" > "Select Data". Below this, there is a "File Upload" section with a "CSV" icon and a "Drop files to upload, or browse" area. Underneath, an "Uploads" table lists the uploaded files:

File Name	Date Uploaded
GettingStarted ECommerce.csv	2/21/2018

At the bottom of the interface, there are "Union Selected" controls and a "Next" button. A yellow callout with the number "1" points to the file name in the table, and another yellow callout with the number "2" points to the "Next" button.

5. Select the GettingStarted Ecommerce file from the Select Table list. You can preview the data inside it by clicking . This displays some more settings

you can use to customize your data, but for now, click **Done**.

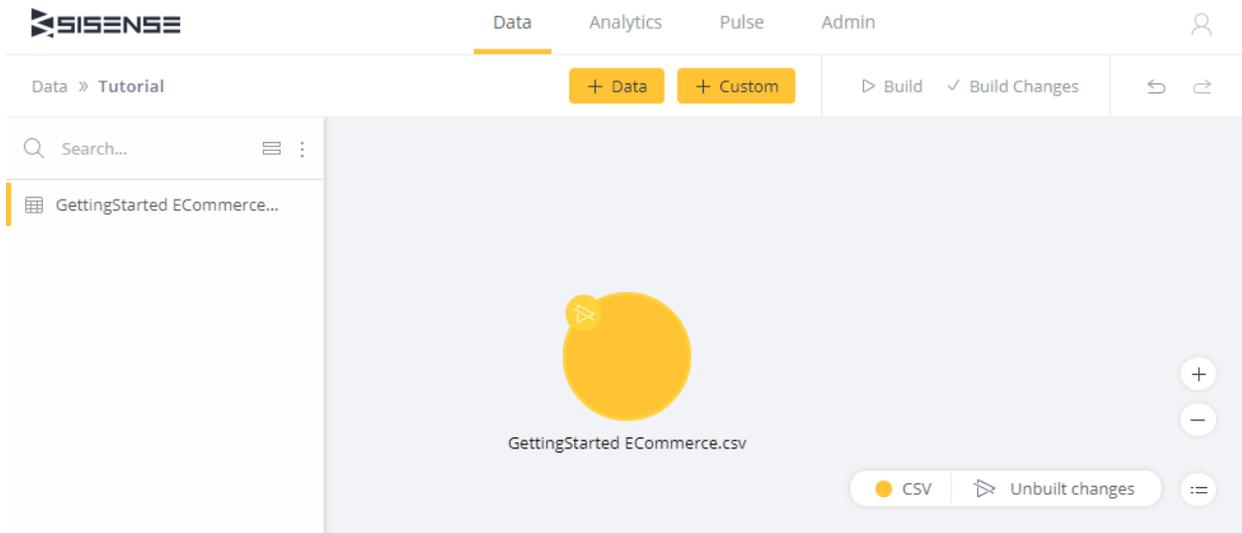
The screenshot shows the 'Select Data' interface. At the top, there are navigation steps: 'Add Data', 'Select Connector', 'Connect', and 'Select Data'. Below this is a 'Select Table' section with a search icon and a table listing data sources. The first table source is selected and highlighted with a yellow box: 'GettingStarted ECommerc...'. Below the table list is a 'Settings' panel with various options:

- Culture: English (United States) (dropdown)
- Text Qualifier: (input field)
- First Row Contains Field Names: (checkbox, checked)
- Delimiter: Comma (,) (dropdown)
- Ignore Rows that Start ...: (input field)
- Ignore Rows that Contain: (input field)
- Ignore First Rows: (input field)

At the bottom right of the interface, there are two buttons: 'Done' (highlighted in yellow) and 'Cancel'.

Congratulations, you have connected to your first data source. You can now see the CSV file in your schema.

The name of the table in the ElastiCube model defaults to the file name in this case. You can double-click the title to rename it and remove the CSV extension. Your ElastiCube model should look like this:



## Table #2 – Excel File

Information about the brands that were sold in the main table exists in another table, this time in an Excel file.

1. Click **Add Data** and select **Microsoft Excel File** from the list.
2. You will be prompted to select an Excel file. Find and select “GettingStarted Brands.xlsx”, which you downloaded in Step 1 (you can also download the file [here](#)).
3. In the new window leave the default settings and click **Add**.

Congratulations, you have now added your second table, an Excel file.

To rename the table, click the table once and select and rename the table to “Brands”.

Your ElastiCube model should look like this:

Search...

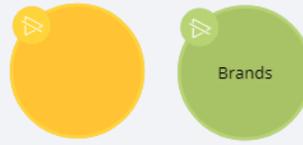


Brands

Brand ID

Brand

GettingStarted ECommerce...



GettingStarted ECommerce.csv



● CSV

● Excel

U Unbuilt changes



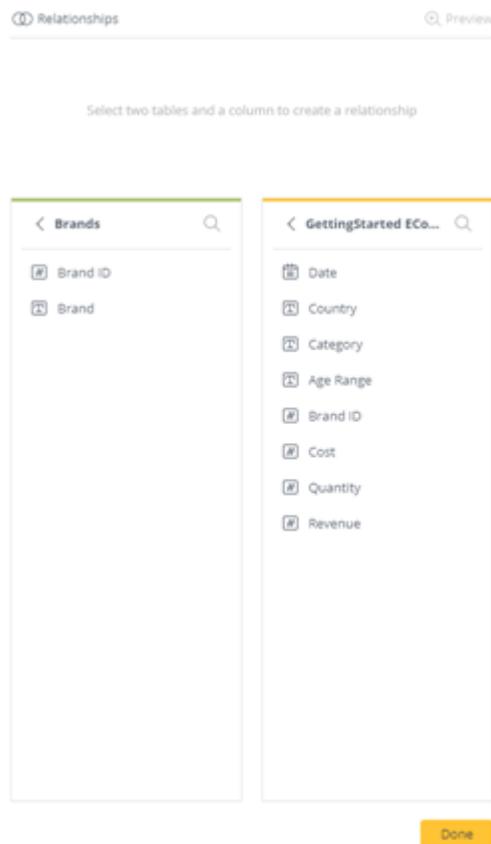
### 3. Connect Different Data Sources

So far you have added two tables from two different data sources. Now let's see how to connect them.

In the ElastiCube model, creating and deleting relationships between tables is as simple as drag and drop.

Any two fields of the same type (numeric/text/date) can be connected. In this sample, both tables have a "Brand ID" column. Let's connect them:

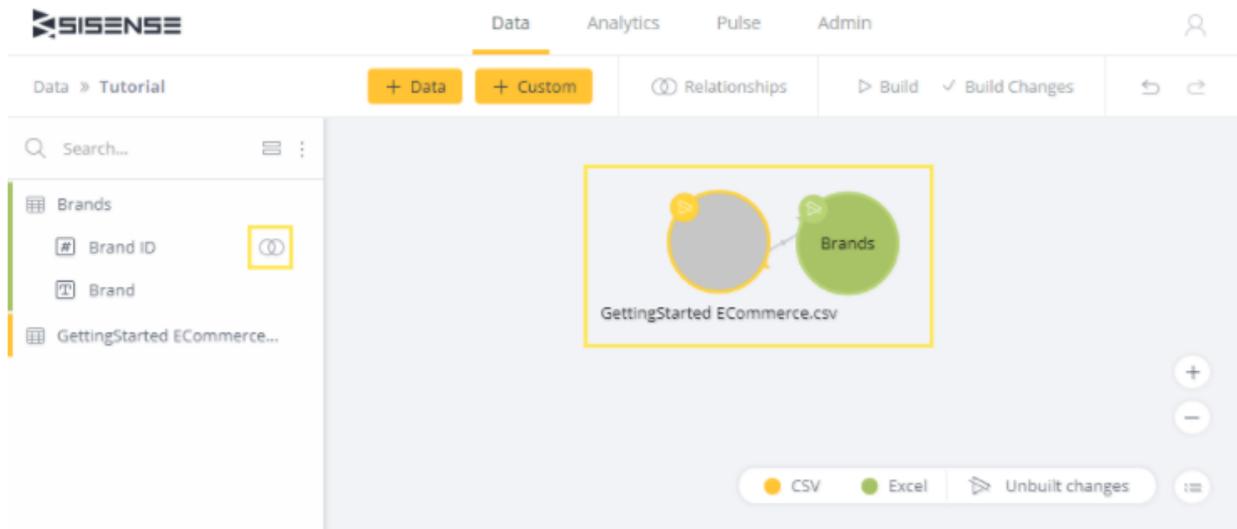
1. Drag the Brands table onto the Ecommerce table. This opens the Relationships pane.



2. Select **Brand ID** from both tables and click **Done**. You should now see the two tables are joined through the column Brand ID.

## 4. Build the ElastiCube

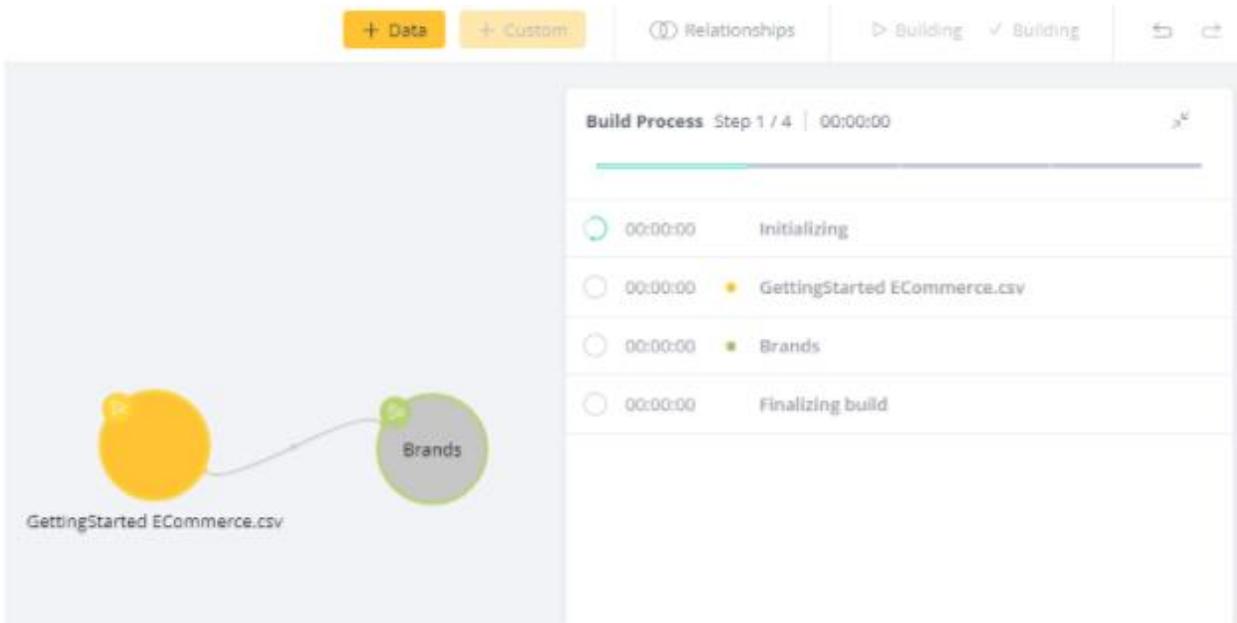
Now that you have defined your ElastiCube model, you are ready to build it. This will pull the data from the data sources into the ElastiCube model.



Click **Build** in the top menu.



The build will start. You will see the progress in Build log. Wait for the build to finish:



Congratulations! You have successfully built your first ElastiCube model.

## 5. Dashboard Time!

Click **Analytics** in the top menu to open the Sisense Web Application and create a new dashboard.

Data **Analytics** Pulse Admin

### *Step 2 Your First Dashboard*

Welcome to Step 2 of the Sisense tutorial. In Step 1 – Connect to Data of this tutorial, you connected to data and built your first ElastiCube model. Now, you are ready to jump right into creating your first dashboard.

Although it is recommended to complete Step 1 in advance, you can jump right into Step 2 if you wish.

If you're arriving directly from Step 1, you should already be in the process of creating a new dashboard after clicking Analytics in the top menu of Sisense:



To Create Your  
Dashboard  
[Click Here](#)

# Introduction

Sisense is designed to let business users easily create powerful and meaningful dashboards, without being a developer or data scientist.

To get a feel for the process, Sisense has put together this tutorial which shows you how to do a quick market analysis to identify what products are meeting customers' needs, and in which markets. You will use appropriate visualizations to create a dashboard that provides answers to four questions:

1. **Sales Trend:** Is annual revenue increasing?
2. **Optimal Product Mix:** What product lines should we grow?
3. **Segmentation:** Which customer segments should we develop?
4. **Market Size and Growth:** Where are our best markets located?

To make things interesting, your data set is based on sales and customer data from an E-commerce site.

# 1 – Create a New Dashboard

1. Click **+** to create a new dashboard on the **Analytics** page.



2. In the **Data Set** field, select the ElastiCube model to which you want to connect. If you've completed [Step 1 – Connect to Data](#), select 'Tutorial' in the Data Set field, otherwise choose 'Sample ECommerce'.
3. A default title name matching the ElastiCube model name will be given to the dashboard. Feel free to change it.

New Dashboard / Please select a Data Set to connect to

---

Data Set: Tutorial

---

Title: Tutorial

Create

Cancel

4. Click **Create**.  
Now that you have a dashboard, you can start adding widgets to it.

## 2 – Create Your First Widget: Sales Trends

Create your first widget to get a general idea of your sales performance.

1. Click Select Data on the left side of the screen:

Welcome to your new dashboard!

To create a new widget, first select the data you would like to visualize.

### New Widget

+ Select Data

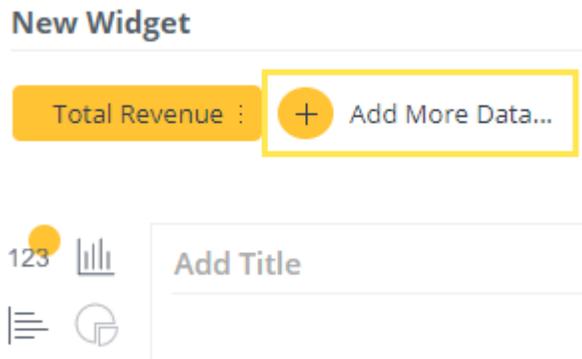
2. A list of available fields will be displayed grouped according to the table they belong to. This window is called 'Data Browser', and lets you easily find the fields available in your ElastiCube model. Select the 'Revenue' field:

The screenshot shows the 'New Widget' interface. On the left, there is a sidebar with a yellow button labeled '+ Select Data'. The main area is titled 'Tutorial / Add a Field' and contains a search bar with the placeholder text 'Type to search for fields'. Below the search bar, the available fields are grouped into two sections: 'Brands' and 'GettingStarted ECommerce.Csv'. The 'Brands' section lists 'Brand' (with an 'A' icon) and 'Brand ID' (with a '#' icon). The 'GettingStarted ECommerce.Csv' section lists 'Age Range' (with an 'A' icon), 'Brand ID' (with a '#' icon), 'Category' (with an 'A' icon), 'Cost' (with a '#' icon), 'Country' (with an 'A' icon), 'Date' (with a calendar icon), 'Quantity' (with a '#' icon), and 'Revenue' (with a '#' icon). The 'Revenue' field is highlighted with a yellow box.

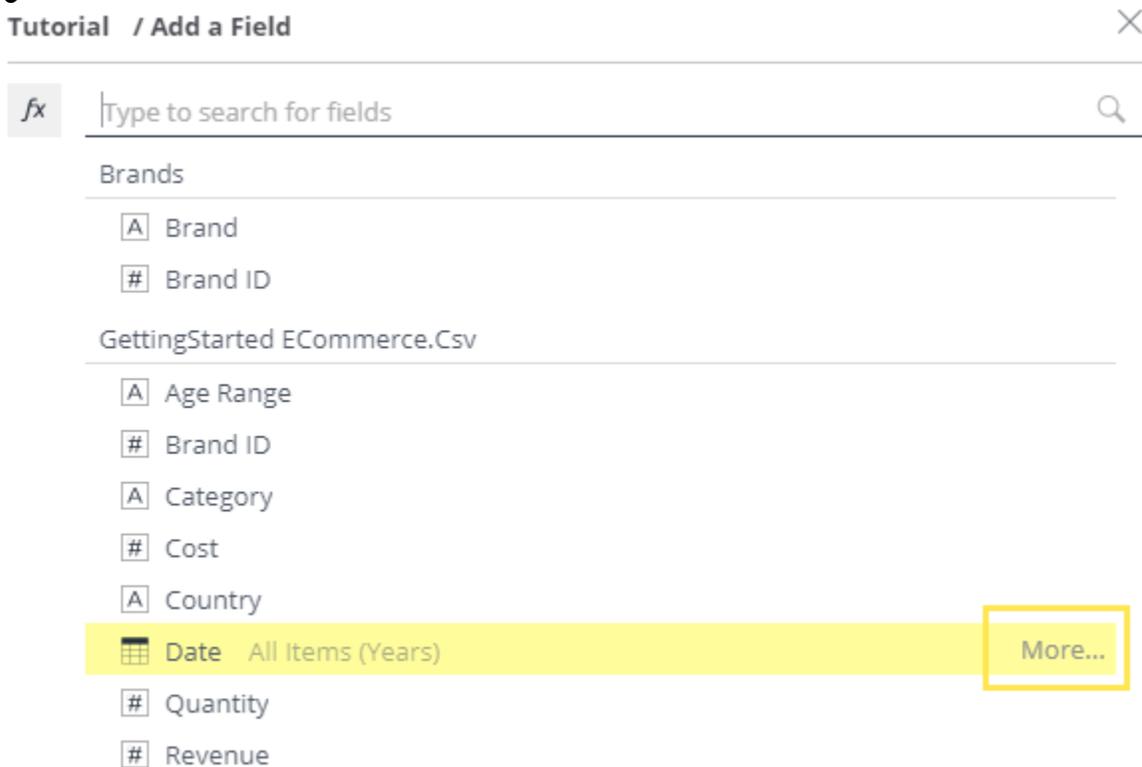
Now you immediately get a ready made widget that shows you all revenue to date. We call this kind of widget an “Indicator”.

But to understand revenue over time, add a time field to the mix:

1. Click **+ Add More Data:**



2. In the data browser, hover over the **Date** field and click **More...** Then select **Quarters:**

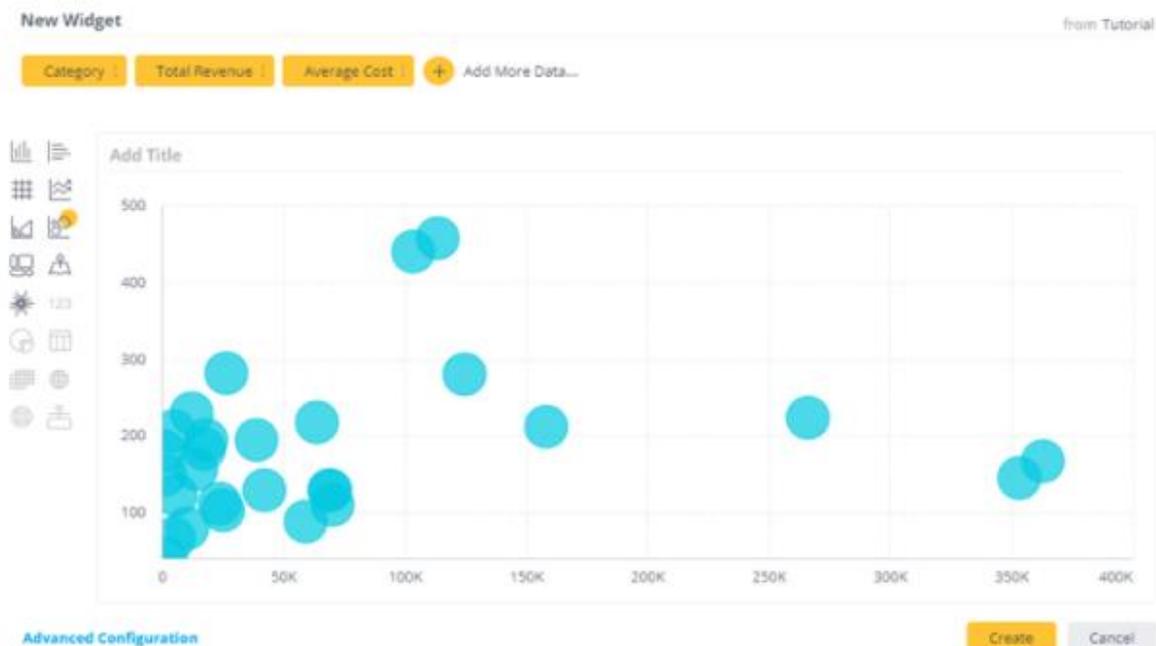


3. Click **Create** to add the line chart widget to the dashboard. The revenue trend is also positive showing annual growth, so now go ahead and explore what’s driving this growth. It would be interesting to uncover profitable product categories.

## 3 – Create a Scatter Chart to See Cost and Revenue

Now try to understand the interaction of cost and revenue to discover the most profitable product lines.

1. On the top of the dashboard click .
2. Click **Select Data** and select the 'Category' field.
3. Click **Add More Data** and select the 'Revenue' field.
4. Click Add More Data again and this time hover over the 'Cost' field and click **More... > Average**.
5. Now select the scatter chart icon  from the widget buttons on the left, and click **Create**.



It's simple to see that product categories on the right generate the most revenue and those products in the lower left have the lowest cost and revenue. Hover over each scatter point to see the category behind it, to discover that Cell Phones and PDAs generate most of the revenue. You now understand which product lines to sell, but to get a complete view of your market, you need to identify optimal customer segments to target in each market – start by creating segments based on age.

## 4 – Creating Pie Charts to Visualize Customer Segmentation

1. On the top of the dashboard, click **New Widget**.
2. Click **Select Data** and select the 'Age Range' field.
3. Click **Add Data** and select the 'Revenue' field.
4. Now select the pie chart icon  from the widget buttons on the left, and click **Create**.



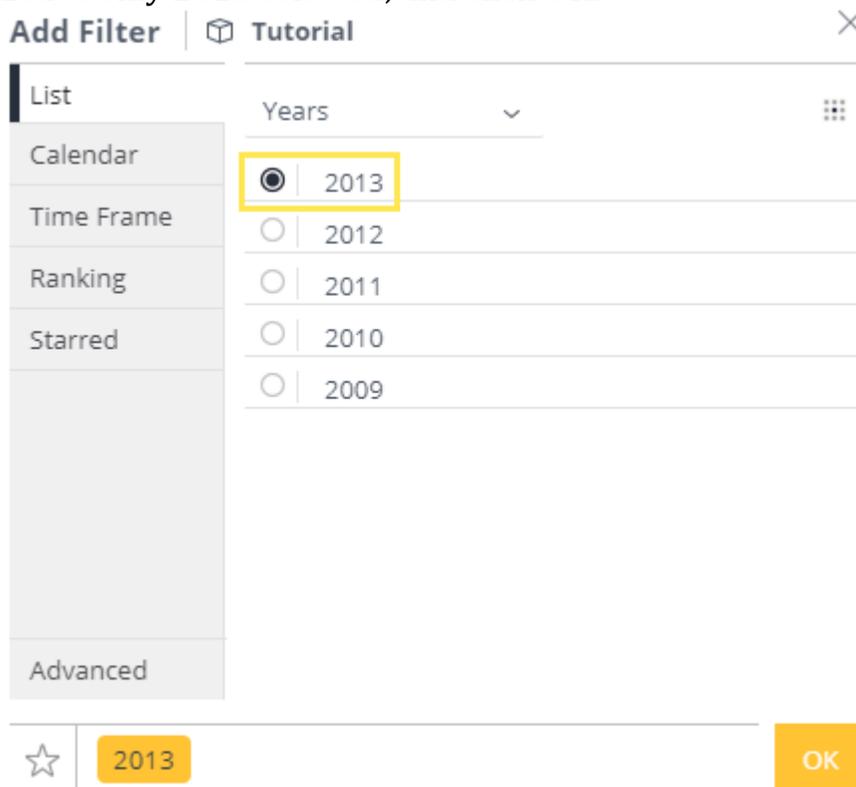
You can clearly see that customers between the ages of 65+ are the most valuable in terms of the revenue they generate. This insight can be used to focus advertising budgets and develop unique campaigns to grow these customer segments. Now, you can turn your attention to identifying which markets are growing at the quickest rate.

## 5 – Creating a Map to See Market Growth

Before you focus on growth, you should focus on a specific year first, by adding a filter.

The **Filter** panel on the right of the dashboards lets you add dynamic filters to the dashboard.

1. Click **Filter Your Dashboard** on the right side of the dashboard.
2. In the Data Browser, click on the **Date** field.
3. Click the multi/single selection button  to make it a single selection filter.
4. Leave only 2013 selected, and click **OK**.

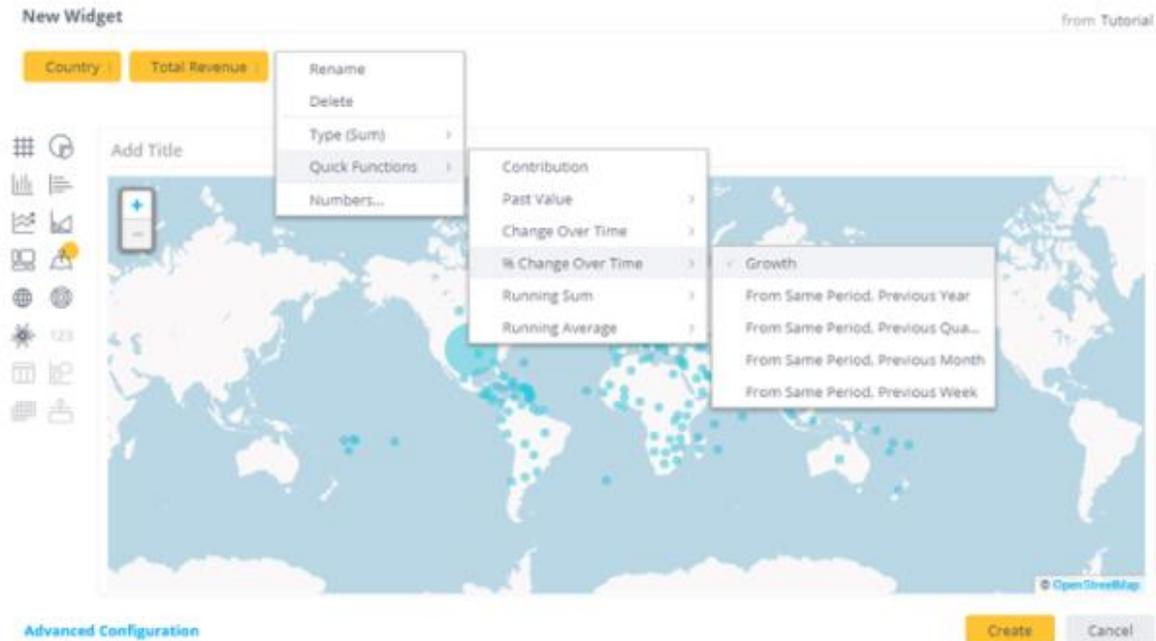


Your dashboard is now filtered to show data only for the year 2013. Now create a map:

5. Click  at the top of the dashboard.
6. Click **Select Data** and select the **Country** field.
7. Click **Add More Data** again and select the **Revenue** field.
8. Click on the scatter map button type  on the left.  
Now you see a map, where the biggest circles represent the countries with

the highest revenue. But if you want to concentrate on the markets with the biggest growth in revenue, its easy:

9. Hover and click to open the menu of the Total Revenue field and select **Quick Functions > % Change Over Time > Growth**.



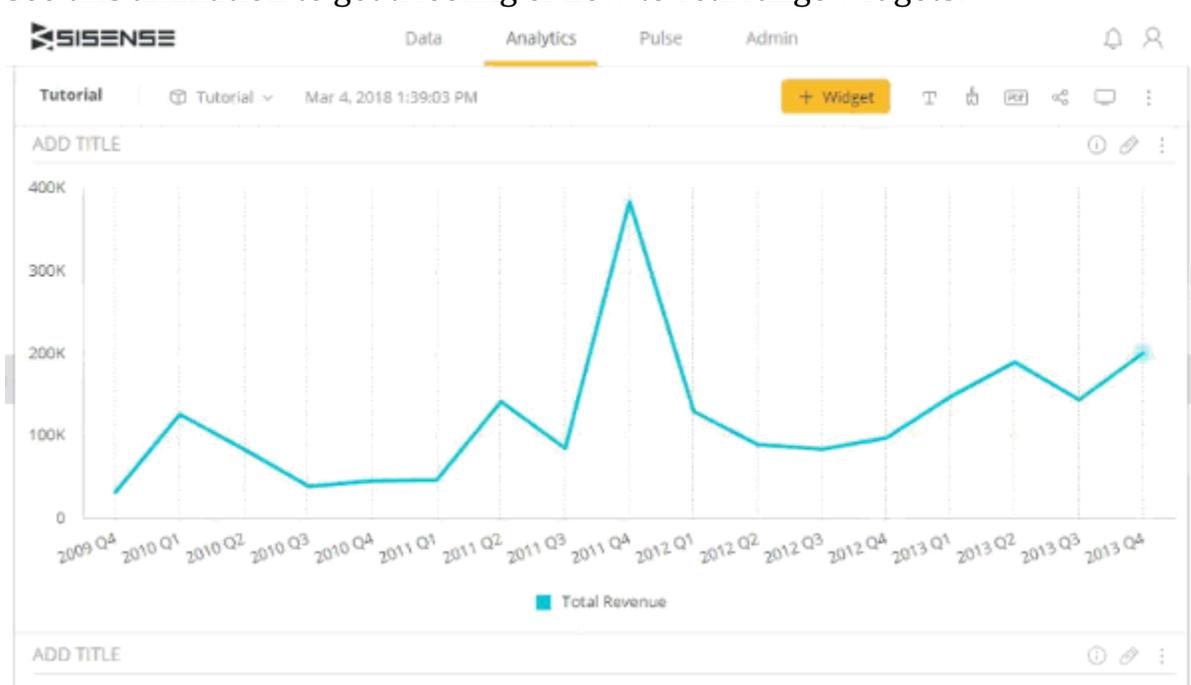
10. Click **Create**.

Now the biggest circles on the map represent countries with highest growth. You can see for example that the United States is very hot right now. You can use this information to focus resources as well as further analyze these markets to understand revenue drivers.

## 6 – Rearrange the Widgets in the Dashboard

You can rearrange the dashboard by dragging and dropping, and resizing the widgets. Dragging one widget on top of another splits the area in the dashboard and allocates space for both widgets.

See this animation to get a feeling of how to rearrange widgets:



Go ahead and rearrange your dashboard however you see fit.

You can now easily see all four visualizations showing sales trends, product mix, customer segments and fast growing markets. You can explore and drill into the details to get further insight.

## 7 – Filtering and Drill Downs

In our pie chart we can clearly see that our strongest age range is 65+. Let's explore this age range a little more:

1. Click the 65+ slice in the pie chart.
2. As a result, a new filter is added to the filter panel on the right, and the entire dashboard is now filtered to focus on this age range.
3. Taking a look at the updated scatter chart, you can see that Monitors are by far the most revenue generating category for this age range.
4. Right-click the 65+ slice in the pie chart and select **Drill Into...** from the menu.
5. A Data Browser will pop up, letting you select any field to drill into. Click on the **Brand** field.
6. You can now see which brands contribute the most to the revenue of your selected age range.  
You can make direct selections by dragging the mouse over the points that interest you in the scatter and line charts as well.

## 8 – So What Did We See Here?

The dashboard you created enables you to interactively analyze three market priorities – where to sell, what to sell and who to sell to. More importantly it makes it simple to see the interaction of product categories, client segments and revenue, to optimize marketing and sales activities and track performance targets.

This was just a quick glimpse of what you can do. Now it's time to start exploring on your own.

### Manage Data

This section describes data models in Sisense, how to connect to data sources, and how to avoid common mistakes when preparing your schemas.

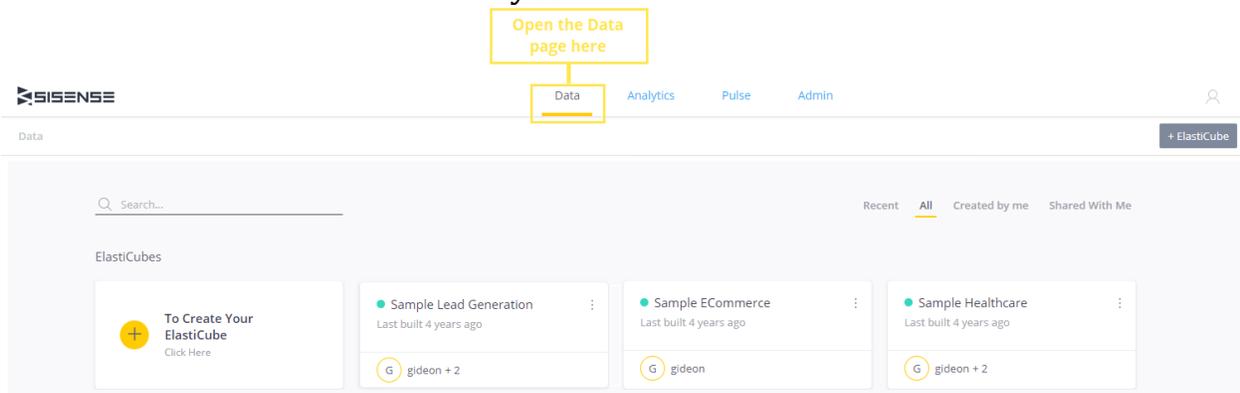
#### Data Models

In Sisense, there are two types of data models, ElastiCube models and live models. ElastiCubes are Sisense's proprietary, high-performance analytical database specifically designed to withstand extensive querying typically required for your business intelligence application. With ElastiCube models, you import data from a variety of data sources into a Sisense ElastiCube, which becomes the database that supports your dashboard. These models then need to be updated as the data in your sources changes.

Unlike ElastiCube models, live models run queries directly against the data source. Sisense performs no additional caching on the data. This provides you with near real-time data updates in your dashboard, though live connections rely on the source database for all queries. This means the queries are only as fast the data source.

In Sisense, Designers create data and live models to represent how data is brought into Sisense and how should it be used when supporting your dashboards. Which model you choose is determined by what you want to do with your dashboard. ElastiCube models tend to be faster when you are working with complex dashboards with large data sets while live models are preferred when you need near real-time updates. ElastiCube models and live models can be used together in a single dashboard to support both use-cases.

From the **Data** page, you can see a list of all the data models that you have created or that were shared with you.



## Related Topics

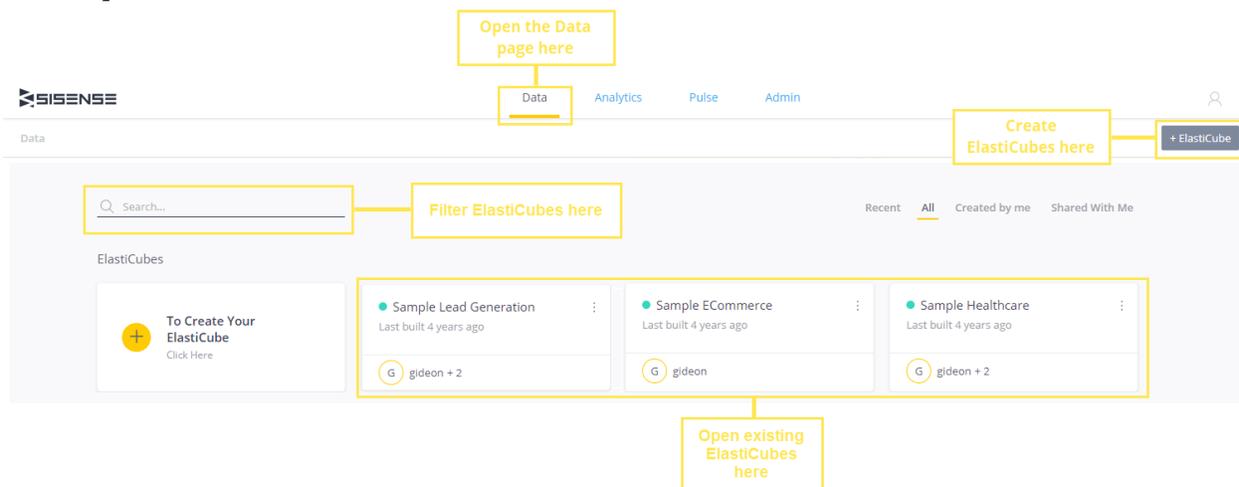
- [Navigating the Data Page](#)

# ElastiCubes

With ElastiCubes, you connect to your data sources and import your data. Once your data is in Sisense, you can then design your ElastiCube model in the Model Editor.

After you have designed your schema, it's time to build the ElastiCube. Building an ElastiCube takes all your data from all your sources, and imports it into the ElastiCube.

The topics below describe ElastiCubes and how to model them.



## Related Topics

- [ElastiCubes](#)
- [Creating ElastiCubes](#)
- [Navigating the Model Editor](#)
- [Introduction to Data Sources](#)
- [Working with Data](#)
- [Transforming and Enriching Data](#)
- [Functions Overview](#)
- [Building ElastiCubes](#)
- [Build Settings and Data Accumulation Behavior](#)
- [Elasticubes for Advanced Business Scenarios](#)

# Live Models

To create dashboards built on live connections to a data source, you create live models. These data models include connection and credential details to the data sources. After you have created your live model, you publish it. Publishing the live model adds it to your list of data models from which you can select when working with dashboards.

The topics below describe live models, and how to create and publish them.

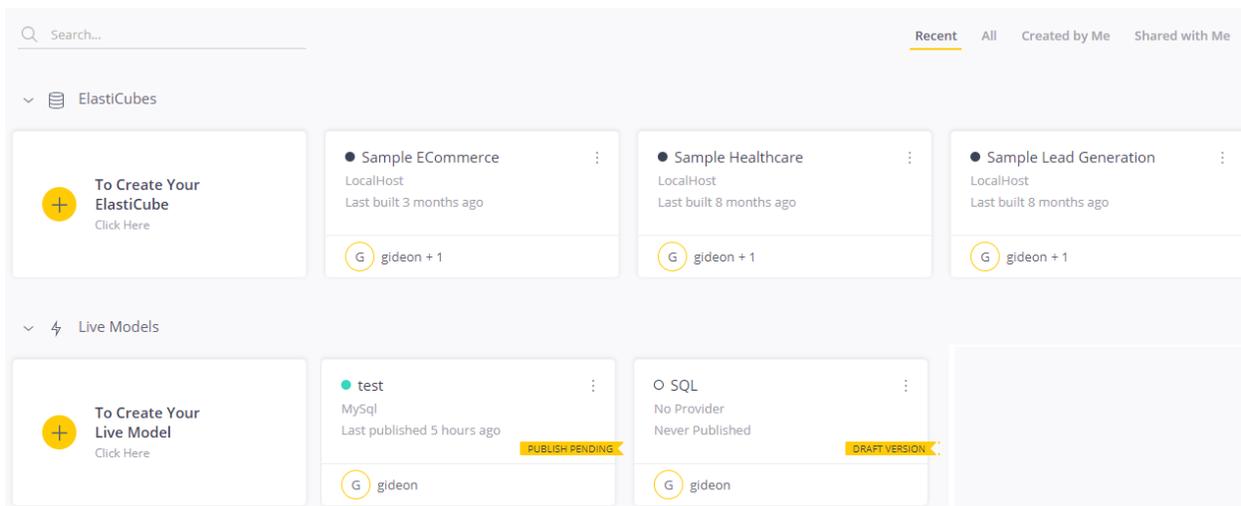
## Related Topics

- [Sisense Live Connect](#)
- [Creating Live Models](#)
- [Adding Live Connections](#)
- [Working with Live Models](#)
- [Publishing Live Models](#)
- [Working with Live Widgets](#)

# Navigating the Data Page

From the **Data** page of ElastiCube Manager, you can manage your ElastiCubes and view their current states.

The **Data** page contains a list of all the ElastiCubes and live models you have created or that have been shared with you.



To locate Elasticubes or live models easily, on the right side of the page, you can view Elasticubes and live models you recently worked on, that were created by you, or that were shared with you. In addition, on the left side, you can locate an Elasticube through the **Search** field.

Each tile contains information about the status of the Elasticube and live model, its owner, and who its shared with it. You can open your Elasticube by clicking **:** on its tile or you can manage your Elasticubes by clicking and selecting one of the following options:

- **Stop:** Click to stop the Elasticube. Elasticubes that have been stopped cannot be queried. This is useful for troubleshooting Elasticubes and issues with your dashboard.
- **Duplicate:** Click to create a copy of a live or Elasticube model. See [Duplicating Data Models](#) for more information.
- **Restart:** Click to restart an Elasticube that you have stopped.
- **Dependencies:** Click to see what entities are using the Elasticube or live model. Entities include data security rules, drill hierarchies, dashboards, widgets (from other dashboards), and Pulse alerts.
- **Schedule Build:** Click to schedule builds to automatically synchronize with the underlying data source to ensure all data is up to date. See [Scheduling Builds](#) for more information.
- **Delete:** Click to delete an Elasticube or live model. If the Elasticube or live model does not have any entities dependent (dependencies) on it, then you will be asked to confirm that you want to delete the Elasticube. Otherwise, you will be prompted to remap your dependencies to another data source. See [Deleting Elasticubes](#) for more information.

- **Cancel all Queries:** Click to cancel all current queries to the ElastiCube.
- **Go to Admin:** Opens the Data Source tab in the **Admin** page where you can manage your ElastiCube or live model.

Some features and functionality supported in the desktop version of Sisense are not supported Sisense currently. These features will be added in the near future. In the meantime, you can find workarounds for missing features at [Sisense V7.0 Backwards Compatibility](#).

## ElastiCubes

# Introduction

The ElastiCube is Sisense's unique, high-performance analytics database with super-fast data stores that are specifically designed to withstand extensive querying typically required by business intelligence applications.

ElastiCubes allow you to bring in data from multiple sources, and then merge, manipulate and query the data as if it was one consolidated data set. ElastiCubes perform so well, that in most cases the creation of dedicated OLAP cubes and/or optimized data marts are completely unnecessary – even when dealing with hundreds of millions of rows of raw data.

One of the biggest advantages of ElastiCubes is the ability to easily mash up multiple data sources. It is made up of fields where each value in one field has a corresponding value in another field. The data for an ElastiCube can come from one source, multiple sources or even from multiple physical locations. Once the data is inside the ElastiCube, it is all the same and every field coming from every table can be analyzed in the context of any other – quickly.

# Benefits

ElastiCube technology make queries over hundreds of millions of rows of raw data return in seconds, with moderate hardware requirements including standard desktop-class computers with commodity hardware. More importantly, ElastiCubes can do this without having to pre-aggregate and pre-calculate the data ahead of time and store it on the hard-drive, thus radically reducing required import/processing time and storage space.

ElastiCubes are most useful when one or more of the following is true:

- Large amounts of data need to be analyzed
- Data for analysis originates from multiple disparate sources

## ElastiCubes – Technical Overview

Relational databases (RDBMS) like SQL Server, Oracle, MySQL and even Access all store tabular data row-by-row. This structure is best for transactional/operational systems that require large numbers of concurrent insertions. With indexes, it can also provide realistic query response times for row-based queries that do not frequently require aggregations or joining of many tables.

Data analysis often requires aggregation of data as well as merging of data located in multiple disparate tables. When dealing with these types of queries, relational databases reach their limits pretty quickly. The only way to extend these limits is by putting in stronger hardware and pre-aggregating data to reduce the amounts of calculations that occur in real time.

# The ElastiCube Columnar Database

ElastiCube data is held in a Columnar Database Management System (CDBMS) that stores data field-by-field. Each field is individually stored in a memory-mapped file, the same mechanism the Windows operating system pagefile uses for memory dumping and loading.

When a query is executed over an ElastiCube, only fields referenced in the query need to be loaded into memory. This leaves enough space for actually processing the query entirely in memory without any read/write to the hard-drive – the prime reason for poor performance of queries. Once a field is no longer used, it is removed from memory and its consumed space is freed.

This approach has several advantages:

## **Query Response Time**

Queries over data sets containing millions of rows of data return in seconds even under modest hardware configurations such as desktop computers.

## **Materialization Time**

ElastiCubes do not require pre-aggregations and/or creation of indexes to assure fast query response, therefore the actual creation of an ElastiCube takes a fraction of the time of a data mart or an OLAP cube.

## **Storage Space**

Pre-aggregations and the creation of indexes are not needed to assure fast query response, making an ElastiCube's size significantly smaller than a datamart or an OLAP cube.

## **High Compression**

This columnar storage strategy makes the data much more suitable for high levels of compression, without loss of detail or accuracy. This means less hardware is needed; less disk space and less RAM than for an equivalent-sized, traditional Business Intelligence DB.

## **64-bit Support**

Written and designed to natively support 64-bit processing, it vastly increases the amount of memory the system can address at any given time. 64-bit architecture means you can work with virtually unlimited amounts of data.

## **True Multi-User, Multi-Application Architecture**

ElastiCubes are not tightly coupled with the application layer of the system. This frees up a single ElastiCube to handle multiple applications and users. Not having to reproduce your data model for every application saves significant time developing and maintaining your dashboards and reports.

## Just-In-Time, In-Memory Processing

### **Smart Cache and Instruction Recycling**

CPU cycles and RAM space are the two most precious resources in any computer, and ElastiCube is designed to use both as efficiently and speedily as possible.

Using our sophisticated caching algorithm, the data is only loaded into memory when it's needed. As part of this algorithm, compute- and time-intense calculations are also intelligently cached to further reduce I/O calls.

### **Cache-aware Algorithm**

Additional sophisticated algorithms further increase Sisense's performance. Once data is loaded into memory, the main performance bottleneck becomes CPU cache misses that naturally come with random access. The ElastiCube is specifically designed to minimize these errors by employing a unique cache-aware algorithm, further increasing Sisense's performance by an additional order of magnitude.

### **Compressed Calculations**

Every DB compresses data to save disk space and RAM. ElastiCube is designed to work directly on this compressed data, so that the need for decompression is virtually eliminated, further increasing ElastiCube's performance.

## Designed with Standard Hardware in Mind

Just about every new computer on the market—even portables like iPhones and iPads—are built with very powerful multi-core processors, putting several CPUs into one. ElastiCube was built specifically to take advantage of these powerful CPUs, further increasing Sisense's performance on standard hardware, enabling you to run multiple applications and support multiple users.

### **Highly RAM-efficient**

The thing we know for sure about DBs is that they grow. Fast. So no matter how much fancy footwork is done with completely in-memory DBs, eventually you run out of RAM space and need to upgrade—at least your RAM (best case) or your entire hardware platform (worst, very expensive, case). At Sisense we know this, so we spent years designing the ElastiCube to be able to handle terabytes—billions of rows—of data efficiently and quickly, even on standard PC hardware.

## Unified Analytics Engine

Sisense can execute queries against a wide variety of data sources as if they were all of the same type, essentially making the individual characteristics of each physical data source unimportant. Our Unified Analytics Engine is what makes this possible.

When Sisense imports data, the Unified Analytics Engine creates a metadata layer, or abstraction layer, which is then used to formulate queries across any number of tables from any number of data sources in any number of formats. It even supports the combined querying of resident and external (live) database sources without first loading data into the database!

These capabilities provide the user with unparalleled flexibility and speed in creating, executing and sharing highly complex reports, dashboards, and analytic applications, with any number and variety of data sources.

## Compliant with Industry Standards

### **Supports SQL-92 Standard**

Even with all this advanced technology, we knew that none of it would be any good if our users couldn't access their existing data. So, we built in an SQL layer to the system, which allows users to integrate Sisense to external applications without needing to learn new scripting languages.

### **Seamless Integration with Existing Data Sources**

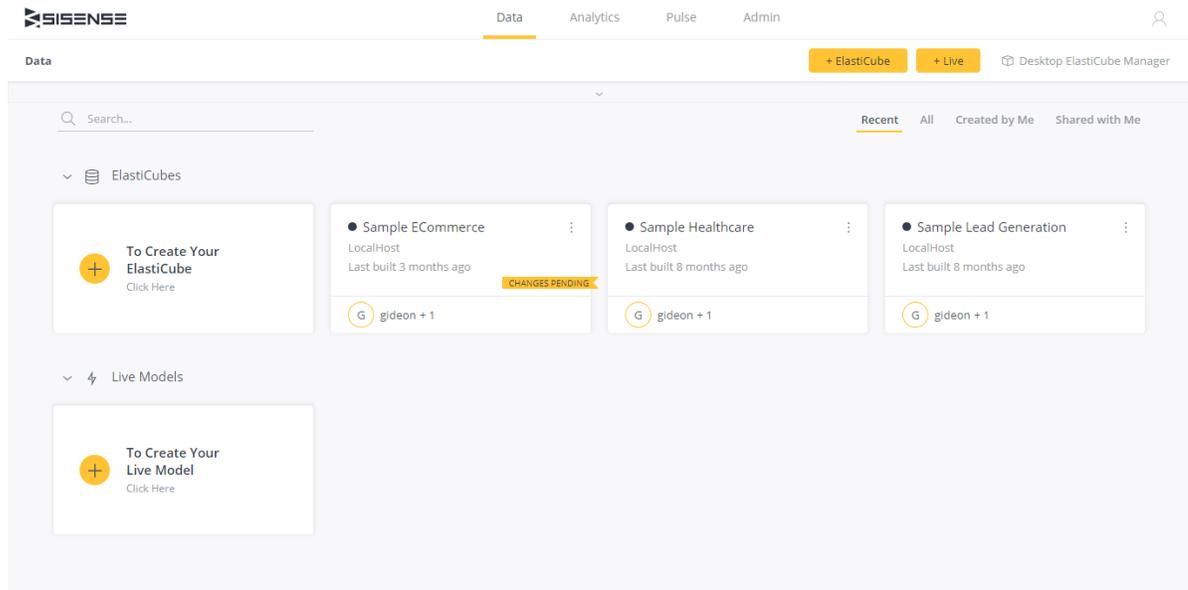
Got an ODBC/OleDB compliant DB today? Great, we built in the ability to access those, too. ElastiCube will seamlessly connect to those data sources so, again, there is no need to learn a new language or write special code to connect to your existing data. With ElastiCube there's no need to start over, you just get faster, easier, and more scalable, with minimal need for IT.

## Creating ElastiCubes

Before you can analyze dashboards, you need to add data to Sisense and then model it. The first step then is to create an ElastiCube. Once you create your ElastiCube, the Model Editor displays an empty canvas where you can begin to import and model your data. For a short video overview of creating an ElastiCube, click [here](#).

**To create an ElastiCube:**

1. Open Sisense and click **Data** in the top menu. Your **ElastiCubes** and live models are displayed.



2. Click **+ ElastiCube**. The **Add new ElastiCube** dialog box is displayed.
3. Name the new ElastiCube and click Save.
4. The Model Editor is displayed, where you can start creating your data model.

## Navigating the Model Editor

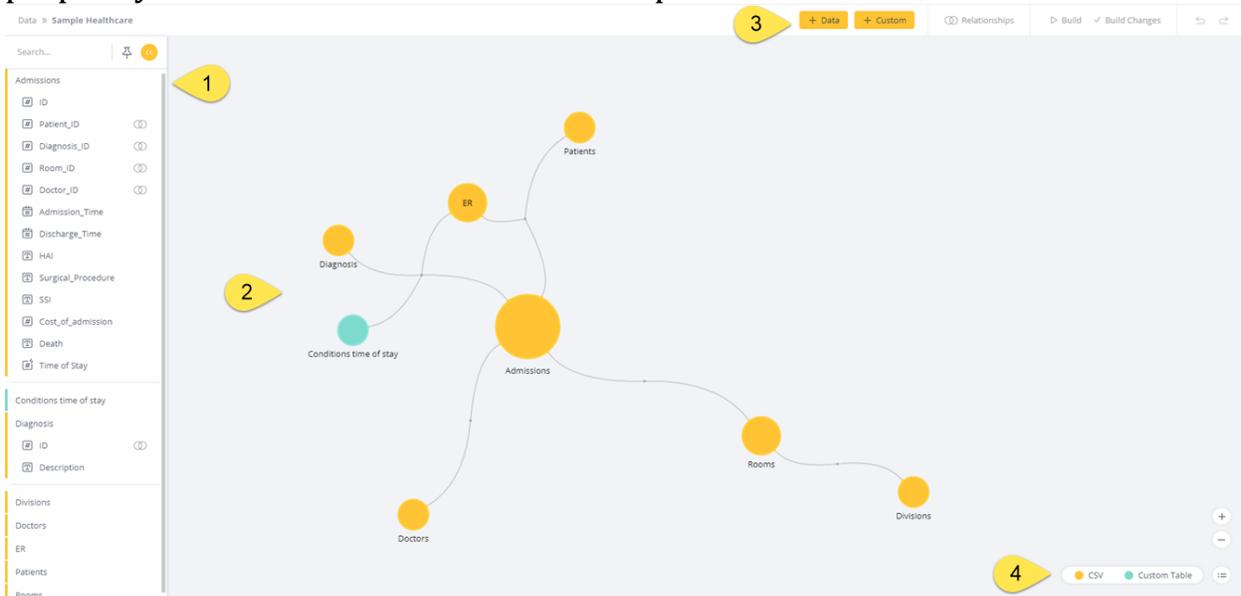
The first time you create an ElastiCube or a live data model, its empty until you start adding some data. Once you connect to your data source and select what data is going to be imported or queried, its represented in Sisense as circular nodes. Each node is a table. The color of a node is determined by its data source, for example, data imported from a CSV file is one color while data imported via SQL is another. The colors of the data source are described in the legend on the bottom right of the editor.

The size of the table reflects the number of relations and the number of columns the table has. So, a table with several relations and a lot of columns will be larger than a table with a single relation and fewer columns. For example:



The 'Conditions time of stay' table is a custom table with a single relation and the 'Admissions' table contains data from a CSV file and has four relations.

The Model Editor contains a variety of information and functionality that lets you prepare your data model as needed. A sample screen is shown below.



1. **Navigation Pane:** This area contains a list of your tables and their columns. On the left side of the table name is an icon that indicates what type of data is included, Date, Numeric or Text. On the right side is a join icon  when the column is connected to another table. See [Finding Tables and Fields](#) more information.
2. **Schema:** The schema contains your data model that represents all the data to be added to the ElastiCube and the relationships between the table. Some of your tables might appear with icon attached them. These indicate the status of your build and table. For example,  indicates that the table has changed since the last ElastiCube build while  indicates that a custom expression is currently in draft mode. See [Working with Data](#) for more information.

3. **Schema Menu:** This menu lists the following buttons:
  - Data:** Click to add a new table to your ElastiCube.
  - Custom:** Click to add a custom table to your ElastiCube.
  - Relationships:** Click to open a preview window where you can join two tables. See [Creating Relationships](#) for more information.
  - Build:** Click **Build** to initiate an ElastiCube build.
  - Undo/Redo:** Click these buttons to undo or redo any recent changes to your schema.
4. **Legend:** The legend lists a description of all your data sources in the ElastiCube and the color that represents them in your schema.

## Introduction to Data Sources

Connecting to your data is the first step in analyzing your business information. Sisense has native connectors to many SQL databases, NoSQL sources, and popular web applications. For a short video overview of connecting to data, click [here](#).

There are two ways you can connect to your data sources. You can either import the data into the ElastiCube or you can connect directly to the data source, which is known as a live connection.

Which connection type you choose depends on your use case and the data source itself. Importing your data into the ElastiCube takes advantage of Sisense high performance database. However, if your data changes frequently and your dashboard must reflect this, than a live connection is probably the best choice. Keep in mind however that live connections work only as well as the data source. The tables below provide a list of supported data sources and links to the documentation for ElastiCube and live data sources.

# Live Connectors

Connector	Download Link	More Information
Google BigQuery	Pre-Installed	<a href="#">Documentation</a>
MySQL	Pre-Installed	<a href="#">Documentation</a>
Oracle	Pre-Installed	<a href="#">Documentation</a>
PostgreSQL	Pre-Installed	<a href="#">Documentation</a>
Redshift	Pre-Installed	<a href="#">Documentation</a>
Snowflake	Pre-Installed	<a href="#">Documentation</a>
SQL Server	Pre-Installed	<a href="#">Documentation</a>

# ElastiCube Connectors

Connector	Download Link	More Information
Access *	<a href="#">Download</a>	<a href="#">Documentation</a>
Bing Search*	<a href="#">Download</a>	<a href="#">Documentation</a>
Box *	<a href="#">Download</a>	<a href="#">Documentation</a>
CSV	Pre-Installed	<a href="#">Documentation</a>
Custom REST	Pre-Installed	<a href="#">Documentation</a>
DynamoDB *	<a href="#">Download</a>	<a href="#">Documentation</a>
Exact Online *	<a href="#">Download</a>	<a href="#">Documentation</a>
Excel	Pre-Installed	<a href="#">Documentation</a>
Generic JDBC Driver	Pre-Installed	<a href="#">Documentation</a>
Generic ODBC Driver	Pre-Installed	<a href="#">Documentation</a>
Generic OLE DB Driver	Pre-Installed	<a href="#">Documentation</a>
Gmail *	<a href="#">Download</a>	<a href="#">Documentation</a>
Google AdWords	Pre-Installed	<a href="#">Documentation</a>
Google Analytics	Pre-Installed	<a href="#">Documentation</a>
Google BigQuery	Pre-Installed	<a href="#">Documentation</a>
Google Sheets	Pre-Installed	<a href="#">Documentation</a>
Heroku Postgres	Pre-Installed	<a href="#">Documentation</a>
Hive	<a href="#">Download</a>	<a href="#">Documentation</a>
HubSpot *	<a href="#">Download</a>	<a href="#">Documentation</a>
Intacct *	<a href="#">Download</a>	<a href="#">Documentation</a>

Connector	Download Link	More Information
MailChimp *	<a href="#">Download</a>	<a href="#">Documentation</a>
Microsoft Active Directory *	<a href="#">Download</a>	<a href="#">Documentation</a>
Microsoft Dynamics CRM *	<a href="#">Download</a>	<a href="#">Documentation</a>
Microsoft Dynamics Nav *	<a href="#">Download</a>	<a href="#">Documentation</a>
MongoDB	<a href="#">Download</a>	<a href="#">Documentation</a>
MySQL	Pre-Installed	<a href="#">Documentation</a>
Oracle	Pre-Installed	<a href="#">Documentation</a>
PostgreSQL	Pre-Installed	<a href="#">Documentation</a>
Quickbooks Desktop *	<a href="#">Download</a>	<a href="#">Documentation</a>
Quickbooks Online *	<a href="#">Download</a>	<a href="#">Documentation</a>
Redshift	Pre-Installed	<a href="#">Documentation</a>
Salesforce	Pre-Installed	<a href="#">Documentation</a>
ServiceNow *	<a href="#">Download</a>	<a href="#">Documentation</a>
SharePoint Excel Services *	<a href="#">Download</a>	<a href="#">Documentation</a>
Sisense ElastiCube	Pre-Installed	<a href="#">Documentation</a>
Snowflake	Pre-Installed	<a href="#">Documentation</a>
Splunk	Pre-Installed	<a href="#">Documentation</a>
Stripe *	<a href="#">Download</a>	<a href="#">Documentation</a>
SQL Server	Pre-Installed	<a href="#">Documentation</a>
Xero *	<a href="#">Download</a>	<a href="#">Documentation</a>
YouTube *	<a href="#">Download</a>	<a href="#">Documentation</a>
YouTube Analytics *	<a href="#">Download</a>	<a href="#">Documentation</a>

\* Certified connectors are provided by our certificated partner. Sisense carefully evaluates and certifies each external connector to work with Sisense. This helps us ensure that our external connectors are always up-to-date and fully supported.

# Technology Partner Connectors

Technology partners provide the service of connecting to data sources via the data pipeline, managing API changes and limits, and delivering comprehensive support as part of their service. Import your data into a data warehouse (Redshift, Google BigQuery, Snowflake, SQL Server, MySQL, PostgreSQL, and more) to access your data with either ElastiCube or live data models. Contact your Account Executive or Customer Success Manager to discuss the full functionality our technology partners provide, as well as to start your free trial. For more information, click [here](#).

Connector	Connector	Connector
Adobe Analytics	Apple Search Ads	AppsFlyer
DoubleClick	Facebook	Facebook Ads
Freshdesk	Jira	LinkedIn Ads
Magento	Marketo	Netsuite
Pardot	Shopify	Twitter Ads
Zendesk	And more...	

# Vendor JDBC Connectors

With the Sisense's [generic JDBC driver](#) you can leverage native JDBC connectors provided by the data source vendors to connect to additional data sources.

Connector	More Information
Athena	<a href="#">Download</a>
PrestoDB	<a href="#">Download</a>
Teradata	<a href="#">Download</a>
Vertica	<a href="#">Download</a>

# Connecting to More Data Sources and Frameworks

You can connect to many additional data sources by leveraging Sisense's generic frameworks such as the [generic ODBC driver](#), [JDBC driver](#), or [Custom REST Framework](#).

Sisense supports these frameworks with frequent upgrades and new functionality. However, full connectors to specific data sources created through these frameworks are not supported.

## Changing Connectivity Settings for Data Sources

This topic describes how to change the connectivity settings for an existing data source. For a short video overview of changing connectivity settings, click [here](#). After you have modeled your data in the ElastiCube, a data source may have changed location or you may want to change a table without importing a new table into your schema from scratch. If this happens, you can update your connection settings to the data source.

When updating connectivity settings, you can choose one of the following:

- **Change Provider:** This enables you to change the source of the data being accessed, for example changing a provider from SQL Server to MySQL. After you select the new provider, you need to choose the relevant database and tables from the new provider. This is useful if you want to replace a table in your schema with a new data source, but don't want to change your schema.
- **Change Connection:** This enables you to use the same provider, but change your credentials to the data source or the database previously selected.
- **Change Table:** This enables you to use the same provider and credentials, but change the database previously selected.

If a table has changed, but the source and its settings are the same, for example a new column was added, and you want to update your schema without importing the data into the schema all over again, click **Refresh** in the table’s menu.

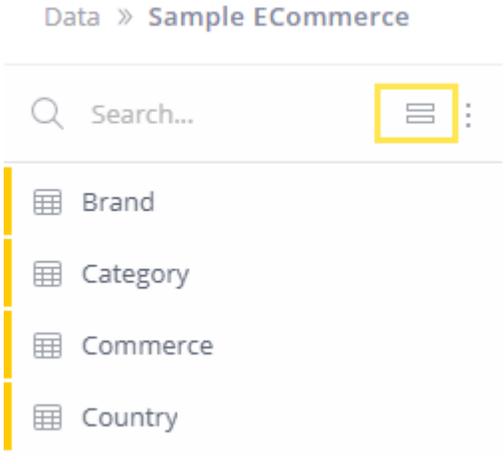
**Note:** The data source name, columns, and column types must remain the same for existing widgets in the dashboard to be able to reference the data correctly.

The table below describes which sources support changing connectivity settings in the web-based ElastiCube Manager:

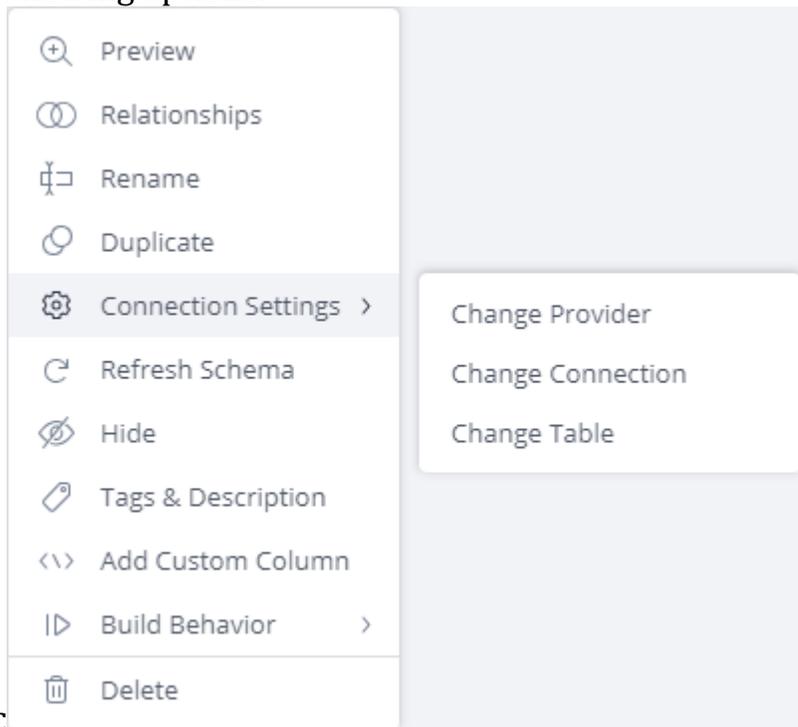
Data Source	Connection Settings	Change Database	Table Settings	Sync Columns
Excel	N/A	N/A	Yes	Yes
CSV	N/A	N/A	Yes	Yes
Google Sheets	Yes	Yes	Yes	Yes
MySQL	Yes	Yes	Yes	Yes
SQL Server	Yes	Yes	Yes	Yes
PostgreSQL	Yes	Yes	Yes	Yes
Oracle	Yes	Yes	Yes	Yes
Redshift	Yes	Yes	Yes	Yes

**To change connectivity settings:**

1. In your schema, select the Data Source view in the Navigation Pane.



2. For the relevant data source, select **Connection Settings** and one of the following options:



c

**Change Provider:** Selecting this option opens the first step of the Connection Wizard where you select the data source. In the Connection Wizard, click to select a data source connection type, for example, change MySQL to an Oracle database. When done, enter the connection credentials and select the relevant tables.

**Change Connection:** Selecting this option opens the second step of the Connection Wizard where you define the connection settings of the data source. In the Connection Wizard, click to edit the login details to the data source. When done, select the relevant tables.

**Change Table:** Selecting this option opens the third step of the Connection Wizard where you select the relevant database from a list of databases in your data source. In the Connection Wizard, select the relevant database.

3. After you have selected the relevant database in the Connection Wizard, click **Done**. The connection settings are updated.

## Working with Data

Once your data is imported into Sisense, Designers can organize the data in a variety of ways. One of the most common ways is to create a relationship

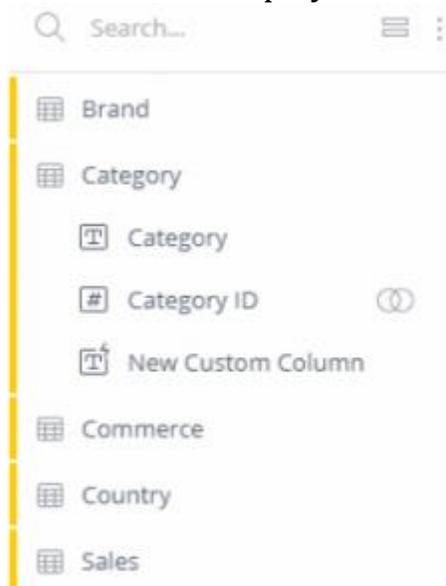
between tables. After you create a relationship, the tables behave as a single table, which can make analyzing your data more easy and accurate.

The following topics describe how you can create relationships with your data and how to avoid some of the pitfalls of working with complex data:

- [Navigating Sisense](#)
- [Finding Tables and Columns](#)
- [Creating and Removing Relationships](#)
- [Previewing Data in a Table](#)
- [Managing Tables and Columns](#)
- [Handling Relationship Cycles](#)
- [Many-to-Many Relationships](#)
- [Chasm and Fan Traps](#)

## Finding Tables and Columns

Through the ElastiCube Search tool, you can locate field tables and fields. The Search tool is displayed in the Navigation Pane of the ElastiCube Manager.



When you perform a search, all the relevant tables and columns are returned as you begin to type.

To organize the results displayed in the pane, toggle between the Data Source icon  and the Tables icon .

Each result has an icon on the left side of it. These icons indicate what is being returned.

The meaning of each icon is as follows:

 Data source

 Table

 Text data

 Numeric data

 Date data

### **To find a table or field in your ElastiCube:**

- On the left-side of the schema, enter the table or field in the **Search** field. As you begin typing the letters or name of the table/field you are searching, the results are displayed below and the relevant table or field is highlighted in the schema.

## Creating and Removing a Relationship between Tables

Relationships specify the connection between tables and enable you to pull data together in meaningful ways. For example, order information is more useful when you know which customer placed each order.

However, you don't need to store both the customer and order information in the same table. Customer and order data can be stored in two related tables with a relationship specified between the two tables to view each order and its corresponding customer information.

In practical terms relationships allow you to query and combine data from multiple tables in your dashboard.

### **To manually create a relationship between tables:**

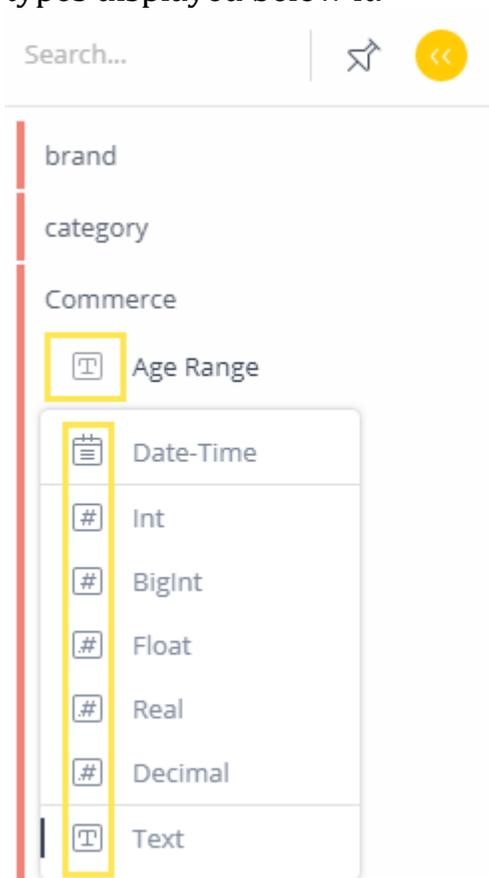
1. Drag a table with the relevant field onto another table with the relevant field.

The Relationship preview window is displayed.

2. Click the columns to be joined from each table. Ensure both fields that you are using to create the relationship are of the same type, for example decimal fields.

See the video below for a quick example.

3. To check or change the field type, in the left pane, click the field type of the field you want to change and select the new type from the list of possible types displayed below it.



4. Click **Save** to join the tables.

#### **To delete an existing relationship:**

1. Click on any part of the existing relationship. The Delete icon is displayed.
2. Click  to delete the relationship.

#### Previewing Data from a Table

The Preview table lets you see all the values for all the rows and columns in a table. This includes any custom columns, tables, and ETL processes that have been applied to your table.

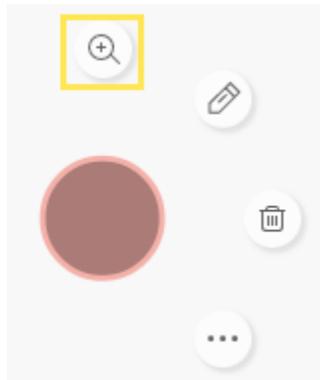
The Preview table makes it easy to verify that the data in your table is accurate and complete.

In addition to displaying a preview of your data, from the Preview table, you can view which columns are connected to other tables and add custom columns.

**Note:** To view underlying table data, you will first need to build your ElastiCube.

**To review the underlying table data:**

- Click the relevant table and select the magnifying glass icon to display the Preview table.



The Preview table contains a variety of information and functionality that relate to your table. A sample preview is displayed below.

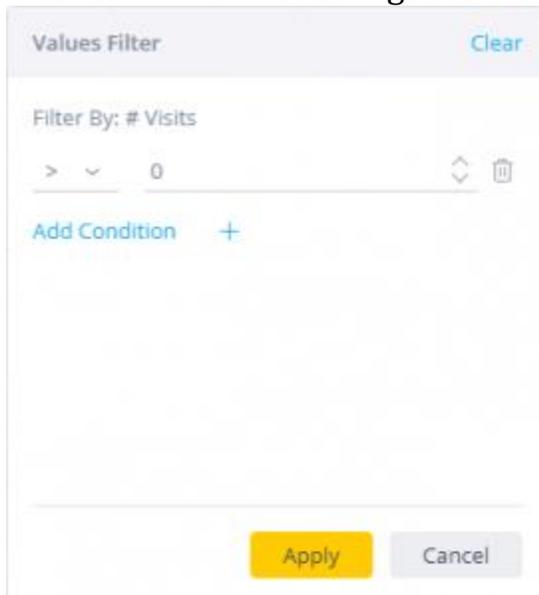
Age Range	Cost	Brand ID	Category ID	Country ID	Date	Quantity	Revenue	Visit ID
25-34	187.483	3	15 Unspecified	1	7/29/2010, 12:00...	Female	374.967	26.215
19-24	43.984	46	9 Used	3	8/18/2010, 12:00...	Male	43.984	26.216
35-44	108.2	122	19 Unspecified	3	8/16/2010, 12:00...	Male	108.2	26.217
35-44	154.282	145	8 Unspecified	2	6/17/2010, 12:00...	Female	154.282	26.218
55-64	1.995	4	6 Unspecified	3	6/30/2010, 12:00...	Male	1.995	26.219
55-64	58.547	3	6 Unspecified	3	7/9/2010, 12:00...	Male	58.547	26.220
	62.803	3	9 Unspecified	1	7/19/2010, 12:00...	Male	62.803	26.221
	224.071	16	11 Unspecified	3	7/30/2010, 12:00...	Unspecified	224.071	26.222
	916.831	3	9 Unspecified	1	8/8/2010, 12:00...	Unspecified	3.667.325	26.223
	43.619	16	6 Unspecified	2	8/18/2010, 12:00...	Unspecified	43.619	26.224
	135.385	5	22 Unspecified	3	6/1/2010, 12:00...	Unspecified	135.385	26.225
65+	124.134	5	11 Unspecified	3	6/9/2010, 12:00...	Female	124.134	26.226
45-54	128.058	18	9 New	3	6/23/2010, 12:00...	Male	128.058	26.227

	Name	Description
1	Column Filter	In the left pane, you can search for columns in your table. As you begin to type, any relevant results are displayed in the left pane. This does not affect the results displayed in the preview.
2	Connected Table	This icon indicates that the column is connected to another table. Click the icon to disconnect the relationship or to view

	Name	Description
		more information about the connection. See <a href="#">Creating and Removing a Relationship between Tables</a> for more information.
3	Table Details	This information describes the number of columns and rows included in the table.
4	Menu Options	This is a list of tasks you can perform on a column. Hover over the column to display the menu and click it to display the options.
5	Filter Column	Click to apply filters to the data in the column. You can filter numeric data and text data. See <a href="#">Filtering Columns</a> for more information.
6	Sort Column	Click to sort a column. Hover over the column name to display the sort icon. You can sort the column by date, numerically, or alphabetically depending on the type of column.
7	Add Custom Column	Click to add a custom column to the table. See <a href="#">Adding a New Custom Column</a> for more information.
8	Sample Data	Click to see a sample of your data. See <a href="#">Sampling Data</a> for more information.

# Filtering Columns

You can filter data displayed in your Preview table's columns by defining conditions for numeric data and text data. To limit what data is displayed in the preview, you can add multiple conditions that you define. For example, a Text filter can limit what strings are included in the preview.



Filtering the data in the Preview table does not affect the data itself, just what you see in the preview.

Filtering your preview is useful if you need to check specific data to verify its accuracy of your table's completeness.

## **To filter entire rows or columns of data from your preview:**

1. In the Preview table, hover over the header in the of the relevant column to display the Filter icon.
2. Click the Filter icon to define the filter's conditions.

<b>Text</b>	<b>Numeric</b>
-------------	----------------

The image shows a user interface for configuring filters, divided into two main sections: 'Text Filter' and 'Values Filter'. Each section has a 'Clear' button in the top right corner.

**Text Filter:** The 'Filter By' is set to 'Country'. Below this, there is a dropdown menu currently showing 'Containing' and a text input field with the placeholder 'Type here...'. To the right of the input field is a trash icon. Below the input field is a blue link 'Add Condition' followed by a plus sign '+'. At the bottom of the section are two buttons: a yellow 'Apply' button and a grey 'Cancel' button.

**Values Filter:** The 'Filter By' is set to '# Visits'. Below this, there is a dropdown menu showing '>' and a numeric input field with the value '0'. To the right of the input field are up/down arrow icons and a trash icon. Below the input field is a blue link 'Add Condition' followed by a plus sign '+'. At the bottom of the section are two buttons: a yellow 'Apply' button and a grey 'Cancel' button.

3. Define your filters. Click **Add Condition** to add more conditions to the filter or click **Clear** to erase all the conditions.
4. After you have defined your conditions, click **Apply** to update the preview.

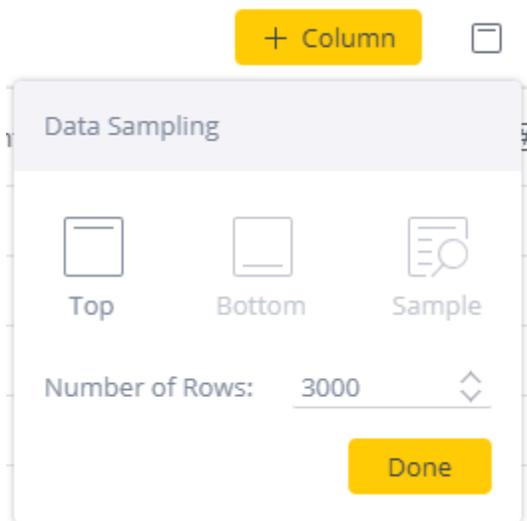
# Sampling Data

If you are working with a large dataset, you may want to restrict the amount of data displayed in your Preview table. Sisense provides three options for displaying a sample of your data:

**Top:** The first rows that are contained at the top of your data.

**Bottom:** The last rows that are contained at the bottom of your data.

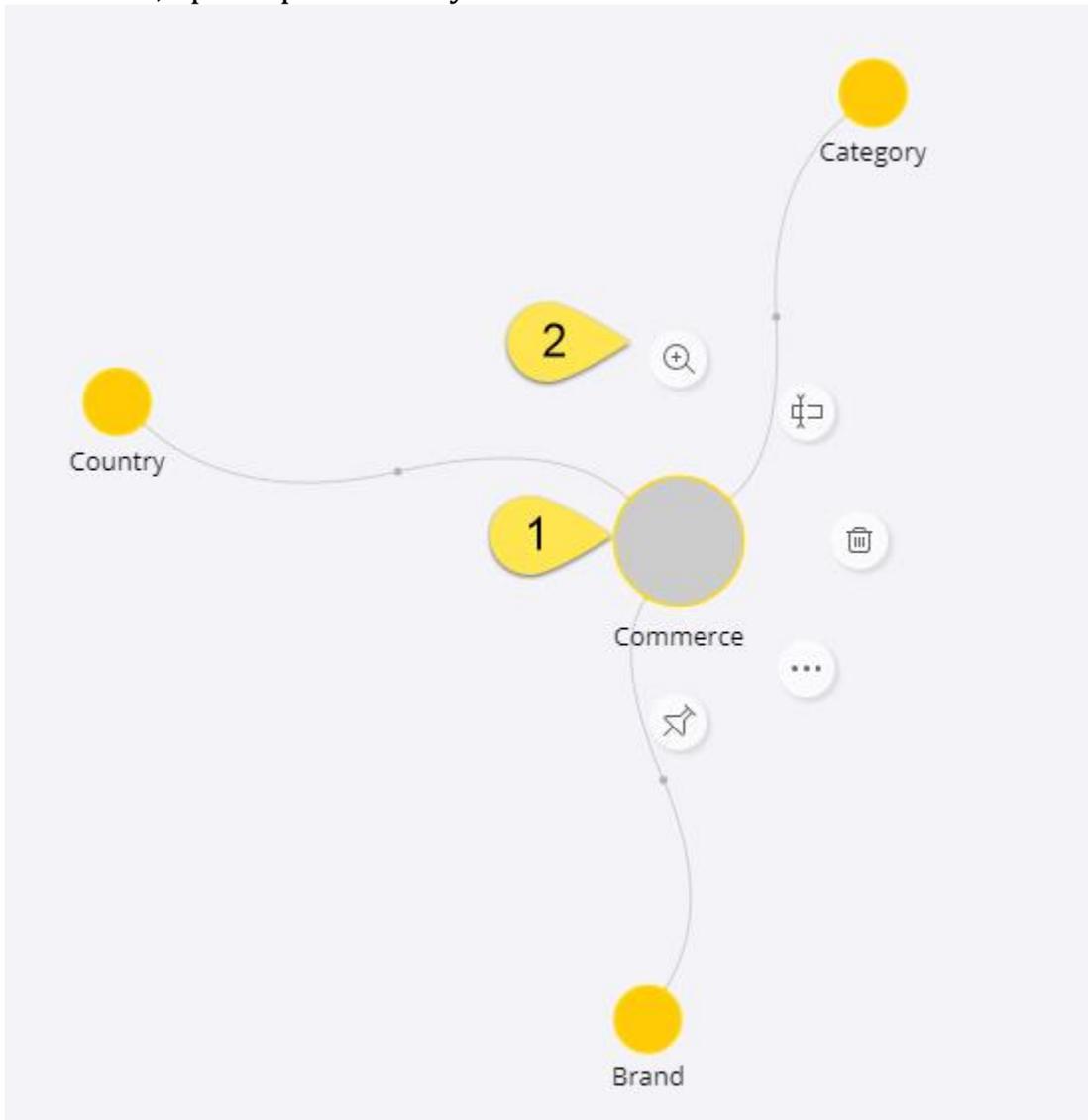
**Sample:** A random selection of rows contained in your table. Sisense selects a sample of your data randomly by rows. If you open the preview again, the same rows will be displayed until your next ElastiCube build.



With each sampling option, you can determine how many rows are displayed according to the value of the **Number of Rows** field. You can modify this value to display as few as 10 rows or include as many as you like up to the full amount of rows included in the table.

**To view a sample of your data:**

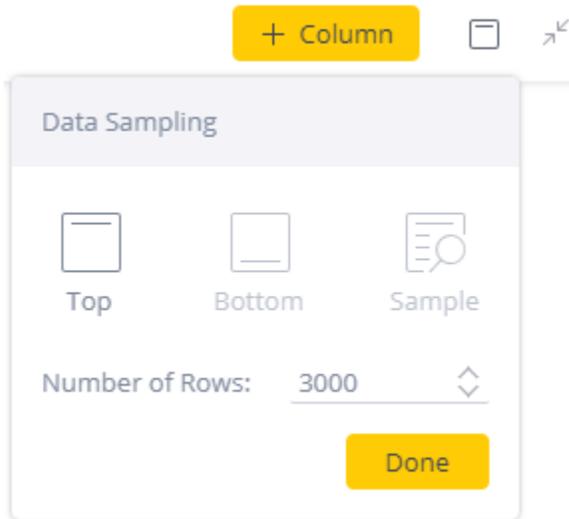
1. In Sisense, open a preview of your table.



2. In the top-right corner of the Preview table, click the **Sampling** Icon.



3. Select the relevant sampling option for your data and enter the amount of rows to be included in the sample in the **Number of Rows** field.



4. Click **Done**. Your Preview table is updated.

## Managing Tables and Columns

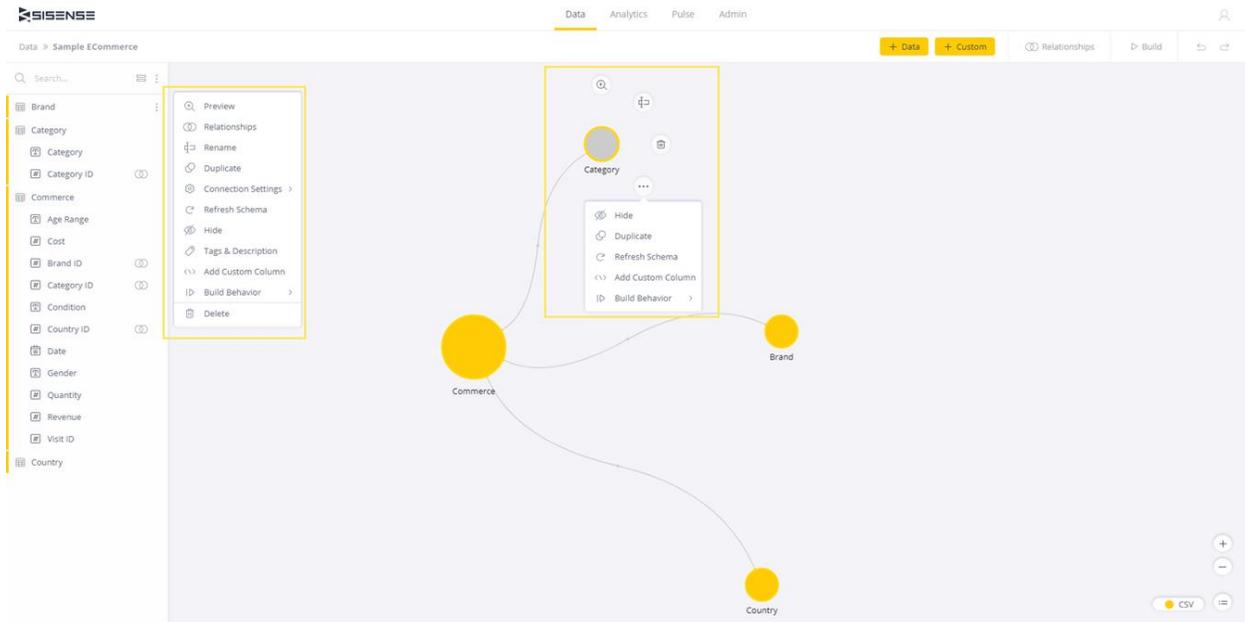
After you have imported your data into Sisense, it's time to shape it into a data model that can support your dashboard and provide accurate information to your Viewers.

You have a variety of options you can use to model your data from the Navigation Pane on the left side or in the schema itself.

You can modify your data on two levels, the Table level and the Column level. Any changes you make to your schema are reflected in the ElastiCube only and do not affect the data in your data source.

# Table Level

You can manage tables imported into the ElastiCube through the Navigation Pane or through the table itself in the schema.



The table below describes the menu options available from the Navigation Pane and the corresponding icons available in the table's menu. Either method allows you to modify and manage your tables.

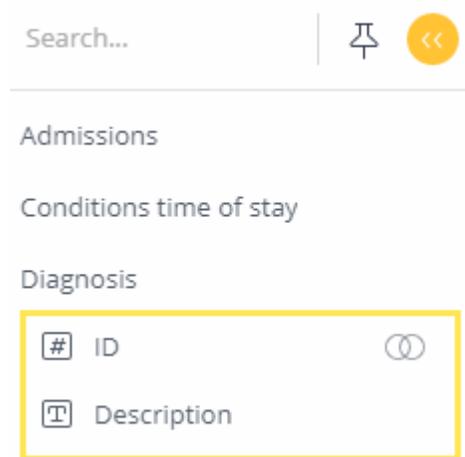
Navigation Pane	Table Icon	Description
Preview & Edit		Displays a preview of your table's data. See <a href="#">Previewing your Data</a> for more information.
Relationships		Displays related tables together side by side. See <a href="#">Creating Relationships</a> for more information.
Rename		Allows you to rename the table.

Navigation Pane	Table Icon	Description
Duplicate	... >  Duplicate	Allows you to duplicate the table.
Refresh Schema	... >  Refresh Schema	Allows you to update a table's columns if there was a change on the data source without having to add the table again.
Tags & Description	... >  Tags & Description	Allows you to tag a table with metadata that you can use to organize your data and search for it later. See <a href="#">Tagging Your Data</a> for more information.
Hide	... >  Hide	Allows you to hide a table. Hidden tables are not built with the rest of the ElastiCube.
Add Custom Column	... >  Add Custom Column	Allows you to add a custom column to the ElastiCube. See <a href="#">Adding Custom Columns</a> for more information.
Build Behavior	... >  Build Behavior	Allows you to define the build behavior for a specific table. See <a href="#">Build Settings and Data Accumulation Behavior</a> for more information.
Delete	... > 	Allows you to delete a table.
--		By default, tables are pinned when you import your data. You can click this icon to unpin your tables so your tables are organized by Sisense around other pinned tables.

To view your table's meta, including its name, path, provider, location, and the number of columns it contains, hover over the table in the schema or in the Navigation Pane.

# Column Level

You can manage tables imported into the ElastiCube through the Navigation Pane. Columns that are part of a table are displayed below the table. On the left side of the column is an icon that indicates what type of data is in the column, Numeric, Date, or Text. You can click this icon to convert the column into another data type.



On the right side of the column's name is an icon that indicates if the column is conducted to another column through a relationship. You can hover over the icon to view more details or disconnect the relationship.

For each column, you can hover over its name to display a menu from where you view the following options for managing your columns:

Menu Option	Description
Relationships	Displays related columns together side by side. See <a href="#">Creating Relationships</a> for more information.
Edit Relationship	Allows you to edit a column's relationship. This option is only available when the column is connected to another table.
Rename	Allows you to rename the column.

Menu Option	Description
Duplicate	Allows you to duplicate the column in the table. This is useful if you need the same column with some modifications.
Hide	Allows you to hide a column. Click this option if you want to import the data associated with this column, but do not want this column available to dashboard designers and viewers.
Indexed	Allows you to index the column data for quicker processing in subsequent builds.
Drop	Allows you to remove column from the table.
Change Data Type	Allows you to change the data type of your column in the ElastiCube. This does not change the column type in the original data source.
Description	Allows you to tag a column with metadata that you can use to organize your data and search for it later. See <a href="#">Tagging Your Data</a> for more information.

## Creating Data Dictionaries

To provide you with greater flexibility and improved ease-of-use when modeling your data, Sisense allows you to tag your data and add descriptions. Tags and descriptions allow you to group tables and columns by defining metadata and providing the basis of a data dictionary without affecting your actual data. For example, you can tag several tables with a unique word or description, and then locate that group of columns through the **Search** field to see all the tagged tables across your schema regardless of which tables the columns belong to. While both are searchable, the difference between a tag and a description is that tags are

associated with tables and descriptions are free text fields associated with columns or tables.

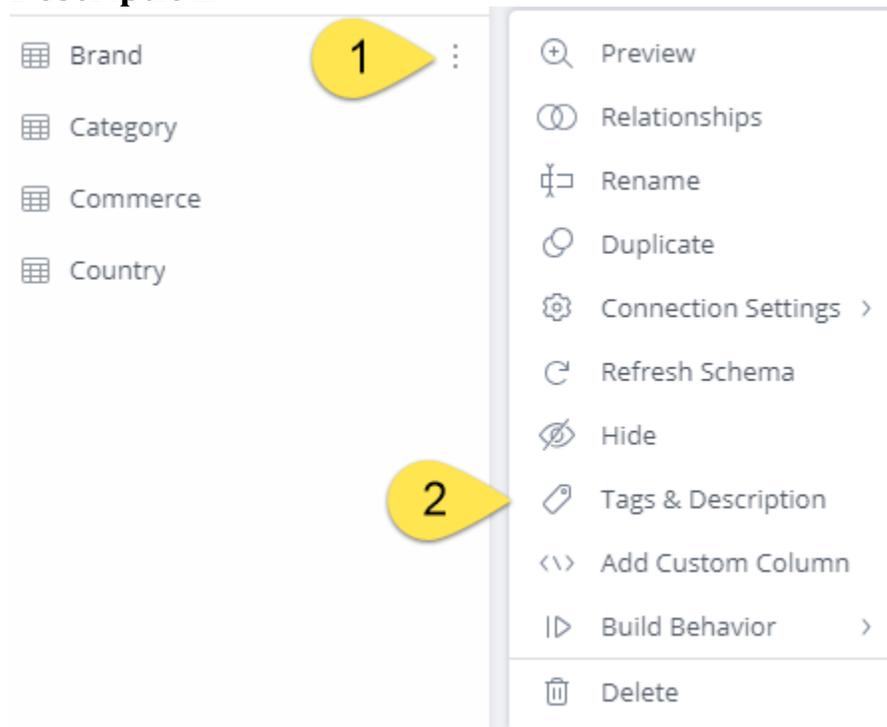
After you have tagged your data, you can easily locate your tagged columns and tables in the **Search** field by entering #, which displays a list of all your tags that you can select to filter your data.



**To tag or add a description to your data:**

**For Tables:**

1. Open your data model.
2. In the Navigation Pane, click the relevant table's menu and select **Tags & Description**.



3. In the **Tags** field, enter a tag and click + to add the tag. You can repeat this step to add multiple tags.

Tags & Description ×

Add a Tag and Press Enter +

Select tag color ● ● ● ● ● ●

Description

Add Description

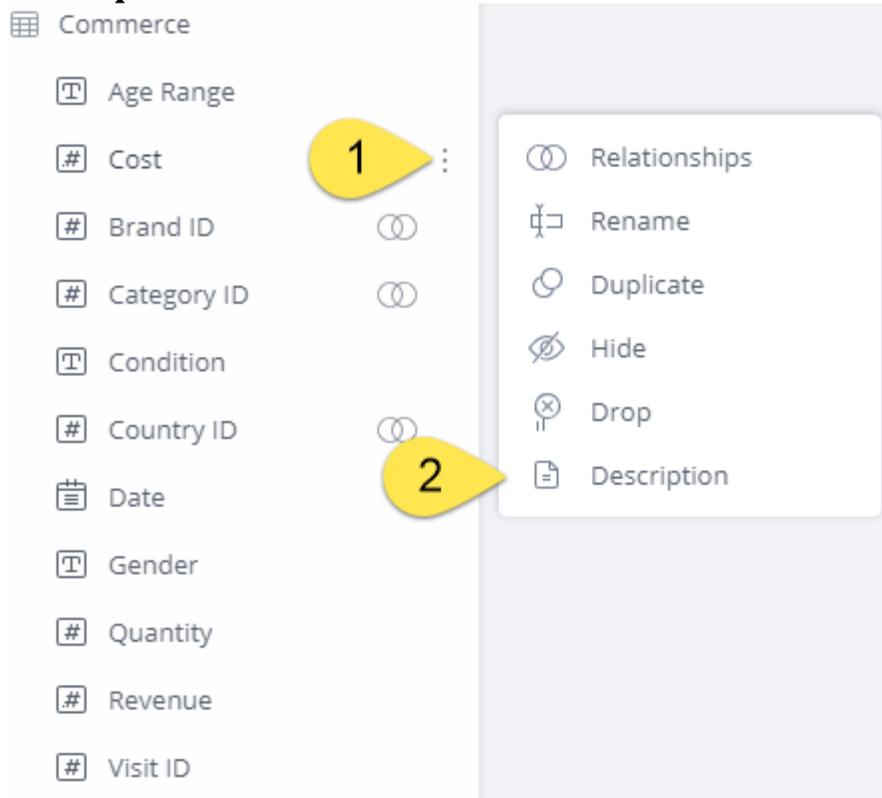
Done

4. Select a color for the tag. The color of a tag is to help you organize related tags.
5. Click **Done** to save your changes.

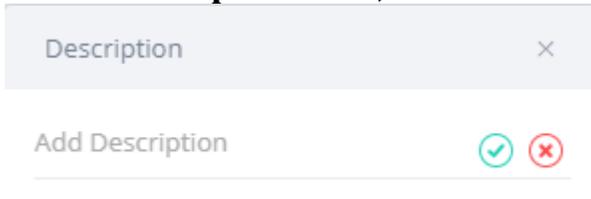
**For Columns:**

1. Open your data model.

2. In the Navigation Pane, click the relevant column's menu and select **Description**.



3. In the **Description** field, enter a description for your column.



4. Click  to save your changes.

## Importing and Exporting ElastiCube Data

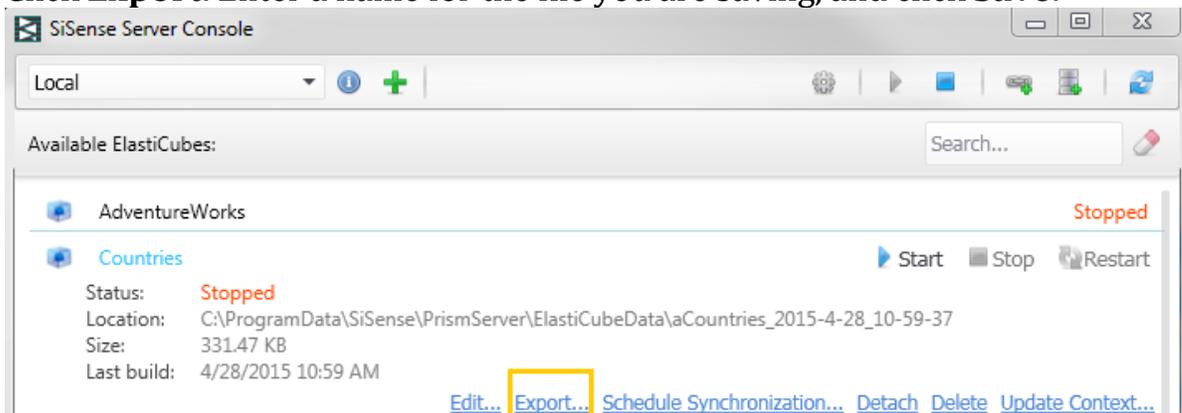
**Note:** This feature is currently available in the desktop version of Sisense. This feature will be migrated to the web-based ElastiCube Manager in the near future. For more information, see [Sisense V7.0 Backwards Compatibility](#).

ElastiCube data can be transferred to different machines. This is useful when you are deploying on a new machine or need to transfer the data to a different server.

The transferred data is stored in a compressed ecdata file. The file contains all of the information contained in an ElastiCubeData folder (including the source data), and is used by the Sisense Server Console to build a new ElastiCube.

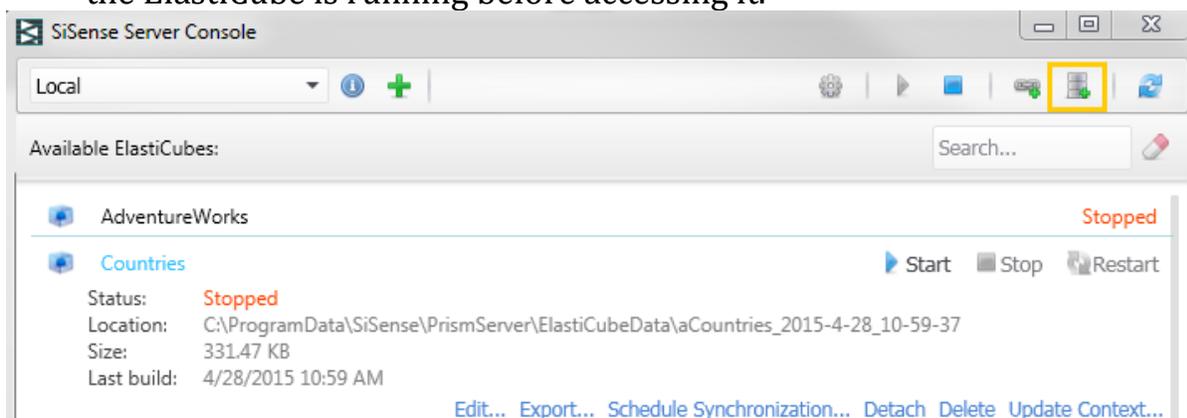
### To export data:

1. Open the Sisense Server Console.
2. Click on the relevant ElastiCube.
3. Click **Stop** to stop the specific ElastiCube (not the entire server).
4. Click **Export**. Enter a name for the file you are saving, and click **Save**.



### To import data:

1. Open the Sisense Server Console.
2. Click on the import ecdata file icon .
3. Select the location of the ecdata file and click **Open**.
4. A window will appear showing the import progress. Once complete, the new ElastiCubes will be added to the list of existing ElastiCubes. Make sure the ElastiCube is running before accessing it.



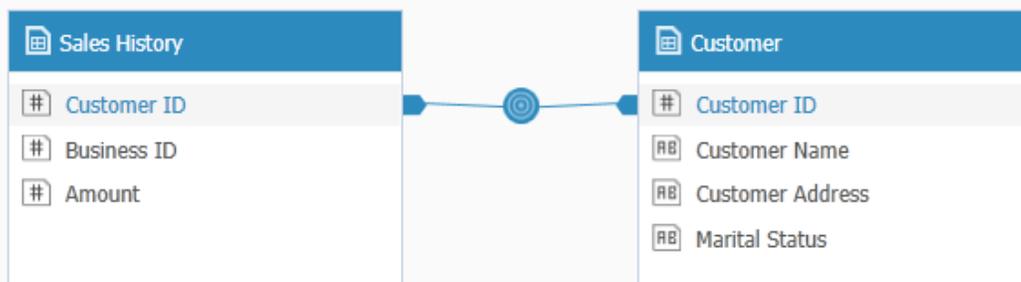
**Note:** You can also transfer data by copying the folder where Sisense stores the data locally, instead of exporting and importing the compressed file.

## Handling Relationship Cycles

**Note:** The image on this page were taken in the desktop version of Sisense, however, the same principles described on this page also apply to the ElastiCube Manager.

Good relationships between tables are the key to pulling in data in logical ways. ElastiCubes are usually created from more than one table. To perform calculations on fields that reside in separate tables, you must define a relationship between the tables so the calculation mechanism knows how to navigate from one field to another.

When a single possible path exists between two fields, there is no issue. For example, performing a query involving the Customer Name and Amount field on the ElastiCube schema below can yield only one result, since there is only one possible path leading from the Customer Name field to the Amount field.

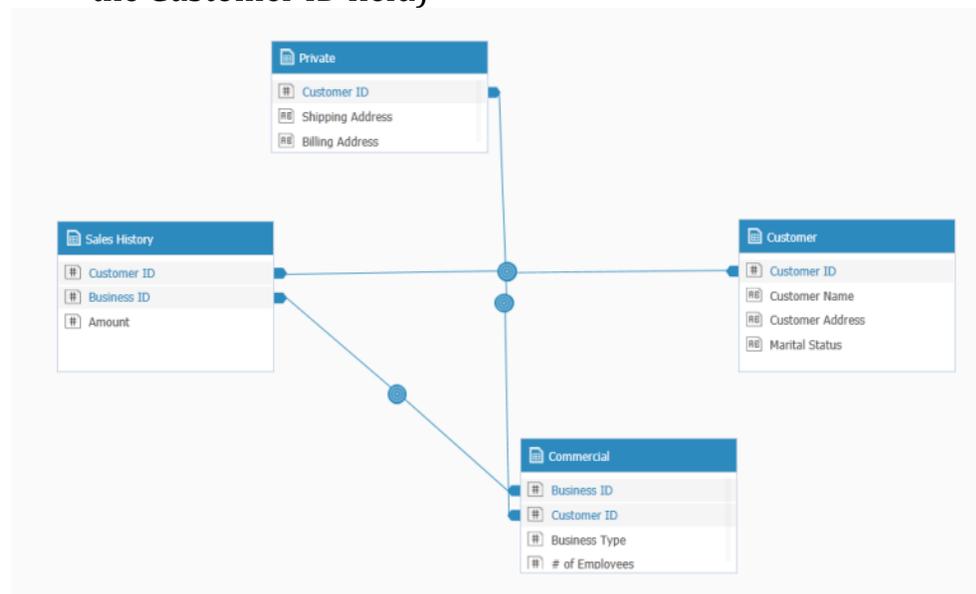


# Relationship Cycles

In some cases, more than one path exists leading from one field to another. Sometimes this is due to poor database design, and other times it is just a realistic necessity due to the way the data is structured. In such cases, there is no absolute way to determine the required path to take to calculate results.

For example, in the ElastiCube schema below, the Customer Name and Amount fields now have multiple paths between them:

- Customer —> Sales History (over the Customer ID field)
- Customer —> Commercial (over the Customer ID field) —> Sales History (over the Business ID field)
- Customer —> Private (over the Customer ID field) —> Sales History (over the Customer ID field)



# Automatically Dealing with Relationship Cycles

Theoretically, any one of these possible paths could be the correct one in terms of the results you are trying to calculate. When the calculation engine encounters numerous possible paths, it picks the shortest path, which will have less impact performance-wise on query processing. Below are some general guidelines.

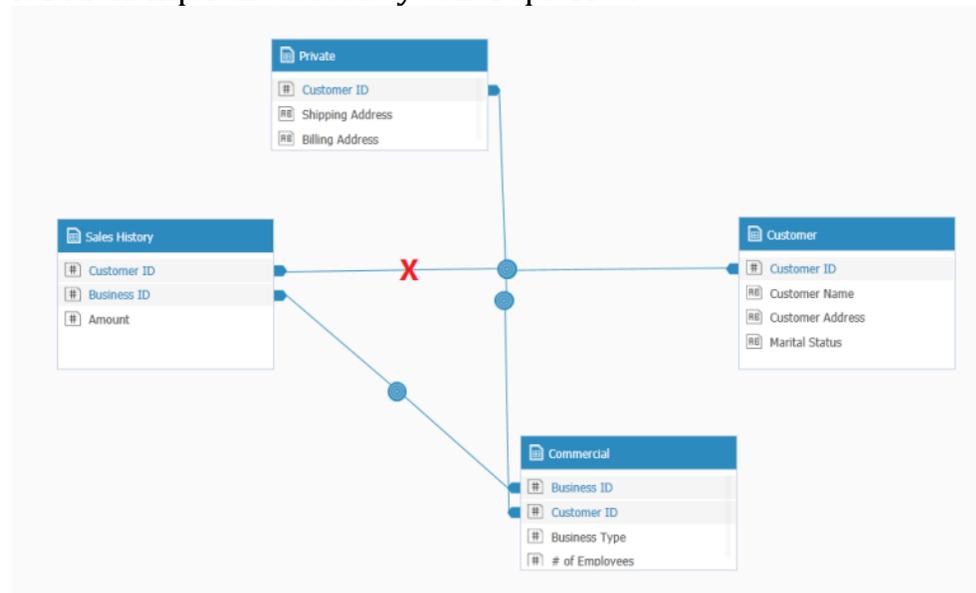
- The calculation engine will always prefer paths that have no many-to-many relationships. If all possible paths contain many-to-many relationships, the path with the least number of this type of relationship is preferred.
- Similarly, the engine will prefer one-to-one relationships over one-to-many relationships.
- If at the end of the first selection process more than one possible path still exists, the engine will prefer the path containing the least number of tables.
- Finally, if numerous paths are still possible, one is picked at random.

# **Breaking Relationship Cycles in Sisense**

To ensure the calculation engine takes the path you want it to take, the cycle must be broken. There are a few techniques to do this depending on the required results.

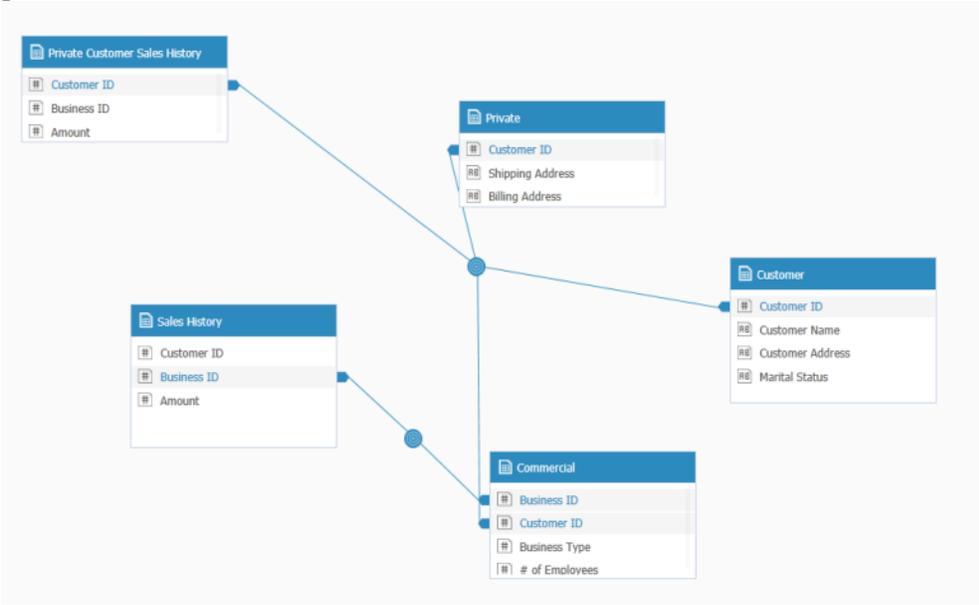
## Removing Relationships

One option is to remove one of the relationships in the cycle. For example, removing the Customer ID relationship from the Sales History table will eliminate the cycle. This approach is only applicable when the removed relationship is not used by other queries.



# Duplicating Table Elements

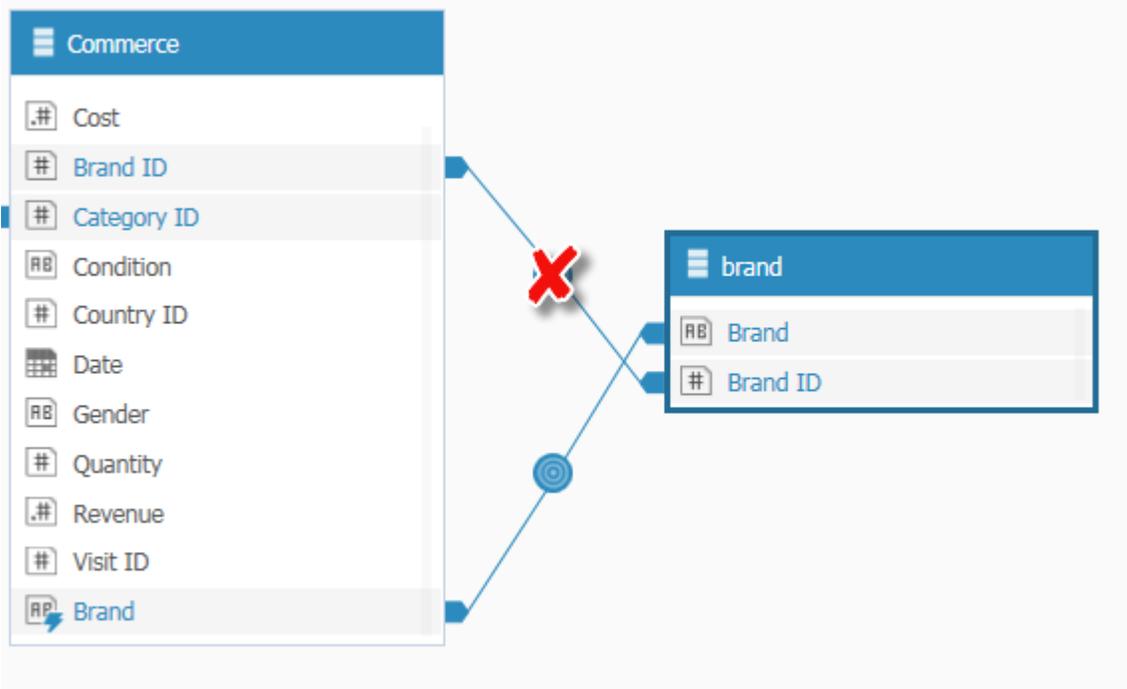
Duplicating table elements is another method to break relationship cycles. In the example below, the Sales History table was duplicated and renamed Private Customers Sales History. In addition, the Customer ID relationship was removed from the original Sales History table. This results in the creation of two separate Amount fields, one for commercial customers residing in the original Sales History table, and one for private customers residing in the new Private Customers Sales History. As the Customer ID relationship was removed from the original table, whichever field you use will determine the correct calculation path.



# Breaking Relationship Cycles in the Sisense Web Application

You can also use the Sisense web app to invoke a specific path to be used during query execution (as opposed to the solutions mentioned above that are defined on the schema level, in Sisense).

Each widget in Sisense has an associated data layout. If you have more than one path connecting corresponding fields between two tables, you can enforce a specific path by filtering the widget, and selecting just the field that you want to use. For example if the Commerce table is connected to Brands via the Brand and Brand ID fields, then add a filter to the widget and select Brand ID to use that path in the query.



## Many-to-Many Relationships

**Note:** The image on this page were taken in the desktop version of Sisense, however, the same principles described on this page also apply to Sisense Online. In databases, a Many-to-Many relationship exists when the value in each field used to create a relationship between tables is included multiple times in each table.

The problem with Many-to-Many relationships is that it can cause duplications in the returned datasets, which can result in incorrect results and might consume excessive computing resources. This section provides solutions and workarounds to common scenarios with many-to-many relationships.

**Example:** A hotel may have a table with reservation data and a table with payment data. Both tables include the name of the guest. A guest can have multiple reservations under their name as well as multiple payments recorded on their name. If a relationship exists between the reservation and payment tables based on the guest's name, a many-to-many relationship is created, as the guest's name appears multiple times in each table.

### Summary of Relationship Types

- **One-to-One Relationship:** In this scenario both sides of the relationship have unique values for every row.
- **One-to-Many Relationship:** In this scenario one side of the relationship will contain unique values for every row, but the other side of the relationship will contain duplicate values for any or all of the corresponding values in the first table.
- **Many-to-Many Relationship:** In this scenario, both sides of the relationship will hold duplicated values, causing excessive calculations for every query run against it.

There are several methods to resolve and bypass a many-to-many relationship. The solution depends on the business model and the logic of the business questions at hand. The following solutions differ by business logic and the schema at hand; each solution can be applied to each schema respectively.

The following sections cover:

- Testing your schema to see if it includes many-to-many relationships
- Understanding which scenario best fits your current schema
- According to your schema logic, applying the respective solution

To check if a relationship is Many-to-Many, you need to check the cardinality of the relationship, and determine the number of unique and duplicate values on each side of the relationship.

## Testing if a Relationship is Many-to-Many

Many-to-Many relationships occur when two tables are joined on a field containing duplicate values on both tables. For example the same guest may have multiple reservations and multiple payments at a hotel, thus joining on the guest between the reservation and payment table would result in a M2M relationship. When testing, if you get the same value for both the unique and duplicate values, then there is no duplication, and this will either be a One-to-Many or a One-to-One relationship. If the number of duplicate values is larger than the number of unique values, then this side of the relationship has duplicated values, and you will need to investigate the other side of the relationship. If the other side of the relationship yields unique values, this is a one-to-many relationship. If not, you have a many-to-many relationship.

Use the following SQL statement to test for potential M2M relationships:

1. In Sisense, open the relevant ecube file.
2. Click **Add Data**> **Custom SQL Expression**.
3. Enter and adjust the SQL statement below.

```
SELECT [Do I have duplications?]
FROM (
SELECT distinct_count(t1.col1)<>count(t1.col1) AS [Do
I have duplications?]
FROM [Table1] t1
UNION all
SELECT distinct_count(t2.col2)<>count(t2.col2)
FROM [Table2] t2) AS temp
GROUP BY [Do I have duplications?]
```

```

SELECT [Do I have duplications?]
FROM (
SELECT distinct_count(t1.col1)<>count(t1.col1)
AS [Do I have duplications?]
FROM [Table1] t1
UNION all
SELECT distinct_count(t2.col2)<>count(t2.col2)
FROM [Table2] t2) AS temp
GROUP BY [Do I have duplications?]

```

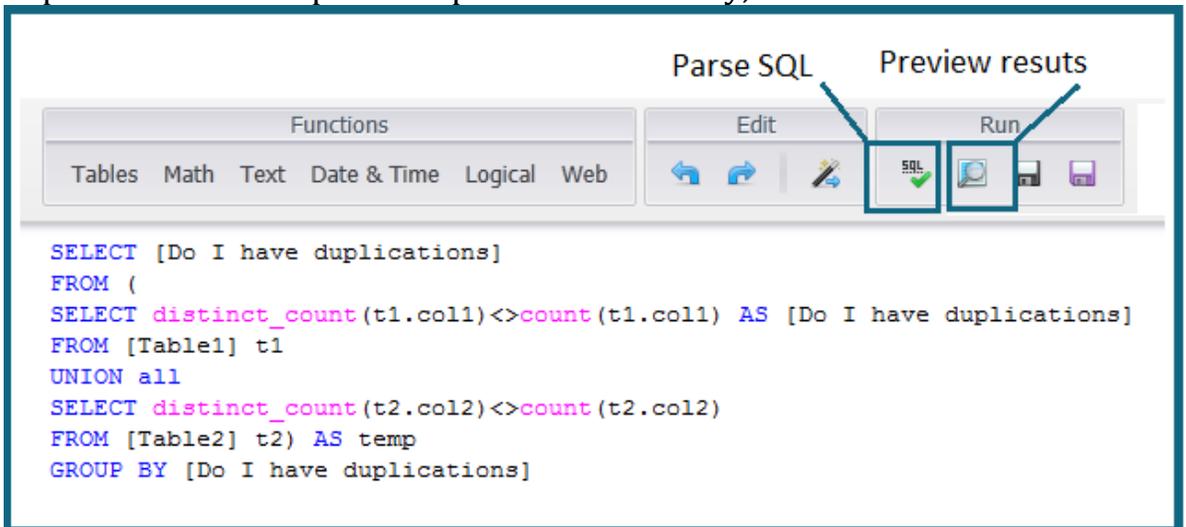
Replace with name of the First Table

Replace with name of merged column from the First Table

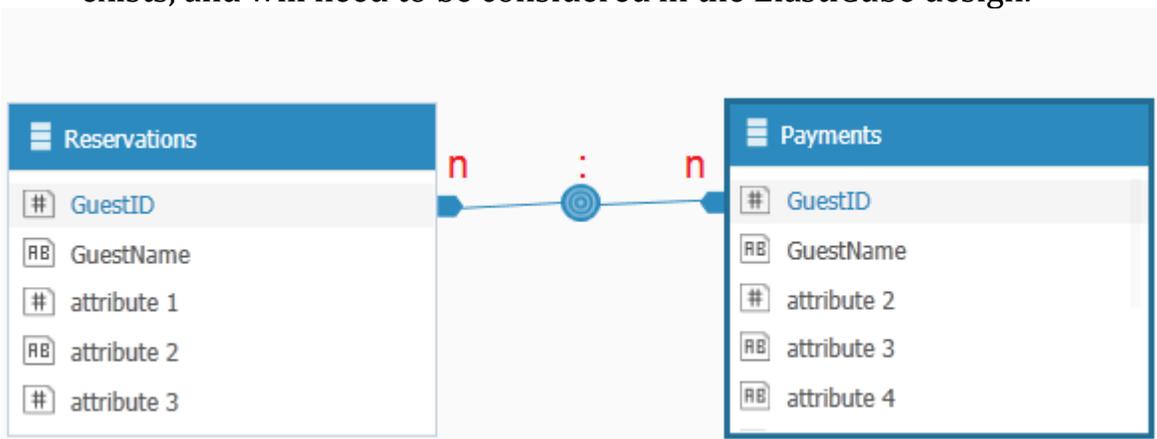
Replace with name of the Second Table

Replace with name of merged column from the Second Table

- In the top right of the expression editor window, click Parse SQL Expression. If the expression parses successfully, click Preview result table.



- If the returned result is 'True' in both lines, a many-to-many relationship exists, and will need to be considered in the ElastiCube design.



## Many-to-Many Relationship Prior to Resolution

If the two values are equal, all guest IDs appear only once, making all values unique. Even if the other side of the relationship has duplicate values for guest ID, this is still a One-To-Many relationship, where the unique values are on the reservations side, and the duplicate values are on the Payments side.

If there are more than two tables connected to this relationship, that is, if there are more than two tables merged on the same field, a few more options exist. The solution for the single many-to-many relationship will be a sub-problem of this scenario. In this case, you'll need to run the test on every table to see the uniqueness or duplication of the merged fields.

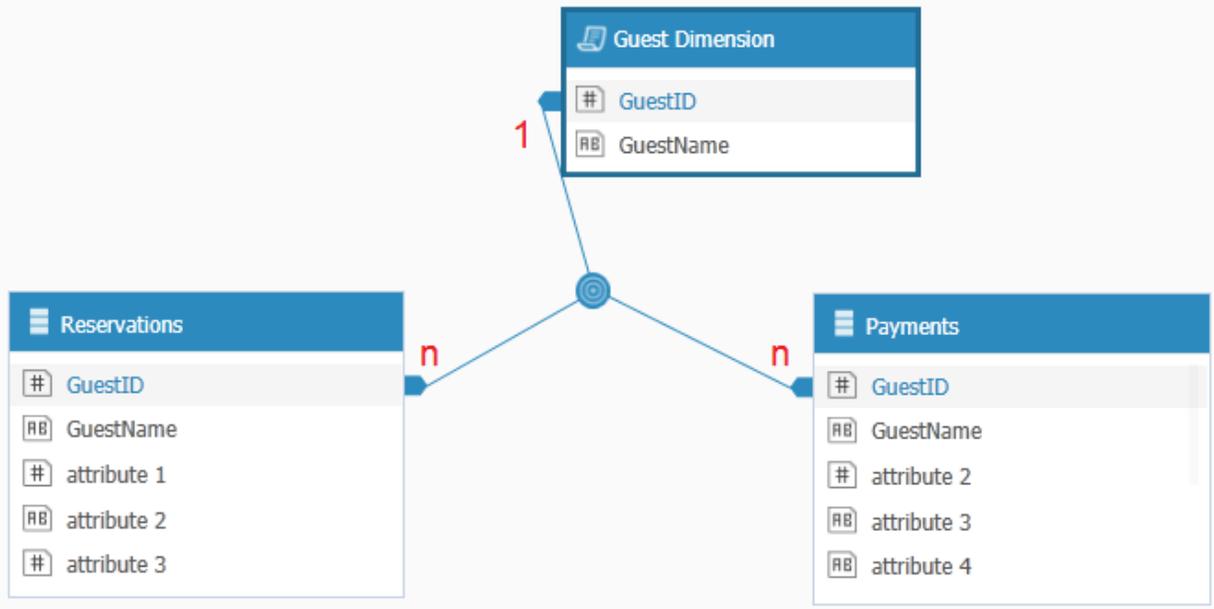
### **Two Tables with One Relationship**

This section describes two possible workarounds when you have a schema that includes two tables with one relationship:

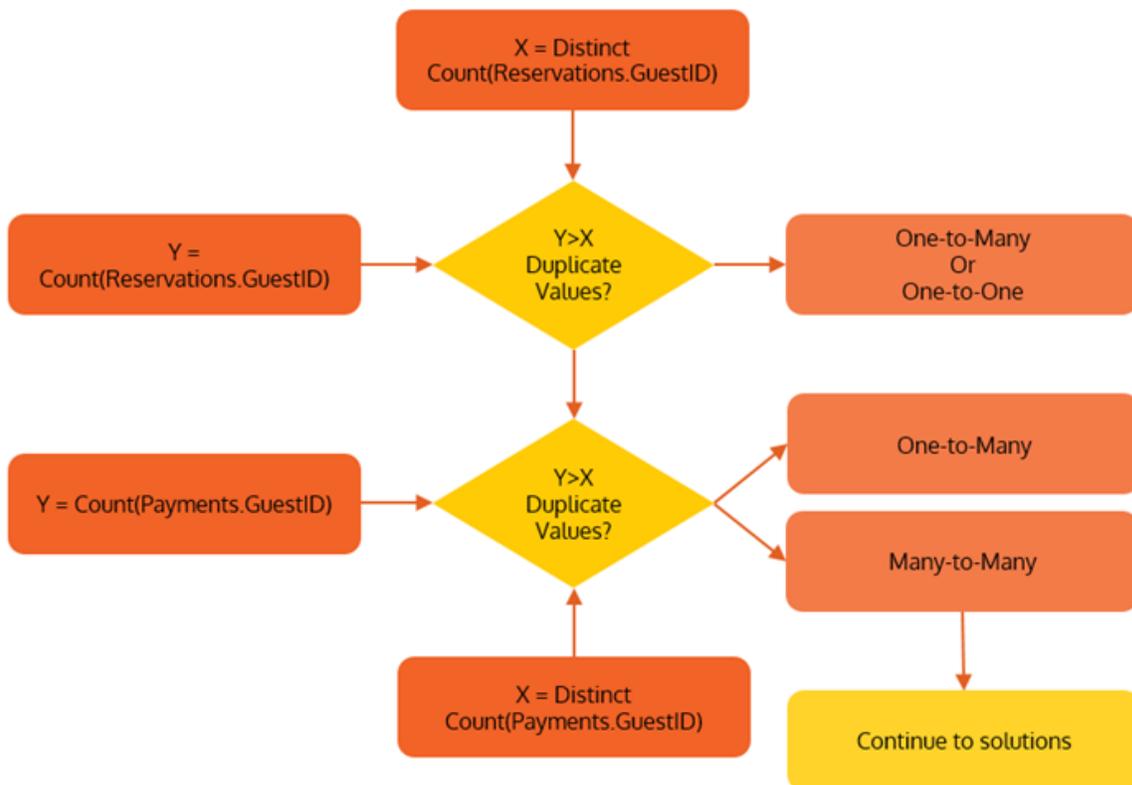
- Two Separate One-to-Many Relationships
- Creating an Aggregated Table

### **Two Separate One-to-Many Relationships**

The direct solution for such a problem is to break this relationship into two separate one-to-many relationships.



You can use the following diagram to understand the logic behind this testing:



1 – Create a custom SQL expression in the Elasticube. In the expression of this table select all the individual values for the identifier column from both sides. The expression should look like this:

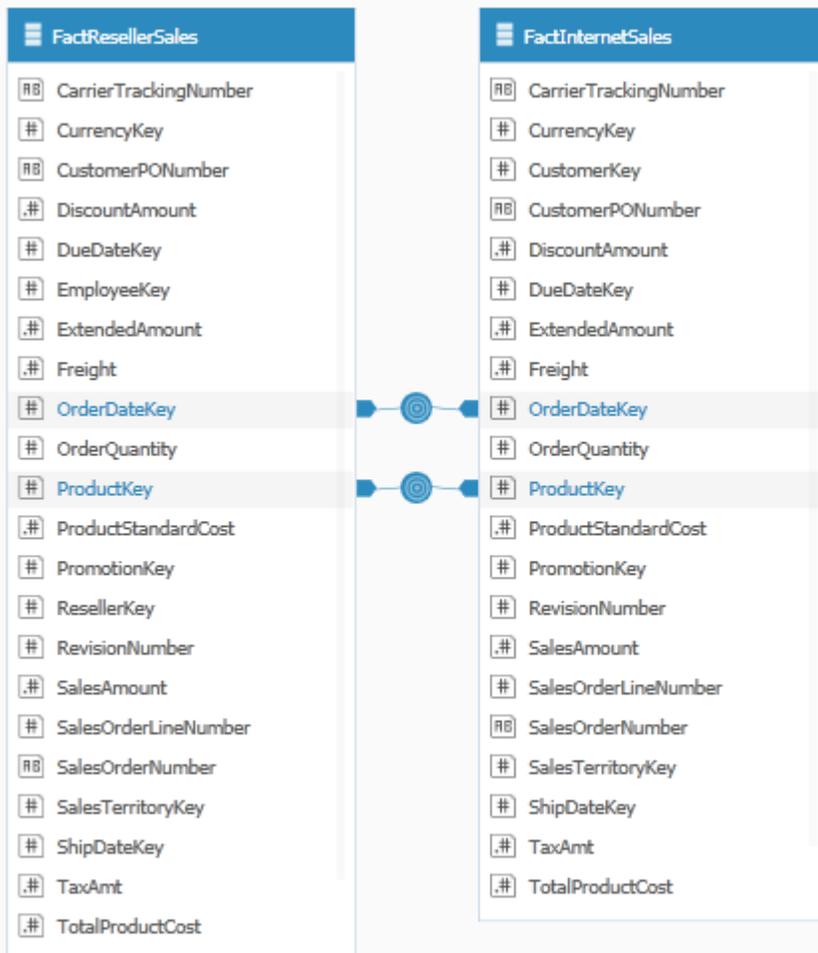
```
SELECT * FROM
(SELECT DISTINCT r.GuestID, r.GuestName
FROM [Reservations] r
UNION
SELECT DISTINCT p.GuestID, p.GuestName
FROM [Payments] p) AS G
```

This query will take all Guest ID values from both tables, and using the UNION statement, will bring in only the unique values from both tables, making this a complete list of all distinct Guest ID values.

2 - Merge the Guest ID field from the new 'linking' table to the other two Guest ID fields from the other two tables, thus creating two One-To-Many relationships. You can now use this Guest ID field as the rows or axes elements of a widget, pulling in the unique values from the new Guest Dimension, with measures from the two other tables. See image above.

### **Creating an Aggregated Table**

In situations where you have more than one fact table (a Fact table is a primary table containing the measures or fields used for calculations in the dashboard) in the Elasticube, there are several situations when an aggregated table can resolve a many-to-many relationship.



## Two fact tables

Assuming you want to segment your data according to a few different dimensions, creating relationships directly between these fields can and will create many-to-many relationships in one of two ways, according to the schema:

- Both tables don't hold unique values, and all values from one table are held in the second table. In this scenario either a linked dimension (as described in the first solution – Two Separate One-to-Many Relationships) or an aggregated table can be created which will hold all the unique values and the desired calculations for one of the tables. To create an aggregate table, create a custom SQL expression and aggregate values from the table that includes all values; its own, and the subset present in the other table with the following expression:

```

SELECT i.OrderDateKey, i.ProductKey,
sum(i.DiscountAmount), sum(i.SalesAmount),
avg(i.UnitPriceDiscountPct)
FROM [FactInternetSales] i
GROUP BY i.OrderDateKey, i.ProductKey
SELECT i.OrderDateKey, i.ProductKey, sum(i.DiscountAmount), sum(i.SalesAmount), avg(i.UnitPriceDiscountPct)
FROM [FactInternetSales] i
GROUP BY i.OrderDateKey, i.ProductKey

```

This custom SQL expression will select the distinct OrderDateKeys and their corresponding ProductKeys from the FactInternetSales, grouped by these fields, together with single value aggregations for the different fields, in this case, Discount Amount, Sales Amount and the average unit Price discount. After merging the OrderDateKey and Product Key to the two other tables, you will be able to pull the values from this new table into the rows or axes panel of a widget in the Sisense Web Application with measures and additional aggregations from the two other tables.

**Note:** The non-aggregated table needs to be a subset in terms of the primary fields from the aggregated table.

- Both tables don't include unique values, and there are different values for several fields in both the tables. Resolving this scenario incorporates both solutions mentioned above. In this scenario, create an aggregated table and a dimension table (both described above). The final resolution should look like this:



## Two Fact tables with a date dimension table and an aggregative Products table

### More than Two Tables with More than One Relationship

This section provides two possible workarounds when you have a schema that includes more than two tables with more than one relationship:

Options include:

- Using the Lookup function
- Concatenating two tables into one

### Using the Lookup Function

In most scenarios, you will aggregate values according to a given ID from the unique side of the relationship to the duplicate side. However in specific cases it'll be vice versa.

For example in the following scenario, in which we have three tables, and between them two one-to-many relationships, this can potentially create a many-to-many relationship, if you query the two leaf tables. This means that the query result table will have multiple rows which won't be distinguishable one from the other.



### Two consecutive M-to-M relationships

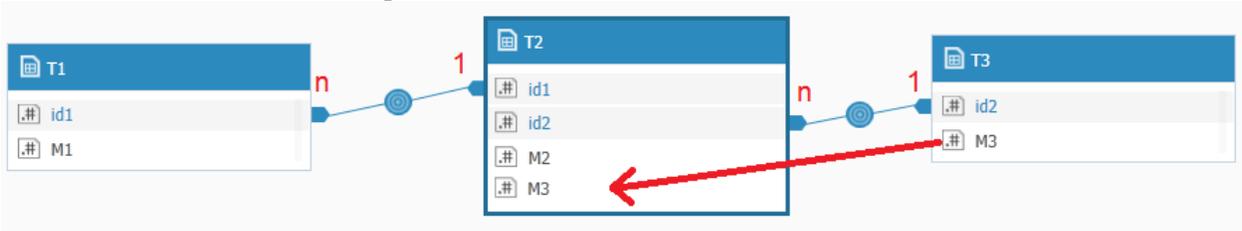
Using the Lookup Function, you can import values from a remote table by matching values in a different column. This will create a new column in the table where you want to perform an aggregation of a given field(s), with the matching value of the identifying field from the other table. Taking the example of tables T1, T2 and T3, we'd like to run a query that will display aggregations from the duplicate IDs from T1, with a measure from T3. If you run the query as is, you will get multiple values for the query's result set, and we will not be able to run this aggregation. To resolve this, use the Lookup function to import the values from T3 into T2 and then re-run the query only on tables T1 and T2. Using the lookup function, available in the 'Miscellaneous Functions' in the custom SQL editor, you can import the values of 'M3' from the 'T3' table into the 'T2' table. Create a new custom column, and use the Lookup function to import the values of attribute, In this case, the Lookup function should look like this:

```
Lookup([T3],[T3].[M3], [T2].id2,[T3].id2)
```

Running this statement in table T2 will import the matching values of M3 from T3 according to the matching results in ID2 between the two tables.

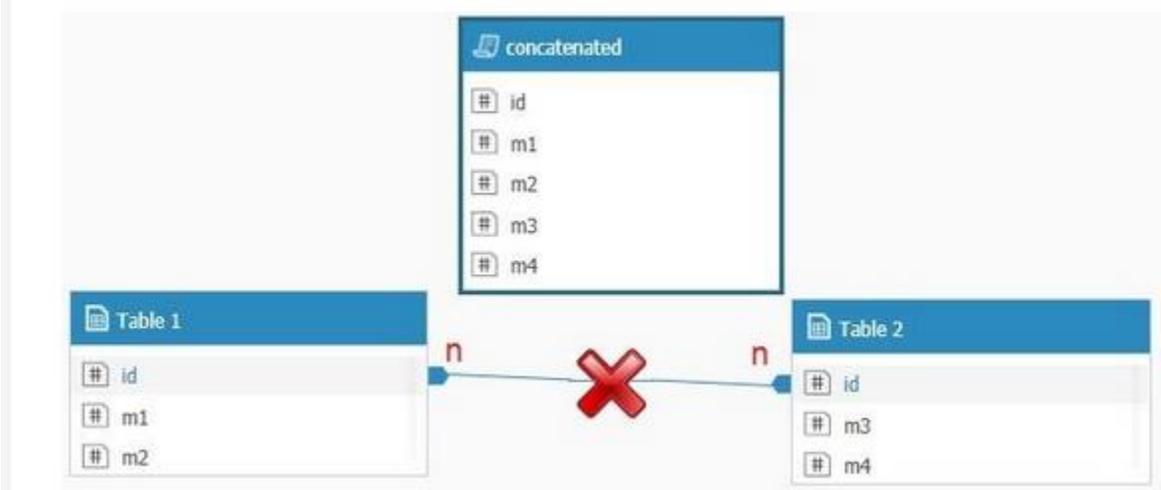
```
LOOKUP(remote_table,remote_result_column,current_match_column,  
remote_match_column)
```

Matches the current value with another value from a remote table. The result will be the value in remote\_result\_column for which the corresponding remote\_match\_column equals the current\_match\_column.



**Two consecutive M-to-O relationships after Lookup fix  
Concatenating Two Tables into One**

Assuming you have two separate tables with duplicate ID values in each, and each including different columns for each ID, you can create a new table including all values for every ID, and pull the aggregations from this new table. Note that the two original tables; Table\_1, Table\_2 have different columns.



**Concatenating tables**

Using the following SQL statement, you can import the data from both tables, with the IDs and the columns respectively:

```
SELECT s.id AS id, s.m1, s.m2, ToInt( NULL ) m3 , ToInt(
NULL ) m4
FROM [Table 1] s
UNION
```

```
SELECT t.id, ToInt( NULL ) , ToInt( NULL ) , t.m3, t.m4
FROM [Table 2] t
```

This will create a table with five columns:

Id

M1 (from table\_1)

M2 (from table\_1)

M3 (from table\_2)

M4 (from table\_2)

The values missing from each table respectively will be NULL's which will result in the following table.

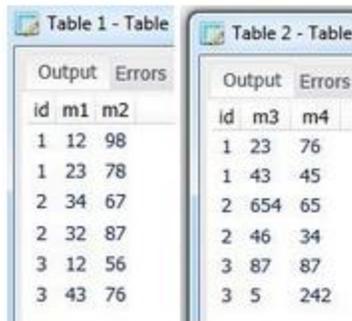
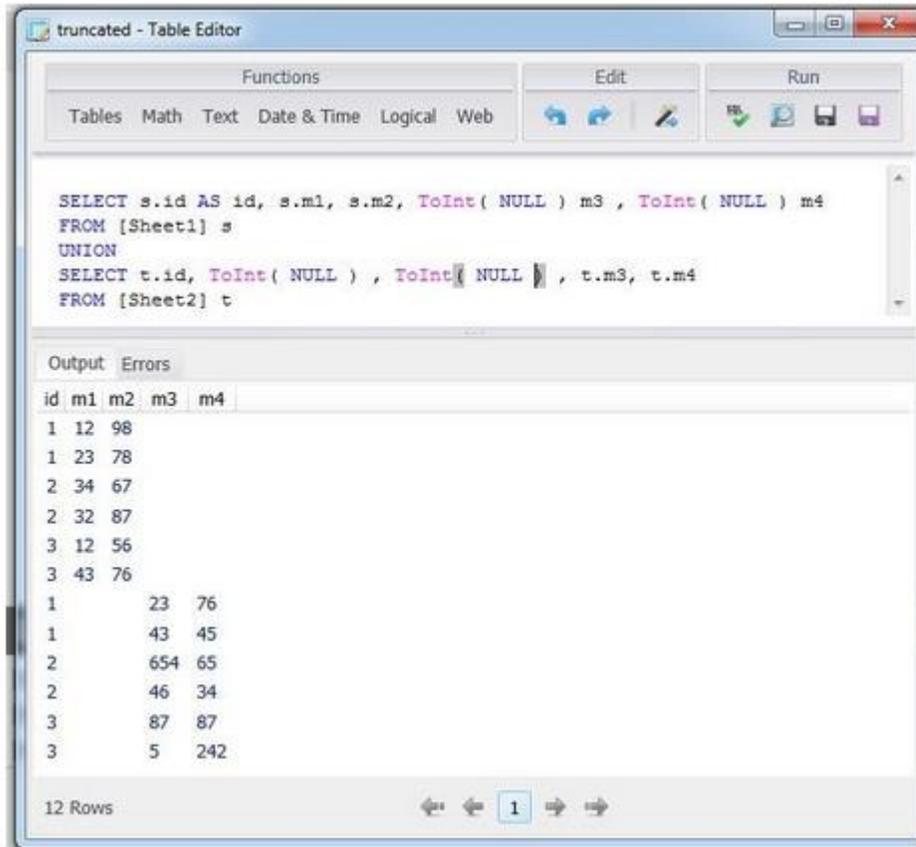
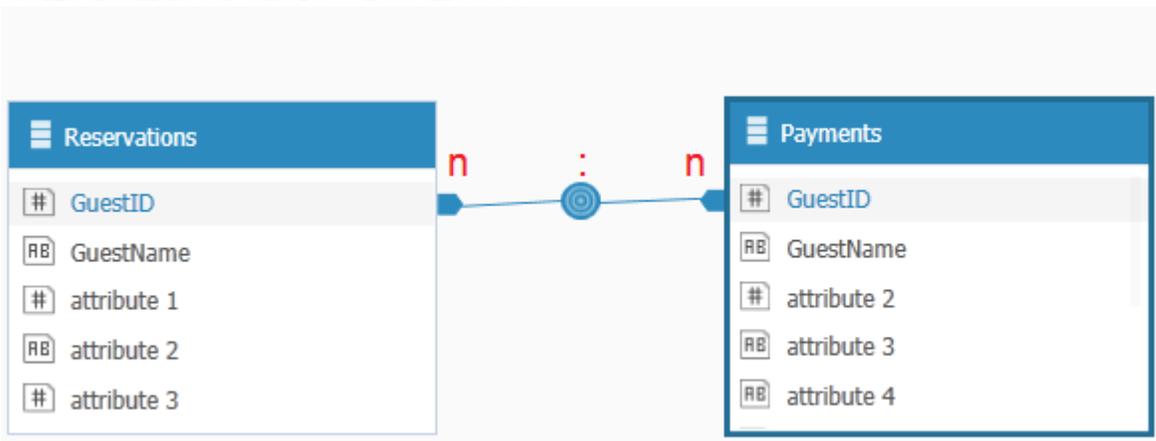


Table 1 - Table		
Output	Errors	
id	m1	m2
1	12	98
1	23	78
2	34	67
2	32	87
3	12	56
3	43	76

Table 2 - Table		
Output	Errors	
id	m3	m4
1	23	76
1	43	45
2	654	65
2	46	34
3	87	87
3	5	242



## Concatenated table – result set



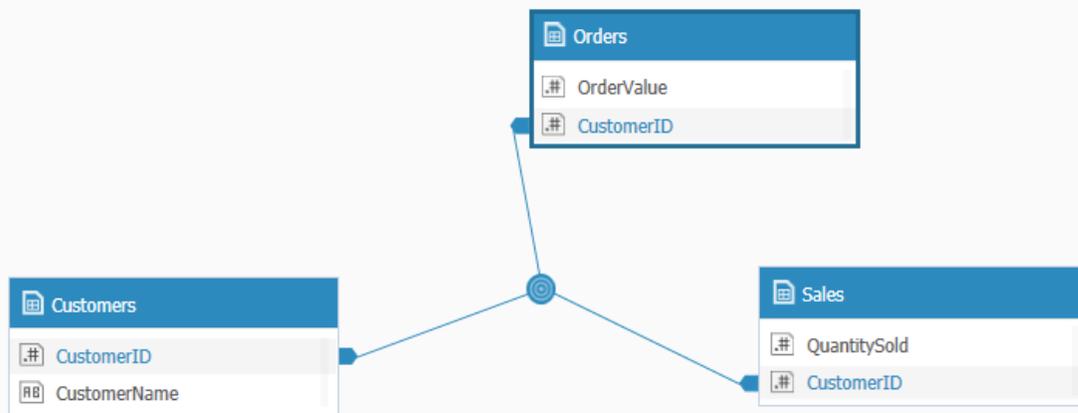
## Chasm and Fan Traps

**Note:** The image on this page were taken in the desktop version of Sisense, however, the same principles described on this page also apply to the web-based ElastiCube Manager.

Chasm and fan traps should be avoided when building your ElastiCube schemas.

# Chasm Traps

A chasm trap occurs when two “Many-to-One” joins converge on a single table, and the query includes measures from both leaf tables. As a result multiple rows are returned from the tables when processing the query.



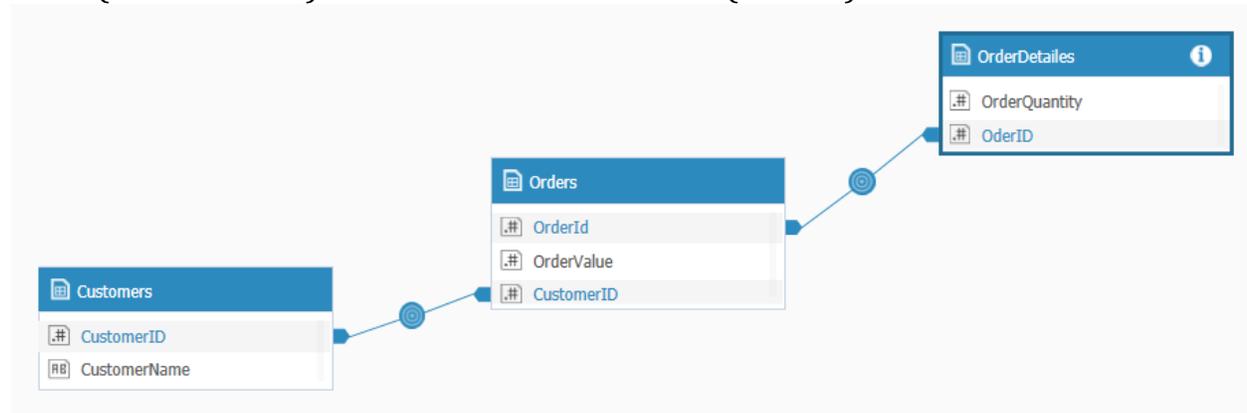
If you were to calculate both measures (Qty and Value) simultaneously, like in the following example, the values for Customers will be multiplied due to the inner join between the leaf tables, and the results may be incorrect:

```
SELECT [Customers].CustomerName,  
       sum ([Orders].OrderValue) AS Value,  
       sum ([Sales].QuantitySold) AS Qty  
FROM [Customers]  
JOIN [Orders] ON [Customers].CustomerID=[Orders].CustomerID  
JOIN [Sales] ON [Customers].CustomerID=[Sales].CustomerID  
GROUP BY [Customers].CustomerName
```

CustomerName	Value	Qty
Jason	1800	180
Daniel	600	60

# Fan Traps

A fan trap occurs when two “many-to-one” joins follow one another in master-detail form (OrderDetails), and the query includes a measure from both the leaf table (OrderDetails) and its immediate master (Orders).



If you try to aggregate both measures simultaneously (using the query below), you will probably get incorrect results:

```
SELECT [Customers].CustomerName,
       sum ([Orders].OrderValue) AS Value,
       sum ([OrderDetails].OrderQuantity) AS Qty
FROM [Customers] JOIN [Orders] ON [Customers].CustomerID= [Orders].CustomerID
join
[OrderDetails] ON [Orders].OrderID=[OrderDetails].OderID
GROUP BY
```

CustomerName	Value	Qty
Jason	700	90
Daniel	400	60
Mike	400	60

The “Qty” measure, corresponding to the leaf measure table (OrderDetails) is calculated correctly, but the “Value” measure, corresponding to the measure held in its master (Orders), is not. This is because we get the “Value” of every OrderID, which may inflate the expected results.

The web application translation module separates the calculations, and unions the results by generating a query for each of the measures' paths (path in terms of tables to go by). Then the web application translation module will group all the measures with the same tables' paths into one query and union it with all the other "same path measures" with different paths.

The described "Chasm Trap" can be prevented like this:

```
SELECT [Customers].CustomerName, sum ([Orders].OrderValue) AS Value, 0 AS Qty
FROM [Customers] JOIN [Orders] ON [Customers].CustomerID=[Orders].CustomerID
GROUP BY [Customers].CustomerName
union
SELECT [Customers].CustomerName, 0 AS Value, sum ([Sales].QuantitySold) AS Qty
FROM [Customers] JOIN [Sales] ON [Customers].CustomerID=[Sales].CustomerID
GROUP BY [Customers].CustomerName
```

CustomerName	Value	Qty
Jason	600	0
Daniel	300	0
Mike	300	0
Jason	0	60
Daniel	0	30

And the "Fan Trap" will be prevented this way:

```
SELECT [Customers].CustomerName, 0 AS Value, sum ([OrderDetails].OrderQuantity) AS Qty
FROM [Customers] JOIN [Orders] ON [Customers].CustomerID= [Orders].CustomerID
join
[OrderDetails] ON [Orders].OrderId=[OrderDetails].OderID
GROUP BY
[Customers].CustomerName
union
SELECT [Customers].CustomerName, sum ([Orders].OrderValue) AS Value, 0 AS Qty
FROM [Orders]
JOIN [Customers] ON [Orders].CustomerID=[Customers].CustomerID
GROUP BY CustomerName
```

CustomerName	Value	Qty
Jason	0	90
Daniel	0	60
Mike	0	60
Jason	600	0
Daniel	300	0
Mike	300	0

## Overview

When you build a data model, Sisense displays an intuitive build log that describes each step of the build.

Build logs are useful for investigating problems that occurred during your build. The log displays each table being built as its being built. You can click the table's name in the log to zoom in on the table in your schema. If a table fails to build, you can click its name in the log and then open the table within your data model to investigate potential causes for the failure.

**Note:** Your build log is available after your first build in Sisense V7.3 and later. Once your build is complete or if the build failed, you can export a detailed log file by clicking in the build log.

The screenshot shows a 'Build Process' window with a progress bar at the top. The progress bar is a horizontal line with six segments, all of which are filled with a teal color, indicating that all tasks are completed. To the right of the progress bar, the text 'Build Process 6 / 6 Tasks Completed | 00:00:45' is displayed. Further right are three icons: a download icon (a square with a downward arrow), a share icon (two arrows pointing outwards), and a close icon (an 'X'). Below the progress bar is a scrollable log area. The log starts with a header 'Build Succeeded' with a green checkmark icon on the left and the timestamp 'October 09, 2018 09:11' on the right. Below the header is a blue link '+ Create Dashboard'. The log then lists three tasks, each with a green checkmark icon, a duration, and a description: 1. '00:00:08 Finalizing build Finalization Completed'; 2. '00:00:02 Country 199 rows Successfully ended'; 3. '00:00:14 Commerce 613,002 rows Successfully ended'.

Build Process 6 / 6 Tasks Completed | 00:00:45

Build Succeeded October 09, 2018 09:11

+ [Create Dashboard](#)

- 00:00:08 Finalizing build  
Finalization Completed
- 00:00:02 Country 199 rows  
Successfully ended
- 00:00:14 Commerce 613,002 rows  
Successfully ended

The exported log is a JSON file that provides details about each table. For information about understanding the log file, see [Working with Build Logs](#). In addition, you can create a new dashboard from the build log or open any existing dashboards related to the data model.

**Build Process** 6 / 6 Tasks Completed | 00:00:45   

---

 **Build Succeeded** October 09, 2018 09:11

[+ Create Dashboard](#)

---

 00:00:08      Finalizing build  
Finalization Completed

---

 00:00:02       Country 199 rows  
Successfully ended

---

 00:00:14       Commerce 613,002 rows  
Successfully ended

**Working with Build Logs**

If you need to troubleshoot an issue with a build, you can download the log from the build log.



✓ **Build Succeeded**
October 09, 2018 09:11

[+ Create Dashboard](#)

---

✓ 00:00:08

Finalizing build

Finalization Completed

---

✓ 00:00:02

●

Country 199 rows

Successfully ended

---

✓ 00:00:14

●

Commerce 613,002 rows

Successfully ended

The build log is a JSON file that contains build objects. The build objects contain key:value pairs that describe the build process for each table.

As the reasons for build failures vary and may be quite complex, there isn't a single solution for resolving build errors. If you are not able to solve the issue yourself, save your logs and provide them to Sisense Support for additional assistance.

The table below describes some of the keys that might be included in your log and their possible values.

Key	Description
timestamp	The date and time the table was built.
verbosity	The type of information provided, such as Info, Warning, or Error. Info indicates that the build was performed. Warning indicates that something unexpected occurred, but the build was able to proceed. Error indicates that an error occurred and the build could not be continued.
type	The type of action being performed by Sisense. There are a variety of

Key	Description
	actions that may be performed, some are relevant to the table, others to Sisense itself.
title	The title of the action being performed.
serverId	Identifier for the server where the build was performed.
serverName	The name of the server where the build was performed.
cubeId	The name of the ElastiCube.
buildSeq	The place in the order of build operations.
typeValue	This is a generic object that contains information about the build process. The keys in this object vary depending on the build stage.
dbfarmpath	The location of the ElastiCube on the server.

## Transforming and Enriching Data

After you have imported data from your data sources, Data Designers can transform and enrich data in the ElastiCube so Data Designers can prepare easy-to-use data sets for Dashboard Designers. Sisense provides several features that make it easy to prepare, blend, and analyze data that can be fed to the Sisense Web Application and consumed by Viewers.

The following topics describe how you can transform and enrich your data in Sisense:

### Related Topics

- [Adding a Custom Column](#)
- [Editing a Custom Field](#)
- [Adding a Custom Table](#)
- [Editing a Custom Table](#)
- [Previewing Results](#)

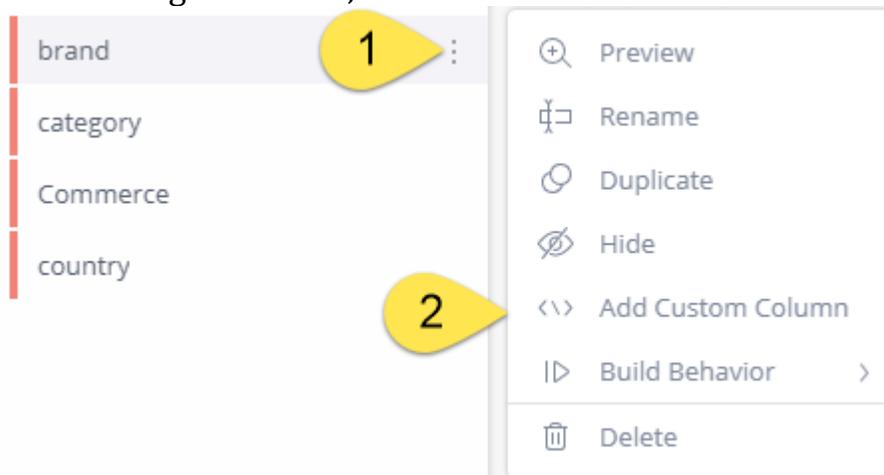
- [SQL Reference](#)
- [Function Reference](#)

## Adding a Custom Column

Sisense makes it easy for you to add new columns to existing tables. This can be useful if you need to combine data from different existing columns, and when you need to cleanse and prepare data. The new columns also provide Designers with additional fields to use as is, or as a basis for even more advanced calculations in their widgets. You can use SQL to customize the values contained within the custom columns.

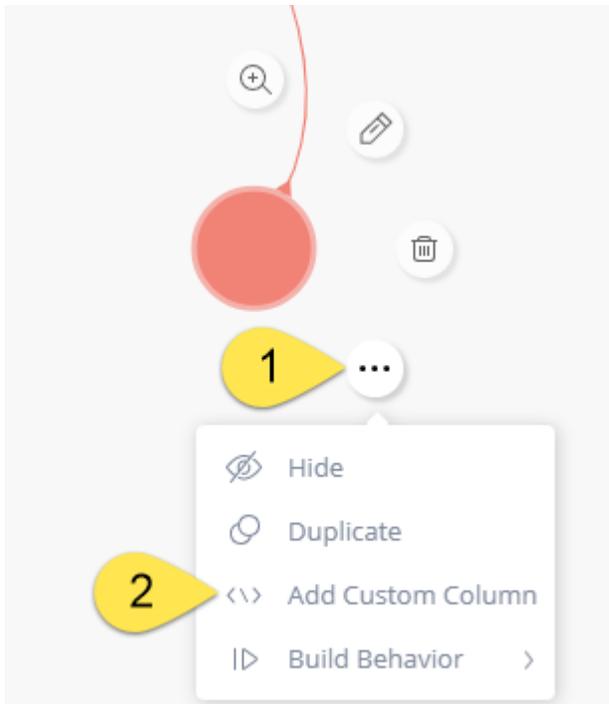
### To add a new column:

1. In the Navigation Pane, select :> **Add Custom Column.**



OR

In the ElastiCube model, select the table and click **...** and **Add Custom Column.**



The new column will be added to the table, and an SQL Editor is displayed where you define your custom SQL expression.

Qlik Sense interface showing the 'New Custom Column' dialog. The dialog has a 'Save' button and a 'Preview' button. Below the dialog, a table is displayed with two columns: 'Brand' and 'Brand ID'.

Brand	Brand ID
Thrubanover Holdings Corp.	1
Undimedat Worldwide	2
Reseller	3
APPLES	4
MINITENDO	5
Gropicklistor Group	6
GOZOROLA	7
Supweran Holdings	8
Tiperimor WorldWide	9
SUNSUNG	10
Haptumonon Inc	11
ABC	12
WHITEBERRY	13
MICRON	14
Unerommazz Direct	15

2. In the **New Custom Column** field, enter the name of your custom column and click .
3. In the SQL Editor, enter your custom expression.
4. Click  to add your column to the table.

## Custom Field Use Cases

Here are some typical scenarios for creating custom fields.

### Row Level Calculations

Calculate revenue from your sales data.

```
Gross Revenue = (Unit Price * QuantitySold)
```

```
Net Revenue = (Unit Price * QuantitySold) - (Unit Cost *  
QuantityPurchased)
```

### Time Difference Calculations

Work out the arrival time based on due and actual arrival dates.

```
daydiff (DueDate, ReceivedAt)
```

### Price Comparisons

Compare your price to the competitor's price.

```
((CompetitorPrice-MyPrice) / ((CompetitorPrice+MyPrice) / 2))
```

### Buckets

Create data buckets according to your required business break points.

```
CASE  
WHEN [PriceVsCompetitor%] < -0.05 THEN '-5%'  
WHEN [PriceVsCompetitor%] >= -0.05 AND  
[PriceVsAvgCompetitor%] <= 0.05 THEN '-5%/+5%'  
WHEN [PriceVsCompetitor%] > 0.05 THEN '+5%'  
end
```

### Attributes

Create attributes to translate data talk to more coherent categories.

```
CASE  
WHEN Region = 1 THEN 'USA'  
WHEN Region = 2 THEN 'EUR'  
WHEN Region = 3 THEN 'ASIA'
```

end

## **Date Conversions**

Convert dates into a numeric representation.

```
getyear(Date)*10000+getmonth(Date)*100+getday(Date) AS  
DateNum
```

Convert text into dates.

```
createdate( toint('20'+RIGHT([Timesheet_Date],2)),  
toint(LEFT(Timesheet_Date,2)),  
toint(StrBetween(Timesheet_Date,'/','/'))  
)
```

## **Date Period Comparisons Over Time**

Create month to date and year to date calculations.

[Click here](#) to read our support article.

## **Surrogate Keys**

```
EmployeeID+tostring([DateNum])+tostring(CustomerID)+tostring([Project_ID])
```

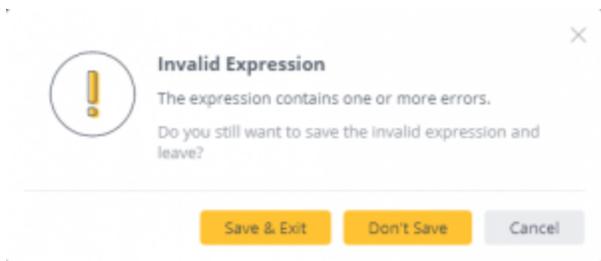
## **Importing from Another Table**

Use the lookup function to import a column from a different table. [Click here](#) to read more.

## **Editing Field Formulas**

Sisense makes it easy to edit custom tables. You can use SQL to customize the fields contained within a custom table.

When you edit a custom table, any changes you make are saved when you click **Save** unless the expression did not successfully parse. In this case, you are prompted to save a draft of the invalid expression.



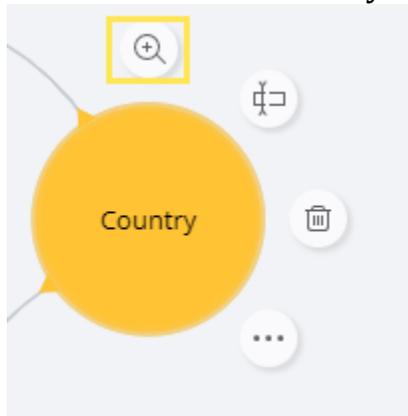
If you save the invalid expression, Sisense continues to use the last valid expression that was saved when building an ElastiCube, but the next time you open a preview of the table, the invalid expression will be displayed. This allows you to continue building working ElastiCubes while letting you save incomplete expressions for later.

If you wish to display the last valid expression, you can click the **Refresh** button in the menu bar and then press **Save**.



### To edit a custom table's SQL expression:

1. Select the custom table you want to edit and click .



The custom table area is displayed

2. Enter SQL statements to access tables and fields that exist in the schema. See related topics for more details on [SQL](#) and [Function References](#).
3. To view results based on the SQL statement, click  Preview .  
**Note:** At least one build including the relevant base table must have been completed in order to preview results.
4. Click **Save** to save your changes.

## Adding a Custom Table

With Sisense, you can easily create custom tables that can combine existing data. This is useful for preparing and cleansing data. Follow the steps below to learn how to add custom tables using SQL.

### To create a custom table:

1. In Sisense, open the schema of the ElastiCube you want to add to a custom table to.
2. In the menu bar of the schema, click  Custom . The New Custom Table area is displayed.
3. Enter the name of the custom table, and click .
4. Enter SQL statements to access tables and fields that exist in Sisense schema. See also [SQL Reference](#) and [Function Reference](#).
5. To view results based on the SQL statement, click  Preview .  
**Note:** You can save an incomplete or invalid SQL expression, however when you build the ElastiCube, only the last valid expression will be included. If you have not built an ElastiCube with a valid expression, then the invalid or incomplete expression will not be saved.
6. Click **Save** to save your changes.

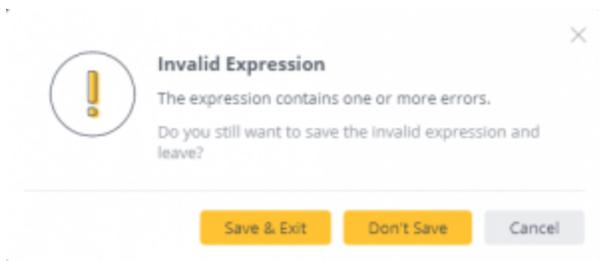
### Tip

- Press Ctrl + Space to complete an SQL statement.
- To make it easier to access table fields, give the table name an alias and reference the alias to bring up the related fields. For example SELECT A. FROM A.
- You can add comments in expressions by using the ‘-’ for a single line or /.../ for blocks.

## Editing SQL Statements

Sisense makes it easy to edit custom tables. You can use SQL to customize the columns contained within a custom table.

When you edit a custom table, any changes you make are saved when you click **Save** unless the expression did not successfully parse. In this case, you are prompted to save a draft of the invalid expression.



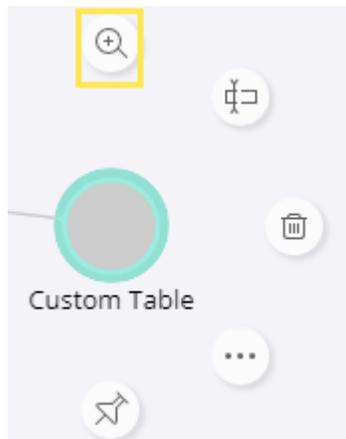
If you save the invalid expression, Sisense continues to use the last valid expression that was saved when building an ElastiCube, but the next time you open a preview of the table, the invalid expression will be displayed. This allows you to continue building working ElastiCubes while letting you save incomplete expressions for later.

If you wish to display the last valid expression, you can click the Refresh button in the menu bar and then press Save.



### To edit a custom table's SQL:

1. Select the custom table you want to edit and click .



The custom table area is displayed

2. Enter SQL statements to access tables and fields that exist in the schema. See related topics for more details on [SQL References](#) and [Function References](#).
3. To view results based on the SQL statement, click  **Preview**. (**Note:** At least one build including the relevant base table must have been completed in order to preview results.)
4. Click **Save** to save your changes.

### Tips

- Press Ctrl + Space to complete an SQL statement.
- To make it easier to access table fields, give the table name an alias, and reference the alias to bring up the related fields. For example Select A. From table as A.
- You can add comments in expressions by using the ‘-’ for a single line or /.../ for blocks.
- A good reference on compatible SQL commands is available on [Microsoft MSDN](#).

## Previewing Results

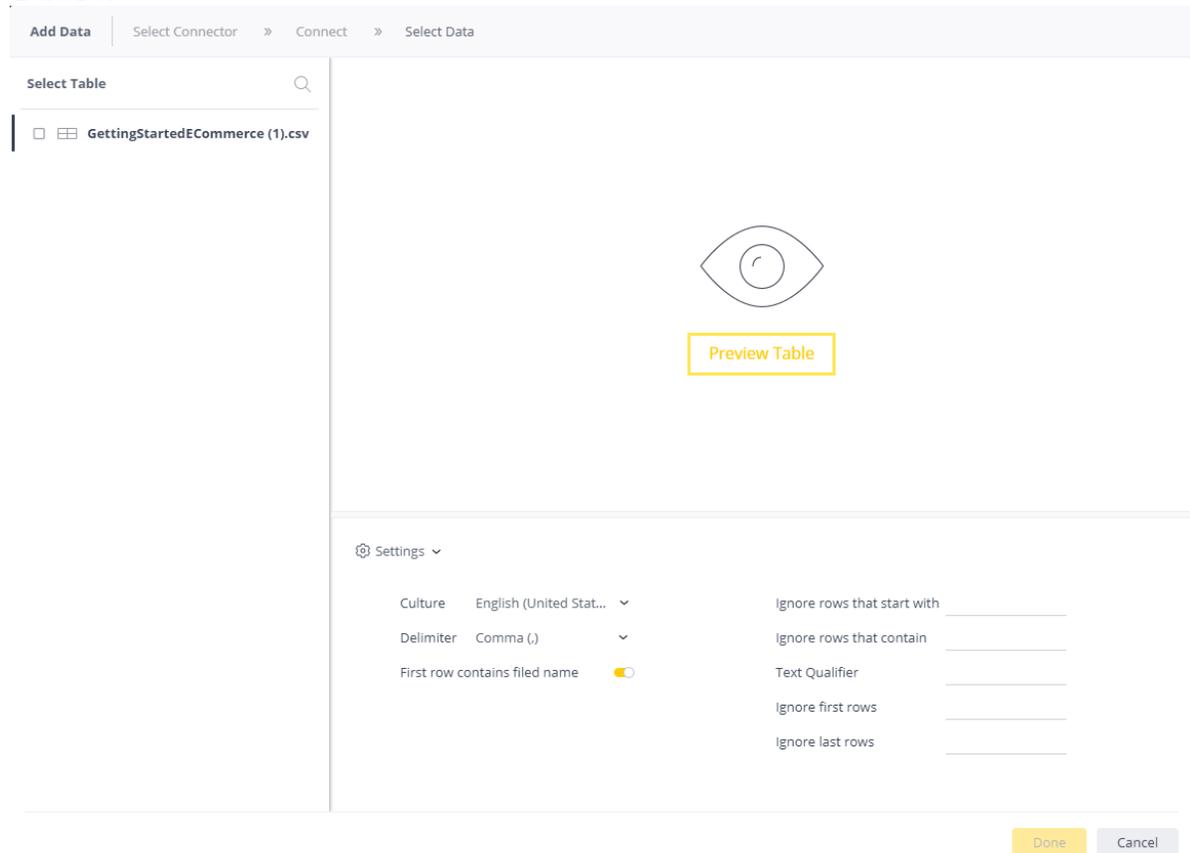
You can preview results as follows:

**Note:** If the results are not displayed for a custom query, this may indicate you have a many-to-many relationship in your data model. Review your model for any problems.

### To view results before connecting to a table:

1. In the schema of your ElastiCube, click .
2. Select the relevant connector.
3. Enter your login details and connect to the source data.
4. All tables and views associated with the database will appear in a new window. To preview data contained in a particular table, click **Preview**

## Table.



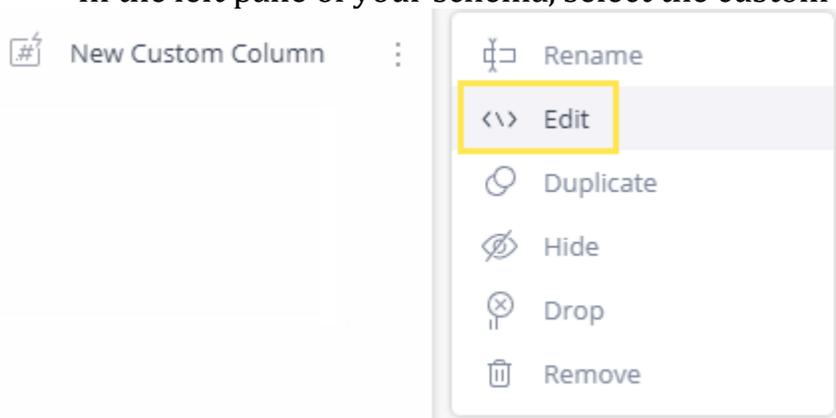
### To view results in a custom table:

- In the schema of your ElastiCube, click the custom table that you want to preview and click .

**Note:** At least one build including the relevant base table must be completed in order to preview results.

### To view results in a custom field:

- In the left pane of your schema, select the custom field and click **Edit**.



## SQL Reference

You can use SQL to create new tables and fields in Sisense.

SQL Structure	SQL Example
<pre>SELECT FieldName(s), Function(s) FROM TableName 1 INNER JOIN\ LEFT JOIN\RIGHT JOIN\FULL JOIN TableName 2 ON TableName 1. JoinField = TableName 2. JoinField WHERE FieldName Condition AND\OR FieldName Condition GROUP BY FieldName(s) ORDER BY FieldName(s) HAVING FieldName(s) Condition</pre>	<pre>SELECT CustomerName, ContactEmail, Count(Orders) FROM Customer INNER JOIN Order ON Customer.CustomerID = Order.CustomerID WHERE Order.OrderId BETWEEN 10 AND 100 AND Customer.CustomerName IN ('John','Mary','David') OR Customer.CustomerLastName LIKE 'Harrison' GROUP BY Customer.CustomerName ORDER BY Customer.CustomerLastName HAVING Count(Orders) &gt; 3</pre>

## Basic SQL Syntax Guide

SELECT STATEMENT	
SELECT	SELECT column_name(s) FROM table_name
SELECT *	SELECT *FROM table_name
SELECT DISTINCT	SELECT DISTINCT column_name(s) FROM table_name

Note: Nested SELECT statements will not work with an AND clause in Sisense.

FUNCTIONS AFTER SELECT CLAUSE	
FUNCTIONS	AVG() – Returns the average value COUNT() – Returns the number of rows MAX() – Returns the largest value MIN() – Returns the smallest value SUM() – Returns the sum

TABLE JOIN FUNCTIONS	
INNER JOIN	SELECT column_name(s) FROM table_name1 T1 INNER JOIN table_name2 T2 ON T1.column_name=T2.column_name
LEFT JOIN	SELECT column_name(s) FROM table_name1 T1 LEFT JOIN table_name2 T2 ON T1.column_name=T2.column_name
RIGHT JOIN	SELECT column_name(s)

	<pre>FROM table_name1 T1 RIGHT JOIN table_name2 T2 ON T1.column_name=T2.column_name</pre>
FULL JOIN	<pre>SELECT column_name(s) FROM table_name1 T1 FULL JOIN table_name2 T2 ON T1.column_name=T2.column_name</pre>
AS (alias)	<pre>SELECT column_name AS column_alias FROM table_nameor SELECT column_name FROM table_name AS table_alias</pre>
WHERE	<pre>SELECT column_name(s) FROM table_name WHERE column_name operator value</pre>

FUNCTIONS AFTER THE WHERE CLAUSE	
AND / OR	<pre>SELECT column_name(s) FROM table_name WHERE condition AND   OR condition</pre>
BETWEEN	<pre>SELECT column_name(s) FROM table_name WHERE column_name BETWEEN value1 AND value2</pre>
IN	<pre>SELECT column_name(s) FROM table_name WHERE column_name</pre>

	IN (value1,value2,..)
LIKE	SELECT column_name(s) FROM table_name WHERE column_name LIKE pattern

ORDERING AND GROUPING AFTER WHERE CONDITIONS	
GROUP BY	SELECT column_name, aggregate_function(column_name) FROM table_name WHERE column_name operator value GROUP BY column_name
ORDER BY	SELECT column_name(s) FROM table_name ORDER BY column_name [ASC   DESC]
HAVING	SELECT column_name, aggregate_function(column_name) FROM table_name WHERE column_name operator value GROUP BY column_name HAVING aggregate_function(column_name) operator value

CLAUSES TO COMBINE TABLES	
UNION	SELECT column_name(s) FROM table_name1 UNION SELECT column_name(s) FROM table_name2

UNION ALL	SELECT column_name(s) FROM table_name1 UNION ALL SELECT column_name(s) FROM table_name2
-----------	-----------------------------------------------------------------------------------------------

## See Also

[Basic SQL Guide](#)

[MSDN SQL Reference](#)

### **ElastiCube for Advanced Business Scenarios**

This section provides examples of more advanced scenarios, and the recommended methods for implementing the required business logic.

The examples are categorized into the following data manipulation methods: Integrating, Formatting and Enhancing.

# Integrating Data

Integrate and merge data from different sources into a single ElastiCube structure by identifying common keys between the different tables. Proper planning is important for merging the data; on the one hand, you need to avoid creating unnecessary relationships, and on the other hand, make sure you don't have any many to many relationships. Examples include:

- [Creating a Common Date Selection](#): Create a common date field from multiple date sets (from multiple data sources), and still keep the ability to use each original date field individually.
- [Financial Reporting](#): Bring in an additional data source to help analyze data from transaction systems. For example, Financial GL data will include all transactions, but may not have all the income statement or balance sheet reporting definitions.
- [Looking Up Values](#): Look up a value from one table and bring it into another table. For example, knowing how much a marketing campaign costs versus the sales opportunity amount is an important KPI to measure.

# Reformatting Data

Reformat field data to free space, and make fields more readable and usable. For example, convert a date field to numeric, or reduce the precision of real numbers.

You can reformat fields within the ElastiCube using a custom SQL expression.

- [Numeric Representation of Date Fields](#): Create a date table that is represented by a numeric representation instead of a date field to improve the query performance, as well as provide more flexibility, including the ability to filter a date range.

# Enhancing Data

[Enhance data](#) by adding attributes/records that did not exist in the original data source.

- [Calculating Derived Facts](#): Derived facts are additional facts that we calculate while importing or delivering the data.
- [Calendar vs. Fiscal Year](#) : Align a fiscal calendar with a Gregorian calendar.
- [Time Zone Conversion](#): Use a source table to convert dates and times from different time zones into a uniform data set.
- [Currency Conversion](#): Convert one currency into another using custom fields and a currency exchange rate table.
- [Current vs. Previous Period for Specific Date Range](#): Compare data such as sales between a current period and a past period.
- [Calculating the Number of Open Orders per Day](#): Check open sales orders where the order has been placed, but has not yet been delivered.
- [Slowly Changing Dimensions](#): Transactional data does not usually change, however the data that describes the associated dimensions may change. See how to manage dimensions that may be updated with new values within the data warehouse at different points in time.

## Integrating Data

**Note:** The images on this page were taken in the desktop version of Sisense, however, the same principles described on this page also apply to the web-based version of Sisense.

The following examples explain how to integrate and merge data from different sources into a single Elasticube structure. This requires properly planning how to merge the data; to avoid creating unnecessary relationships, while avoiding many to many relationships. Examples in this section:

- [Creating a Common Date Selection](#)
- [Financial Reporting](#)
- [Looking Up Values](#)

# Creating a Common Date Selection

## Business Case

When pulling together data from multiple sources, you will have a number of different dates. Marketing has a Campaign Date, Sales has an Opportunity Date and Finance has a GL Date.

## Modeling Challenge

This type of data leaves us with three sets of dates. Modeling it properly will allow you to select from one common date field while still leaving you the option to choose from one of the three date fields individually.

## Solution

Create a custom table that retrieves a unique list of the dates used between the three tables.

GL Entries	Marketing Campaigns	Sales Opportunities
GL Date	Marketing Campaign ID	Opportunity ID
GL Acct No	Marketing Campaign Date	Opportunity Stage
Region	Campaign Type	Opportunity Date
Amount	Campaign Cost	Amount

Use the following syntax:

```
Select [GL Date] AS [Common Date]
FROM [GL Entries]
```

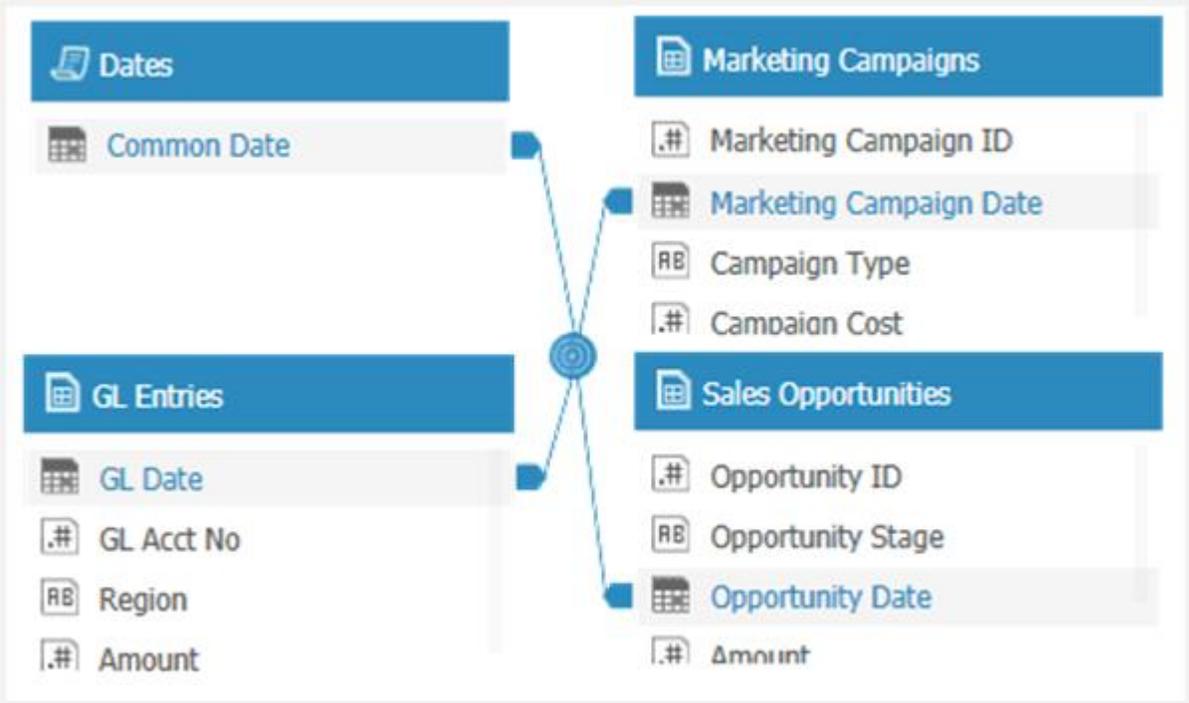
Union

```
Select [Marketing Campaign Date] AS [Common Date]
FROM [Marketing Campaigns]
```

Union

```
Select [Opportunity Date] AS [Common Date]
FROM [Sales Opportunities]
```

Link the four tables together:



## Results

This gives us the common date field to use. This allows, for example, to select a month that will narrow down the selections across all three tables.

# Financial Reporting

## Business Case

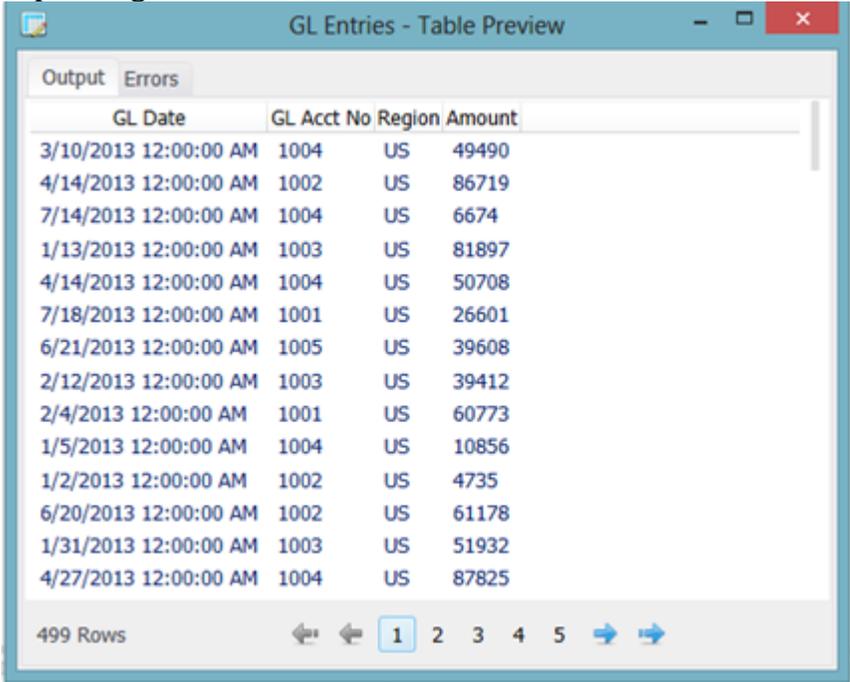
Transactional systems are meant for handling transactions and not for reporting and analysis. For example, Financial GL data will include all the transactions but may not include all income statement or balance sheet reporting definitions.

## Modeling Challenge

This data usually resides in other data sources or tables.

# Solution

Here is what our sample GL entries look like. It is not very useful to analysis and reporting.



GL Date	GL Acct No	Region	Amount
3/10/2013 12:00:00 AM	1004	US	49490
4/14/2013 12:00:00 AM	1002	US	86719
7/14/2013 12:00:00 AM	1004	US	6674
1/13/2013 12:00:00 AM	1003	US	81897
4/14/2013 12:00:00 AM	1004	US	50708
7/18/2013 12:00:00 AM	1001	US	26601
6/21/2013 12:00:00 AM	1005	US	39608
2/12/2013 12:00:00 AM	1003	US	39412
2/4/2013 12:00:00 AM	1001	US	60773
1/5/2013 12:00:00 AM	1004	US	10856
1/2/2013 12:00:00 AM	1002	US	4735
6/20/2013 12:00:00 AM	1002	US	61178
1/31/2013 12:00:00 AM	1003	US	51932
4/27/2013 12:00:00 AM	1004	US	87825

Another data source is needed to help define how the data will be analyzed and reported.

Start by bringing in another Data Source that contains details about the accounts and how they are categorized:



The two tables are linked on the account number. As there is one record in the GL Categories for multiple rows in the GL Entries, this is known as a One-to-Many Relationship.

The end results show that we can easily view the data in an organized fashion for analysis and reporting.

Revenue, COGs and Expenses <span style="float: right;">(i)</span>				
AcctGrouping2	AcctGrouping1	AcctNo	Sum Amount	
COGS	COGS	1009	35,434	
		1010	231,231	
		2001	44,634	
	Salaries	3001	3,543,432	
		3002	242,423	
		3003	523,543	
		3004	91,343	
	Expenses	Operationg Expenses	3009	8,546,214
3010			8,214,443	
Travel & Entertainment		3005	123,134	
		3006	54,354	
		3007	234,234	
		3008	4,374,573	
Revenue		License	1001	799,231
			1002	2,264,080
	Maintenance	1005	996,276	
		1006	1,286,252	
		1007	321,347	
		1008	6,534,321	
	Services	1003	3,319,673	
		1004	2,335,265	
Sum Amount			44,115,437	

# Looking Up Values

## Business Case

Sometimes it is necessary to look up a value from one table and bring it into another table. For example, knowing how much a Marketing Campaign costs versus the Sales Opportunity amount.

## Modeling Challenge

These two amounts typically reside in different systems. We need to look up the value from the Marketing system and bring it into the main Sales table.

## Solution

In the Sales Opportunities table, create a custom field that will allow to look up a value from the Marketing Campaigns table based on the Marketing Campaign ID. You can do this using the lookup function.

### **Reformatting Data**

By reformatting a field, you can create a more readable, and more usable format for analysis, as well as less space consuming in some cases. For example, convert a date field to a numeric field. You can reformat fields within the ElastiCube using a custom SQL expression. An example:

# **Numeric Representation of Date Fields**

## Business Case

Create a “Date” table using a numeric representation of a Date field instead of the Date field itself to improve query performances. In addition, a numeric representation of a Date can provide the ability of filtering the data by time range.

## Solution

To convert a Date (“4/21/2012 12:36:56 AM”) to an integer (20120421), use the following syntax:

```
10000*getyear (Date) +100*getmonth (Date) +getday (Date)
```

To get the hours and minutes too, use a bigint numeric representation. So, “4/21/2012 12:36:56 AM” will become: 201204210036, using the following syntax:

```
tobigint (100000000*getyear (DateTime) *+getmonth (DateTime) *  
1000000+getday (DateTime) *10000+100*gethour (DateTime) +getm  
inute (DateTime) )
```

### Enhancing Data

**Note:** The images on this page were taken in the desktop version of Sisense, however, the same principles described on this page also apply to the online version of Sisense.

The following examples explain how to add attributes and/or records that did not exist in the data source. Examples include:

- [Calculating Derived Facts](#)
- [Calendar vs. Fiscal Year](#)
- [Time Zone Conversion](#)
- [Currency Conversion](#)
- [Current vs. Previous Period for Specific Date Range](#)
- [Calculating the Number of Open Orders per Day](#)
- [Slowly Changing Dimensions](#)

# Calculating Derived Facts

## Business Case

Derived Facts are additional facts that we calculate while importing or delivering the data. For example:

$$\textit{Amount} = \textit{Qty} * \textit{UnitPrice}$$

$$\textit{Profit} = \textit{Amount} - \textit{Cost}$$

$$\textit{Inventory Ratio} = \frac{\textit{Sum}(\textit{Qty Sold})}{\textit{Sum}(\textit{Qty Ordered})}$$

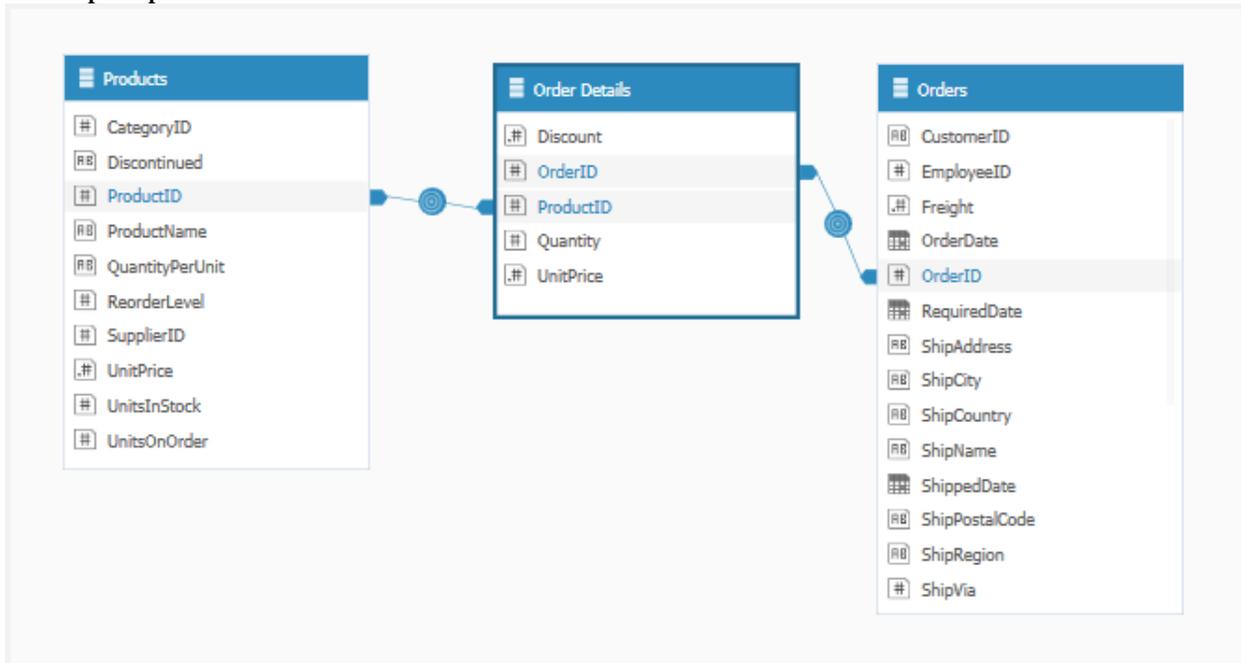
$$\textit{AVG Price} = \frac{\textit{Sum}(\textit{Qty} + \textit{UnitPrice})}{\textit{Sum}(\textit{Qty})}$$

## Modeling Challenge

You must decide whether to calculate the derived facts “on demand”, meaning in the web application, or in advance in the ElastiCube. Take into consideration that calculating ‘On Demand’ Derived Facts in the web application can enable more dynamic filtering, while calculating them in the ElastiCube stage will save query time when retrieving the data, and enforce calculation consistency, especially with non-trivial facts. This is due to the fact that the dashboard designer/end users will receive consistent results for measures, instead of having to create the complex measures individually, by their own understanding.

## Solution

In the following schema you can create a derived fact to calculate the inventory ratio per product.



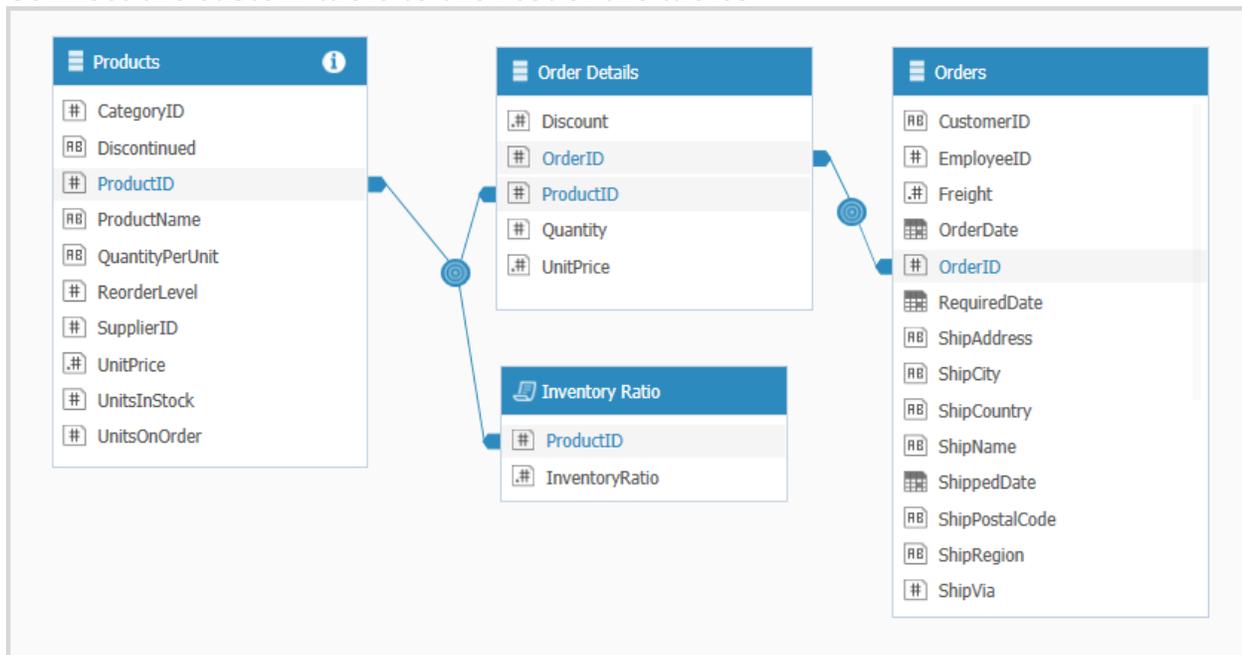
Create a [custom table](#) using an SQL Expression that joins the “Order Details” table with the “Products” table and returns the division result of “Quantity” and “UnitOnOrder”, with the following Syntax:

```
SELECT
[Products].ProductID,
toFloat (sum (UnitsOnOrder)) / tofloat (sum (Quantity)) AS
InventoryRatio
FROM [Products] JOIN [Order Details]
ON [Products].ProductID=[Order Details].ProductID
GROUP BY [Products].ProductID
HAVING
toFloat (sum (UnitsOnOrder)) / tofloat (sum (Quantity)) > 0
```

The result table will give the desired results:

ProductID	InventoryRatio
11	1.614731
31	2.555476
49	2.423077
74	0.8754209
2	1.66509
32	2.020202
21	1.535433
37	2.4
70	0.4773562
56	0.3958828
43	0.4827586
68	0.4255319
3	2.560976
64	3.243243
66	3.34728
45	1.929134
48	3.043478

Connect the custom table to the rest of the tables:



**Note:** You can also add the “InventoryRatio” measure to the “Products” table using the Lookup() function by “ProductID”.

# **Calendar vs. Fiscal Year**

## Business Case

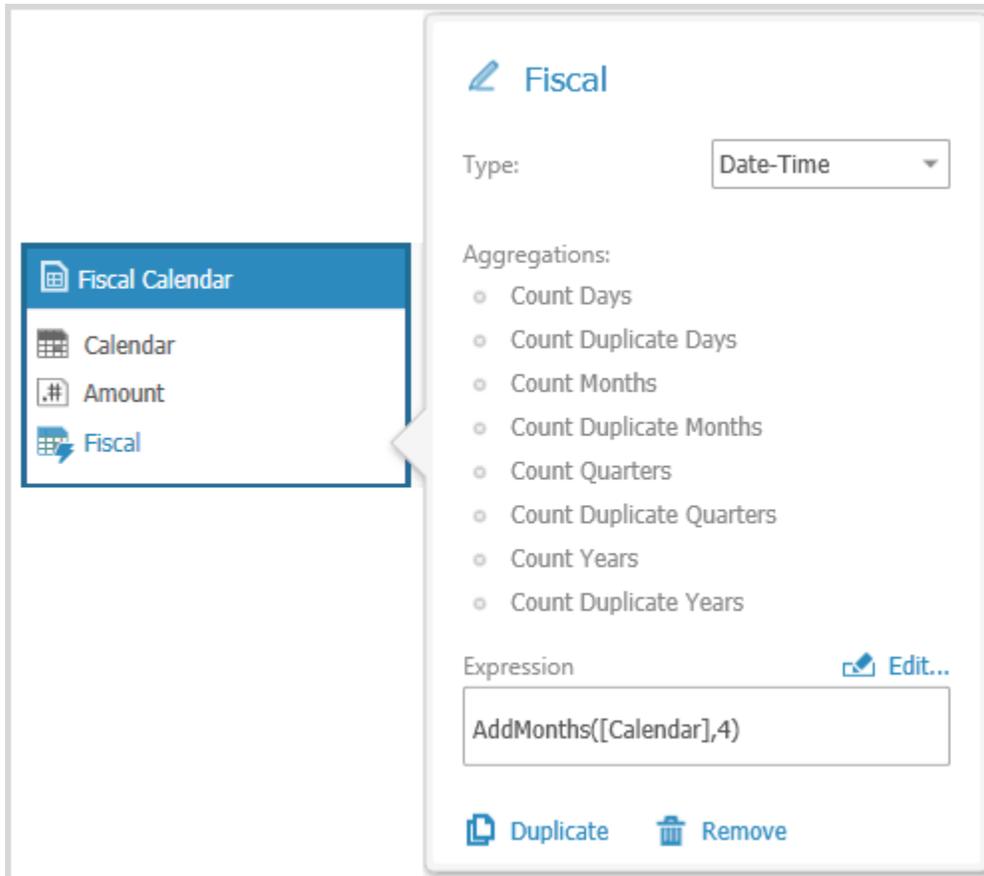
A large number of companies use a fiscal calendar that does not comply with the Gregorian 12-month calendar.

## Modeling Challenge

This requires modeling the data properly so that the data can be reported or analyzed via the normal calendar or via the revised fiscal calendar.

## Solution

In this example, let's assume that the Fiscal Calendar starts on September 1st. So if we are in the calendar year of 2013, then the fiscal year of 2014 starts September 1st. To accomplish this, we create a custom field that takes the date field and adds four months to it.



The screenshot shows a configuration window for a field named "Fiscal". On the left, a list of fields includes "Fiscal Calendar", "Calendar", "Amount", and "Fiscal". The "Fiscal" field is selected. The configuration panel on the right shows the following details:

- Type:** Date-Time
- Aggregations:**
  - Count Days
  - Count Duplicate Days
  - Count Months
  - Count Duplicate Months
  - Count Quarters
  - Count Duplicate Quarters
  - Count Years
  - Count Duplicate Years
- Expression:** AddMonths([Calendar],4)
- Buttons:** Duplicate, Remove

When you create a pivot table in the web application, you will see that the new year (2014) starts in September using the Fiscal field.

Calendar Vs. Fiscal Year i

Days in Date	Calendar Year	Fiscal Year	Total Amount
08/29/2013	2013	2013	35,756
08/30/2013	2013	2013	3,687
08/31/2013	2013	2013	35,617
09/01/2013	2013	2014	1,108
09/02/2013	2013	2014	91,387
09/03/2013	2013	2014	11,330
09/04/2013	2013	2014	69,273

# **Time Zone Conversion**

## Business Case

In many cases, we need to generate reports based on data from different time zones.

## Modeling Challenge

When working with different time zones, the challenge is to store all of the business transactions in an absolute time reference that does not change with the seasons, locations (for instance – GMT), or daylight saving. Therefore, the absolute transition time is a combination of location and date.

## Solution

The aim is to add an “absolute time” field to every business transaction, based on its location and time.

### Step 1 – Create a Reference Source Table

Create a source table (database table / Excel / CSV) that contains the countries and cities that exist in the database, a numeric representation of timestamp range to determine if the transaction belongs to daylight savings time or not (see this [web site](#)), and the UTC to allow the conversion to GMT.

For example:

Country	City	DST_From	DST_To	UTC
USA	Seattle	20120311.2	20121103.1	-7
USA	Seattle	20121103.1	20130310.2	-8
USA	Seattle	20130310.2	20131027.1	-7
USA	Seattle	20131027.1	20140309.2	-8
UK	London	20120325.1	20121028.2	0
UK	London	20121028.2	20130330.1	1
UK	London	20130330.1	20131027.2	0
UK	London	20131027.2	20140330.1	1

### Step 2 – Add a Numeric Representation of the OrderDate

To associate the Order Date with its UTC, create a custom field of type “Decimal” with a numeric representation of the Date timestamp, using this SQL statement:

```
getyear (OrderDate) *10000+getmonth (OrderDate) *100+getday (OrderDate) +ToDouble (gethour (OrderDate) ) /100
```

The result table should look like this:

CustomerID	OrderID	EmployeeID	OrderDate	DateNum	ShipAddress	ShipCity	ShipCountry	Freight
VINET	10248	5	7/4/2012 12:00:00 AM	20120704	59 rue de l'Abbaye	Reims	France	32.38
TOMSP	10249	6	7/5/2012 12:00:00 AM	20120705	Luisenstr. 48	Münster	Germany	11.61
HANAR	10250	4	7/8/2012 12:00:00 AM	20120708	Rua do Paço, 67	Rio de Janeiro	Brazil	65.83
VICTE	10251	3	7/8/2012 12:00:00 AM	20120708	2, rue du Commerce	Lyon	France	41.34
SUPRD	10252	4	7/9/2012 12:00:00 AM	20120709	Boulevard Tirou, 255	Charleroi	Belgium	51.3
HANAR	10253	3	7/10/2012 12:00:00 AM	20120710	Rua do Paço, 67	Rio de Janeiro	Brazil	58.17
CHOPS	10254	5	7/11/2012 12:00:00 AM	20120711	Hauptstr. 31	Bern	Switzerland	22.98
RICSU	10255	9	7/12/2012 12:00:00 AM	20120712	Starenweg 5	Genève	Switzerland	148.33
WELLI	10256	3	7/15/2012 12:00:00 AM	20120715	Rua do Mercado, 12	Resende	Brazil	13.97
HILAA	10257	4	7/16/2012 12:00:00 AM	20120716	Carrera 22 con Ave. Carlos Soublette #8-35	San Cristóbal	Venezuela	81.91
ERNSH	10258	1	7/17/2012 12:00:00 AM	20120717	Kirchgasse 6	Graz	Austria	140.51
CENTC	10259	4	7/18/2012 12:00:00 AM	20120718	Sierras de Granada 9993	México D.F.	Mexico	3.25
OTTIK	10260	4	7/19/2012 12:00:00 AM	20120719	Mehrheimerstr. 369	Köln	Germany	55.09
QUEDE	10261	4	7/19/2012 12:00:00 AM	20120719	Rua da Panificadora, 12	Rio de Janeiro	Brazil	3.05
RATTC	10262	8	7/22/2012 12:00:00 AM	20120722	2817 Milton Dr.	Albuquerque	USA	48.29
ERNSH	10263	9	7/23/2012 12:00:00 AM	20120723	Kirchgasse 6	Graz	Austria	146.06
FOLKO	10264	6	7/24/2012 12:00:00 AM	20120724	Åkergatan 24	Bräcke	Sweden	3.67
BLONP	10265	2	7/25/2012 12:00:00 AM	20120725	24, place Kléber	Strasbourg	France	55.28
WARTH	10266	3	7/26/2012 12:00:00 AM	20120726	Torikatu 38	Oulu	Finland	25.73
FRANK	10267	4	7/29/2012 12:00:00 AM	20120729	Berliner Platz 43	München	Germany	208.58
GROSR	10268	8	7/30/2012 12:00:00 AM	20120730	5ª Ave. Los Palos Grandes	Caracas	Venezuela	66.29
WHITC	10269	5	7/31/2012 12:00:00 AM	20120731	1029 - 12th Ave. S.	Seattle	USA	4.56
WARTH	10270	1	8/1/2012 12:00:00 AM	20120801	Torikatu 38	Oulu	Finland	136.54
SPLIR	10271	6	8/1/2012 12:00:00 AM	20120801	P.O. Box 555	Lander	USA	4.54
RATTC	10272	6	8/2/2012 12:00:00 AM	20120802	2817 Milton Dr.	Albuquerque	USA	98.03
QUICK	10273	3	8/5/2012 12:00:00 AM	20120805	Taucherstraße 10	Cunewalde	Germany	76.07
VINET	10274	6	8/6/2012 12:00:00 AM	20120806	59 rue de l'Abbaye	Reims	France	6.01
MAGAA	10275	1	8/7/2012 12:00:00 AM	20120807	Via Ludovico il Moro 22	Bergamo	Italy	26.93
TORTU	10276	8	8/8/2012 12:00:00 AM	20120808	Avda. Azteca 123	México D.F.	Mexico	13.84
MORGK	10277	2	8/9/2012 12:00:00 AM	20120809	Heerstr. 22	Leipzig	Germany	125.77

### Step 3 – Join between the Two Tables

The third step includes creating a custom SQL expression that joins between the two tables and creating the “Absolute Time” custom field within it (“GMTDate”). This is to create a synchronization between all the transactions. The custom field will be created using the “add hours” function with the matching UTC value. See the following script:

```
SELECT
[Orders].CustomerID,
[Orders].EmployeeID,
[Orders].Freight,
[Orders].OrderDate,
[Orders].OrderID,
[Orders].ShipAddress,
```

```

[Orders].ShipCity,
[Orders].ShipCountry,
AddHours([Orders].OrderDate,[GMT Conversion.csv].UTC)
AS GMTDate
FROM [Orders]
JOIN
[GMT Conversion.csv]
ON
[Orders].ShipCity=[GMT Conversion.csv].City AND
[Orders].ShipCountry=[GMT Conversion.csv].Country AND
[Orders].DateNum>=[GMT Conversion.csv].DST_From AND
[Orders].DateNum<[GMT Conversion.csv].DST_To

```

The result table will look like this:

CustomerID	EmployeeID	Freight	OrderDate	OrderID	ShipAddress	ShipCity	ShipCountry	GMTDate
WHITC	5	4.56	7/31/2012 12:00:00 AM	10269	1029 - 12th Ave. S.	Seattle	USA	7/30/2012 4:00:00 PM
WHITC	5	4.56	7/31/2012 12:00:00 AM	10269	1029 - 12th Ave. S.	Seattle	USA	7/30/2012 5:00:00 PM
WHITC	5	4.56	7/31/2012 12:00:00 AM	10269	1029 - 12th Ave. S.	Seattle	USA	7/30/2012 4:00:00 PM
WHITC	5	4.56	7/31/2012 12:00:00 AM	10269	1029 - 12th Ave. S.	Seattle	USA	7/30/2012 5:00:00 PM
BSBEV	7	22.77	8/26/2012 12:00:00 AM	10289	Fauntleroy Circus	London	UK	8/26/2012 1:00:00 AM
BSBEV	7	22.77	8/26/2012 12:00:00 AM	10289	Fauntleroy Circus	London	UK	8/26/2012 12:00:00 AM
BSBEV	7	22.77	8/26/2012 12:00:00 AM	10289	Fauntleroy Circus	London	UK	8/26/2012 1:00:00 AM
BSBEV	7	22.77	8/26/2012 12:00:00 AM	10289	Fauntleroy Circus	London	UK	8/26/2012 12:00:00 AM
WHITC	4	23.29	11/1/2012 12:00:00 AM	10344	1029 - 12th Ave. S.	Seattle	USA	10/31/2012 4:00:00 PM
WHITC	4	23.29	11/1/2012 12:00:00 AM	10344	1029 - 12th Ave. S.	Seattle	USA	10/31/2012 5:00:00 PM
WHITC	4	23.29	11/1/2012 12:00:00 AM	10344	1029 - 12th Ave. S.	Seattle	USA	10/31/2012 4:00:00 PM
WHITC	4	23.29	11/1/2012 12:00:00 AM	10344	1029 - 12th Ave. S.	Seattle	USA	10/31/2012 5:00:00 PM
SEVES	5	288.43	11/21/2012 12:00:00 AM	10359	90 Wadhurst Rd.	London	UK	11/21/2012 1:00:00 AM
SEVES	5	288.43	11/21/2012 12:00:00 AM	10359	90 Wadhurst Rd.	London	UK	11/21/2012 12:00:00 AM
SEVES	5	288.43	11/21/2012 12:00:00 AM	10359	90 Wadhurst Rd.	London	UK	11/21/2012 1:00:00 AM
SEVES	5	288.43	11/21/2012 12:00:00 AM	10359	90 Wadhurst Rd.	London	UK	11/21/2012 12:00:00 AM
EASTC	1	71.97	11/26/2012 12:00:00 AM	10364	35 King George	London	UK	11/26/2012 1:00:00 AM
EASTC	1	71.97	11/26/2012 12:00:00 AM	10364	35 King George	London	UK	11/26/2012 12:00:00 AM
EASTC	1	71.97	11/26/2012 12:00:00 AM	10364	35 King George	London	UK	11/26/2012 1:00:00 AM
EASTC	1	71.97	11/26/2012 12:00:00 AM	10364	35 King George	London	UK	11/26/2012 12:00:00 AM
SEVES	1	22.21	12/9/2012 12:00:00 AM	10377	90 Wadhurst Rd.	London	UK	12/9/2012 1:00:00 AM
SEVES	1	22.21	12/9/2012 12:00:00 AM	10377	90 Wadhurst Rd.	London	UK	12/9/2012 12:00:00 AM
SEVES	1	22.21	12/9/2012 12:00:00 AM	10377	90 Wadhurst Rd.	London	UK	12/9/2012 1:00:00 AM
SEVES	1	22.21	12/9/2012 12:00:00 AM	10377	90 Wadhurst Rd.	London	UK	12/9/2012 12:00:00 AM
SEVES	2	34.86	12/19/2012 12:00:00 AM	10388	90 Wadhurst Rd.	London	UK	12/19/2012 1:00:00 AM
SEVES	2	34.86	12/19/2012 12:00:00 AM	10388	90 Wadhurst Rd.	London	UK	12/19/2012 12:00:00 AM
SEVES	2	34.86	12/19/2012 12:00:00 AM	10388	90 Wadhurst Rd.	London	UK	12/19/2012 1:00:00 AM

## **Step 4 - Make Schema Adjustments**

For the next step, do the following:

- Replace the current Orders table with the new one,
- Refer to the new “Absolute Time” custom field (“GMTDate”) as the leading date field
- Make the reference tables (“Orders” and “GMT Conversion.csv”) invisible.

# Currency Conversion

## Business Case

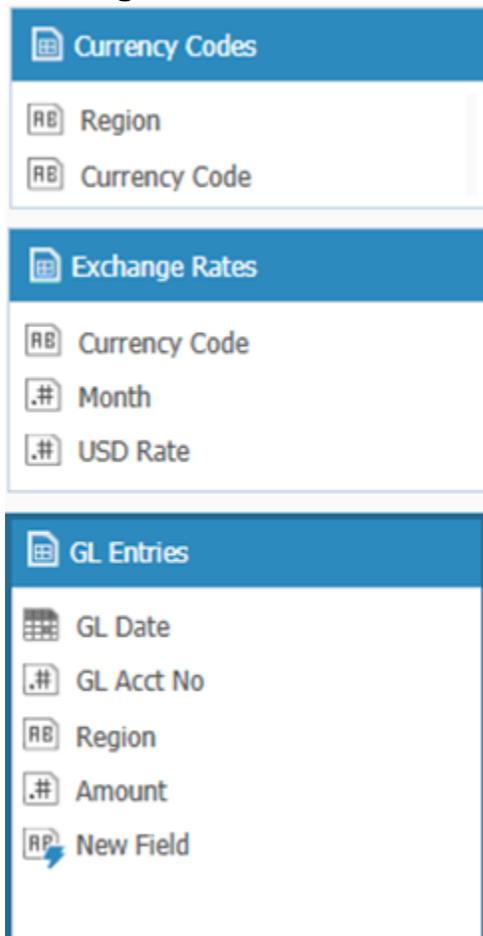
Most data for entities is recorded in their local reporting currency (ie \$ for United States, £ for UK). Here we want to convert all the amounts to USD.

## Modeling Challenge

This requires determining the Currency Rate of the region and then multiplying the value in local currency by the associated Exchange Rate by Month.

## Solution

Create two custom fields in the GL Entries. The first will look up the Currency code of the region. This field will be used along with a month field to link to the Exchange Rates table.



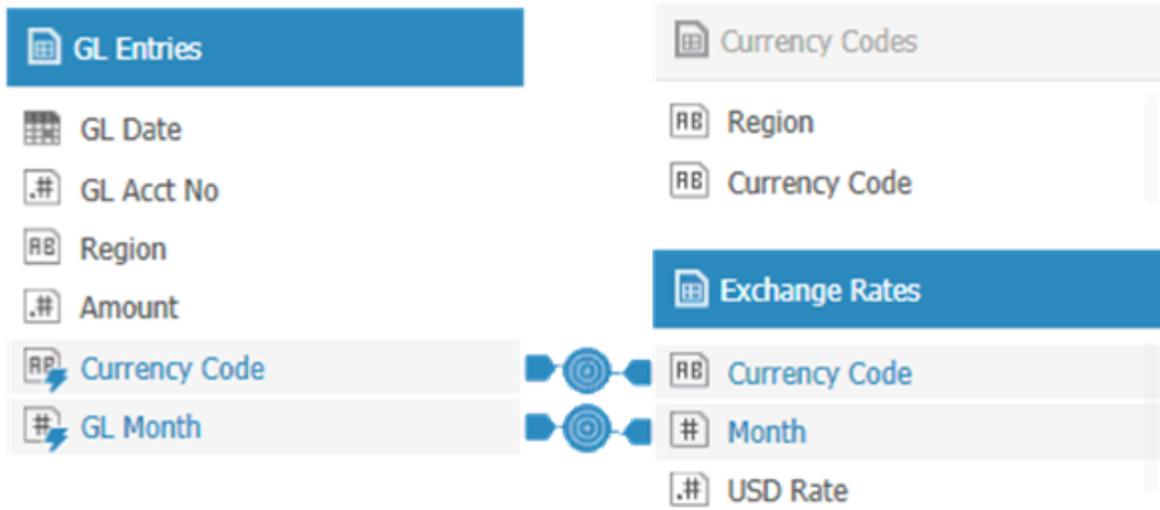
The first field in the GL Entries is created using the lookup function to retrieve values from the Currency Codes table.

```
Lookup([Currency Codes],[Currency Code],Region,Region)
```

Then create a second Custom Field for the Month of the GL Date.

```
GetMonth([GL Date])
```

Next, link the fields together (note that both Month fields were set to Integer and the Currency Codes table to Invisible).



The Local Amount multiplied by the Exchange Rate gives the Converted USD Amount.

### LOCAL and USD

Region	GL Month	Local Amount	Exchange Rate	Converted USD Amount
IL	1	1,303,490	.25	325,873
	2	833,853	.27	225,140
	3	1,915,644	.28	536,380
	4	1,303,288	.27	345,371
	5	1,232,185	.29	357,334
	6	1,743,100	.28	479,353
	7	1,387,776	.28	388,577
	8	17,116	.29	4,964
US	1	939,352	1.00	939,352
	2	752,517	1.00	752,517
	3	604,556	1.00	604,556
	4	648,043	1.00	648,043
	5	769,723	1.00	769,723
	6	884,302	1.00	884,302
	7	550,515	1.00	550,515

# **Current vs. Previous Period for Specific Date Range**

## Business Case

In many cases we would like to compare our business' performance last week, to the week before, or maybe we would like to see a percentage of sales growth for the current month/quarter compared to the previous month/quarter.

## Modeling Challenge

Since we want the compared time range to be as flexible as possible, the solution has to include both layers – ElastiCube and web application.

## Solution

Create a custom table in the ElastiCube to summarize the totals/counts per day for the source table:

```
SELECT
a.Date,
sum(a.Revenue)AS value
FROM [Accord 2011 Client List] AS a GROUP BY a.Date
```

Create a custom table in the ElastiCube with current vs. previous values, by adjusting the script below:

```
SELECT
curr.Date AS date,
curr.value AS current,
prev.value AS prev
FROM [sum] curr
LEFT JOIN [sum] AS prev
ON curr.Date = addyears (prev.Date,1)
UNION
SELECT
addyears (prev.Date,1) AS date,
curr.value,
prev.value
FROM [sum] prev
LEFT JOIN [sum] AS curr
ON prev.Date= addyears (curr.Date,-1)
```

In the web application, add a 'date range picker' using the days from the custom table. Then add two new numeric indicators. In the first numeric picker add the

'sum of the current value', in the second numeric picker, add the 'sum of the previous value'.

In the date range picker, select the days of interest and you will see the current and previous values.

# **Calculating the Number of Open Orders per Day**

## Business Case

An open sales order is where the order has been placed but has not yet been delivered. If for example there is an order for 100 items and against this order only 50 items have been delivered (it is partially delivered). A high level of open orders per day may indicate that something is wrong with orders handling.

## Modeling Challenge

We cannot just count the number of orders per day because it will exclude orders that were open on a certain day and are already closed. Therefore, we will need to create a snapshot of the number of open orders per day.

## Solution

1. Import an Excel file with all dates listed in the Orders table into the ElastiCube.
2. To improve query performance, convert all the date fields into numeric representations (for more information, see [Numeric Representation of Date fields](#)).
3. Create the following custom table:

```
SELECT
s.Dates,
tm.Created_At,
tm.Closed_At,
tm.TicketId
FROM [All Dates] s LEFT JOIN [Orders] tm
ON s.DateInt >= tm.CreatedAtInt
AND (tm.SolvedAt IS NULL OR s.DateInt <= tm.SolvedAtInt)
```

# **Slowly Changing Dimensions**

## Business Case

Transactional data typically does not change, however the data that describes the associated dimensions may change. This example demonstrates how to manage dimensions that may be updated with new values within the data warehouse at different points in time.

For example, a customer that was living in NYC and moved to LA earlier this year.

Date	Customer	City
------	----------	------

1/1/2011	John Smith	NYC
----------	------------	-----

1/1/2013	John Smith	LA
----------	------------	----

## Modeling Challenge

Following the example above, when the transactions were made last year, the customer was living in NYC. Later this year, the customer moved to LA.

If you decide to refer only to the last city and summarize revenue by city, the NYC transaction will be credited to LA only because the customer currently lives there.

If you connect the “Customer” field within the above table to the “Customer” field in the fact table, you will create a “Many to many” relationship because “Customer” is not a unique identifier of this table.

## Solution

The solution is to change the level of granularity of the “Customer\_City” table and add the “Date” field to the key –

1. Concatenation of the Slowly Changing Dimension table’s unique identifier (for example – Customer\_ID + Date)
2. Creation of the same concatenated field in the transactions table.
3. Merge between the 2 keys.
4. This way, you can associate the [Customer\_ID + Date] key of every transaction with the relevant customer city.

### Functions Overview

Sisense supports many functions that you can use for data calculations.

The following topics describe the functions Sisense supports:

- [Date and Time Functions](#)
- [Logical Functions](#)
- [Mathematical Functions](#)
- [Miscellaneous Functions](#)
- [String and Text Functions](#)
- [Web Functions](#)

### Date and Time Functions

You can use date and time functions in custom tables and fields. See [Adding a New Custom Table](#) and [Defining and Editing Field Formula](#) for further details.

Below are explanations of available date and time functions.

**ADDDAYS(datetime, number)**

**ADDHOURS(datetime, number)**

**ADDMONTHS(datetime, number)**

**ADDQUARTERS(datetime, number)**

**ADDEXTENDED(datetime, number)**

**ADDMINUTES(datetime, number)**

**ADDSECONDS(datetime, number)**

Adds a given number of years|quarters|months|days|hours|minutes|seconds to a specified date/time. An example of when to use this is when the fiscal year is not the same as the Gregorian calendar's beginning and end dates.

**CREATEDATE(year,month,day)**

Creates a timestamp from a given year, month and day. Time is set to midnight.

**CURRENTDATE()**

Returns the current date.

**CURRENTTIME()**

Returns the current time.

**CURRENTTIMESTAMP()**

Returns the current timestamp.

**GETYEAR(datetime)**

**GETQUARTER(datetime)**

**GETMONTH(datetime)**

**GETDAY(datetime)**

**GETHOUR(datetime)**

**GETMINUTE(datetime)**

**GETSECOND(datetime)**

Returns a number that represents the year|quarter|month|day|hour|minute|second in a given date/time.

**DAYOFWEEK(datetime)**

Returns the day, out of seven days in a week, represented by a given date/time.

**DAYOFYEAR(datetime)**

Returns the day, out of 365 days in a year, represented by a given date/time.

**WEEKOFYEAR(datetime)**

Returns the week, out of 52 weeks in a year, represented by a given date/time.

**DAYNAME(number)**

Returns the name of the day represented by the given number.

**YEARDIFF(end,start)****QUARTERDIFF(end,start)****MONTHDIFF(end,start)****DAYDIFF(end,start)****HOURLDIFF(end,start)****MINUTEDIFF(end,start)****SECONDDIFF(end,start)**

Returns the difference in years|quarters|months|days|hours|minutes|seconds

**TODATETIME()**

Converts a specified value to a DateTime value. Sisense supports the string format: YYYY-MM-DD HH:mm:ss:ms and YYYY-MM-DD HH:mm:ss. If not HH:mm:ss are entered, Sisense assumes that the time is 12:00AM.

**Logical Functions**

You can use mathematical functions in custom tables and fields. See [Adding a New Custom Table](#) and [Defining and Editing Field Formula](#) for further details. Below are explanations of available logical functions.

**IFBIGINT(condition, true value, false value)**

Returns the first value if the boolean evaluate is true, else returns the second value.

**IFDATETIME(condition, true value, false value)**

Returns the first value if the boolean evaluate is true, else returns the second value.

**IFDOUBLE(condition, true value, false value)**

Returns the first value if the boolean evaluate is true, else returns the second value.

**IFINT(condition, true value, false value)**

Returns the first value if the boolean evaluate is true, else returns the second value.

**IFSTRING(condition, true value, false value)**

Returns the first value if the boolean evaluate is true, else returns the second value.

**INRANGE(value, start, end)**

Returns true if a given value is between start and end.

**ISNULL(value)**

Returns true if a given value is null.

**LOOKUP(remote\_table,remote\_result\_column,current\_match\_column, remote\_match\_column)**

The Lookup function imports a field from one table into another table, by matching two other corresponding fields from both tables. The result will be the value in remote\_result\_column for which the corresponding remote\_match\_column equals the current\_match\_column.

**Mathematical Functions**

You can use mathematical functions in custom tables and fields. See [Adding a New Custom Table](#) and [Defining and Editing Field Formula](#) for further details.

Below are explanations of available mathematical functions.

**ACOS(number)**

Returns the angle, in radians, whose cosine is the given number. Also referred to as arccosine.

**ASIN(number)**

Returns the angle, in radians, whose sine is a given number. Also referred to as arcsine.

**ATAN(number)**

Returns the angle in radians whose tangent is a given number. Also referred to as arctangent.

**CEILING(number)**

Returns the smallest integer that is greater than, or equal to, the a given number.

**COS(number)**

Returns the trigonometric cosine of a given angle, in radians.

**COSH(number)**

Returns the hyperbolic cosine of a given number.

**COT(number)**

Returns the cotangent of a given number.

**DIV(number1, number2)**

Returns the number of times number2 fits completely in number1.

**EXP(number)**

Returns the exponential value of a given number.

**FLOOR(number)**

Returns the largest integer less than or equal to the given numeric expression.

**ISINFINITY(number)**

Returns -1 if the number evaluates to negative infinity, 1 if it evaluates to positive infinity, and 0 otherwise.

**ISINTEGER(number)**

Returns whether a given number is a whole number.

**LOG(number)**

Returns the natural logarithm of a given number.

**LOG10(number)**

Returns the base-10 logarithm of the given float expression

**MAXVAL(number1,number2)**

Returns the highest of two numbers.

**MINVAL(number1,number2)**

Returns the smallest of two numbers.

### **MOD(number1,number2)**

Returns the remainder of dividing number1 by number2 (modulo).

### **PI()**

Returns the constant value of the mathematical PI.

### **POWER(number1,number2)**

Returns the value of the given number raised to a specified power.

### **RAND()**

Returns a random number between 0 and 32767

### **RANK()**

Returns the rank (i.e. row number) of the current record in a table.

### **RANKASC(order\_by\_field)**

Returns the row number ordered by given field in ascending manner.

Example: (3, 5, 5, 19) is ranked in ascending order (1, 2, 3, 4).

You can also sort by multiple columns:

```
RANKASC ([partitionA], [partitionB]...[partitionN], [orderByField])
```

Example: RANKASC ([EmployeeID], [CategoryID], [Quantity])

### **RankCompetitionAsc(partition fields list, order field)**

Returns the rank of each row within the partition of a result set. The rank of a row is one plus the number of ranks that come before the current row ordered by the rightmost field parameter in ascending order.

Leftmost field parameters (optional) define the partitions in which each ordering starts.

Example: (3, 5, 5, 19) is ranked in ascending order (1, 2, 2, 4).

### **RankCompetitionDesc(partition fields list, order field)**

Returns the rank of each row within the partition of a result set. The rank of a row is one plus the number of ranks that come before the current row ordered by the rightmost field parameter in descending order.

Leftmost field parameters (optional) define the partitions in which each ordering starts.

Example: (3, 5, 5, 19) is ranked in descending order (4, 2, 2, 1).

### **RankDenseAsc(partition fields list, order field)**

Returns the dense rank of rows within the partition of a result set without any gaps in the ranking. The rank of a row is one plus the number of distinct ranks that come before the current row ordered by the rightmost field parameter in ascending order.

Leftmost field parameters (optional) define the partitions in which each ordering starts.

Example: (3, 5, 5, 19) is ranked in ascending order (1, 2, 2, 3).

### **RankDenseDesc(partition fields list, order field)**

Returns the dense rank of rows within the partition of a result set without any gaps in the ranking. The rank of a row is one plus the number of distinct ranks that come before the current row ordered by the rightmost field parameter in descending order.

Leftmost field parameters (optional) define the partitions in which each ordering starts.

Example: (3, 5, 5, 19) is ranked in descending order (3, 2, 2, 1)

### **RANKDESC(order\_by\_field)**

Returns the row number ordered by given field in descending manner.

Example: (3, 5, 5, 19) is ranked in ascending order (4, 3, 2, 1).

You can also sort by multiple columns:

```
RANKDESC([partitionA], [partitionB]...[partitionN], [orderByField])
```

Example: RANKDESC ([EmployeeID], [CategoryID], [Quantity])

**ROUND(number,precision)**

Returns the given number, rounded to a specified precision.

**SIN(number)**

Returns the trigonometric sine of a given angle, in radians.

**SINH(number)**

Returns the hyperbolic sine of a given number.

**SQRT(number)**

Returns the square root of a given number.

**TAN(number)**

Returns the tangent of a given number.

**TANH(number)**

Returns the hyperbolic tan of a given number.

**TOBIGINT(string)**

Converts a string representing a valid BigInt value to BigInt type.

**TODOUBLE(string)**

Converts a string representing a valid double value to Double type.

**TOINT(string)**

Converts a string representing a valid Int value to Int type.

**String and Text Functions**

You can use mathematical functions in custom tables and fields. See [Adding a New Custom Table](#) and [Defining and Editing Field Formula](#) for further details.

Below are explanations of available string and text functions.

**ASCII(char)**

Returns the Int ascii code of a given character.

**CHAR(number)**

Returns a character for a given ascii code.

**CONCAT(string1,string2)**

Concatenates string1 and string2.

**INDEXOF(string1,string2,n)**

Returns the index of the n-occurrence of string2 in string1.

**INSERT(string1,string2,overrides,index)**

Inserts string2 into string1 at a specified index, possibly overriding a specified number characters in string1.

**INSERTSTR(string1,string2,index)**

Inserts string2 into string1 at specified zero-based index.

**ISNULLOREMPTY(string)**

Returns true if the string is null or empty.

**LEFT(string,length)**

Returns a substring of a specified length, starting at the first character.

**LENGTH(string)**

Returns the length of a given string.

**LOWER(string)**

Returns a lowercase version of a given string.

**LTRIM(string)**

Removes whitespaces from the start of the string.

**REMOVEAT(string,index)**

Removes a character at a specified index from a given string.

**REMOVERANGE(string,index,count)**

Removes a specified number of characters from a given string starting at a specified index.

**REPEAT(string,number)**

Returns a concatenation of a string repeated a given number of time.

**REPLACEALL(string,oldstring,newstring)**

Replaces all occurrences of old string with new string.

**RIGHT(string,length)**

Returns a substring of a specified length, end with the last character.

**RTRIM(string)**

Removes whitespaces from the end of the string.

**SOUNDEX(string)**

Returns a four-character (SOUNDEX) code to evaluate the similarity of two objects.

**STRBETWEEN(string, startstring, endstring)**

Returns the substring contained between startstring and endstring.

**STRPARTS(string,delimiter,n)**

Returns the string delimited by a specified delimiter. The search ends after the n-occurrence of the delimiter is found.

**SUBSTRING(string,index,length)**

Returns the substring of a specified length, starting at a specified index.

**TOSTRING(object)**

Converts a given value to a string.

**TRIM(string)**

Removes whitespaces from both ends of a string.

**UPPER(string)**

Returns an uppercase version of a given string.

**Web Functions**

You can use web functions in custom tables and fields. See [Adding a New Custom Table](#) and [Defining and Editing Field Formula](#) for further details. Below are explanations of available web functions.

**Note:** In all circumstances, the URL must include 'http://'. For example, if the URL is www.sisense.com, the syntax should be GetFile ('http://www.sisense.com')

**DomainIs: true/false DomainIs (URL,compared domain)**

Get whether the given url's domain matches the given domain. For example 'http://www.wikipedia.org' matches org, wikipedia, www.wikipedia.org.

**GetFile: string GetFile (url)**

Extract the filename portion of the path in the given url and returns it. The filename is everything after the last slash in the path.

**GetHost: string GetHost (url)**

Extract host from the given url, for example 'www.wikipedia.org' in 'http://www.wikipedia.org'

**GetPassword: string GetPassword (url)**

Extract password from the given url.

Specified in the url following the user name, with leading colon (':') and trailing at sign (@), as in 'http://user:password@host/'

**GetPath: string GetPath (url)**

Extract path from the given url.

Path is the url portion following the host name, for example

'/wiki/Business\_intelligence' in

'http://www.wikipedia.org/wiki/Business\_Intelligence'

**GetPort: string GetPort (url)**

Extract port from the given url, parsed as integer, for example 123 in

'http://www.wikipedia.org:123/'

**GetQuery: string GetQuery(url)**

Extract the query from the given url.

Specified in the url after the '?' and before (optional) the '#' sign, for example

'par=val' in 'http://www.wikipedia.org/?par=val#ref'.

**GetQueryParamValue (url,param name): string GetQueryParamValue (url,param name)**

Extract param value from the given url's query, by the given param name.

Param/Value pairs when supplied, are delimited by an ampersand or a semicolon.

**GetRef: string GetRef (url)**

Extract ref from given url.

Specified in the url starting after last appearance of the '#' sign, for example 'ref' in 'http://www.wikipedia.org/?par=val#ref'

**GetScheme: string GetScheme (url)**

Extract scheme from given url.

Skips leading spaces and followed colon (':') sign, for example 'http://www.wikipedia.org/'

**GetUserName: string GetUserName (url)**

Extract user name from given url.

Specified in the url before the host name and delimited by an '@' as in 'http://user@host/'

**HasHost: true\fast HasHost (url)**

Get whether the given url specifies a host.

**HasPassword: true\fast HasPassword (url)**

Get whether the given url specifies a password.

**HasPath: true\fast HasPath (url)**

Get whether the given url specifies a path.

**HasPort: true\fast HasPort (url)**

Get whether the given url specifies a port.

**HasQuery: true\fast HasQuery (url)**

Get whether the given url specifies a query.

**HasRef: true\fast HasRef (url)**

Get whether the given url specifies a ref.

**HasScheme: true\fast HasScheme (url)**

Get whether the given url specifies a scheme.

**HasUsername: true\fast HasUsername (url)**

Get whether the given url specifies a user name.

**HostIsIpAddress: true\fast HostIsIpAddress (url)**

Get whether the given url host name is an ip address.

**IsEmptyUrl: true\fast IsEmptyUrl (url)**

Get whether the given url is empty.

**IsFileUrl: true\fast IsFileUrl (url)**

Get whether the given url is a file url.

**IsSecuredUrl: true\fast IsSecuredUrl (url)**

Get whether the given url scheme indicates a secured connection(https).

**IsStandardScheme: true\fast IsStandardScheme (url)**

Returns true if the scheme for the current urk is a known 'standard' scheme.

Standard schemes have an authority and a path section (including file:).

**IsValidUrl: true\fast IsValidUrl (url)**

Get whether the given url has a valid form.

**PathForRequest: true\fast PathForRequest (url)**

Returns the path that should be sent to the server.

This is the path, parameter, and query portions of the url.

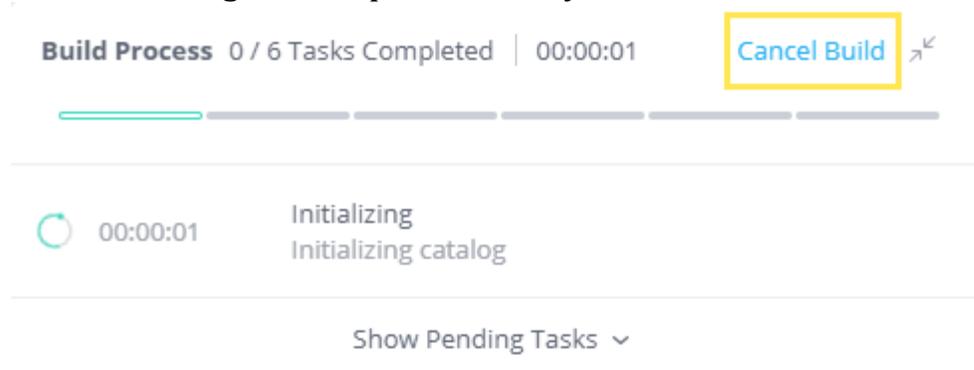
**SchemeIs: true\fast SchemeIs (url)**

Get whether the given url scheme portion matches the given scheme.

## Building ElastiCubes

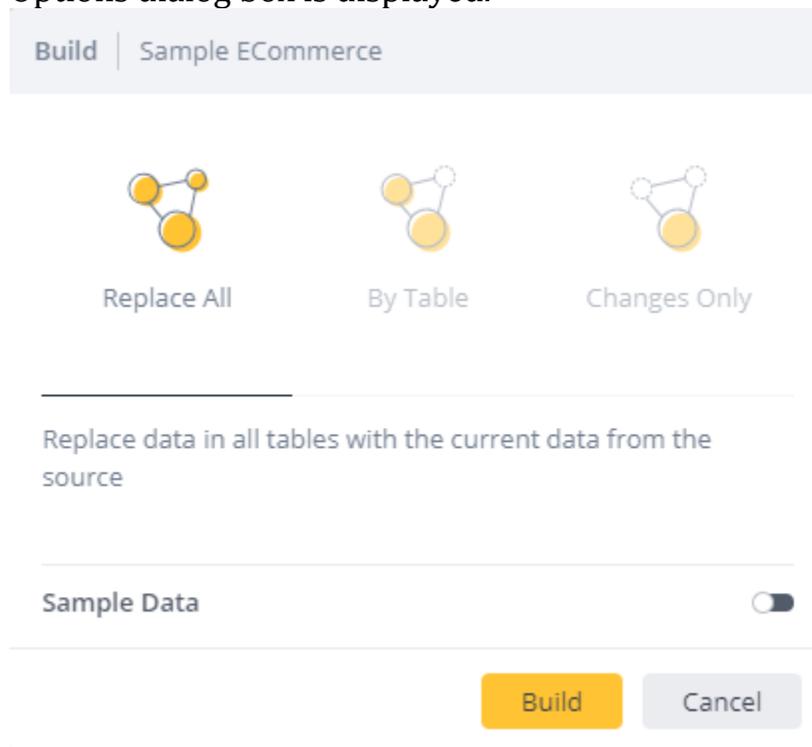
After defining a schema, you need to build your ElastiCube. Building an ElastiCube imports the data from your data sources that you added to your schema. You must build an ElastiCube at least once before the ElastiCube data can be used in a dashboard.

If your data set is very large, a build may take some time. You can cancel the build in the build log that is opened when you run a build.



### To build an ElastiCube:

1. In the **Data** page, open the relevant ElastiCube and click **Build**. The Build Options dialog box is displayed.



2. Select **Replace All**, **By Table**, or **Changes Only**. **Replace All** builds the ElastiCube from scratch with the current data from all the sources in the schema. **By Table** builds the ElastiCube as defined on the table level where all tables not defined are built from scratch. **Changes Only** builds the ElastiCube with data for tables that were added to the schema or modified since the last build. For an in-depth description of these settings and the implications of each setting, see [ElastiCube Build Settings](#).

3. Toggle the switch below to limit the amount of rows imported for each table.

Sample Data

Limit the number of imported rows per table:

Number of rows

---

This is useful for building samples of your data and reduces the overall amount of time it takes to build an ElastiCube.

4. Click **Build**.  
A log is displayed that describes the build process. When the build is complete, the following message appears in the logs: Build successfully ended.

## Building ElastiCubes Remotely

**Note:** This functionality is only available in the desktop version of Sisense. This functionality will be available in Sisense in the near future.

You can build the ElastiCube locally on your machine or build the ElastiCube remotely from another machine in which Sisense is installed. This is useful for example when your configuration supports test and production environments. For more information on setting up production environments, click [here](#).

When connecting remotely, your ElastiCube server must have Sisense installed and a unique Sisense license. While you can have multiple Administrators working on and building an ElastiCube, only one Administrator can work on an ElastiCube at a time to prevent data corruption. In addition, the machine in which the build is performed, must have sufficient resources to be able to handle all of your data. See Sisense's [minimum requirements](#) for more information.

### **To build an ElastiCube remotely:**

1. On the ElastiCube server where the ElastiCube is to be built, open ports 811 and 812 for inbound and outbound access.
2. In Sisense, open the Sisense Server Console by clicking **+**.
3. In the **Connection Settings** window, enter a name for the server and its IP address.
4. Click **Connect**. The list of ElastiCubes for that server are displayed in the Sisense Server Console.
5. In Sisense, click **Build**.
6. Under Target Server, select the new server you have just added.
7. Define your build settings and click **Build**.

## ElastiCube Build Settings

When you build your ElastiCube, data is imported from your data sources into the ElastiCube according to settings you define on the model level and on the table level.

This topic describes how data can be imported on each level and the implications of each of the ElastiCube build settings.

# Model Level

On the model level, you have three choices for importing data, Replace All, By Table, and Changes Only.

Build | Sample ECommerce

Replace All      By Table      Changes Only

---

Replace data in all tables with the current data from the source

---

Sample Data

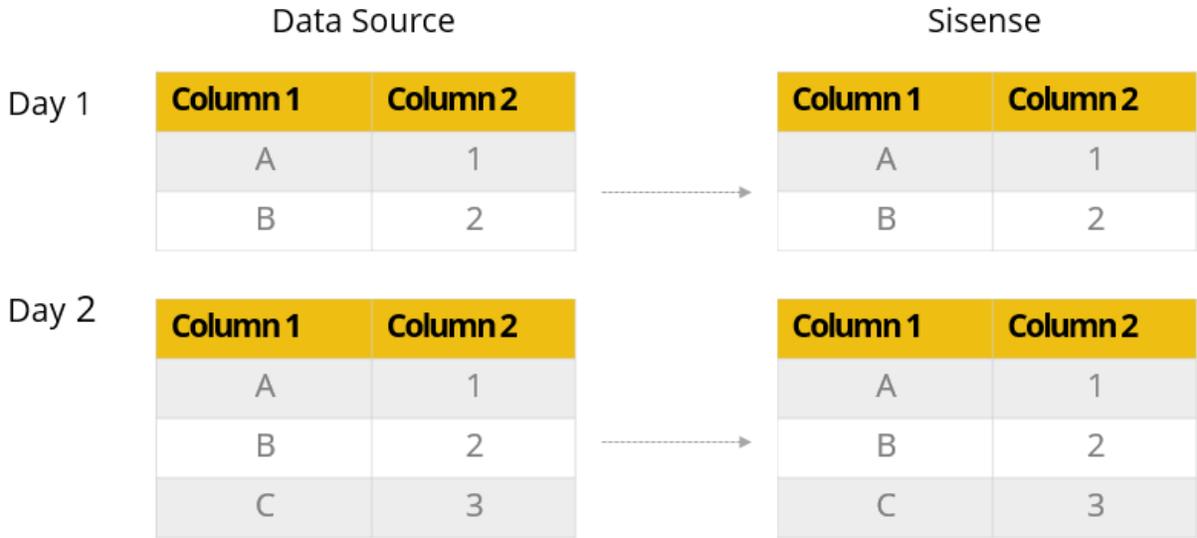
---

**Build**    Cancel

# Replace All

When you replace all your data on the model level, any existing data imported into your ElastiCube is overwritten and any custom tables or columns are recalculated. Sisense imports an exact copy of your data from the source. If the size of your data is large or the connection speed to the source slow, it can take a long time to build your ElastiCube.

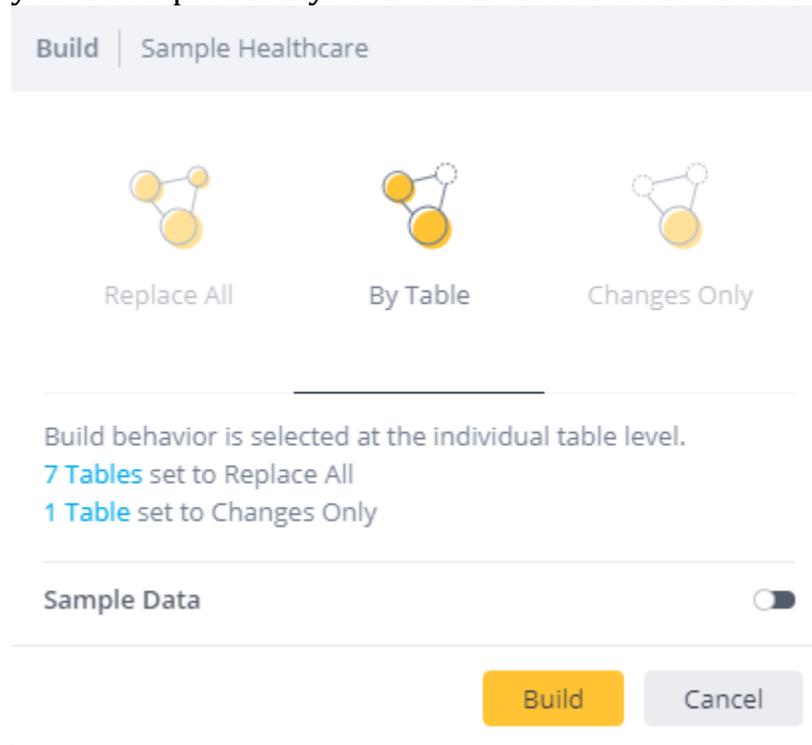
The diagram below illustrates what happens before and after a Replace All build:



**Note:** While building the ElastiCube with schema changes or entire ElastiCube builds, you can continue to run queries. In the event that the build fails, Sisense restores the original version of the ElastiCube and attempts the build again. While Sisense restores the original ElastiCube, some downtime may occur while Sisense copies the ElastiCube to your drive. The downtime is the amount of time it takes to copy the ElastiCube locally. If you require high availability for your data, Sisense provides this through ElastiCube Sets. When Sisense restores an ElastiCube a copy of the original ElastiCube is created. You should verify that your server can store multiple copies ElastiCubes until the restoration process is complete.

## By Table

When you build By Table, all your data is overwritten except for tables where you have specifically defined their build behavior in the schema.



This option is only displayed when a table's build behavior has been defined in the schema. When you have customized a table's build behavior, an icon is displayed that indicates the table's build behavior, for example, **Changes Only**  or **Append** . These icons are also described in the legend in the bottom-right corner of the schema.

In the Build Settings dialog box, you can click the table links to see which tables have been configured or are set to be overwritten.

For more information about table build behavior, see [Table Level](#) below.

## Changes Only

When you build by Changes Only, you build only new tables or tables that have changed since the last build. This is useful when you frequently need to refresh a large data source. For example, if you have a data source that is updated daily, rather than rebuild the entire ElastiCube daily, you can just import the new data added each day. This option can significantly reduce the amount of time it takes to complete a build.

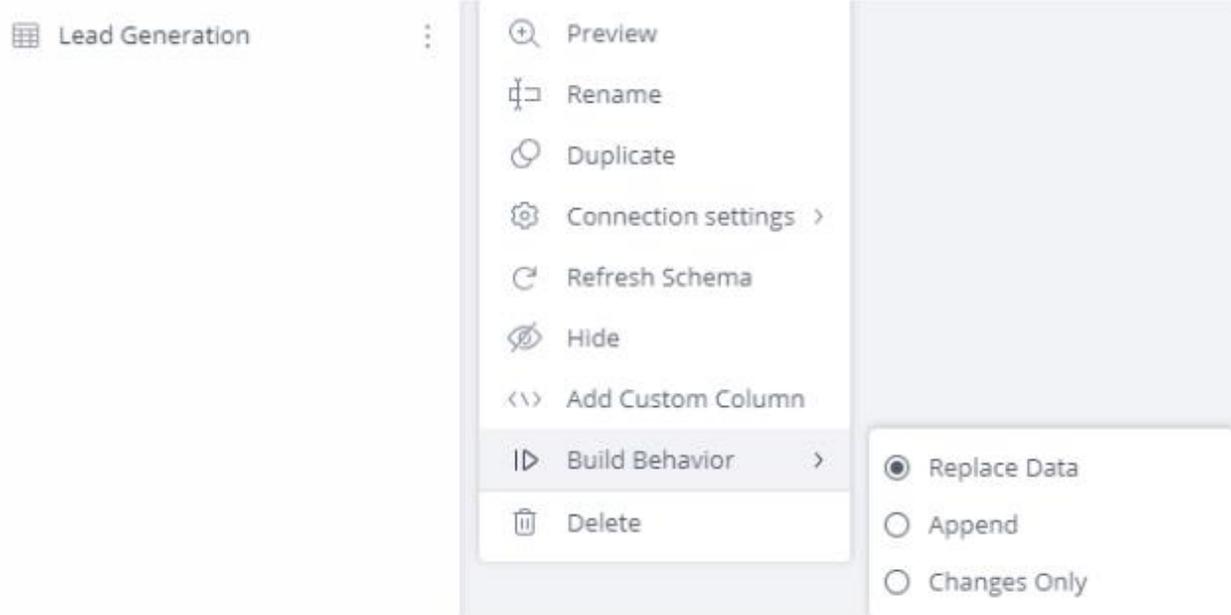
Tables that have changed since the last build include the following:

- New tables that were added
- Tables where columns were added or dropped
- Custom tables where the expression has changed
- Tables with custom columns where the expression has changed
- Custom tables that have SQL expressions which reference other tables that have changed

If your table meets any of the above criteria, then the table's data is overwritten and rebuilt from scratch.

# Table Level

You can define your build's behavior per table by selecting the table's menu > **Build Behavior**, and selecting the relevant option.



Then, when you define what type of build you want to do, select the **By Table** option.

Build | Sample Healthcare



Replace All



By Table



Changes Only

Build behavior is selected at the individual table level.

7 Tables set to Replace All

1 Table set to Changes Only

Sample Data



Build

Cancel

You can set the build behavior of a table to one of the following options:

## Replace All (Default)

Replaces all data at the time of the build. This is recommended for dimension tables, for example: store attributes, or dimensions, which describe the objects in a fact table.



## Changes Only

Does not import any data unless changes have occurred in the table. This is recommended with summary/snapshot fact tables and with data marts (smaller subsets of data, tailored for specific needs).

Tables that have changed since the last build include the following:

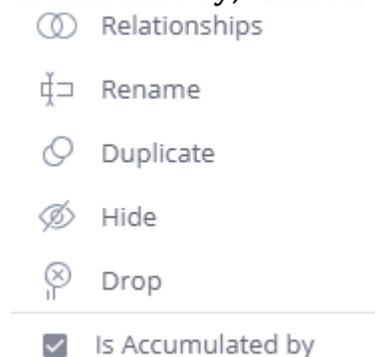
- New tables that were added
- Tables where columns were added or dropped
- Custom tables where the expression has changed
- Tables with custom columns where the expression has changed
- Custom tables that have SQL expressions which reference other tables that have changed

## Accumulate By

Adds additional rows of data incrementally to an existing table according to the data in a specified integer or date column that acts as an index. You can select a specific integer or date column that will be used to determine whether to accumulate data at the time of the build. The column acts as an index, and if the index value is greater in a subsequent build, then data is accumulated for the table. This is recommended for detailed fact tables, for example: store quantitative information for analysis.

**Note:** Only date and integer columns are supported.

To select the integer or date column within a table as the parameter to accumulate by, click on the column's menu, and then select **Accumulate By**.



Accumulation behavior for integers and for dates are as follows:

When you select an integer, only source rows with a value greater than the maximum index value in the ElastiCube table will be inserted. Existing data in the ElastiCube table will not be modified or deleted.

The diagram below illustrates what happens before and after an Accumulative build:



## Accumulative Build Support

Some data sources (for example, CSV files) do not support accumulating data on the column level, and in such cases data will be duplicated. The table below describes which data providers support accumulative builds and appending data to tables:

Data Source Provider	Append to Table	Accumulate by Column
Oracle	Yes	Yes
MySQL	Yes	Yes
SQL Server	Yes	Yes
PostgreSQL	Yes	Yes
ODBC	Yes	No
OleDb	Yes	No
Hive	Yes	Yes
HubSpot		Yes
MS Excel	Yes	No
CSV file	Yes	No
MS Access	Yes	No
Salesforce	Yes	No
Google AdWords	Yes	No
Google Analytics	No	No
Google Spreadsheets	Yes	No
Amazon Redshift	Yes	Yes
Heroku Postgres	Yes	Yes

Data Source Provider	Append to Table	Accumulate by Column
DB2	Yes	Yes
Teradata	Yes	Yes
MongoDB (ODBC)	Yes	Yes

## Scheduling Data Loads

After you build an ElastiCube, you can schedule future builds to automatically synchronize with the underlying data source to ensure all data is up to date.

There are two options you can choose when scheduling a build, by time or by intervals. Builds scheduled by time occur according to the days and hour you define. You can configure Sisense to build an ElastiCube automatically any or every day of the week at the same hour.

Builds scheduled by intervals occur after the defined intervals passes. You can set a single interval for an ElastiCube.

**Note:** Only one option can be selected at a time. The last configuration you set is the configuration that will be used.

If you want to cancel a scheduled build, select **Disable** and click **Save**.

By default, scheduled builds are Entire builds unless any table within the ElastiCube is set to Accumulative or a table is set to be ignored.

Your ElastiCube's build configuration determines how the scheduled build is to be performed. For more information about how to build ElastiCubes click [here](#).

**To schedule a build:**

1. In the **Data** page, open the menu of the relevant ElastiCube and select **Schedule Build**. The Schedule Build window is displayed.

**Schedule Build**

---

ElastiCube: Tutorial Last Build: 03/04/2018 1:39 PM

Choose one of the scheduling options:

  
Disable

  
Daily

  
Interval

---

Select the days and time to build your ElastiCube automatically and click **Save**.

Monday

Tuesday

Wednesday

Thursday

Friday

Saturday

Sunday

12:00 AM v

---

Save Cancel

2. Select the relevant scheduling option:  
**Disable:** Disables a schedule build.  
**Daily:** Select the relevant days and define the hour when the build is to occur. For each selected day, the build will occur at the same hour defined.  
**Interval:** Select the time in between automatic builds. The first automatic build begins after you save your settings and the interval passes.
3. Click **Save**.

## Changing an ElastiCube

This topic describes how to make changes in your ElastiCube.

### To update an ElastiCube:

- Open the **Data** page and click the relevant ElastiCube. The schema of the ElastiCube is displayed.

Changes are saved as you work and take affect when you build the ElastiCube.

## Live Connect

Traditionally, your data was only as fresh as your latest ElastiCube build. With live connections, you can refresh your dashboards manually or set them to refresh automatically every few seconds. Sisense Live Connect queries are run directly against the data source. Sisense performs no additional caching on the data.

The most significant difference between dashboards built on ElastiCubes and dashboards connected to live data sources is how the load is managed. With an ElastiCube, you refresh the data when you build an ElastiCube and query the ElastiCube, not the data sources that define your ElastiCube. With live connections, queries are run against the data source itself. This means that you don't have to wait for an ElastiCube to finish building before you see the results, however, your data source must be able to handle the potential load of multiple refreshes.

To create dashboards built on live connections, you create live models. These data models include connection and credential details to the data sources. After you have created your live model, you publish it. Publishing the live model adds it to your list of data sources from which you can select when working with dashboards.

## Limitations

The following features are not supported in live connections:

1. Pulse alerts
2. Box & Whisker plot
3. Certain analytical functions such as Mode and Standard Deviation
4. Changing live data sources from dashboards
5. Custom SQL tables and columns

## Frequently Asked Questions

**Q:** I cannot connect to my data source?

**A:** As the connection is live, the data source must be available for Sisense to connect. Also, if your credentials changes, you must update your data set so Sisense can continue to connect and refresh the data.

**Q:** No data sources are displayed when trying to add one?

**A:** In Windows Services, verify that the following are running, and if not, restart them:

- Sisense.CLRConnectorsContainer
- Sisense.Discovery
- Sisense.ECMServer
- IIS

**Q:** If I use Windows Authentication with my SQL server, what do I enter in the Username and Password fields?

**A:** Enter the Location and leave the Username and Password fields blank.

**Q:** After entering a data set name, the Done button is not displayed:

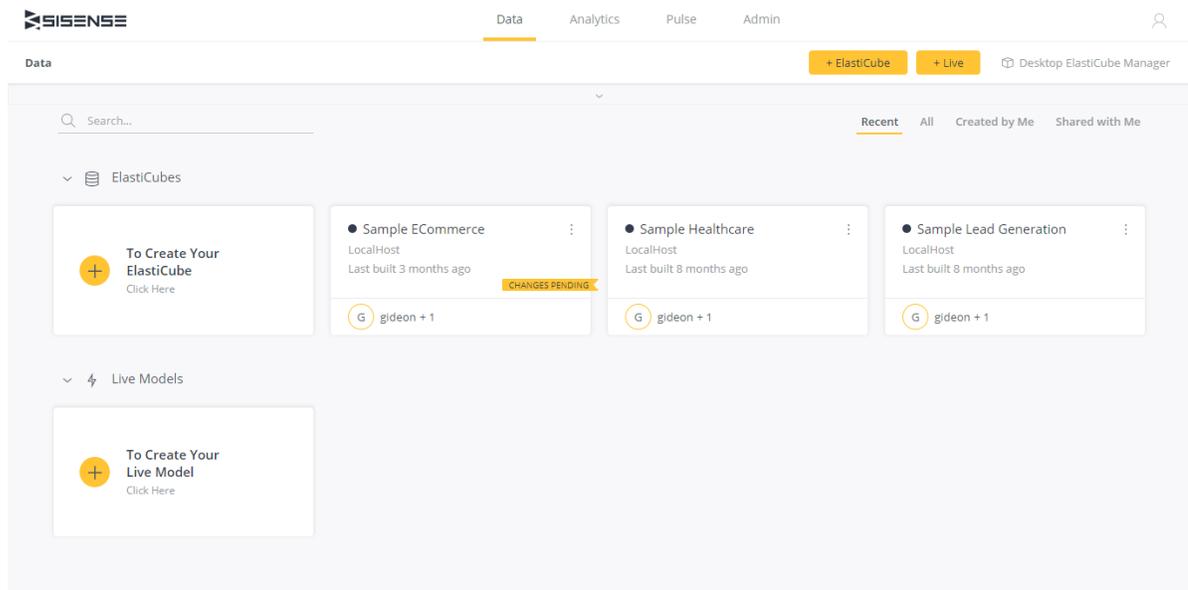
**A:** Make sure the data set's name is unique. You cannot create data sets with the same name.

## Creating Live Models

Before you can analyze dashboards, you need create a live model.

**To create a live model:**

1. Open Sisense and click **Data** in the top menu. Your ElastiCubes and live models are displayed.



2. Click **+ Live**. The **Add new Live Connection** dialog box is displayed.
3. In **Title**, enter a name for your live model and click **Save**. An empty model is displayed in the Model Editor.

## Adding Live Connections

An alternative to importing data into an ElastiCube is to connect directly to the source through a live data connection. With Live Connect, queries can be processed directly in the data source. This allows you to work with only the result set from a query, rather than doing large imports from the source. You may want to leverage this method if you have invested in a high performance database or want to tap into real-time functionality. For a short video overview of working with live data, click [here](#).

Each live data source has its own requirements. For documentation for the relevant live data sources, see the table below:

Connector	More Information
Google BigQuery	<a href="#">Documentation</a>

Connector	More Information
MySQL	<a href="#">Documentation</a>
Oracle	<a href="#">Documentation</a>
PostgreSQL	<a href="#">Documentation</a>
Redshift	<a href="#">Documentation</a>
Snowflake	<a href="#">Documentation</a>
SQL Server	<a href="#">Documentation</a>

## Working with Live Models

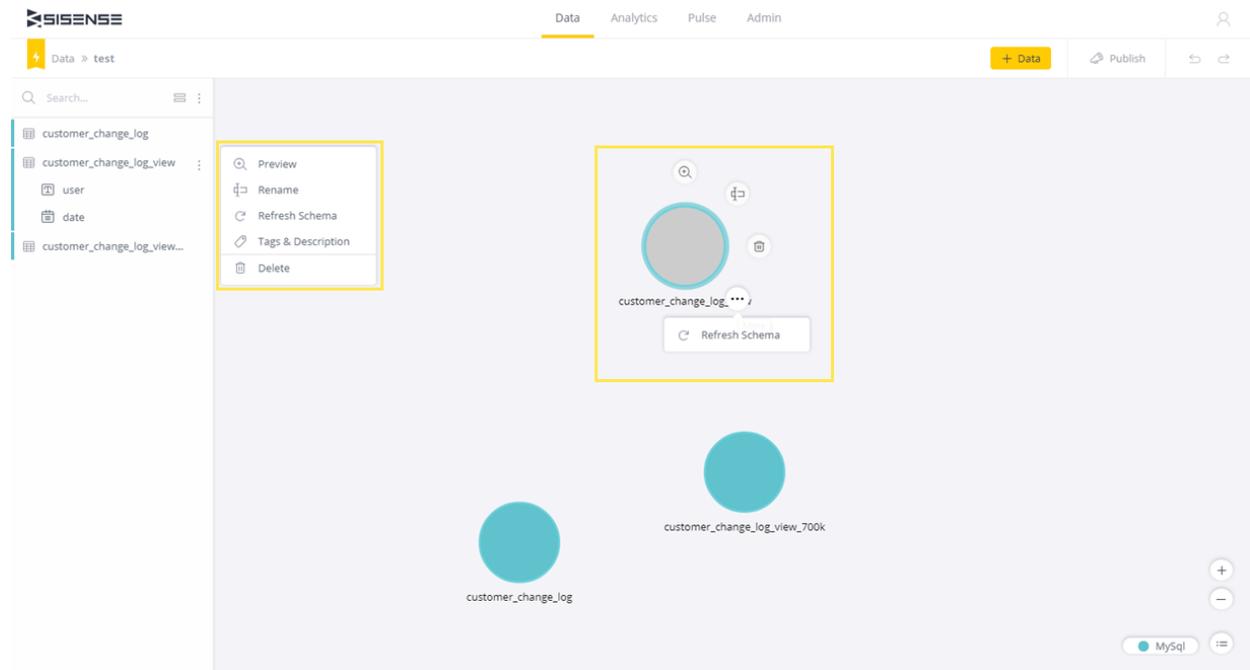
In the Model Editor, you have a variety of options for working with tables you have connected to from a live data source.

You can manage your data on two levels, the Table level and the Column level.

Any changes you make to your schema are reflected in the dashboard or widgets of the live model only after you publish the changes.

# Table Level

You can manage tables connected to a live model through the Navigation Pane or through the table itself in the schema.



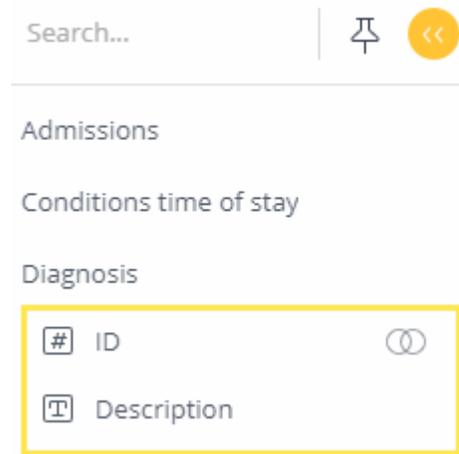
The table below describes the menu options available from the Navigation Pane and the corresponding icons available in the table's menu. Either method allows you to modify and manage your tables.

Navigation Pane	Table Icon	Description
Preview		Displays a preview of your table's data.
Refresh Schema	Refresh Schema	Allows you to update a table's columns if there was a change on the data source without having to add the table again.
Delete		Allows you to delete a table.

To view your table's metadata, including its name, path, provider, location, and the number of columns it contains, hover over the table in the schema or in the Navigation Pane.

# Column Level

Columns that are part of a table are displayed below the table in the Navigation Pane. On the left side of the column is an icon that indicates what type of data is in the column.



On the right side of the column's name is an icon that indicates if the column is connected to another table through a relationship. You can hover over the icon to view more details or disconnect the relationship.

For each column, you can hover over its name to display a menu from where you view the following options for managing your columns:

Menu Option	Description
Relationships	Displays related columns together side by side. See <a href="#">Creating Relationships</a> for more information. <b>For Amazon Redshift and Microsoft SQL Server only.</b>
Edit Relationship	Allows you to edit a column's relationship. This option is only available when the column is connected to another table.
Description	Allows you to tag a column with metadata that you can use to organize your data and search for it later. See <a href="#">Tagging Your Data</a> for more information.

## Publishing Live Models

After creating and designing a live model, you need to publish it. Publishing a live model adds the model to the list of your data sources from which you can choose when creating a dashboard or changing a data source.

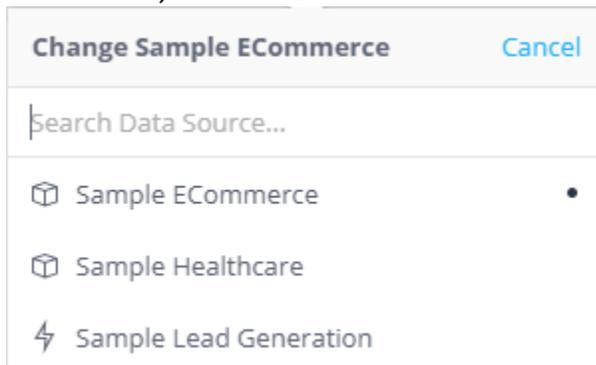
### To publish a live model:

1. In the **Data** page, open the relevant live model.
2. In the Model Editor, click  **Publish**.

## Working with Live Widgets

After you have published your live model, you can begin to create visualizations like you would with any other data source. See [Creating Dashboards](#) and [Creating Filters](#) for more information.

In Sisense, live data sources have the  icon next to their name.



While you can create dashboards from a live data model like you would an ElastiCube, you can also add live data models as a data source to a dashboard that already has a data source such as an ElastiCube. When you have multiple data sources working together on a single dashboard, the data source and the widgets built on it operate independently of each other. This allows you to view widgets that visualize time sensitive information in addition to widgets that are refreshed less frequently. For more information about adding multiple data sources, see [Changing a Dashboard's Data Source](#).

In terms of governance, all existing access rights settings and data security settings can be applied for live connections.

## Load Management

The most significant difference between dashboards built on ElastiCubes and dashboards connected to live data sources is how the load is managed. With an ElastiCube, you refresh the data when you build an ElastiCube and query the ElastiCube, not the data sources that define your ElastiCube. With live connections, queries are run against the data source itself. This means that you don't have to wait for an ElastiCube to finish building before you see the results, however, your data source must be able to handle the potential load of multiple refreshes.

When you refresh the data for a live connection, Sisense uses the connection settings and credentials in the data set to query the live data source. Depending on your Sisense role, there are several options for refreshing the data.

Administrators define the refresh rate in seconds and timeout for each data set. After the refresh rate is set by the Administrator, no user can set a refresh rate shorter than the rate set by the Administrator. Administrators can manually refresh the data set from the **Data Sources** tab in the **Admin** page.

Keep in mind that each time a refresh takes place, your data source is queried, so your data source needs to handle the refresh rate you define. Designers can reduce the frequency of data refreshes or stop automatic refreshes on a per widget level, but they cannot increase the frequency of refreshes beyond the threshold set by an Administrator.

**Add Live Connection**

1 Select Source

2 Connect

3 Select Data

4 Connection Settings

Choose Connector



PostgreSQL



Amazon Redshift



Microsoft Sql Server



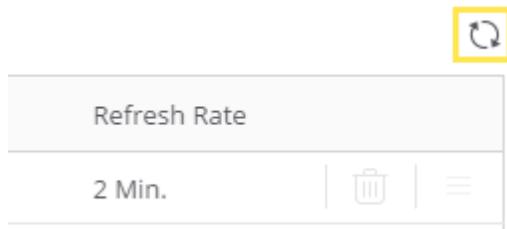
MySQL Database

Cancel

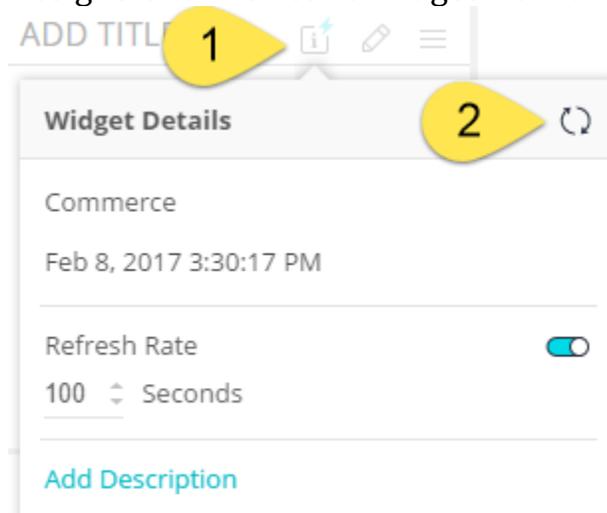
## Refreshing a Data Set

There are several ways in which you can refresh a widget build on a live data model depending on your Sisense user role.

Administrators set the default refresh and can refresh the data manually from the **Data Sources** tab in the **Admin** page of Sisense.

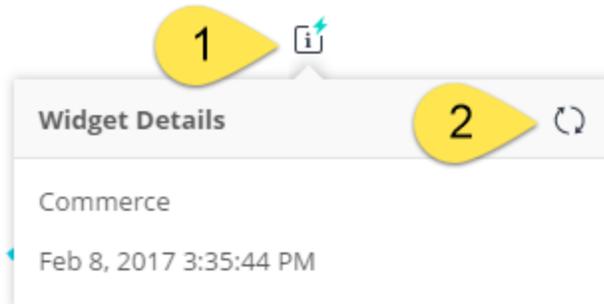


Designers can refresh a widget from the widget's information window .



In addition, Designers can reset the Refresh Rate, however, the rate cannot be lower than the default rate set by the Administrator.

Like Designers, Viewers can refresh a widget from the widget's information window, but they cannot modify the refresh rate.

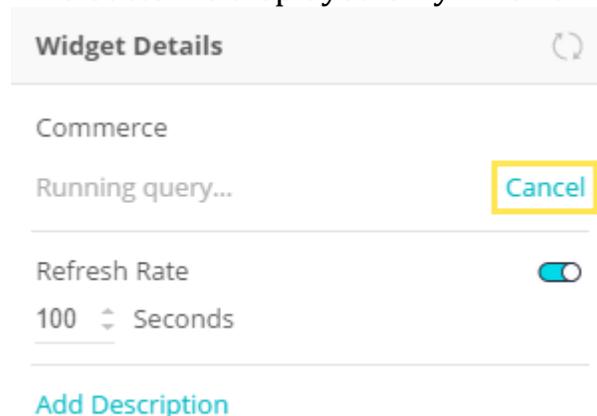


While any users can refresh a widget, Sisense has a minimum 10 second refresh limit. Any refreshes requested within 10 seconds are ignored.

## Canceling a Refresh

If you have a refresh that you want to cancel, you can cancel it by clicking **Cancel** in the widget's information box.

This button is displayed only when a refresh is in progress.



In addition, any time a user leaves a query, for example, to view another page in their browser, Sisense pauses the query.

## Deleting ElastiCubes and Live Models

When you no longer want to use an ElastiCube or live data model, you can delete them from the **Data** page.

Deleting an ElastiCube or live model removes it from the server and the ElastiCube's folder saved locally on your hard drive. After they have been deleted, they cannot be restored.

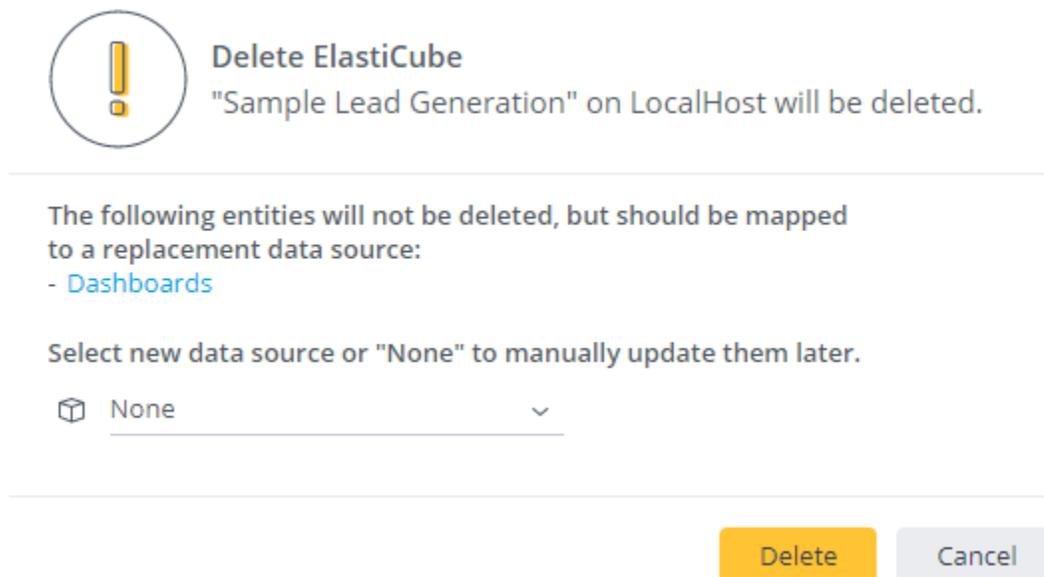
If your ElastiCube or live model has any dependencies, or entities such as dashboards, that are currently connected to it, some of these entities will also be deleted and you will be prompted to remap to another data source. Entities that are deleted when you delete an ElastiCube include:

- Data security rules
- Build alerts
- Drill hierarchies

Entities that are not deleted, but should be remapped to another data source include:

- Dashboards
- Widgets (from other dashboards)
- Pulse Alerts

Entities that you can remap to another data source are clickable links that when clicked, display a list of all the entities that need to be remapped.



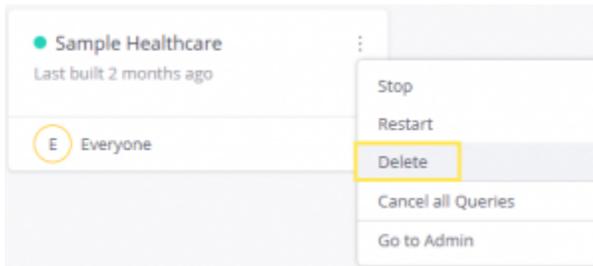
You can choose to remap the entities to another data source manually or select the new data source from the list on the bottom of the Delete ElastiCube dialog box. If you do not remap the entities, the dependencies will not work until you apply a new data source.

**Note:** Any temporary folders that were saved locally during failed builds are not deleted when the ElastiCube is deleted and must be removed manually.

**To delete an ElastiCube or live model:**

1. In Sisense, click **Data** in the top menu.

2. In the **Data** page, click the menu of the relevant ElastiCube or live model and select **Delete**.



3. Click **Yes** to confirm you want to delete the ElastiCube or live model.  
OR  
If you ElastiCube or live model has any dependencies, remap those entities or select None to remap them later.

## Sisense Server Console

**Note:** This feature is currently available in the desktop version of Sisense. This feature will be migrated into Sisense in the near future. For more information, see [Sisense V7.0 Backwards Compatibility](#).

You can use the Sisense Server Console to perform various management tasks on your ElastiCubes, including:

- [Scheduling Data Loads](#)
- [Importing and exporting data](#)
- [Attaching and detaching directories](#)
- [Changing the Location of your ElastiCube Data Storage](#)
- [Canceling Running Queries](#)

### To open the Sisense Server Console:

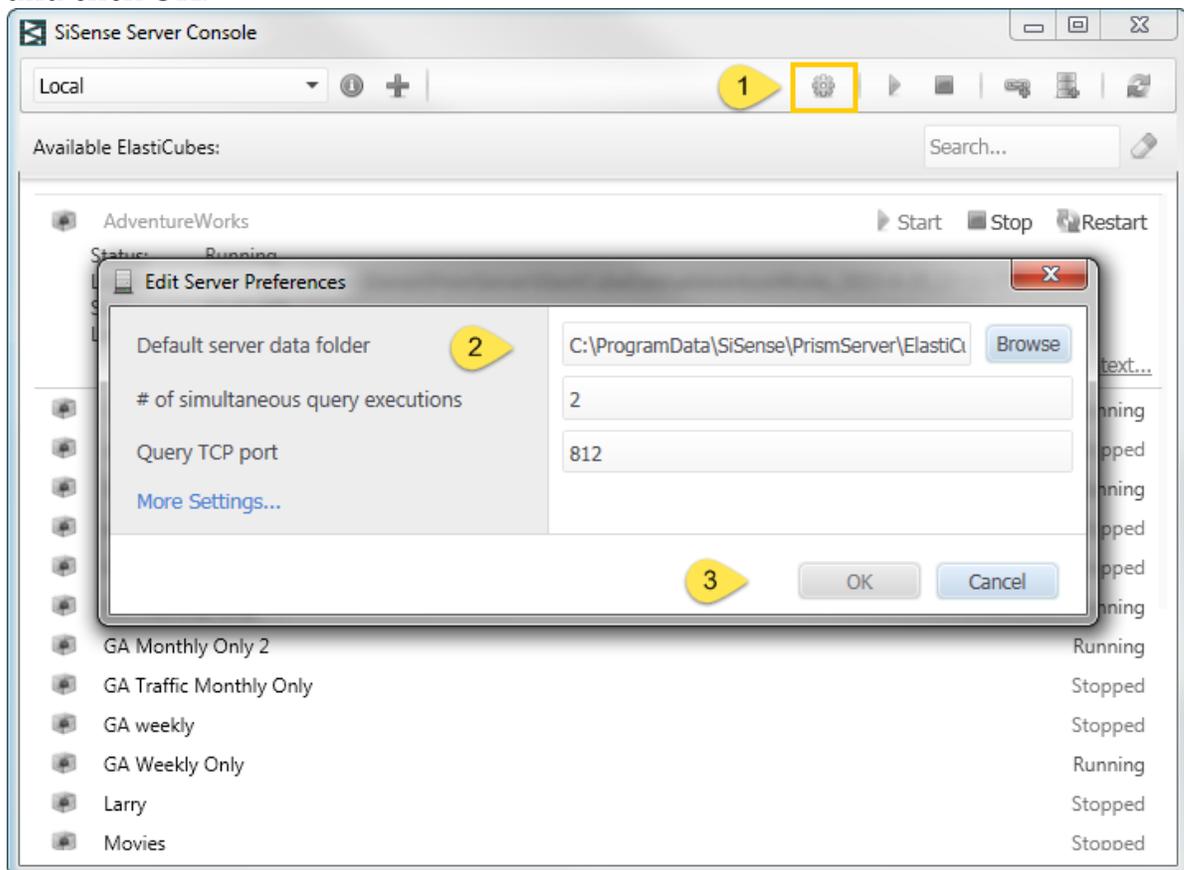
- Right-click the systems tray and double-click the console icon .
- or  
Click on the Windows start menu, and select **All Programs** > Sisense > **Sisense Server Console**.

ElastiCube data is stored in a default folder, which you can change in the Sisense Server Console.

# Changing the Location of the ElastiCube Data Storage

## To change the folder:

1. From the Windows start menu, open the Sisense Server Console.
2. In the top panel, click on the cog icon to view/change server preferences.
3. Click **Browse** next to **Default server data folder**.
4. Select the location of the folder where you want to store ElastiCube data, and click **OK**.

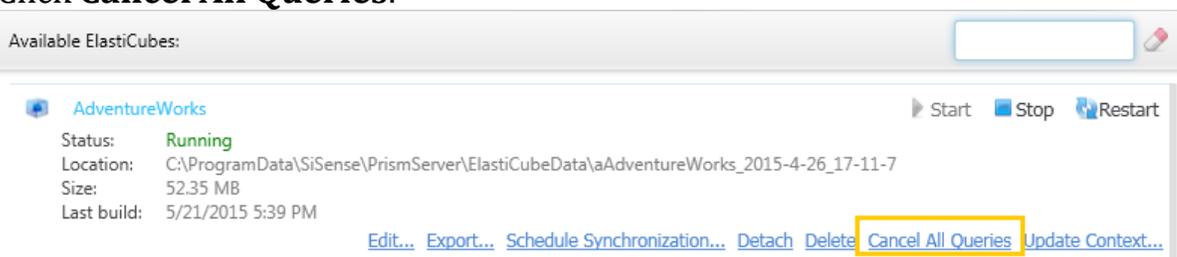


# Canceling Running Queries

You can cancel all running queries in your ElastiCube to free up resources. This is useful in cases where your ElastiCube is consuming too many resources, and as a result, causing the Sisense Web Application to be unresponsive.

## To cancel all queries:

1. Click on the Windows start menu, and select **All Programs > Sisense > Sisense Server Console**.
2. Click the relevant ElastiCube to view the menu panel.
3. Click **Cancel All Queries**.



# Dashboards

After importing data into Sisense, the next step is to create dashboards that Viewers can interact with.

# This section will show you how to get started working with your dashboards.

The image shows a screenshot of a Sisense dashboard interface. The dashboard is titled "Sample - Ecommerce" and displays various data visualizations. Annotations in yellow boxes provide instructions on how to interact with the dashboard.

**Annotations:**

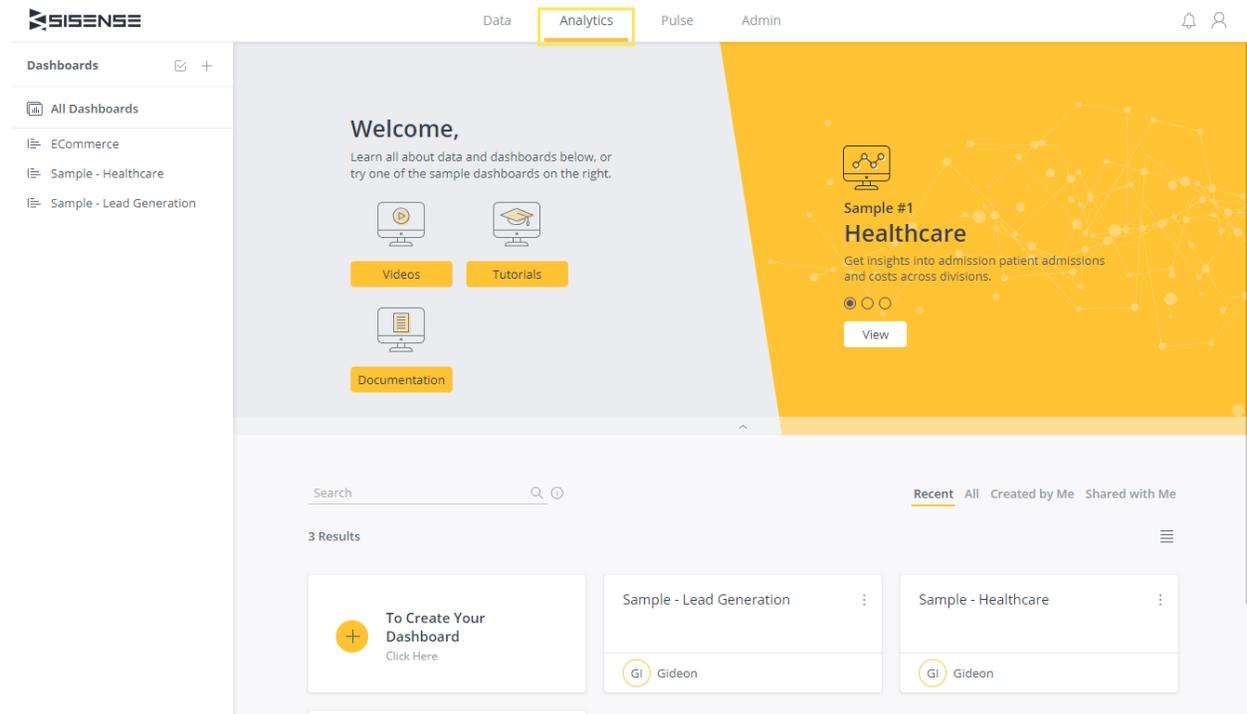
- Click here to access your dashboards:** Points to the top navigation bar.
- Manage Dashboards here:** Points to the "Dashboards" sidebar menu.
- Analyze your data here:** Points to the main dashboard area.
- Change data sources, add widgets, export to PDF, share dashboards and more:** Points to the widget configuration menu.
- Filter your dashboard from the Filter Pane:** Points to the "Filters" sidebar.

**Dashboard Content:**

- Summary Widgets:** Four gauge charts showing "TOTAL REVENUE" (1.53M), "TOTAL SALES" (2,195), and "TOTAL BRANDS" (222).
- REVENUE vs. UNITS SOLD:** A combined bar and line chart showing sales over time for Total Revenue and Total Quantity.
- TOP 3 CATEGORIES BY REVENUE AND AGE:** A horizontal stacked bar chart.
- GENDER BREAKDOWN:** A pie chart showing Male (83%) and Female (17%).
- Filters:** A sidebar on the right with filters for COUNTRY, YEARS, SALES REVENUE RANGE, and Age Range.

# Creating Dashboards

Designers and Dashboard Designers can create dashboards from the **Analytics** page where all your existing dashboards are displayed.



When you create a dashboard, your dashboard is initially empty. The next step is to add widgets, or visualizations, to your dashboard to make it meaningful.

The topics below provide information about creating dashboards, adding widgets to it, and customizing the data displayed in the dashboard.

- [Creating Dashboards](#)
- [Adding Widgets to a Dashboard](#)
- [Managing Widgets](#)
- [Filtering Dashboards](#)
- [Formulas](#)
- [Working with Time](#)
- [Customizing the Dashboard Layout](#)
- [Changing the Color Palette](#)
- [Changing a Dashboard's Datasource](#)

# Managing and Sharing Dashboards

As a Designer, you can manage dashboards that you have created, or that have been shared with you; these topics explain how.

- [Organizing Dashboards](#)
- [Locating Dashboards](#)
- [Deleting Dashboards](#)
- [Copying Dashboards](#)
- [Sharing Dashboards](#)

# Interacting with Dashboards

Now that you've created your dashboards, it's time to start interacting with them.

The topics below will help you learn how to get the most out of your dashboards.

- [Viewing Dashboards](#)
- [Interacting with Filters as a Viewer](#)
- [Changing Sisense's Language](#)
- [Drilling Down in a Widget](#)
- [Making Selections in a Widget](#)
- [Exporting Dashboards](#)

## Creating Dashboards

There are two ways you can create dashboards, first, through the Sisense **Analytics** page, and the second, through the Sisense REST API.

The difference between these two methods is that the **Analytics** page provides an interface where you populate the dashboard with widgets.

To create a dashboard through the REST API, you define the dashboard and then populate it with widgets that already exist through their widget IDs.

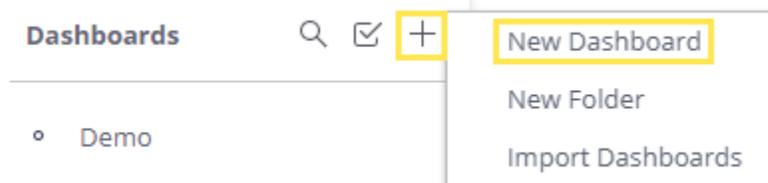
# Creating a Dashboard

Each Dashboard is based on the data selected from one or more ElastiCubes or Live data sources. The instructions below assume that the required ElastiCube(s) and Live data model(s) have already been created. You can also use the sample ElastiCubes that are installed with Sisense if you have not yet created your own.

## To create a dashboard:

1. In the **Analytics** page, click **+** above the Dashboards list.  
OR

Right-click on the folder menu and select **New Dashboard**.



The following window appears:

**New Dashboard / Please select a Data Set to connect to**

---

Data Set: Sample ECommerce

---

Title: Sample ECommerce

**Create** **Cancel**

2. Click the name of the **Data Set** displayed to select the ElastiCube or Live data model you want to work with. ElastiCubes have the  next to their name while Live data models have the  icon next to their name.
3. Click the name of the **Title**, and enter a name for the dashboard. This name will appear at the top of the dashboard, and in the **Dashboards** list.
4. Click **Create**.  
You are then automatically guided through the process of creating your first

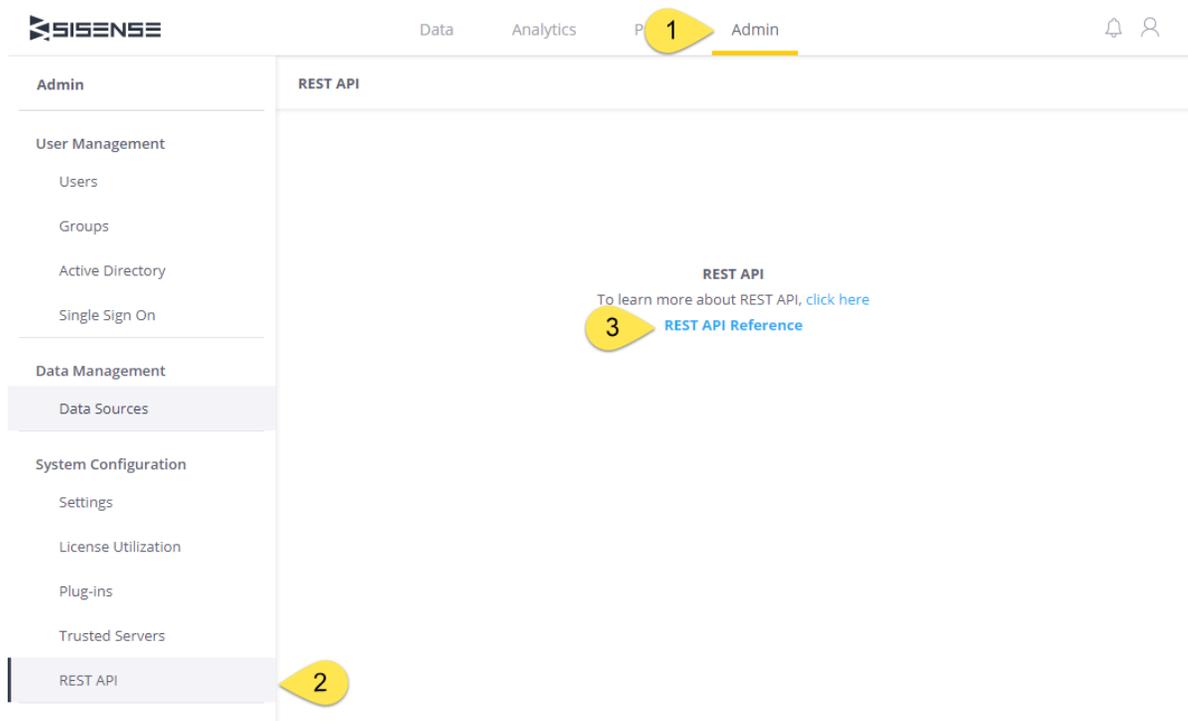
widget in the Widget Wizard, as described in [Adding Widgets to a Dashboard](#).

# Creating Dashboards through the REST API

The Sisense REST API enables you to programmatically create dashboards by defining the basic properties of a dashboard such as Title, datasource, and owner. After defining the properties of the dashboard, you define its visualizations by including the widget IDs of pre-existing widgets.

## To create a dashboard through the REST API:

1. In Sisense, click **Admin** at the top, and then **REST API**, followed by REST API Reference.



2. Select **Dashboards** then **POST /dashboards**.
3. In the body of your call, define the values of the following keys:

Key	Value
Title	String that defines the name of the dashboard as displayed in the Sisense Web Application.
desc	String that represents a short description of the dashboard.

Key	Value
datasource	<p>Object that defines the data source where the dashboard pulls its data.</p> <p>This object contains the following elements:</p> <p><b>title:</b> The name of the data source.</p> <p><b>id:</b> The address and name of the data source where the dashboard queries its data from.</p> <p><b>address:</b> The address of the dashboard.</p> <p><b>database:</b> The name of the data source where the data is pulled from by the dashboard.</p> <p><b>fullname:</b> The name of the data source.</p>
filters	<p>This object defines the dashboard filters that the user add goes to the "filters" object in the .dash file.</p> <p>This object contains the following elements:</p> <p><b>datatype:</b> The type of data affected by the filter, for example, text.</p> <p><b>title:</b> The name of the filter displayed in the Filters pane.</p>
layout	<p>Object that defines what widgets are displayed in the dashboard and their layout.</p> <p>This object contains the following elements:</p> <p><b>type:</b> String that defines how the layout of the dashboard. This value should be "columnar".</p> <p><b>columns:</b> Array objects that defines the width of each column in the dashboard and how widgets are displayed in each column.</p> <p><b>cells:</b> Object that contains the subcells.</p> <p><b>subcells:</b> Object that contains the elements objects where</p>

Key	Value
	widgets are defined. <b>elements:</b> This objects contains the widget and defines its dimensions within the data source.
defaultFilters	This object defines the default filters of a dashboard. When a user selects "Set as my default filters" from the Filters menu in Sisense, the filters are defined in this object.

4. Click **POST**. The Dashboard is added to the Dashboards list in the **Analytics** page.

## Adding Widgets to a Dashboard

This topic describes how to add widgets to your dashboard through the Widget Wizard.

The Widget Wizard automatically guides you through a very simple process of creating a widget, while offering the best recommendations for displaying the data that you select.

Almost always, simply accepting the wizard's recommendations will help you create a great widget. However, if you want more control over your widget's appearance and behavior, click the **Advanced Configuration** option in the bottom left of the wizard to access a rich variety of additional design options, as described in [Using the Widget Designer](#).

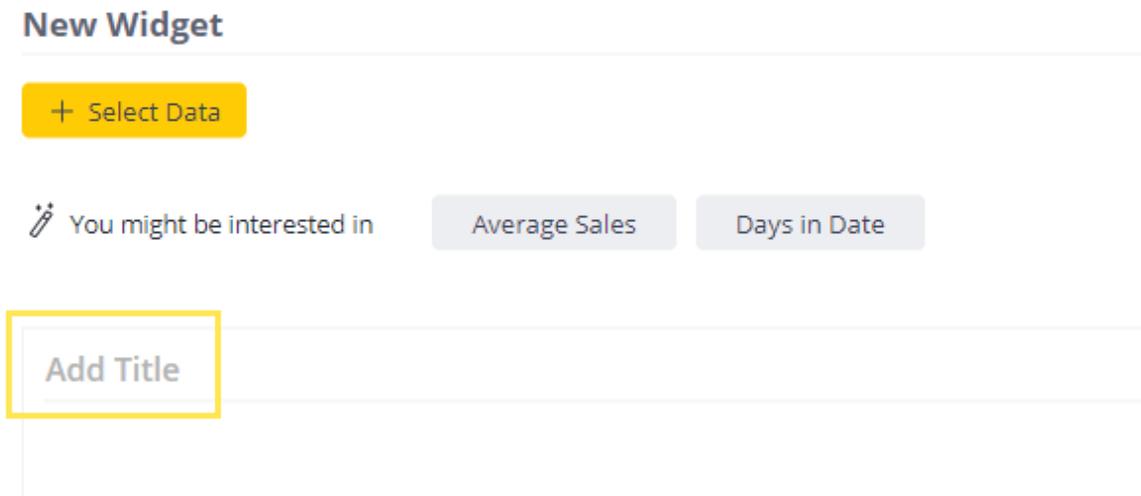
# Using the Widget Wizard

## To add a widget to the dashboard:

1. If this is your first widget, click **+ Select Data**. The Sisense Data Browser opens, and from there you can select one of the fields (columns) from this dashboard's ElastiCube or Live data source. If you already have widgets in your dashboard, then click the **+ Widget** button.

**Note:** To select data from a different ElastiCube, click **from** on the right side of the **New Widget** window, and select an ElastiCube.

2. You can add a title to your widget now or after creating your widget.



3. In the Data Browser, select a field to add to your widget. The field that you select is automatically displayed in a suggested widget. When only a single numeric field is selected, it is displayed as an Indicator widget. A single descriptive field is displayed as a Pivot widget.

## New Widget

from Sample Healthcare

Total Time of Stay + Add More Data...

You might be interested in

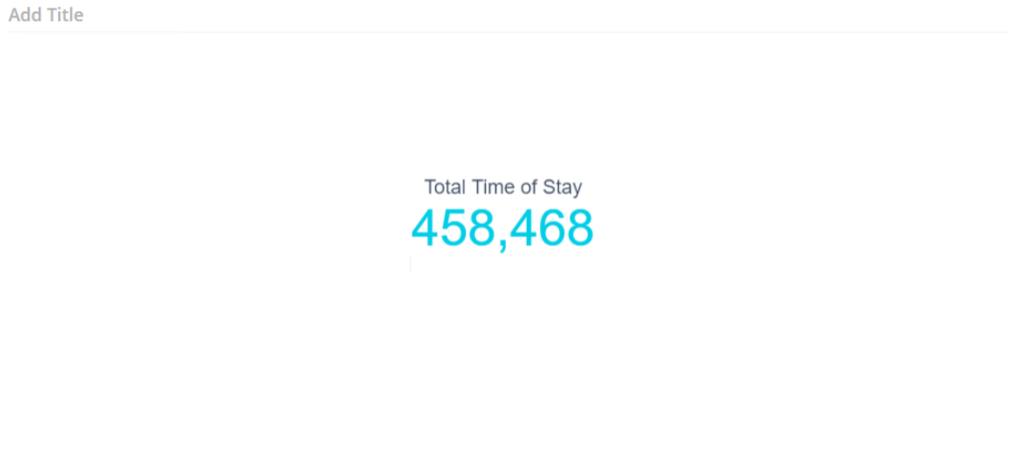
Average Cost\_of\_admission

# of unique Patient\_ID

DIAGNOSIS

Months in Admission\_Time

Diviso >



Advanced Configuration

Create

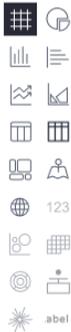
Cancel

## Single Numeric Field

## New Widget

from Commerce Demo

Brand + Add More Data...



Add Title

Brand
ABC
Adbananor WorldWide
Adcadopazz WorldWide Inc
Addimantor WorldWide
Addimegistor Inc
Addimin
Adericator
Aderor WorldWide
Adhupar
Adhupommazz Corp.
Adkilazz Holdings
Adkilewor WorldWide Corp.
Adnipax Direct

Rows 1-25 (752 Total)

Advanced Configuration

Create

Cancel

## Single Descriptive Field

But your widget is not really interesting yet, because it has only one field!

**New Widget** from Commerce Demo

Brand Total Revenue + Add More Data...

**Add More Fields**

**Selected Fields**

**Change Widget Type**

Brand	Total Revenue
ABC	251,866.95
Adbananor WorldWide	4,340.14
Adcadopazz WorldWide Inc	300.17
Addimantor WorldWide	6,726.56
Addimegistor Inc	200
Addimin	56.08
Adericator	99.69
Aderor WorldWide	3,166.59
Adhupar	10.96
Admazz Corp.	13,362.71
Adkilazz Holdings	5,603.10
Adkilewor WorldWide Corp.	2,822.54
Adnipax Direct	2,600.84

Rows 1-25 (752 Total)

Advanced Configuration Create Cancel

Repeat the step above to add more fields to the widget.

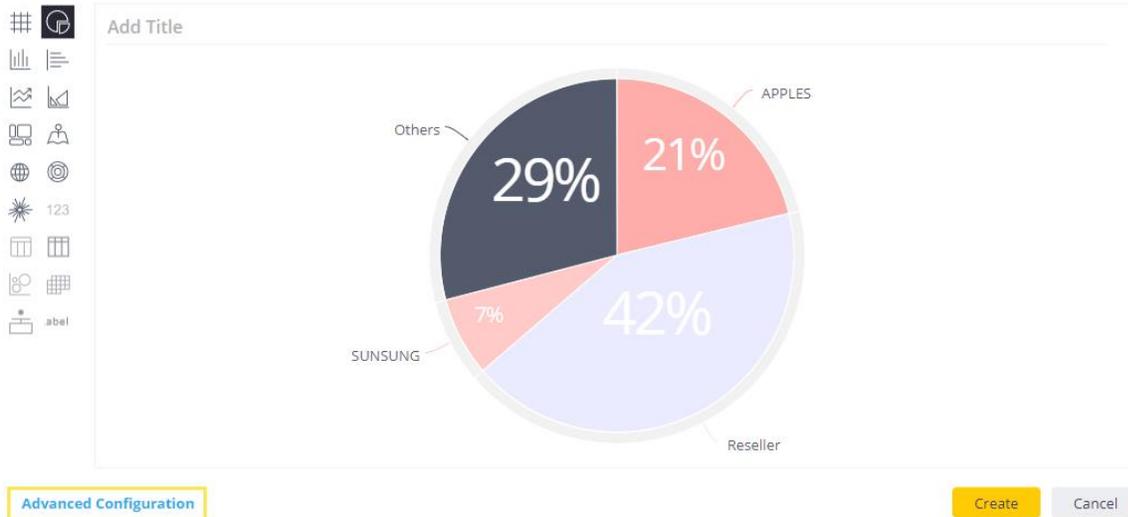
As you add fields, the relevant options are displayed as buttons to the left of your selection.

The fields that you select are listed across the top left of the wizard.

You can click on each visualization button to display the selected fields in that visualization/chart.

For example, the same widget as above could be displayed as a Pie chart:

Brand : Total Revenue : + Add More Data...



Advanced Configuration

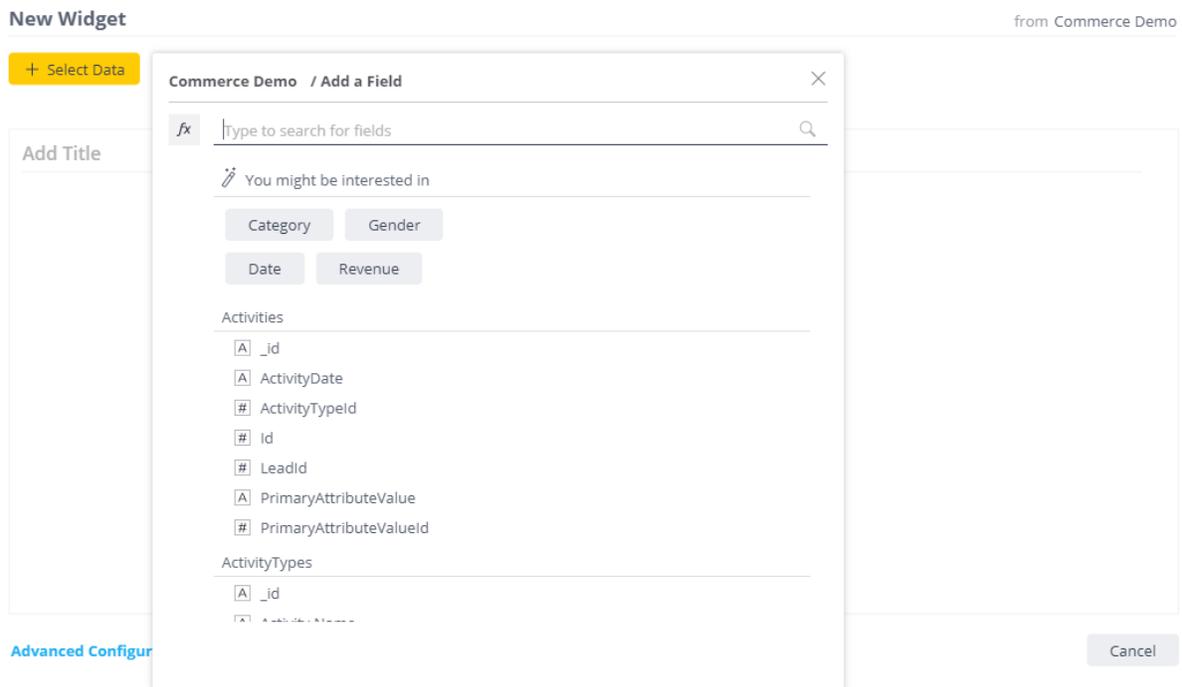
Create

Cancel

**Tip:** Sisense provides its best recommendation for your widget design. However, if you want more control, you can click the **Advanced Configuration** option on the bottom left of the window to provide a rich variety of additional design options, as described in [Using the Widget Designer](#).

# Data Browser

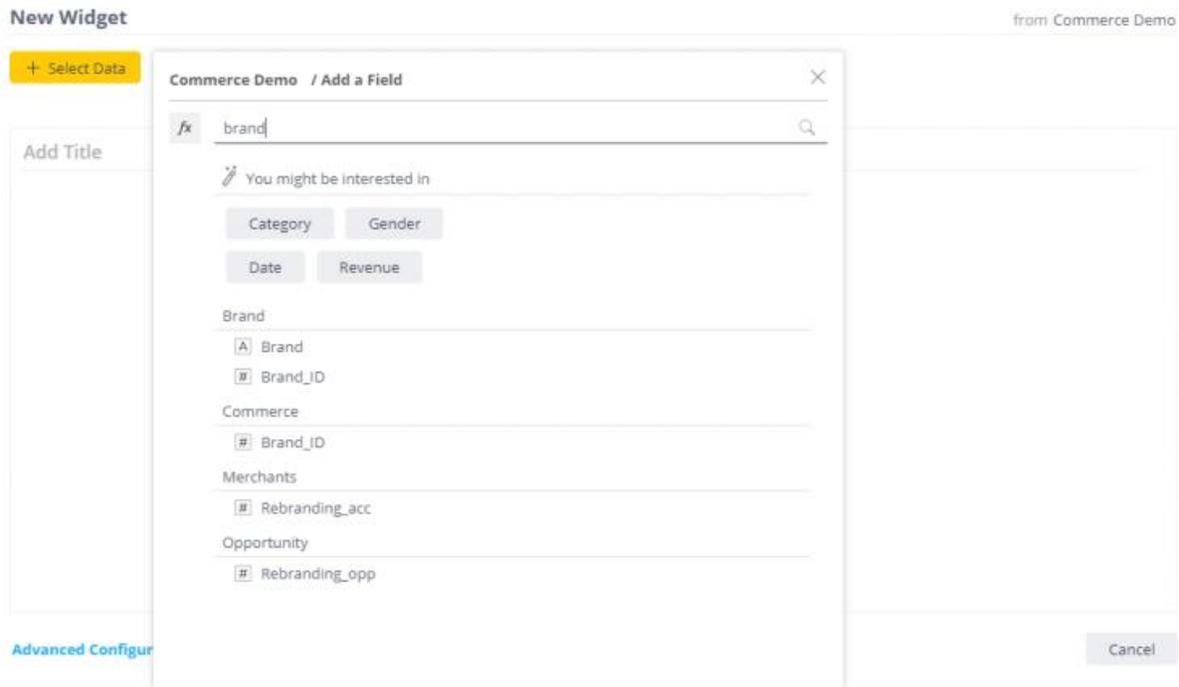
In the Data Browser, you can select and add columns (sometimes called fields) from an ElastiCube or a Live data source to your widget.



An icon appears to the left of each field to indicate its data type:

-  Date
-  Alphanumeric string
-  Numeric

As you type into the Data Browser, the list is dynamically filtered to only show the fields that contain the text you typed.



Simply clicking a field to add it to the widget generally provides great results, however there are more options in the Data Browser.

- Hover over a field in the Data Browser and click **More ...** to display additional aggregation (quick functions) and filtering options.
- Click the  button to define formulas (free-form expressions) that define the field values and filters of a widget. A rich variety of functions are provided for you to use in the formula that you define.

The screenshot shows a 'Commerce Demo / Add a Field' dialog box. At the top, there is a search bar with 'brand' entered. Below the search bar, there are several categories of fields:

- You might be interested in:** Category, Gender, Date, Revenue
- Brand:** Brand, Brand\_ID
- Commerce:** Brand\_ID
- Merchants:** Rebranding\_acc Sum (highlighted in yellow), More...
- Opportunity:** Rebranding\_opp

Two yellow arrows point to specific fields:

- An arrow labeled 'More Options' points to 'Brand\_ID'.
- An arrow labeled 'Default Behavior' points to 'Rebranding\_acc Sum'.

At the bottom left, there is a link for 'Advanced Configur...' and at the bottom right, there is a 'Cancel' button.

To learn more about these options, see [Using Formulas](#).

## Area Chart

An Area Chart is very similar to a Line Chart except that the areas under each line are filled in (colored), and it is possible to display them as stacked. The chart is recommended for displaying absolute or relative (stacked) values over a time period.

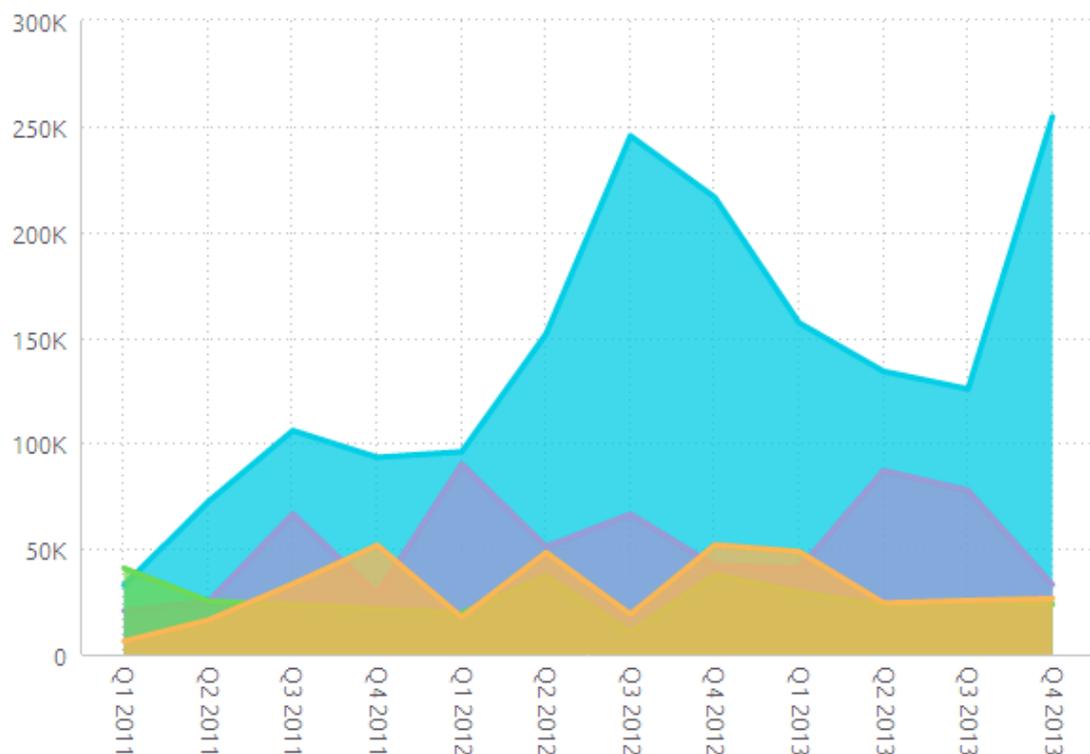
# Adding Data

1. In the Widget Designer, click **+ Select Data** to select the field(s) whose values will be placed on the X-Axis.  
Typically, the X-Axis of an Area Chart is used to represent time. You must add at least one item to the X-Axis. You can add at most two items. When you add two X-axis items, the chart's two X-Axis are combined.
2. In the **Values** area, click **Add+** to select the field whose value determines the height of the area in the chart. Typically, the Y-Axis of an Areas Chart is used to represent numeric data.  
You must add at least one field to **Values**. When you add more than one Value, then each item is represented in the chart by its own color and area.
3. **Break by** (optional): Click **Add +** to select a field by which to break up (group) the data represented in this chart.  
Each field added to **Break By** is represented by its own area in the chart and is automatically added to the chart's legend.  
This is an optional field, and operates under the following conditions:
  - You can add at most one **Break by** field.
  - The **Break by** option is available only when a single field was added to Values (Y-Axis).
  - You can add only a single field to Values, if the **Break by** option was used.

# Designing the Area Chart

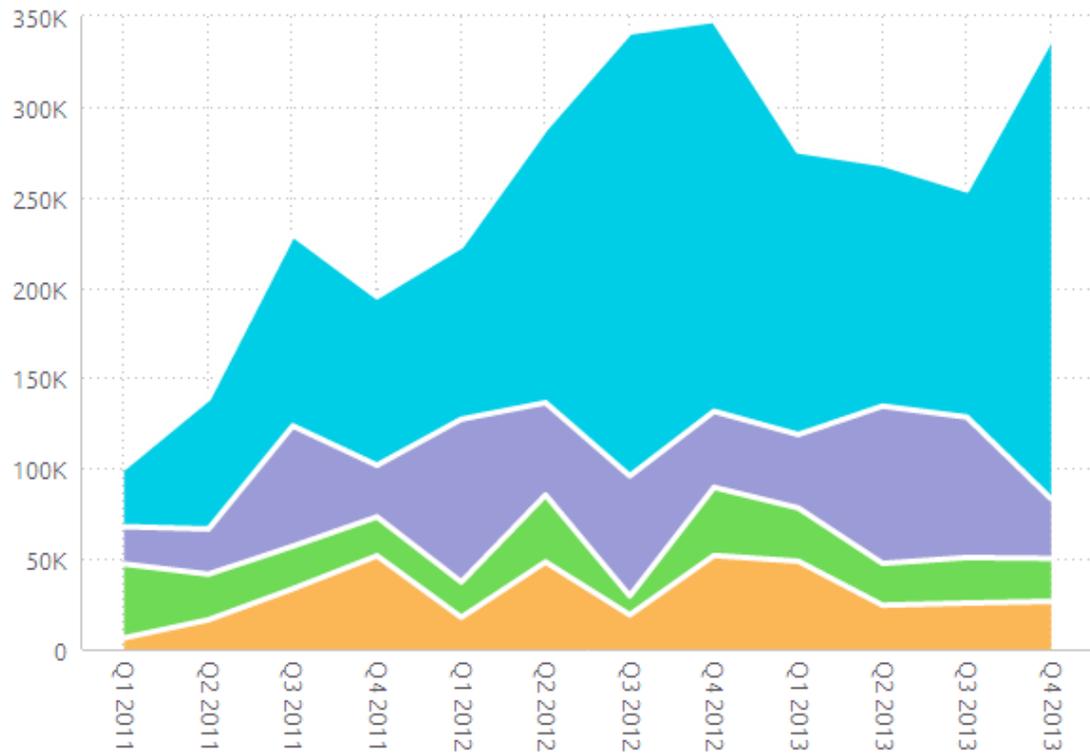
To fine-tune the appearance of the Area Chart widget, open the widget in Edit mode and on the right pane you can modify your Area Chart with the following options:

- **Area Types:** Select how areas are represented in the widget:
  - **Classic:** Areas overlap, meaning that smaller areas appear on top of larger areas and cover them. The colors indicate the distribution between the values.



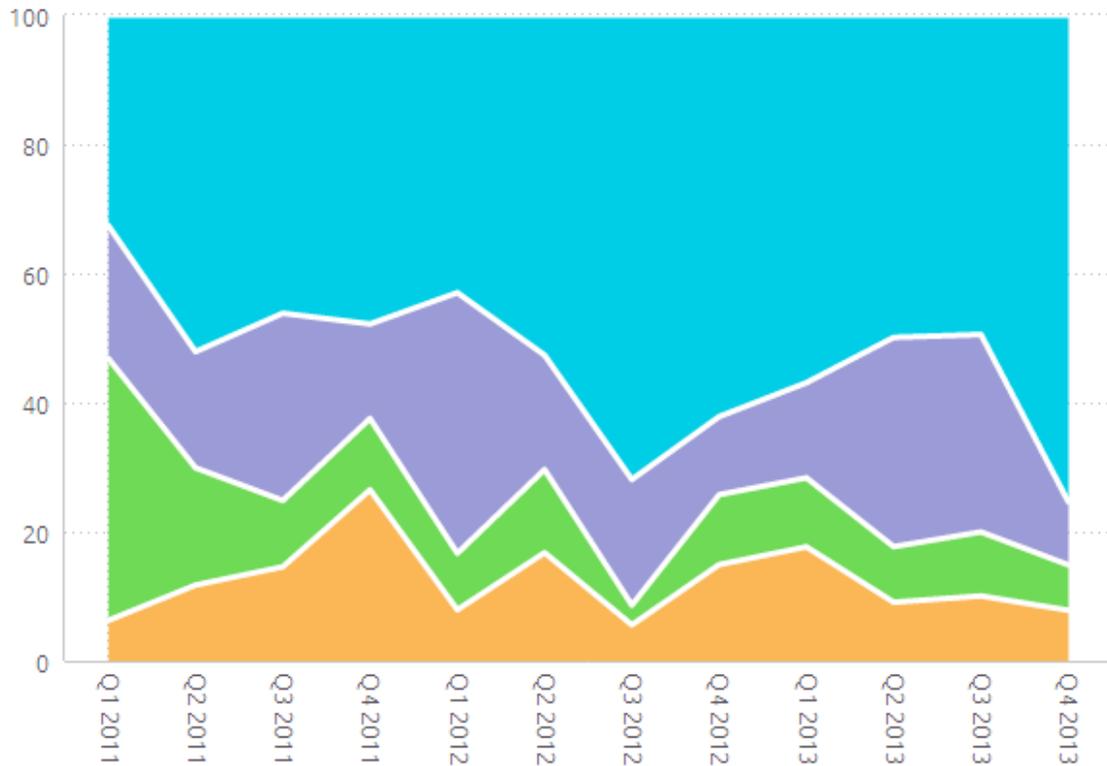
- **Stacked:** Areas are stacked on top of each other and do not overlap. Hover over the columns to see the percentage distribution among the

values.

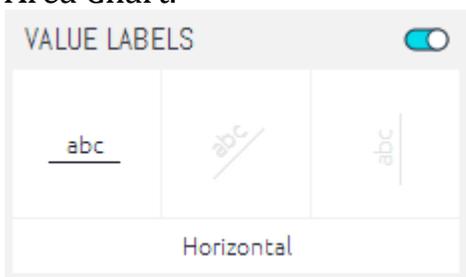


- **Stacked 100:** Areas are stacked on top of each other (but do not overlap). The combined area is stretched to the top in order to represent 100%. This option is most commonly used when the relative

distribution of the values is more important than their aggregation.



- **Line Type:** Select how the line appears in the widget: **Straight** (Default) or **Smooth**.
- **Line Width:** Select either **Thin**, **Bold** (Default) or **Thick**.
- **Legend:** Specify whether to show or hide the Legend and its position.
- **Value Labels:** Specify whether to show or hide labels showing values in the Area Chart.

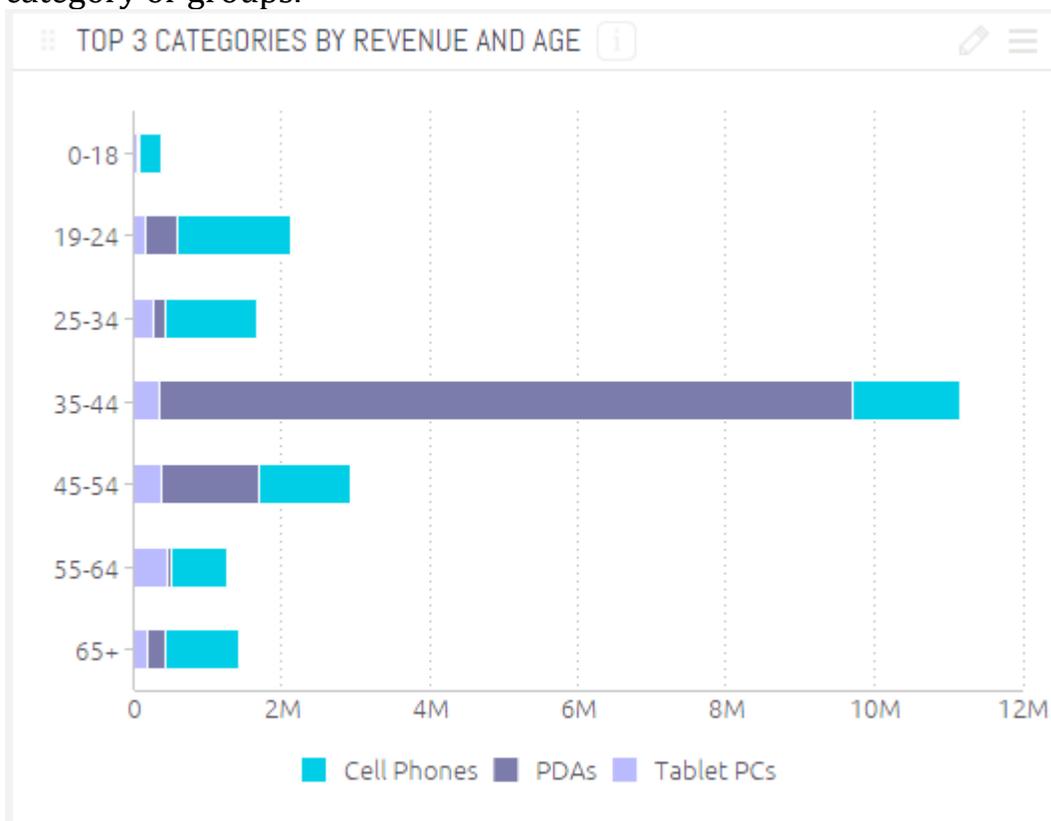


- **Markers:** Specify whether to show or hide markers over data points.
- **X-Axis:**
  - **Grid Lines:** Shows (Default) / hides.
  - **Labels:** Shows (Default) / hides.
  - **Title:** Select the checkbox to display the x-axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.
- **Y-Axis**

- **Grid Lines:** Shows (Default) / hides.
- **Logarithmic:** Displays using orders of magnitude.
- **Labels:** Shows (Default) / hides.
- **Title:** Select the checkbox to display the y-axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.
- **Values on Axis:** To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values. To restore any of the values to their default states, click on the reset button. ↻

## Bar Chart

The Bar Chart is commonly used to compare many items. The Bar Chart typically presents categories or items (descriptive data) displayed along the Y-Axis, with their values displayed on the X-Axis. You can also break up the values by another category or groups.



# Adding Data

1. In the **Categories** panel, click **Add+** to select the field(s) whose values will be placed on the Y-Axis.

Typically, the Y-Axis of a Bar Chart is descriptive data. You must add at least one item to **Categories**, and a maximum of two items. When you add two fields to **Categories**, the chart's Y-Axis are combined.

2. In **Values**, select the fields whose values determine the length of each column. Typically, the values of a Bar Chart are used to represent numeric data.

You must add at least one field to **Values**. When you add more than one value, each item is represented in the chart by its own color and area.

3. (Optional) Select a field by which to break (group) the data represented in this chart by clicking **Add +**.

Each group is represented by a different column and is automatically added to the chart's legend.

Break by is optional, and operates under the following conditions:

- You can add at most one **Break by** field.
- The **Break by** option is available only when a single field was added to **Values** (Y-AXIS).
- You can only add a single field to **Values** if the **Break by** option was used.

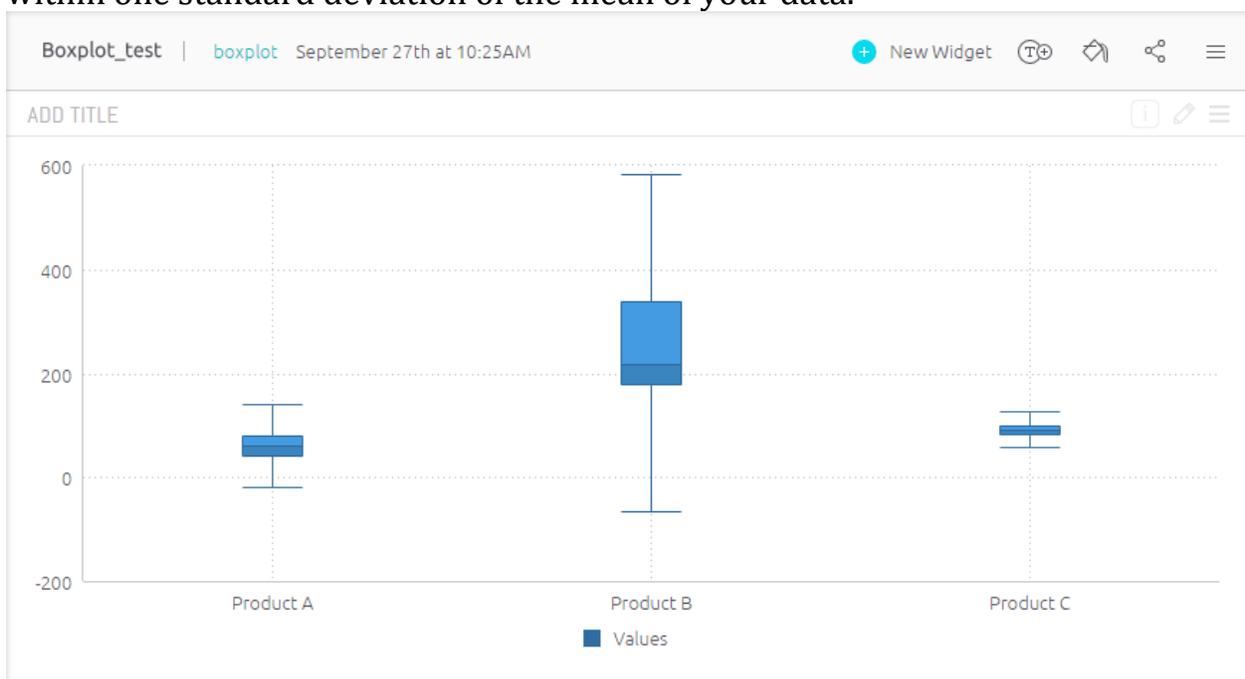
# Designing the Bar Chart

Fine-tune the appearance of the Bar Chart widget. The design options are the same as for [Column Chart](#).

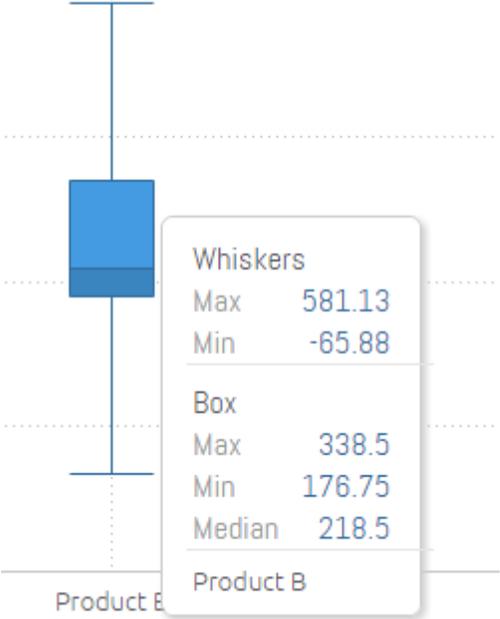
## Box & Whisker Plot

The Box & Whisker Plot, or Box Plot, widget is a convenient way of visually describing the distribution, variability, and center of a data set along an axis. Box Plots are divided into four quartiles. The middle quartiles are represented by a box that contains 50% of the data and the median value. The upper and lower quartiles contain the maximum and minimum values and the remaining 50% of the data. These quartiles are represented by lines called whiskers.

The maximum and minimum values can be adjusted when defining your box plot. By default, these values are within 1.5 times of the IQR (Interquartile Range), however you can set them to the actual maximum and minimum values or to within one standard deviation of the mean of your data.



Box plots divide five descriptive statistics into four equal quartiles. You can view these statistics by hovering over the Box Plot.



# Adding Data

1. In the **Categories** panel, click **Add +** to select the field whose values will be placed on the X-Axis. The X-Axis of a box plot is used to represent numeric data.
2. In the **Values** panel, click **Add +** to select the field whose values will be placed on the Y-Axis.

# Design the Box Plot

You can fine tune the appearance of the Box Plot widget with the following tools.

- **Whisker Values:** Defines how the whiskers are calculated and displayed.
  - **IQR (Interquartile Range):** Default setting. The whisker values are calculated within 1.5 times the IQR (the difference between the 1st and 3rd quartiles).
  - **Extremums:** The whiskers' values include the maximum and minimum values in the calculation.
  - **Standard Deviation:** The whiskers' values are calculated to one standard deviation above and below the mean of the data.
- **Box:** Specify whether to display a full or hollow box.
- **Value Labels:** Specify whether to show or hide labels indicating the values of each point (circle). Select the orientation of these labels: **Horizontal** (Default), **Diagonal** or **Vertical**.
- **Legend:** Specify whether to show or hide the Legend, and select its position.
- **Show Outliers:** Specify whether to show or hide labels indicating the values of each point (circle). Select the orientation of these labels: **Horizontal** (Default), **Diagonal** or **Vertical**.
- **X-Axis and Y-Axis:**
  - **Grid Lines:** Shows (Default) / hides.
  - **Labels:** Shows (Default) / hides.
  - **Title:** Select the checkbox to display the axis title. To edit the title, type in a new title. Click outside the text box to apply the new value. To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values.
  - **Values on Axis:** To restore any of the values to their default states, click the reset button.
- **Auto Zoom:** When a widget contains more data than can comfortably be displayed in one view, Auto Zoom will resize the chart to include more data, and in some cases adds a zoom bar under the widget. The zoom bar enables you to scroll right and left and to zoom in/out of different parts of the width of the widget.

# Calendar Heatmap

The Calendar Heatmap widget visualizes values over days in a calendar-like view, making it easy to identify daily patterns or anomalies.

The screenshot displays the Sisense Calendar Heatmap widget interface. The main area shows a heatmap for the year 2013, with data points represented by colored cells in a calendar grid. The colors range from red (high values) to green (low values). The interface includes a left sidebar for configuration, a top navigation bar, and a right sidebar for additional settings.

**DATE:** Days in Date

**COLOR:** # of Visit ID

**CALENDAR HEATMAP CONFIGURATION:**

- MONTHS: Year view
- CALENDAR TYPE: Classic (selected), Week View
- WEEK BEGINS ON: Sunday (selected), Monday
- GRAYOUT WEEKENDS:
- SHOW DAY NUMBER:
- SHOW DAY NAME:

**HEATMAP DATA (Approximate Values):**

Month	Sun	Mon	Tue	Wed	Thu	Fri	Sat
January	6	13	20	27	3	10	17
February	3	10	17	24	3	10	17
March	10	17	24	31	7	14	21
April	28	5	12	19	26	3	10
May	7	14	21	28	4	11	18
June	25	2	9	16	23	30	7
July	14	21	28	4	11	18	25
August	1	8	15	22	29	5	12
September	19	26	3	10	17	24	31
October	7	14	21	28	4	11	18
November	25	2	9	16	23	30	6
December	13	20	27	3	10	17	24

# Adding Data

1. In the **Date** panel, select the date field that you want to be used as the calendar.  
The calendar's range will be between the minimum and maximum date values within the field.
2. In the **Color** panel, click **Add +** to select the field that determines which days to color.

# Designing the Heatmap

Fine-tune the appearance of the Calendar Heatmap widget, using the following tools.

- **Months:** Select how many months should be presented at once: 1 (One Month View), 3 (Quarter View), 6 (Half-Year View) or 12 (Year View). When more months exist than currently visible, paging arrows are displayed to move between months.
- **Calendar Type:** Select between two types of view modes:
  - Classic – Each month is presented in typical calendar format.
  - Week View – Months are ordered by weekdays from top to bottom.



- **Week Begins On:** Select the first day of the week to be represented in the calendar.

## More Configuration Options

- Grey Out Weekends – If on, the weekend days will be colored grey in the calendar.
- Show Day Number – Toggle whether to show the calendar day numbers inside the widget.
- Show Day Name – Toggle whether to show the day name abbreviations on the calendar.

### Column Chart

The column chart can be used in different business scenarios, especially for comparing items, and comparing data over time. The chart can include multiple values on both the X and Y-axis, as well as a break down by categories displayed on the Y-axis.

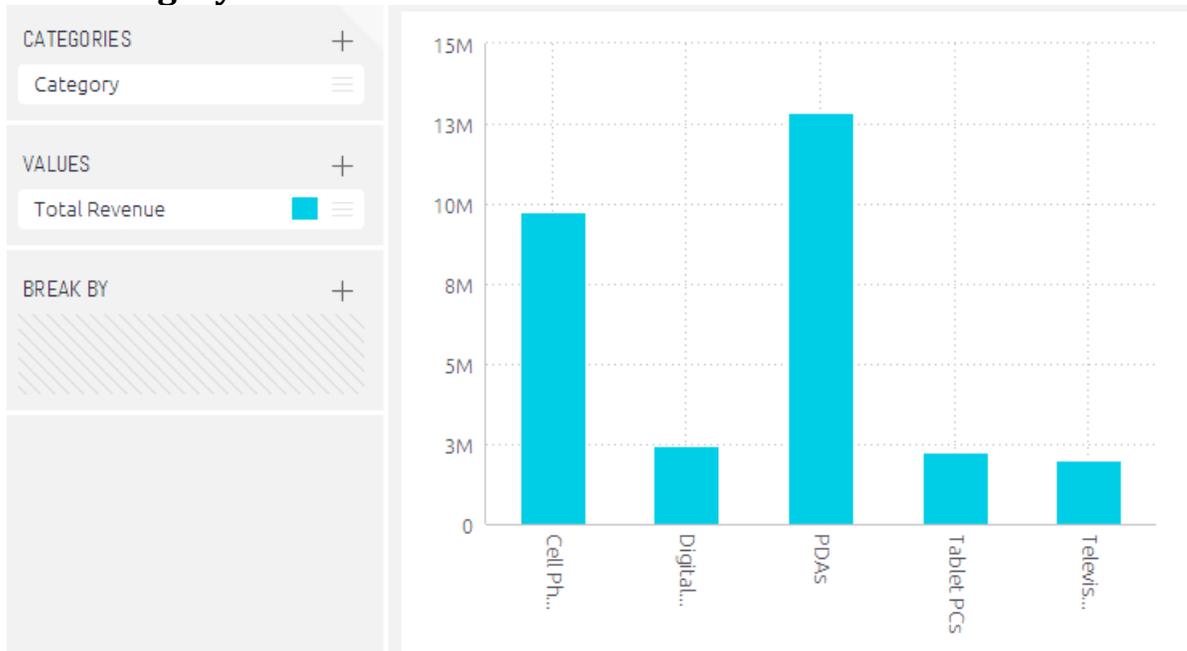
# Adding Data

1. In the **Categories** panel, click **Add +** to select the field(s) whose values will be placed on the X-Axis.

Typically, the X-Axis of a Column Chart is descriptive data. You must add at least one item to **Categories**, and at most, two items.

When two **Category** items are added, then the chart's X-Axes are automatically grouped. To change the order of the categories, drag a category up or down in the Categories list.

## One Category Item



## Two Category Items



2. In the **Values** area, select the field whose value determines the height of the columns by clicking **Add +**. Typically, the Y-Axis of a Column Chart is used to represent numeric data.

You must add at least one field to **Values**. When more than one Value is added, then each item is represented in the chart by its own color and area.

3. **Break by**: Select a field by which to break (group) the data represented in this chart by clicking **Add +**.

Each group is represented by a different column and is automatically added to the chart's legend.

This is an optional field, and operates under the following conditions:

- At most one **Break by** field can be added.
- The **Break by** option is available only when a single field was added to **Values** (Y-Axis).
- Only a single field can be added to **Values**, if the **Break by** option, described below, was used.

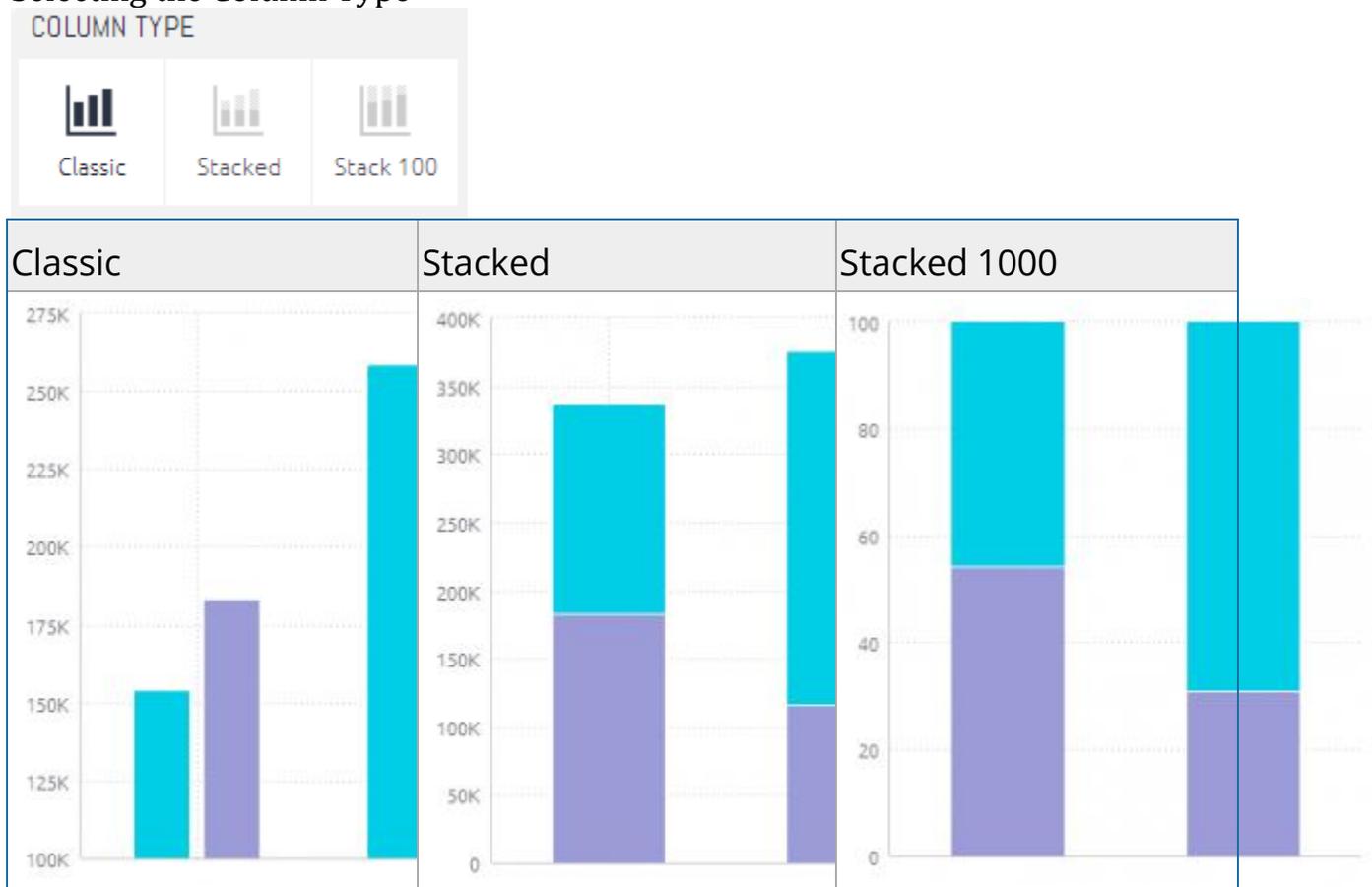
4. Define filters for the widget, as described in [Filtering Dashboards](#).

# Designing the Column Chart

Fine-tune the appearance of the COLUMN CHART Widget, using the following tools:

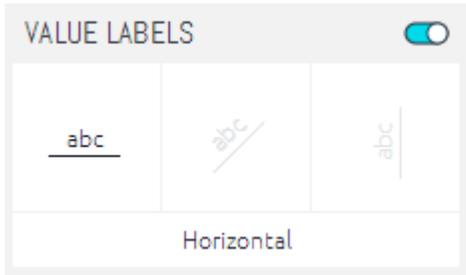
- **Column Type:** Select how columns are represented in the Widget:
  - **Classic:** Columns are displayed side by side.
  - **Stacked:** Columns are stacked on top of each other and do not overlap. The tooltips over the columns show the percentage distribution among the Values.
  - **Stacked 100:** Columns are stacked on top of each other (but do not overlap) and the combined column is stretched to represent 100%. This option is most commonly used when the relative distribution of the values is more important than their aggregation.

Selecting the Column Type



- **Legend:** Specify whether to show or hide the Legend and its position.

- **Value Labels:** Specify whether to show or hide labels showing values in the Column Chart. You can also select the angle of the labels.



- **X-Axis:** Enable or disable the following options:
  - **Grid Lines:** Shows (Default) / hides.
  - **Labels:** Shows (Default) / hides.
  - **Title:** Select the checkbox to display the X-Axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.
- **Y-Axis:** Enable or disable the following options:
  - **Grid Lines:** Shows (Default) / hides.
  - **Logarithmic:** Displays using orders of magnitude.
  - **Labels:** Shows (Default) / hides.
  - **Title:** Select the checkbox to display the y-axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.
  - **Values on Axis:** To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values. To restore any of the values to their default states, click on the reset button.
- **Auto Zoom:** When a widget contains more data than can comfortably be displayed in one view, Auto Zoom will resize the chart to include more data, and in some cases adds a zoom bar under the widget. The zoom bar enables you to scroll right and left and to zoom in/out of different parts of the width of the widget.

## Indicator

The Indicator widget provides various options for displaying one or two numeric values as a number, gauge or ticker. It also provides the option to add additional titles and a color-coded indicator icon representing the value, such as a green up arrow or a red down arrow.

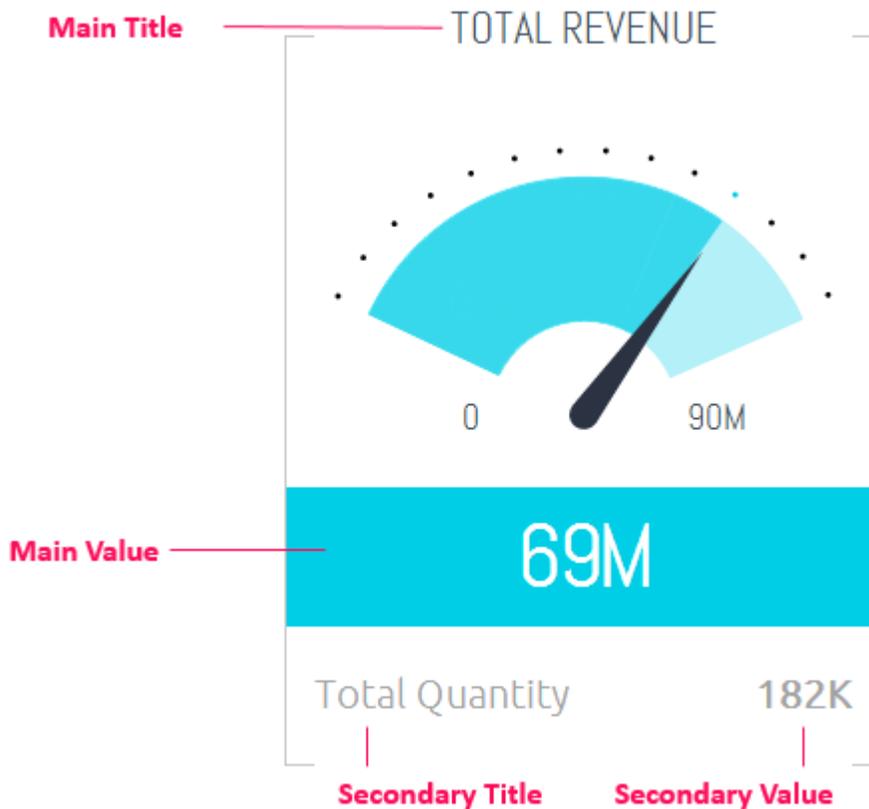
# Adding Data

1. Click **Add +** in the **Value** panel to select one field whose name and value will be shown as the **Main Title** and **Main Value**, as shown below.
2. Click **Add +** in the Secondary panel to select one field whose name and value will be shown as the Secondary Title and Secondary Value, as shown below.

## Numeric Indicator



## Gauge Indicator



3. For a gauge, select the minimum and maximum values that can be represented by the gauge. These values can either be set as a fixed numeric value that you specify or as another numeric field that you select using the Data Browser (thus making the value dynamic).

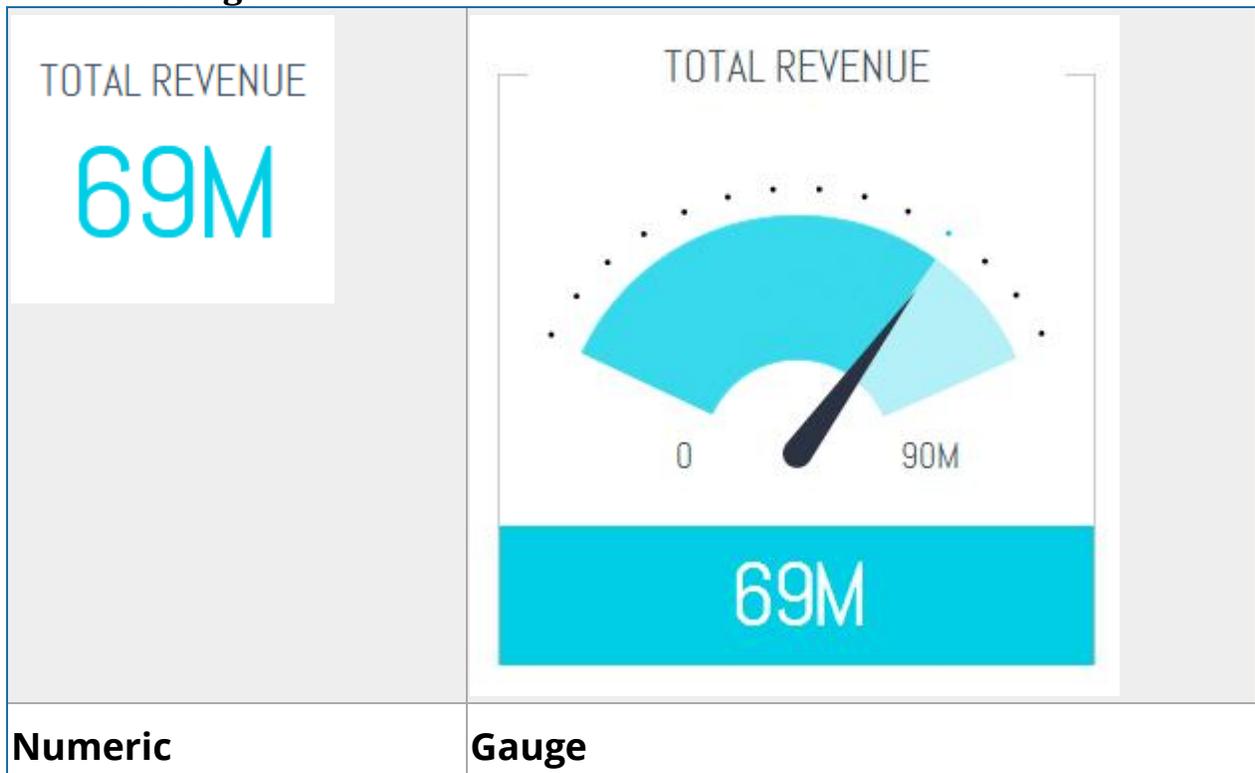
MIN	<input type="text" value="0 (default)"/>
MAX	<input type="text" value="150"/>

In an Indicator widget, you can also define whether an icon appears in the Widget and the icon's color. For example, you can define a red arrow pointing downwards for a negative value. You may refer to [Defining Conditional Coloring - Condition Tab](#) for more information.

# Designing the Indicator

Fine-tune the appearance of the Indicator widget, using the following tools:

- **Indicator Type:** Select whether the Indicator appears in **Numeric** form or as a **Gauge**.



**Note:** If you significantly shorten the height of an Indicator widget in the dashboard, it automatically turns into a Ticker type Widget. You may refer to [Ticker Widgets](#) for more information.

| GROWTH OF VISITORS: 13.2% | # of Visitors: 220K |

- **Skin:** Select the skin of the gauge.
- **Components:** Select which labels to include in the widget.
- **Indicator Sub Type**(for numeric indicator only): You can choose a another skin option.

## Line Chart

The line chart can be used for various business cases, including:

- Comparing data over time, for example: to analyze sales revenue for the past year.
- Comparing changes over the same period of time for more than one group or category. Example: Analyze expenditures of different business units for the past year.

# Adding Data

1. Add data to the X-axis.

Click **Add +** to select the field(s) that will be represented on the X-Axis.

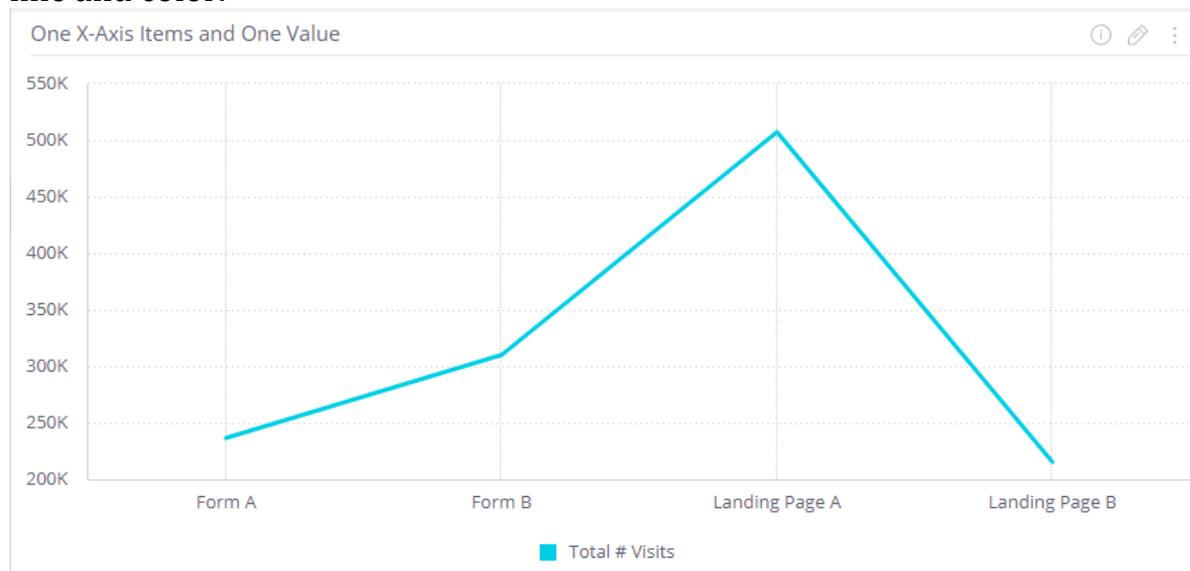
Typically, the x-axis of a Line chart is used to represent time.

You must add at least one item to the x-axis, and at most, two items.

When you add two x-axis items, the chart's x-axes are combined and a separate plot (line) is displayed for each unique value of the second selected x-axis.

2. Add data to the Y-axis.

In the **Values** area, click **Add +** and select the field whose value determines the height of the lines in the chart. Typically, the Y-Axis of a line chart is used to represent numeric data. You must add at least one field. If you add more than one field, then each item is represented in the chart by its own line and color.



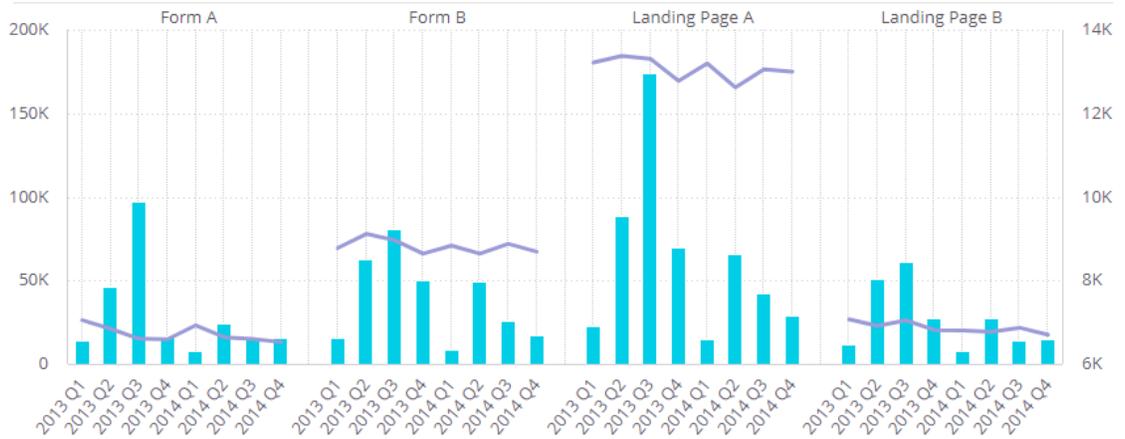
Two X-Axis Items and Two Value



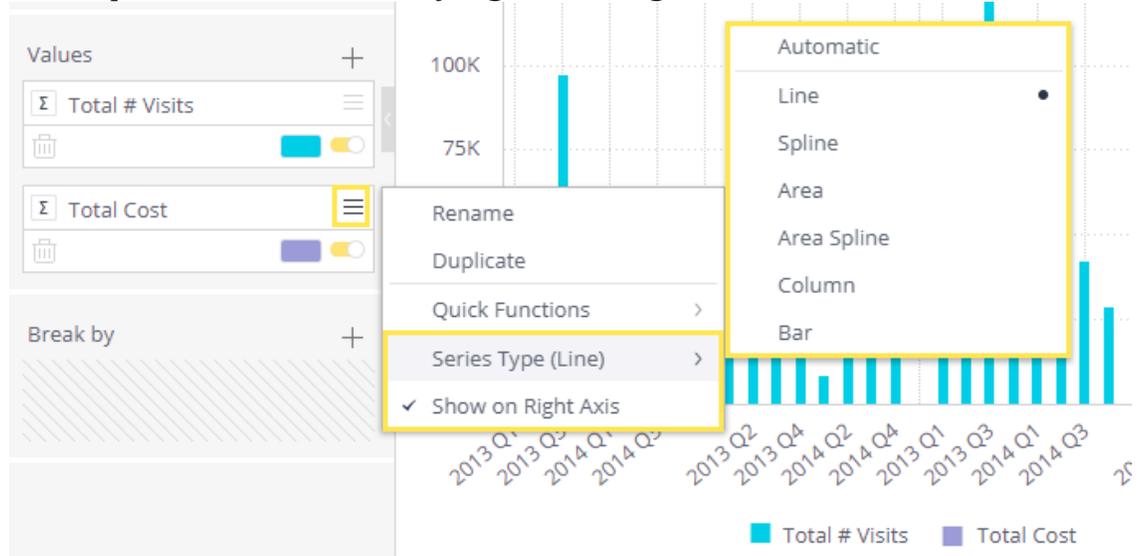
Another version of the same chart includes the following differences:

- The second value has been changed into a column chart.
- The second value has been added to the right axis. This is a useful option when the values are using a considerably different scale.

Two X-Axis Items and Two Value



Both options are available by right-clicking on the value.



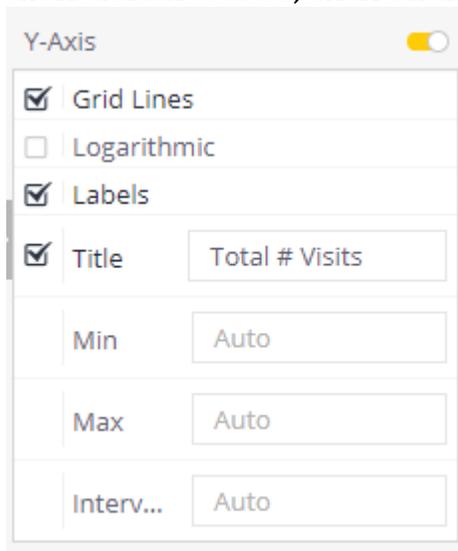
3. (Optional) Select a field by which to break up the data. Click **Add +**. Each item that is a member of the field added to **Break by** is represented by its own line in the chart, and is automatically added to the chart's legend. You can add only one **Break by** field. The **Break by** option is only available when you have a single field in the Y-Axis.

# Designing the Line Chart

Fine-tune the appearance of the line chart, using the following tools:

- **Line Width:** Select either **Thin**, **Bold** (default) or **Thick**.
- **Line Type:** Select how the line appears in the widget: **Straight** (default) or **Smooth**.
- **Legend:** Toggle the legend on or off, and select its position in the chart.
- **Value Labels:** Toggle labels on or off, and define how labels are displayed: **Horizontal**, **Vertical** or **Diagonal**.
- **Markers:** Select whether to show or hide markers over data points. Select **Full** or **Hollow**, and **Small** or **Large** markers.
- **X-Axis:**
  - **Grid Lines:** Show (default) or hide grid lines.
  - **Labels:** Show (default) or hide labels.
  - **Title:** Select the checkbox to display the x-axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.
  - **Values on Axis:** To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values. To restore any of the values to their default states, click on the reset button. reset button
- **Y-Axis**
  - **Grid Lines:** Show (default) or hide grid lines.
  - **Logarithmic:** Displays a logarithmic scale instead of a linear scale on the axis.
  - **Labels:** Show (default) or hide labels.
  - **Title:** Select the checkbox to display the y-axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.
  - **Values on Axis:** To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values. To restore any of the values to

their default states, click on the reset button. ↻



The image shows a settings panel for a Y-axis. At the top, it is titled "Y-Axis" and has a yellow toggle switch that is turned on. Below the title, there are several rows of settings, each with a checkbox on the left and a text input field on the right:

- Grid Lines
- Logarithmic
- Labels
- Title: Total # Visits
- Min: Auto
- Max: Auto
- Interv...: Auto

## Pie Chart

The Pie chart is used to display proportional data, and/or percentages.

# Adding Data

1. In the Categories panel, click **Add +** to select the data that will be displayed in the Pie Chart. You can add only one field.

Typically, the categories of a Pie Chart contain descriptive data.

**Note:** All slices that are smaller or equal to 3% are grouped together into a slice called **Other**. You can hover over this slice to display its breakdown.

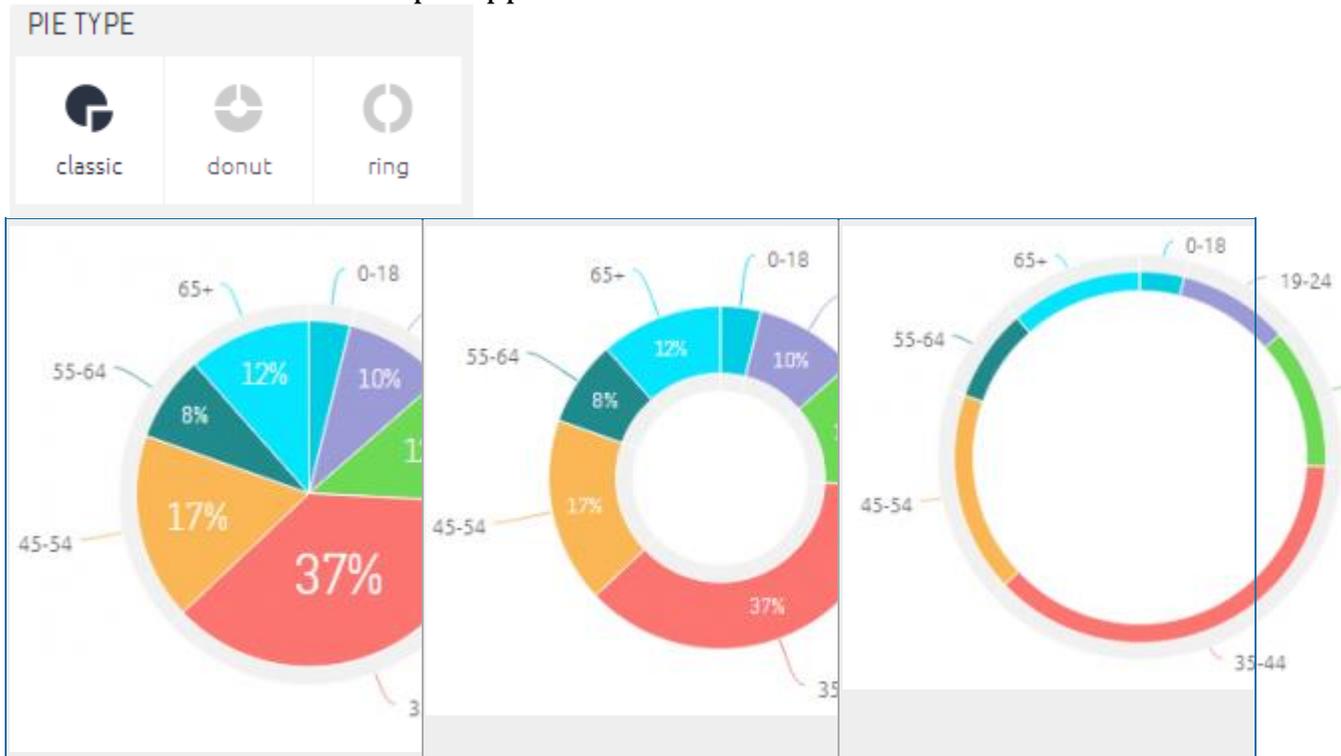
2. In the **Values** area, select the field whose values determine the size of the pie slice by clicking **Add +**. Values are typically used to represent numeric data.

**Note:** You can add only one field.

# Designing the Pie Chart

Fine-tune the appearance of the Pie Chart widget, using the following tools:

- **PIE TYPE:** Select how the pie appears.

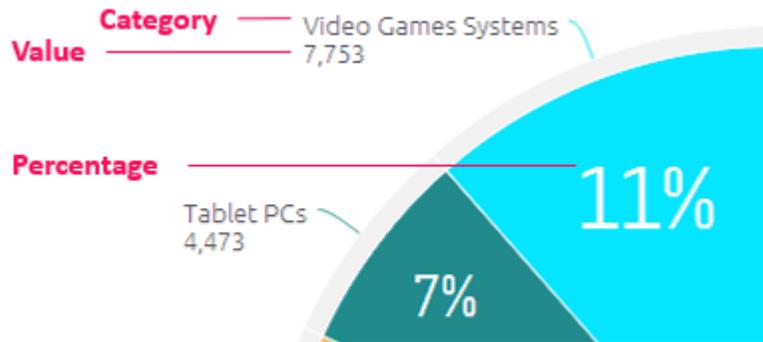


- **Legend:** Specify whether to show or hide the Legend, and select its position.

- **Labels:** Select which labels to display on the pie chart, as indicated below:

Labels

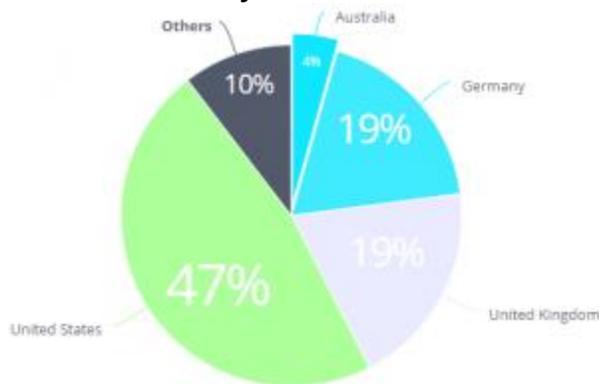
<input checked="" type="checkbox"/> Categories
<input type="checkbox"/> Value
<input checked="" type="checkbox"/> Percent
<input type="checkbox"/> Decimals



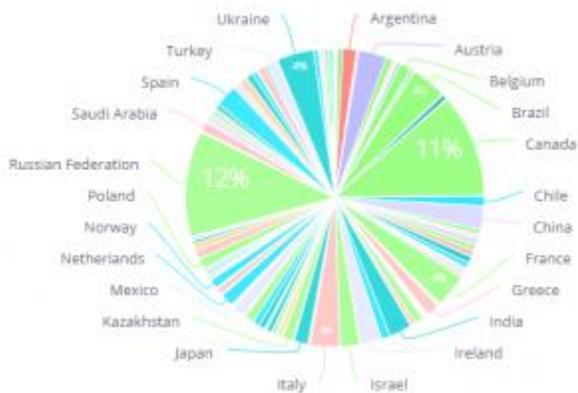
# Exploring Pie Charts

You can [drill down](#) into pie charts just like other widgets. In addition, when your pie chart has so many smaller slices that they are aggregated into a single slice called Other, you can click this slice to display a second pie chart containing a breakdown of the Other slice.

For example, the Pie chart below displays a breakdown of the top 5 slices in a pie chart with many smaller slices included in the Others slice.



By clicking the Others slice, you can see a breakdown of all the slices included in the Others slice.



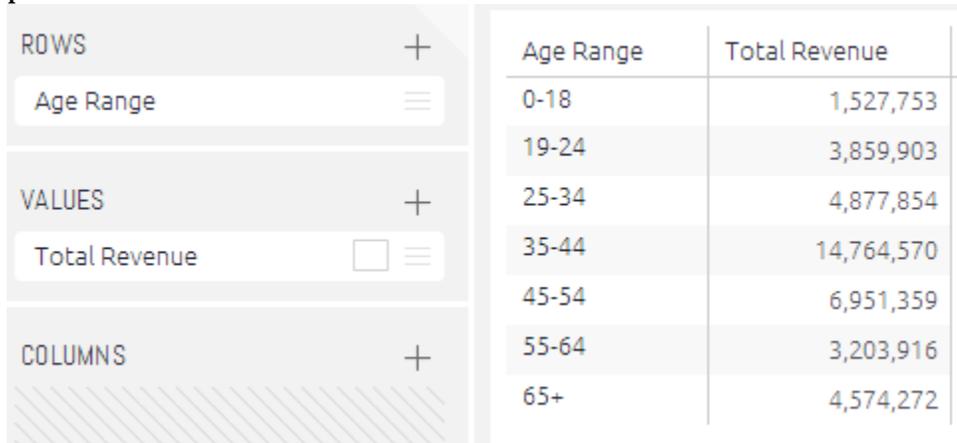
## Pivot

Pivot tables are one of the most useful widgets for visualizing data. They enable you to quickly summarize and analyze large amounts of data.

# Adding Data

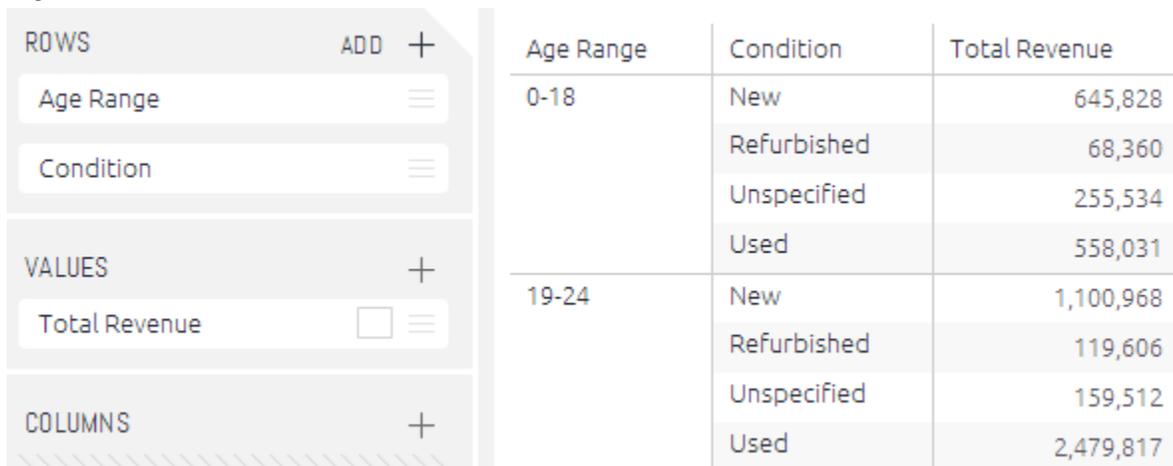
Select the data to appear in the Pivot.

1. In the **Rows** panel, click **Add +** to select the field(s) whose values will be placed in the rows of the Pivot table.



Age Range	Total Revenue
0-18	1,527,753
19-24	3,859,903
25-34	4,877,854
35-44	14,764,570
45-54	6,951,359
55-64	3,203,916
65+	4,574,272

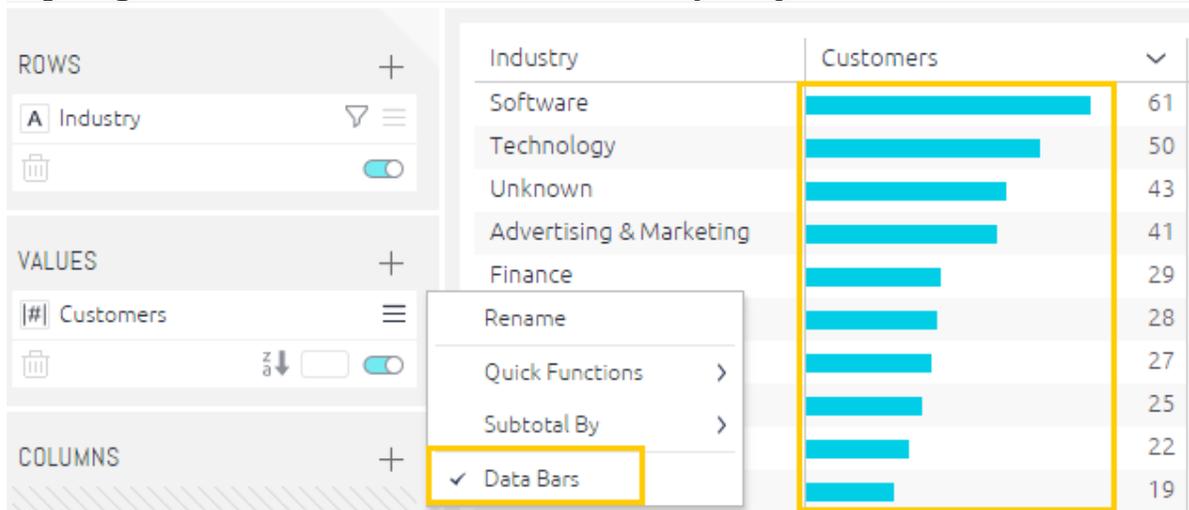
When more than one Row is selected, the Rows are broken into sub-rows (groups). For example, the following shows **Condition** added as a second Row:



Age Range	Condition	Total Revenue
0-18	New	645,828
	Refurbished	68,360
	Unspecified	255,534
	Used	558,031
19-24	New	1,100,968
	Refurbished	119,606
	Unspecified	159,512
	Used	2,479,817

2. In the **Columns** panel, click **Add +** to select the field(s) whose values will be placed in the columns of the Pivot table.  
You can drag and drop the fields that you added to **Columns** to the **Rows** area and vice versa.
3. In the **Values** panel, select a field whose values will appear in the Rows and Columns of the Pivot table.

**Tip:** Right-click the value to add data bars to your pivot table.

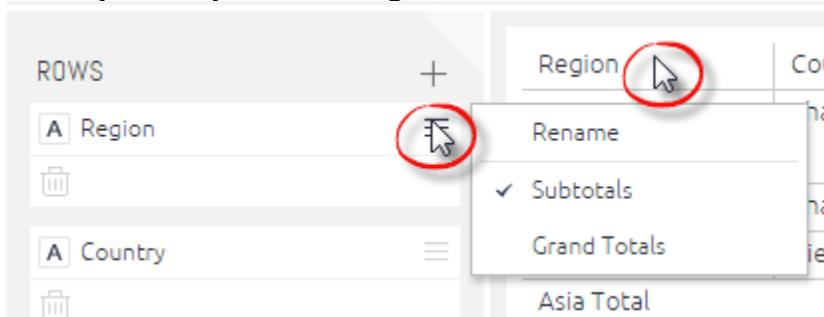


# Subtotals

You can add subtotals to one or more rows, and define how to calculate subtotals.

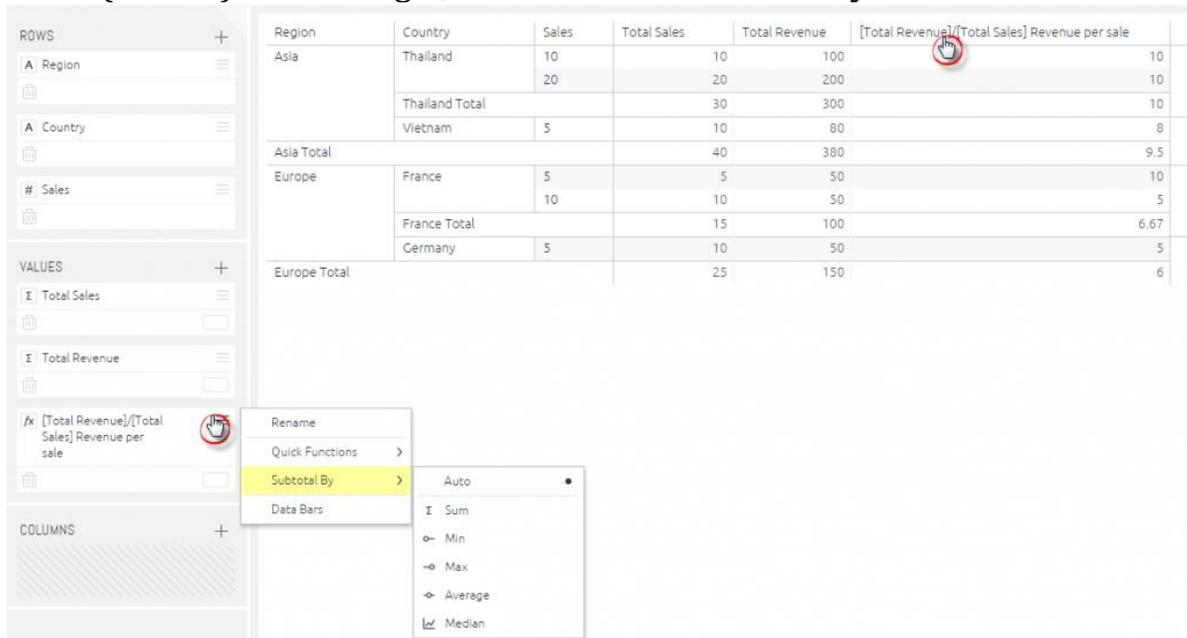
## To add subtotals to rows:

- Click on the menu of the Row in the Data Panel, or right-click on the row name (header) in the widget, and select Subtotals.



## To define how to calculate the subtotals:

- Click on the menu of the Value in the data panel, or right-click on the value name (header) in the widget, and then click **Subtotal by** and the method.



The default option is Auto, which aggregates all the data. The other custom options (Sum, Min, Max, etc.) calculate only the values in the rows above the subtotal.

Look at the following example:

Region	Country	Total Sales	Total Revenue	Ave. Revenue per sale
Asia	Cambodia	20	300	15
	Thailand	30	300	10
	Vietnam	15	1,050	70
Asia Total		65	1,650	<b>AUTO</b> → 25.38

Region	Country	Total Sales	Total Revenue	Ave. Revenue per sale
Asia	Cambodia	20	300	15
	Thailand	30	300	10
	Vietnam	15	1,050	70
Asia Total		65	1,650	<b>AVERAGE</b> ↶ ↷ 31.67

In the top example, using the **Auto** option, the subtotal for average revenue aggregates all the sales and revenue data, thus representing a true subtotal of the average revenue for the Asia region.

In the bottom example, **Average** is used to calculate the subtotal, and therefore calculates the average of all the average revenue values in the rows above the subtotal.

# Grand Totals

Grand totals aggregate the rows in the pivot table.

**To add grand totals:**

- Click on the menu of the row in the data panel, or right-click on the row name (header) in the widget, and then click **Grand total**.

Grand totals is effected by subtotals if defined. In the following example, the Grand total for Average Sales is 10, representing the average sales for two regions (Asia and Europe). In the example below, Average Sales has a subtotal (see above) set to Sum, and therefore the Grand total is the total of the rows above.

Region ^	Total Sales	[Average Sales]	Total Revenue
Asia	65	13	1,650
Europe	35	7	250
Grand Total	100	<b>AUTO</b> 10	1,900

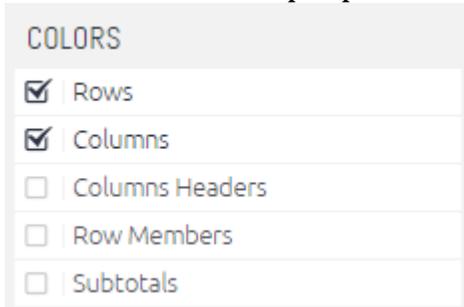
  

Region ^	Total Sales	[Average Sales]	Total Revenue
Asia	65	13	1,650
Europe	35	7	250
Grand Total	100	<b>SUM</b> 20	1,900

# Designing the Pivot Table

Fine-tune the appearance of the PIVOT table, using the following tools.

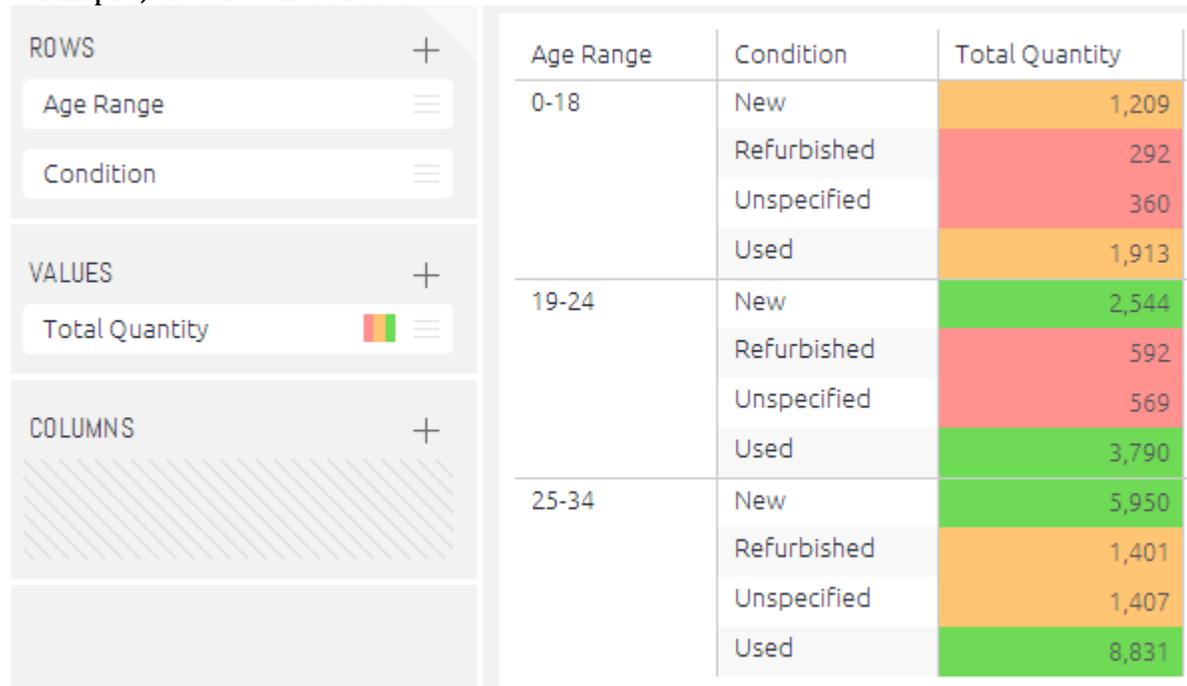
- **Page Size:** Specify how many rows appear in each page. Paging options are provided accordingly.
- **Colors:** Select the properties in the table to which you want to add color.



Select how the rows and columns of the Pivot table are highlighted.

## Exception Highlighting

Conditional formatting can be used for exception highlighting in a Pivot table. For example, as shown below:



The image shows a PivotTable interface with three sections: ROWS, VALUES, and COLUMNS. The ROWS section contains 'Age Range' and 'Condition'. The VALUES section contains 'Total Quantity'. The COLUMNS section is empty. The PivotTable data is as follows:

Age Range	Condition	Total Quantity
0-18	New	1,209
	Refurbished	292
	Unspecified	360
	Used	1,913
19-24	New	2,544
	Refurbished	592
	Unspecified	569
	Used	3,790
25-34	New	5,950
	Refurbished	1,401
	Unspecified	1,407
	Used	8,831

See [Defining Conditional Coloring – Condition](#) for more information.

**Note:** For more styling options, including font style and colors, see [this article](#).

### Limitations

Sisense supports the exporting of pivot tables of up to 1.5 million cells.

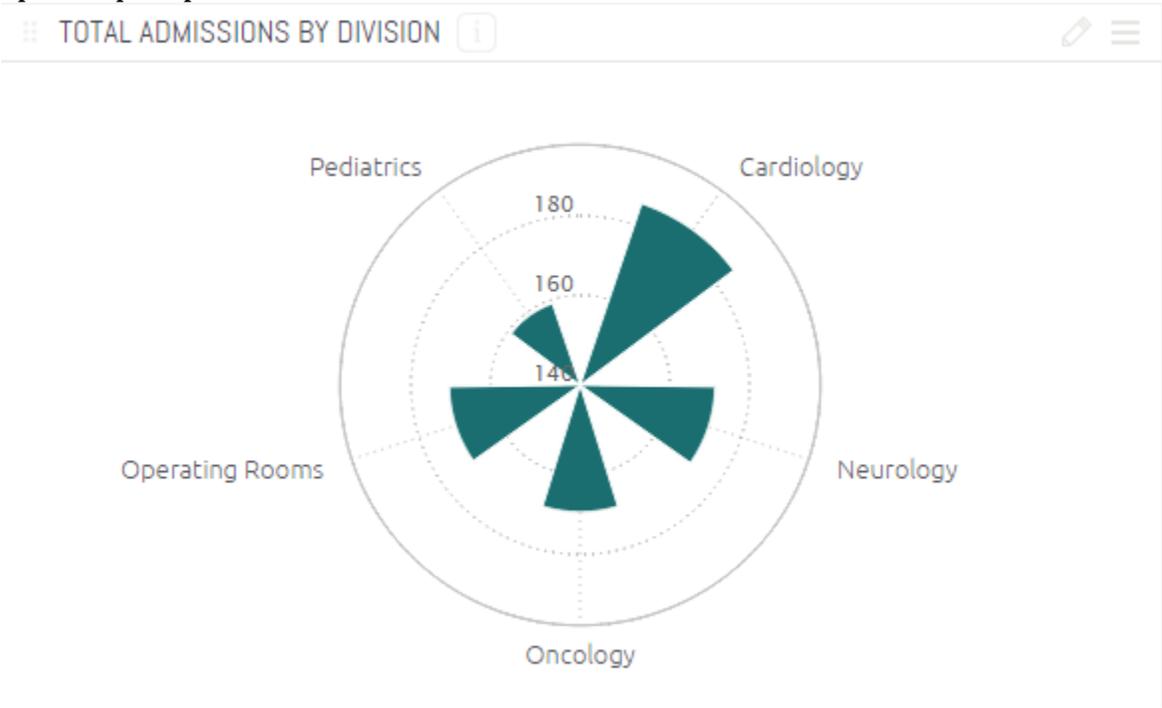
Attempting to export a higher number of cells might result in a timeout. The following properties of a pivot table increase the probability of reaching a timeout when dealing with very large tables:

- Sub totals or Grand totals
- Complex formulas in Values
- Data Security rules
- Widget-level filters

To successfully export pivot tables, you are advised to split pivot table with many columns to separate pivot tables.

# Polar Chart

Use the polar (radar) chart to compare multiple categories/variables with a spacial perspective in a radial chart.



# Adding Data

1. Click **Add +** to select the field whose values will be represented as columns in the circle. Each Category gets its own data point.  
Typically, the Categories of a Polar Charts are descriptive data.
2. In the **Values** area, click **Add+** to select the field whose values determine the distance of the column from the center of the polar circle. This is typically used to represent numeric data.  
When you select more than one value, they are stacked on top of each other.
3. Select a field by which to break (group) the data in this chart by clicking **Add +**.  
Each field added to Break by is represented by its own column and color in the chart and is automatically added to the chart's legend.  
This is an optional field. You can add at most one **Break by** field. The **Break by** option is available only when a single field was added to **Values** (Y-AXIS).

# Designing the Polar Chart

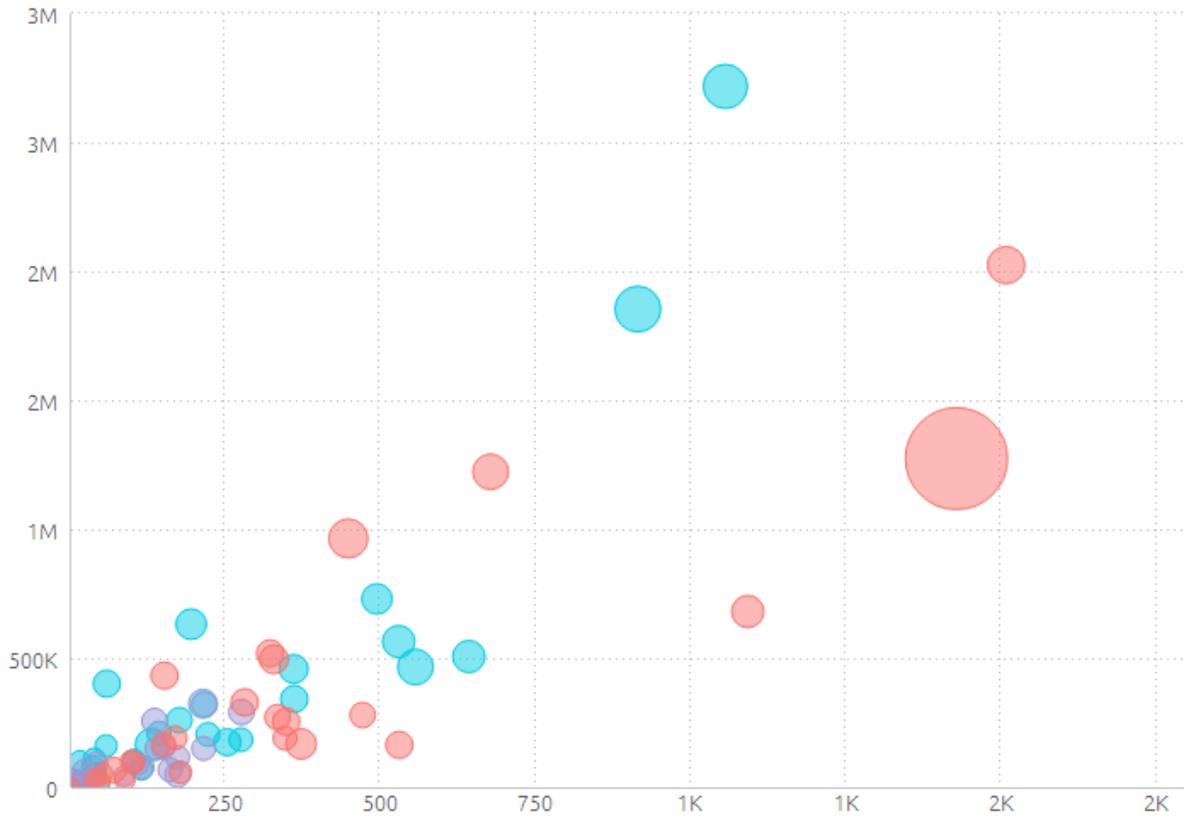
Fine-tune the appearance of the Polar Chart widget, using the following tools.

- **Polar Type:** Select the type of chart.
- **Legend:** Specify whether to show or hide the LEGEND, and select its position.
- **Value Labels:** Specify whether to show or hide labels indicating the values. Select the orientation of these labels: **Horizontal** (Default), **Diagonal** or **Vertical**.
- **Categories:**
  - **Grid Lines:** Shows (Default) / hides.
  - **Labels:** Shows (Default) / hides.
- **Axis:**
  - **Grid Lines:** Shows (Default) / hides.
  - **Logarithmic:** Displays using orders of magnitude.
  - **Labels:** Shows (Default) / hides.
  - **Title:** Select the checkbox to display the axis title. To edit the title, type in a new title. Click outside the text box to apply the new value.
  - **Values on Axis:** To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values. To restore any of the values to their default states, click on the reset button. ↺

## Scatter Chart

The Scatter Chart displays the distribution of two variables on an X-Axis, Y-Axis, and two additional dimensions of data that are shown as colored circles scattered across the chart:

- **Point:** A field that for each of its members a scatter point is drawn.
- **Size:** An optional field represented by the size of the circles. If omitted, all scatter points are equal in size. If used, the circle size is relative to their value.



# Adding Data

1. In the **X-Axis** panel, click **Add +** to select the field whose values will be placed on the X-Axis.  
Typically, the X-Axis of a Scatter Chart is used to represent numeric data. Alternatively, Descriptive data is also supported. You can only add one field.
2. In the **Y-Axis** panel, click **Add +** to select the field whose values will be placed on the Y-Axis.  
Typically, the Y-Axis of a Scatter Chart is used to represent numeric data. Alternatively, descriptive data is also supported. You can only add one field.
3. In the **Point** panel, click **Add +** to select the point field.  
The point area determines the field to be scattered across the chart as circles; one point (circle) appears on the chart for each member of this field. These fields must be descriptive data (not numeric data).  
You can only add a point if either the X-Axis or the Y-Axis was defined to represent numeric data.
4. **Break by/ Color:** Select a field by which to break (group) the fields in the chart. This must be a descriptive field. If you select descriptive data, then the points in the chart are grouped into the members of the field. Each member is represented by a different color.  
OR  
Select a field by which to color the fields in the chart. This must be a numeric field. If you select numeric data, then the point color is a gradient where the highest value is the darkest color and the lowest value is the lightest color.
5. In the **Size** panel, select the field that determines the size of the circle. It must be numeric data.

# Designing the Scatter Chart

Fine tune the appearance of the SCATTER CHART widget, using the following tools.

- **Legend:** Specify whether to show or hide the Legend, and select its position.
- **Marker Size:** Change the size of the circles. If you are not using a Point value, the slider will change the size of all the points on the map. If you are using the Point value, drag each side of the slider to determine the relative minimum and maximum circle sizes.
- **Value Labels:** Specify whether to show or hide labels indicating the values of each point (circle). Select the orientation of these labels: **Horizontal** (Default), **Diagonal** or **Vertical**.
- **X-Axis and Y-Axis:**
  - **Grid Lines:** Shows (Default) / hides.
  - **Logarithmic:** Displays using orders of magnitude.
  - **Labels:** Shows (Default) / hides.
  - **Title:** Select the checkbox to display the axis title. To edit the title, type in a new title. Click outside the text box to apply the new value. To change the default minimum, maximum or interval values on the axis, type in the new values, and click outside the text boxes to apply the new values.
  - **Values on Axis:** To restore any of the values to their default states, click on the reset button. ↻

## Scatter Map

Scatter Maps allow you to visualize geographical data as data points on a map.

The map distinguishes data using different colors and sizes for the data points on the map.

Scatter Map

TOTAL VISITOR BY LOCATION

Apply Cancel

Filters Design

Location

City

Color

Total # Visits

Site

Total # Converted



Markers Style

Full

Marker Size

Update on Every Change

# Adding Data

In the **Location** panel, you can add any field(s) that contain geographic data, and the map will find the most appropriate point. Supported Location types:

- Country
- City
- State/Province
- Latitude/Longitude (for example: 37.5601, -122.50131, but not 37.5601° N, 122.50131° W).

For example, if you have a country field, using it by itself will visualize the countries on the map.

If instead you have a city field, using it will visualize the cities on the map automatically.

# Improving Geographic Accuracy

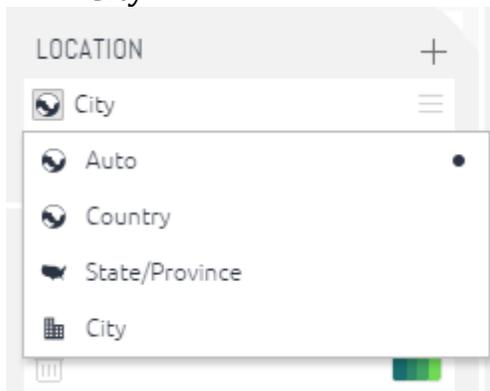
## Use multiple geographic fields

To improve accuracy, it is recommended to use more than one geographic level. For example, if you have both a country and a city field, add both of them to the **Location** panel.

## Define the location type

Clicking on the geographic field's menu will present a location type sub-menu, which contains the following options:

- Auto
- Country
- State/Province
- City



By default, **Auto** is always selected. This option always tries to figure out which fits best. For better accuracy, you can specify the appropriate location type.

## Using Latitude and Longitude data

To visualize latitude and longitude data, you have to add one field containing latitude data, and another field containing longitude data, in this order.

### **Color**

Select a numeric field here to color the scatter points on the map according to their value and color rule.

The value of the field is displayed in the tooltip of the scatter point on the map. Hover over the point to see it.

### **Size**

Select a numeric field here to make the size of the scatter points on the map relative to their value in the field's value range.

### **Details**

You can select additional data (numeric or descriptive) that will appear when hovering over the data point in the map.

LOCATION +

≡

COLOR

≡

SIZE

≡

DETAILS

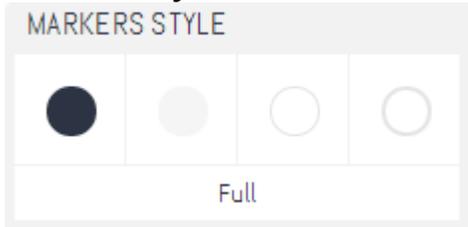
≡



# Designing the Scatter Map

Fine-tune the appearance of the Scatter Map Widget, using the following tools.

- **Marker Style:** Choose between various marker styles.



- **Marker Size:** Select the relative size for the markers.



When the Size has a field in it, the slider has two levers controlling the minimum and maximum size of the points.



## Sunburst Widget

The Sunburst widget is similar to a pie chart but is multi-dimensional. Whereas a pie chart combines one field and one numeric value, the Sunburst widget can display multiple rings, one for each field. Each ring in the Sunburst shows a breakdown of its parent ring slice.

Sunburst 

Age Range vs. Total Cost 

Categories 

A. Age Range

- 5-18
- 19-24
- 25-34
- 35-44
- 45-54
- 55-64
- 65+

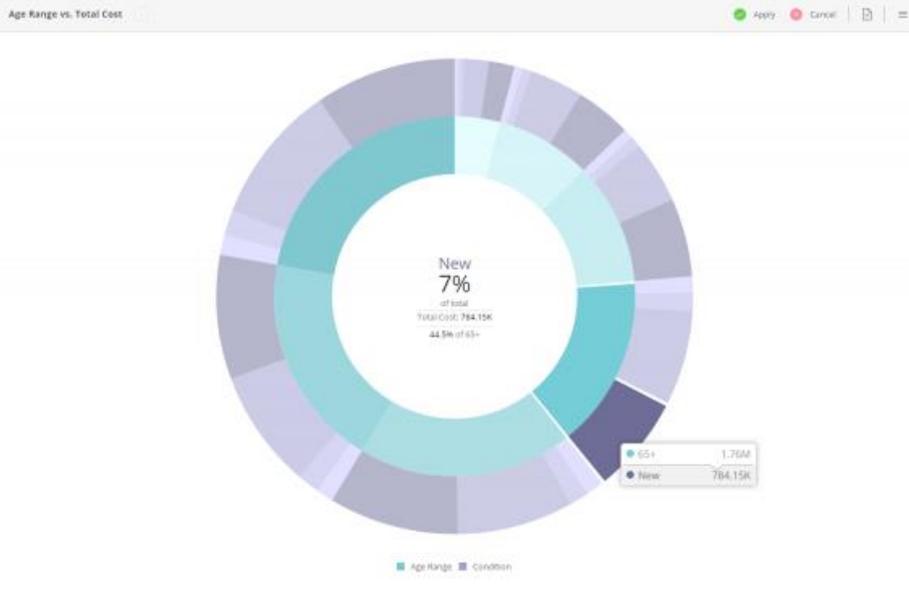
Condition

- New
- Refurbished
- Unspecified
- Used

Values

F. Total Cost

Update on Every Change 



Filters  Design 

Center

- Value
- Contribution
- Contribution to parent

Tooltip

- Value
- Contribution

Legend 

- Top
- Left
- Right
- Bottom

# Adding Data

1. In the **Categories** panel, select the fields whose values will be represented as slices of the ring.

Typically, the **Categories** of a Sunburst widget contain descriptive data. Multiple fields can be added, so that each field is represented as a ring whose values are broken up by its parent ring slice.

You can reorder the fields using drag and drop.

2. In the **Values** panel, click **Add +** to select the field whose values determine the size of the ring. Values are typically used to represent numeric data. You can only add one field.

# Designing the Sunburst Widget

Fine-tune the appearance of the Sunburst widget using the following tools.

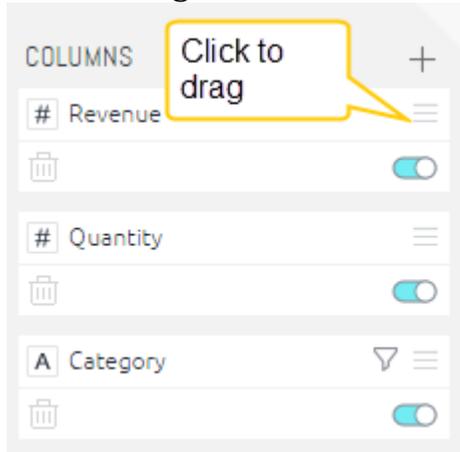
- **Center:** Select which info is presented in the center of the Sunburst widget when hovering over a ring slice:
  - Value – Show the actual numeric value associated with the selected slice.
  - Contribution – Show the contribution (percentage) of the selected slice to the entire field.
  - Contribution to parent – Show the contribution (percentage) of the selected slice to its parent ring slice.
- **Tooltip:** Determine whether to show the actual value or contribution in the slice tooltip when hovering over it.
- **Legend:** Specify whether to show or hide the Legend, and select its position in the widget.

## Table

The Table widget displays a broader view of your data, presenting raw and non-aggregated data in columns, with as much fields and metrics as needed.

# Adding Data

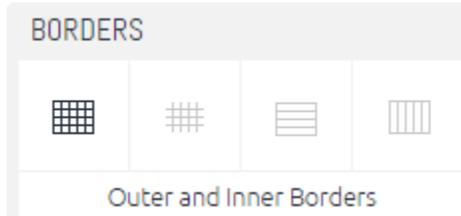
- Click **Add +** to select the field(s) that you want to add to the table.  
To rearrange the order of the columns, click and drag a column up or down.



# Designing the Table Panel

Fine-tune the appearance of the Table widget, using the following tools:

- **Borders:** Select the borders to display in the table.



- **Column Width:** You can resize the column widths as follows:
  - **Fit to Window Size:** Columns are resized automatically to fit the contents in the window.
  - **Fit to Content:** You can manually resize the column widths by dragging the column separator.



- **Colors:** Select whether to color (highlight) alternating columns and rows, and the table header.
- **Word Wrap:** Select whether to wrap texts in headers and rows.

Ca...	Age Range
Monit...	25-34
Monit...	35-44
PDA's	35-44
Proce...	25-34
Table...	25-34
Table...	35-44

Category	Age Range
Monitors	25-34
Monitors	35-44
PDA's	35-44
Processor Upgrades	25-34
Tablet PCs	25-34

- **Infinite Scroll:** Toggle the Infinite Scroll option On or Off:
  - On: Include all items in a single page with scrolling. Additional data loads as you scroll down in the page.
  - Off: Specify how many rows you want in a single page. Above the specified number, you will have paging options.
- Infinite Scroll in the Off position.

## Adding Text Widgets

You can add Text widgets to create titles and texts that stand out, or when you want to add more descriptive explanations to your dashboard and surrounding visualizations.

In the Text widget's settings, you can find plenty of options for creating different text styles. Some examples of what you can do include changing the font color, selecting a background color, adding hyperlinks, and defining the text alignment.



### To add a Text widget:

1. Click on the Text Widget icon in the top menu.



A new Text widget appears in your dashboard.

2. Type in your text. As you begin to type, a tool bar with text and formatting options is displayed.

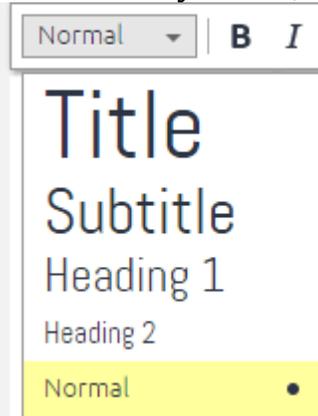


3. Click outside of the widget area to save the text and formatting.

## Examples

### To select a font style:

1. Click anywhere in the relevant paragraph.
2. From the Styles list, select one of the five predefined styles.

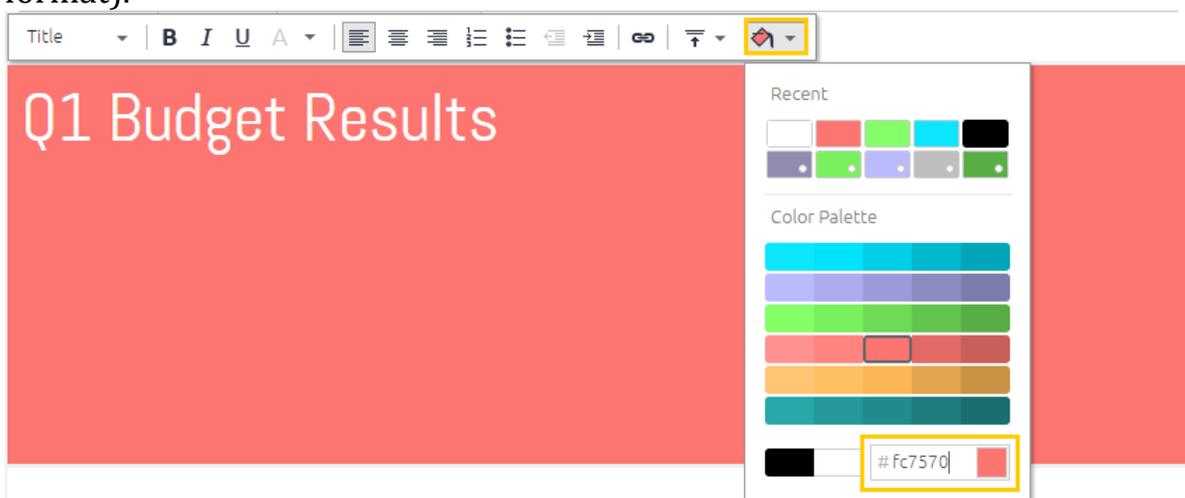


### To add a hyperlink:

1. Select the word or text fragment to which you want to add a link.
2. Click the hyperlink icon and enter the URL.
3. You can also select which text to display instead of the selected (linked) text, and whether to open the link in the same window, or in a new window.

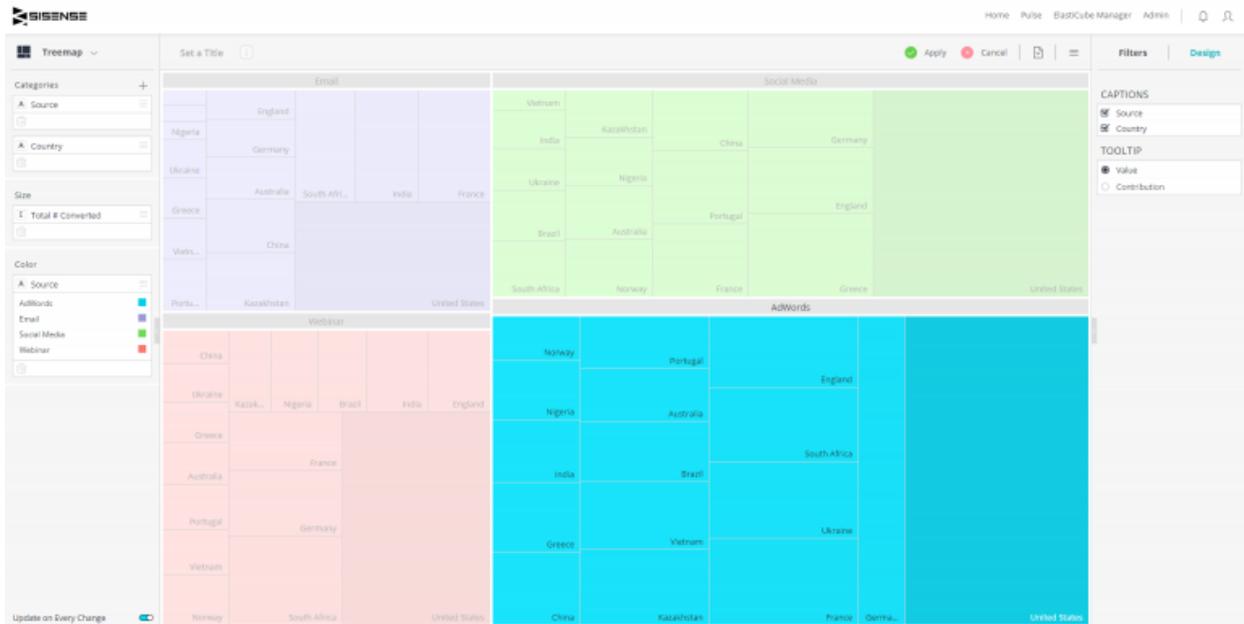
### To change the background color:

1. Click on the paint bucket icon to open the color palette.
2. Select a color from the palette, or enter a custom color (Hexadecimal format).



# Treemap

The Treemap is a multi-dimensional widget that displays hierarchical data in the form of nested rectangles. This type of chart can be used in different scenarios, for example, instead of a column chart if you have to compare too many categories and sub-categories.

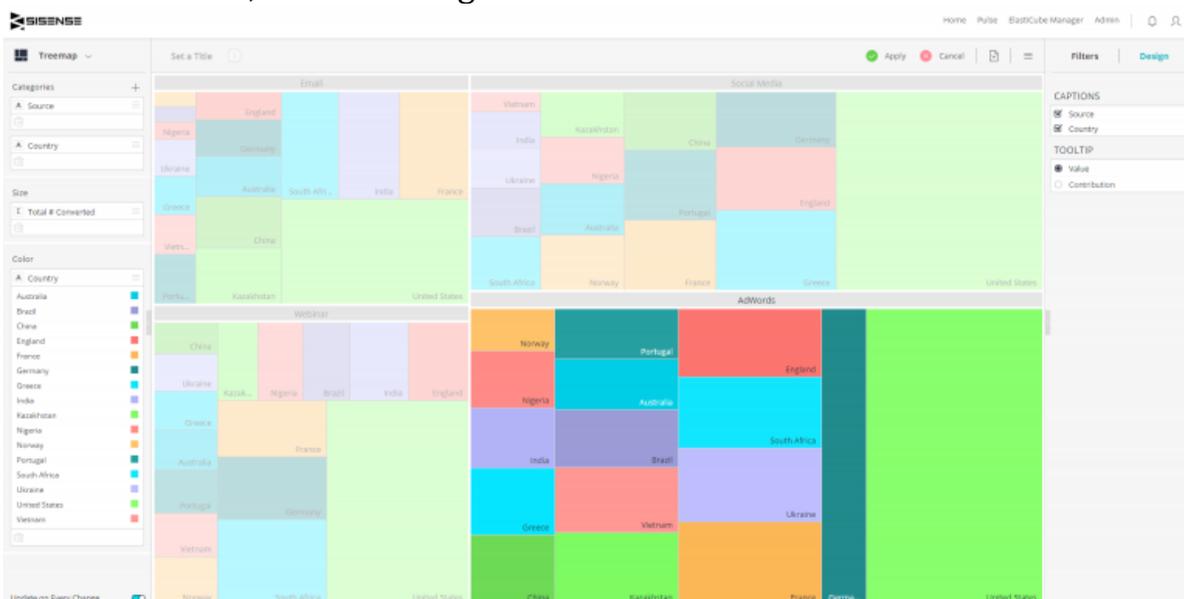


# Adding Data

1. In the **Categories** panel, click **Add+** to select the fields whose values will be represented as rectangles in the chart.  
Typically, the Categories of a Treemap widget contain descriptive data. You can add up to three fields, so that a rectangle hierarchy is created. You can reorder the fields using drag and drop.
2. In the **Size** panel, select the field whose values determine the size of the rectangles by clicking **Add +**. Values are typically used to represent numeric data.  
You can add only one field.
3. In the **Color** panel, click **Add +** to select the field to use as the color basis for your rectangles.

The following coloring scenarios are supported:

- **Color by a numeric value:** If you choose a numeric value, then each rectangle will be colored according to its value in the field's range of values. You can manipulate the coloring rules in the color formatting menu.
- **Color by field members:** If you choose a field, then all rectangles that are members of that field will be colored the same. For example, if you choose to color by the field that is also used as the first category, then the entire Treemap will be split into groups of colors, like in the image at the top of this page. If you choose to color by the field that is used as the second category, then all rectangles that share the same value will also share the same color, as in the image below:



**Note:** You can only color by a field that is also used in the Categories panel. If you choose a field that is not there, it will be automatically added.

# Designing the Treemap

Fine-tune the appearance of the Treemap widget, using the following tools.

- **Captions:** Control which info is presented in the treemap. A checkbox will be visible for each field that is used in the **Categories** panel. Turning a field on/off will determine if the field member name will be visible inside the treemap. Note that this info will always be available in the tooltip when hovering over a rectangle, regardless of the on/off setting.
- **Tooltip:** Select whether to show the actual value or contribution in the rectangle tooltip when hovering over it.

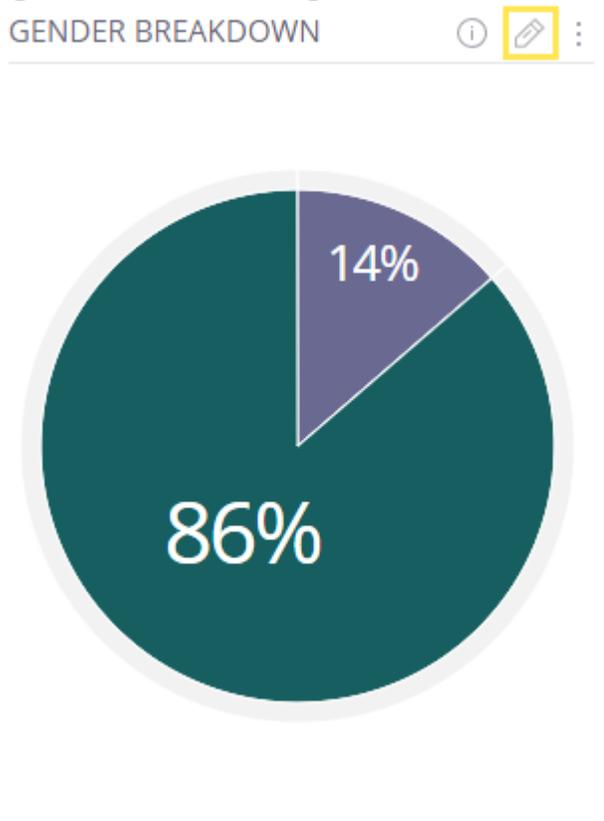
## Widget Designer

The Widget Designer lets you fine-tune a widget's appearance and behavior.

# Opening the Widget Designer

## To display the Widget Designer:

- On the dashboard, click the **Pencil (Edit)**  button that appears in the top-right corner of a Widget.



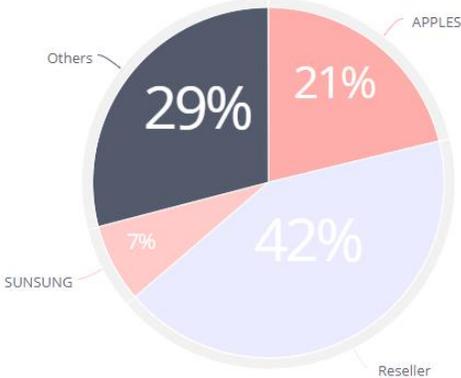
- Alternatively, while creating a new widget, in the Widget Wizard, click the **Advanced Configuration** option that appears at the bottom-left of each window. The Widget Designer is then displayed showing the widget in the

same state as in the Widget Wizard preview.

**New Widget** from Commerce Demo

Brand : Total Revenue : + Add More Data...

#  Add Title



Category	Percentage
Reseller	42%
Others	29%
APPLES	21%
SUNSUNG	7%

[Advanced Configuration](#) Create Cancel

# Navigating the Widget Designer

The Widget Designer enables you to select the data to be included in a widget in addition to providing a variety of options for customizing the visualization used to show the data.



1. **Selecting the Widget Visualization:** Enables you to change the visualization of the widget.
2. **Adding Data to the Widget (Data Panel):** Used for selecting the values that you want to appear in the widget and those needed for grouping the data. The options differ depending on the visualization (chart type).
3. **Previewing the Widget:** The center of the window displays the current design and content of the widget, which automatically changes each time you select a different option in this window.
4. **Adding a title to the dashboard:** Click Set a Title and type in a new title for the widget. You can also add a widget title directly from the dashboard without entering the Widget Designer. [Click here](#) to learn more.
5. **Accessing More Options:** Displays a menu of additional options for the widget.
6. **Filtering the Widget (Filter Panel):** Lets you manage the filters that affect this widget. Learn more in the [filtering page](#).
7. **Designing the Widget (Design Panel):** Provides a variety of options for fine-tuning the appearance of the Widget, including labels, legends, line types and more. Some of these options need to be turned on for you to

configure them. The options differ depending on the visualization (chart type).

8. **Update on every change:** Selected – Data changes are automatically and dynamically updated in the display. Not selected – you have to click **UPDATE** to display your latest changes in the widget. Clearing this checkbox may be useful when you are working with very large datasets where query times might be slower.

# Fine-tuning a Widget

- To learn more about adding data and fine-tuning the design of a specific widget, click on a widget from the list below.
  - [Indicator](#)
  - [Column Chart](#)
  - [Line Chart](#)
  - [Area Chart](#)
  - [Area Map](#)
  - [Bar Chart](#)
  - [Pie Chart](#)
  - [Polar Chart](#)
  - [Scatter Chart](#)
  - [Scatter Map](#)
  - [Pivot](#)
  - [Sunburst](#)
  - [Treemap](#)
  - [Calendar Heatmap](#)
  - [Table](#)
  - [Box & Whisker Plot](#)

See also [Additional Widget Design Options](#) and [Extending Dashboard Functionality with JavaScript](#).

## Selecting Colors in Widgets

Different field values are represented by different colors. The Widget Designer automatically assigns these colors and displays them in the widget legend that is displayed in the Widget Designer and in the dashboard.

The Widget Designer provides the following options for manipulating these colors:

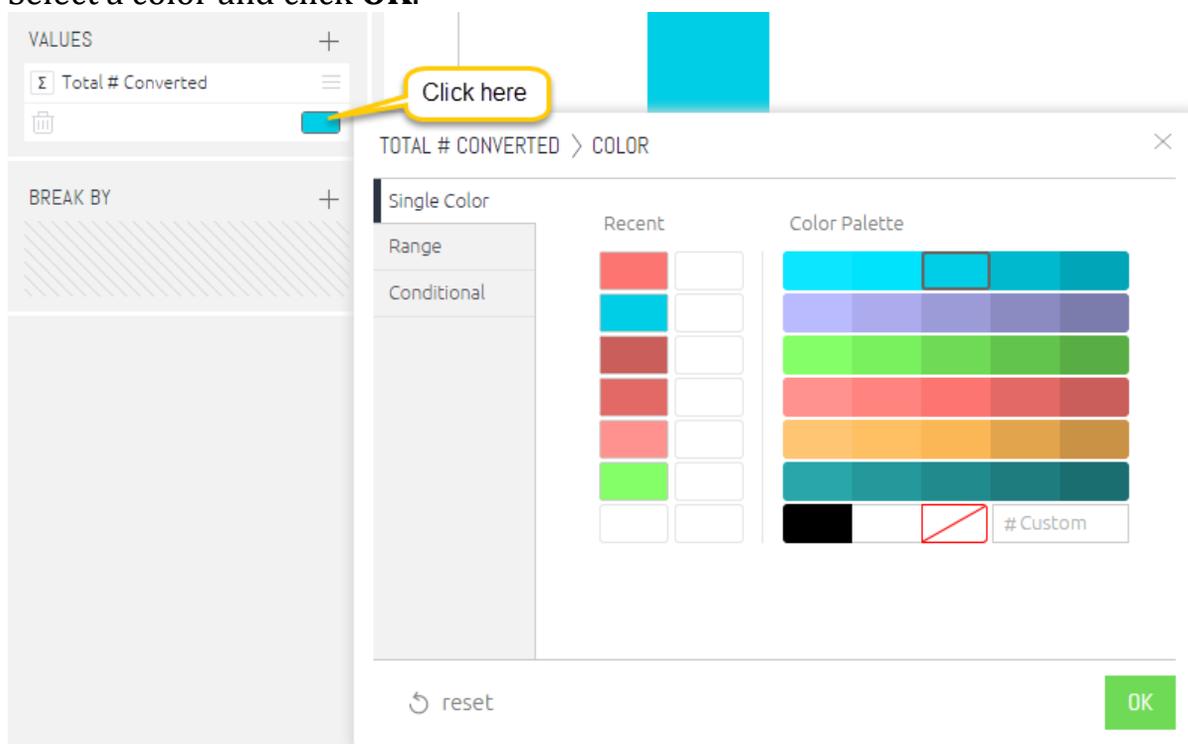
- **Single Color:** For selecting a specific color for all values
- **Range:** For selecting a range of colors
- **Conditional:** For defining colors based on conditional data.

# Selecting a Specific Color – Single Color Tab

The following procedure describes how to assign a specific color to a data value that does not change. This color only applies to the widget in which it is defined.

## To select a specific color:

1. Click on a color in the Widget Designer Data Panel to display a color picker window.
2. Select the **Single Color** tab.
3. Select a color and click **OK**.



**Note:** To select a custom color that does not exist in the palette, enter a hexadecimal value into the # Custom field.

# Selecting a Range of Colors – Range Tab

The **Range** tab is available for the following widgets:

- Column Chart and Bar Chart
- Scatter Map and Area Map

The following procedure describes how to assign a range of colors to a field according to its value.

## To select a color range:

1. Click on a color in the Widget Designer Data panel to display a color picker window.
2. Select the **Range** tab.
3. By default, the range configuration is set to 'Auto', which automatically assigns a color range.

Select **Manual** to manually configure the color range, as follows:

- **Change Left/Right Colors:** Click the color rectangles at the right and left to change the start and end color of the range.
- **Customize Range:** By default, the color range is set to match the minimum and maximum values of the data. Instead, you can manually set a minimum value. All values that are below this value will be assigned to the color on the left. Accordingly, selecting a Max value means that all values larger than it will be assigned the color on the right. By default, the 'Middle' color is grey. All values larger than this will start gradually getting the right color. All values smaller than this will start gradually getting the left color. Changing the 'Middle' value controls which value the color gray is assigned to.
- **Coloring Method:** Selecting **Min/Max** will display the full color range. Selecting **Min** will display all the colors between the minimum and middle values, and higher than the middle value will be gray. Selecting **Max** will display all values between the middle value and the maximum value, while all other values will be displayed in gray.
- **Transition:** Select Steps to define a distinct number of colors in the range. Select **Gradient** for a linear color range.

- Single Color
- Range**
- Conditional

Configuration:  Auto  Manual



Coloring method:  Min  Max  Min / Max

Customize range:

Transition:  Steps   Gradient

↻ reset

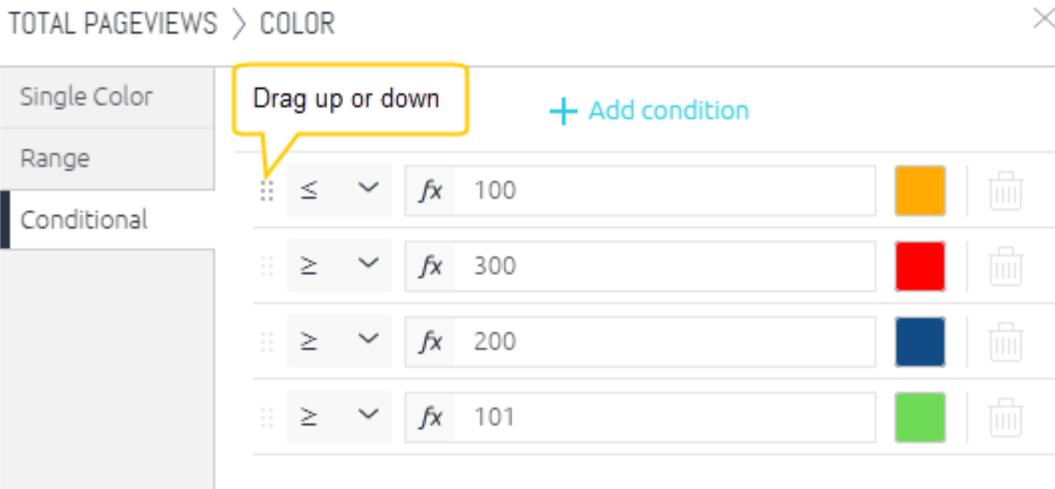
OK

# Defining Conditional Coloring – Conditional Tab

Define colors based on the value of the field as defined by a logical expression. In addition, for Indicator widgets, you can also define whether an icon appears in the widget and select the icon's color. For example, you can define a red arrow pointing downwards for a negative value.

## To set conditional coloring:

1. Click on a color in the Widget Designer Data panel to display a color picker window.
2. Select the **Conditional** tab. Each row in this window represents a condition to which you can apply a value. If the condition is true, then that value is shown in the color that you select in this row. For example, one row can state that a negative value is displayed in red and another row can state that a positive value is green.
3. In each row, define an expression to be applied to the value of this field by selecting an operator and specifying a value, as shown below.
4. Select a color for the field.
5. If required, add and define additional condition rows by clicking **+ Add condition**.
6. In the dashboard, these conditions are evaluated in the order in which they appear in this window – from top to bottom. The first condition that is true, determines the color. Reorder the condition rows as necessary by dragging them up or down.
7. Click **OK**.

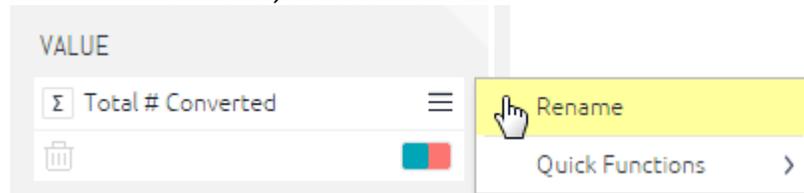


## Renaming Fields in Widgets

The names of fields in widgets are taken from the raw data. To make these names more comprehensible, you can rename the fields in your widget.

### To rename a field in a widget:

1. In the Widget Designer, right-click on a field, for example, in the X-AXIS, Y-AXIS, VALUES, CATEGORIES areas and so on.
2. In the menu, select **Rename**.



**Note:** Renaming a field only affects how it is displayed in the current widget. This field's name is not affected in the Data Browser or in other widgets in the dashboard.

## Formatting Numbers in Widgets

You can change the way numbers are displayed and formatted in widgets. For example, you can display a number as a percentage, and you can define the number of decimal places.

### To format numbers in a widget:

1. In the Widget Designer, for example, in the X-Axis, Y-Axis, Values, Categories areas and so on, hover over, and click **123**.

The screenshot shows the 'FORMAT NUMBER' dialog box in a widget designer. The dialog is open over a widget element labeled '123'. The dialog has a left sidebar with 'Number', 'Currency', and 'Percent' options. The 'Number' option is selected. The main area shows 'Abbreviations' with checkboxes for 1T, 1B, 1M, and 1K, all checked. 'Decimal Places' has radio buttons for 'Automatic' (selected), '.0', and '.00'. 'Thousands Separator' has radio buttons for '1000' and '1,000' (selected). At the bottom, there is a 'reset' button, a preview area showing '1.23B', and an 'OK' button.

2. Select how the number, currency or percentage appears. The preview area at the bottom of this window shows a preview of how each option will appear in the widget.

## Locale Settings

The formats for dates, times and numbers in your dashboards are based on your computer's operating system or browser's locale settings (depending on which browser you use).

The following locales are supported:

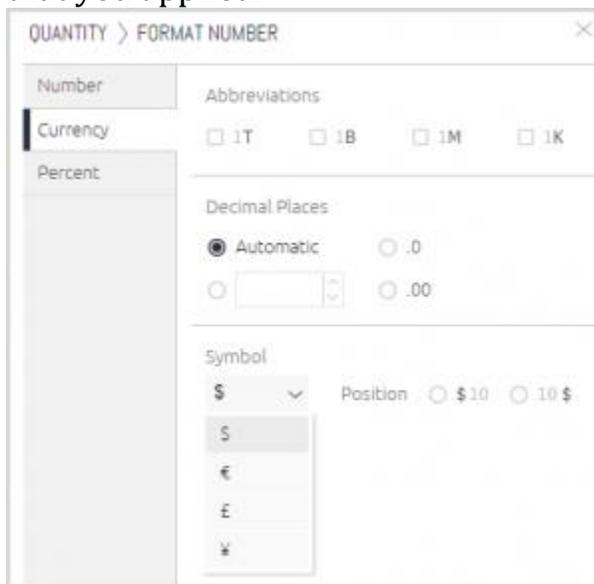
- Unites States
- United Kingdom
- Israel
- Canada
- South Africa
- Australia
- Netherlands
- Germany
- Ireland
- Mexico
- France
- China
- Brazil

**Notes** You can override the default locale settings for all users, by updating a parameter via the Rest API. To use a locale file that is not included in the above list of countries, you can manually add a locale file in the following location: C:\Program Files\Sisense\PrismWeb\client\resources\base\localization. Click [here](#) to access Locale files and view their codes.

## Currency Settings

The default currency symbol used in the dashboard is the US dollar (\$), however you can change the symbol by either selecting another symbol from the drop-down list, or by typing in a different symbol. The list of currency symbols will include common currency symbols, the currency defined in your browser's locale settings, and the symbol that you added (if you did so).

After you select a different symbol, additional viewers will see the same symbol that you applied.



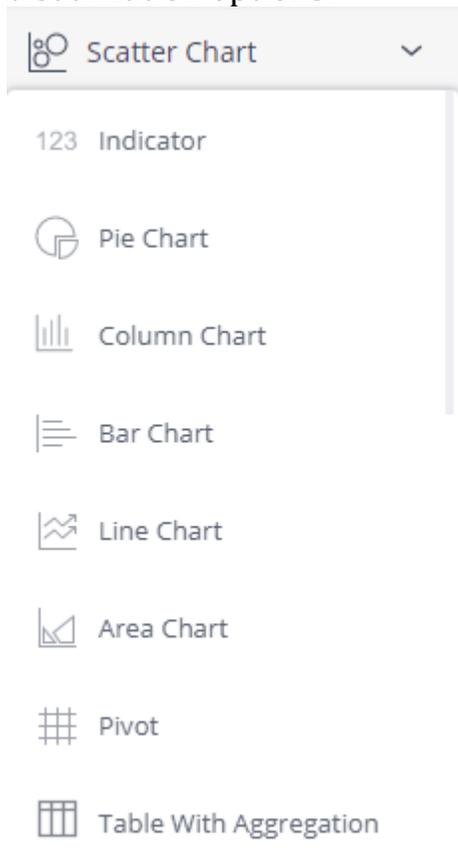
## Changing a Widget's Visualization

You can change the visualization of a widget (for example, from a Line Chart to a Pie Chart). When you change the type, all relevant values and definitions are transferred to the new visualization.

### **To change a widget's visualization:**

1. Open the Widget Designer, as described in [Opening the Widget Designer](#).

2. Click the Visualization selector, shown below, to display a menu of visualization options.

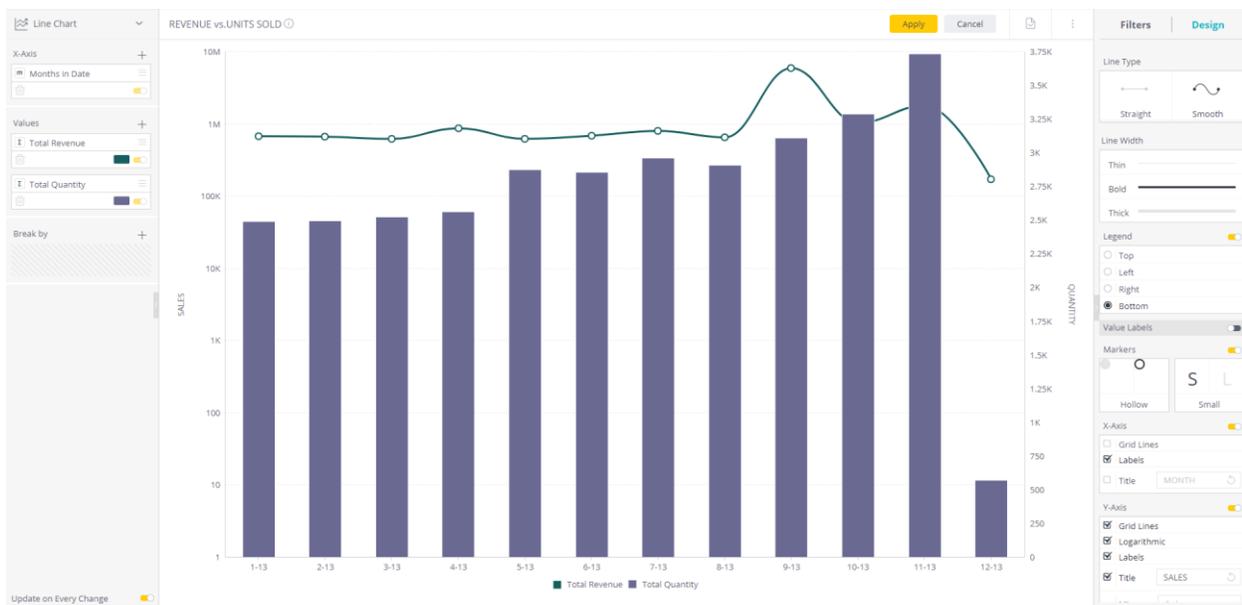


3. Select a different visualization, and click **Apply**.

### Combining Two Types of Visualizations in a Widget

In chart widgets (such as Line Charts, Area Charts, Column Charts and Bar Charts), you can represent one or more selected field(s) using a different visualization than the visualization of the original widget.

For example, the following example shows a Line Chart in which the **Total Revenue** is still represented as a Line Chart (the default), but the **Total Quantity** is represented as a Column Chart.



### To select a different visualization for a field in the widget:

1. In the Widget Designer, in the **Values** area, right-click the field that you want to change.
2. Select **Series Type** from the menu. A menu of alternative visualization types is displayed.
3. Select the visualization to use for this field only.

**Tip:** If necessary, drag the field up or down in the Widget Designer to have a visualization brought to back or front.

**Note:** The chart keeps the same visualization type as was originally created. The Design panel still displays only the design options of the widget's original visualization type.

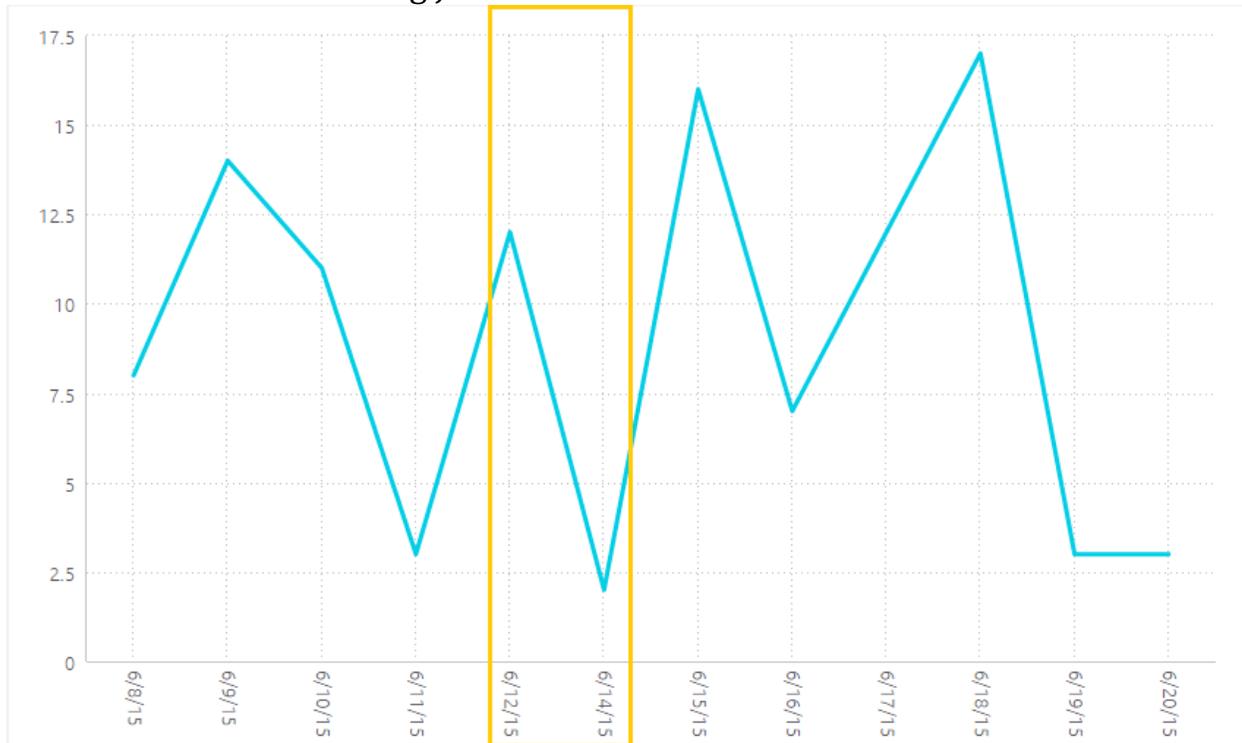
### Creating a Continuous Chart with Missing Date Values

If your data is missing date values, you can add the missing dates to your chart to create a continuous data flow. In addition, you can select whether to display missing dates as gaps in your chart, or include null values as zeros, to avoid gaps, and maintain a continuous data flow.

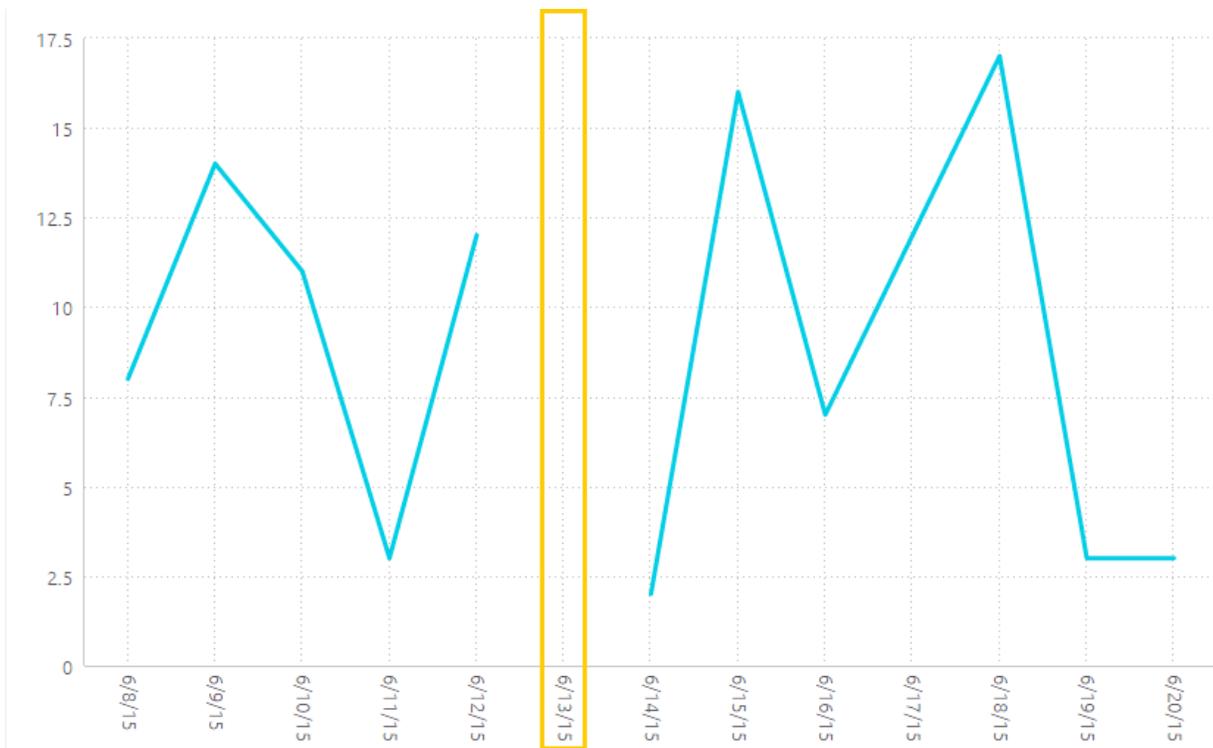
For example, if your chart begins June 8th and ends June 20th, but there is no value on June 13th, you can choose to include the missing date on the x-axis. You

can also choose whether to continue the graph (create zero values) or cut the graph where there are missing values.

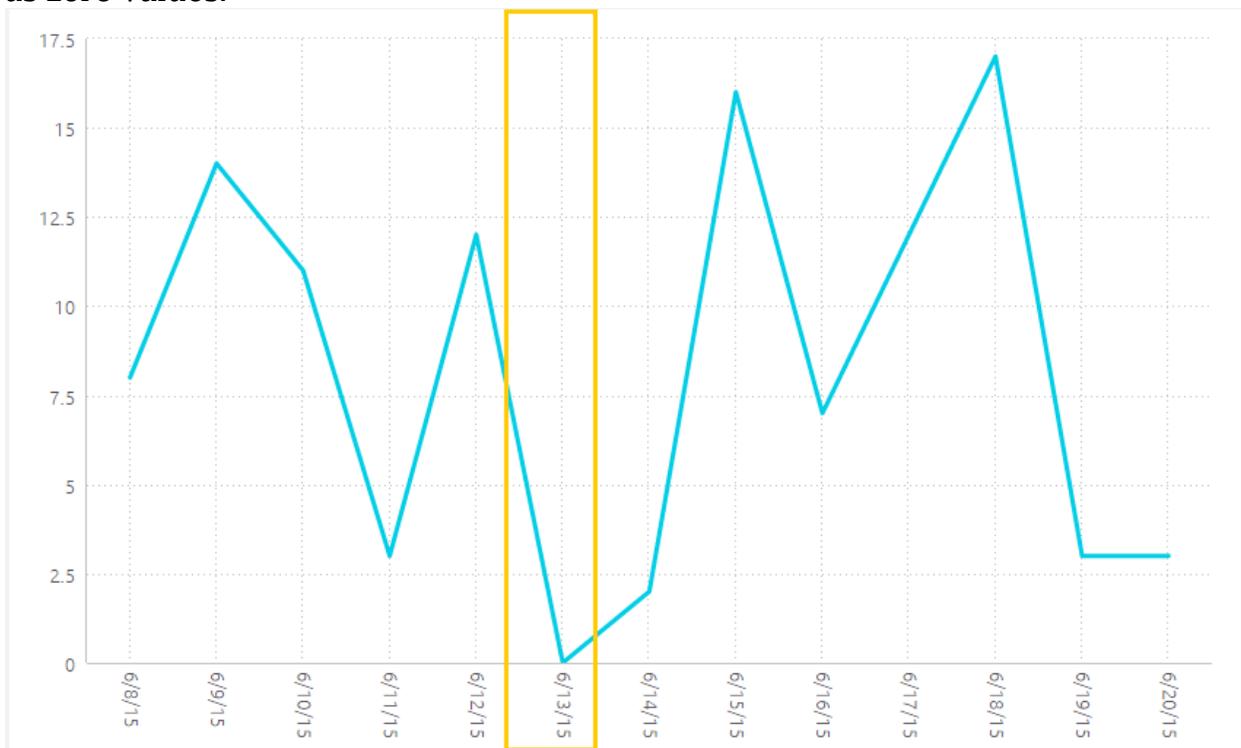
The default chart is missing June 13th.



After selecting the Continuous option, the chart includes the missing date(s).



In addition, you can select to continue the line, by treating null (missing) values as zero values.

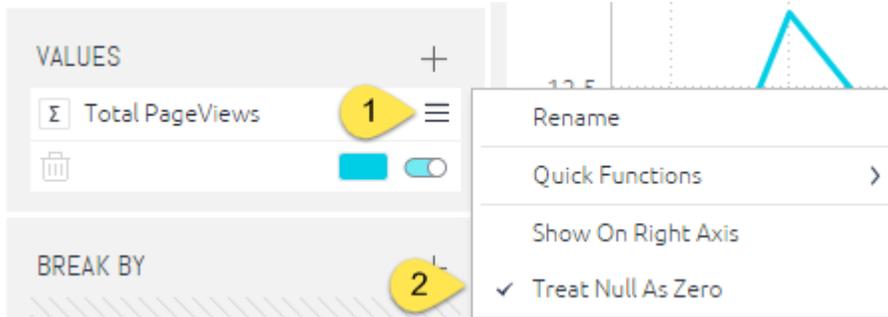


**To create a continuous chart:**

1. In the X-Axis panel, click on the menu, and select **Continuous**.



2. To treat null values as zero values, click on the menu of the Values panel, and select **Treat Null as Zero**.



## Adding Drill Hierarchies to Widgets

Dashboard Viewers have the option to [drill down in a widget](#), and get an in-depth view of a selected value.

While Viewers can select a drill-down path from the complete list of fields, it's easier to select a commonly needed drill hierarchy from a short list. This is especially true when there is a lot of data, and the Viewer needs to remember specific fields, and select them each time.

As a Designer, you can add predefined drill hierarchies to widgets.

**Note:** Your Sisense Administrator must first create the drill hierarchies before you can select and add them in the Widget Designer.

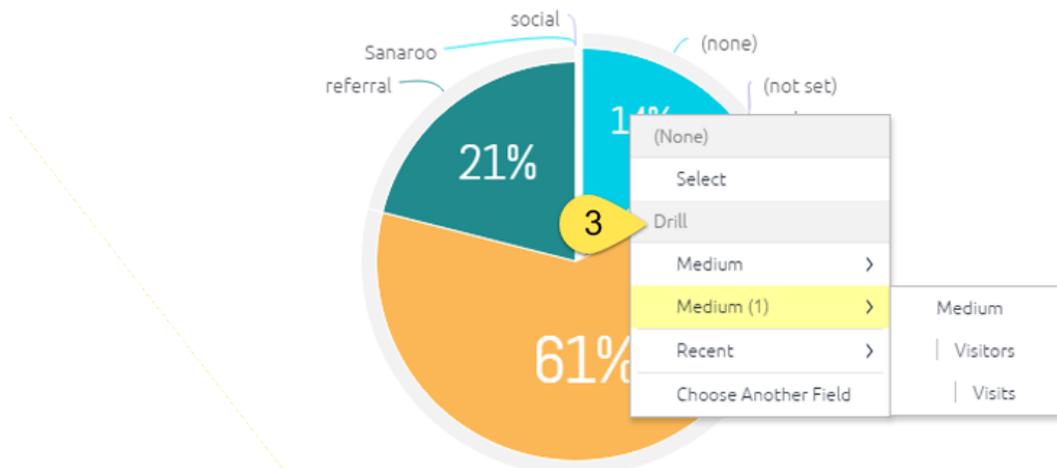
### **To enable drill hierarchies in a widget:**

1. In the Widget Designer, click on the menu of the value (dimension), and select Hierarchies...
2. Select the checkboxes next to the drill hierarchies that you want to make available to dashboard viewers.

**Note:** Hierarchies with a lock icon will always appear in the widget. To

unlock this option, please contact your Sisense Administrator.

The screenshot shows two parts of the Sisense interface. On the left, a 'CATEGORIES' list includes 'Medium' (highlighted with a yellow callout '1'), '(none)', '(not set)', 'banner', 'cpc', 'organic', 'referral', 'Sanaroo', and 'social'. A context menu is open over 'Medium', with 'Hierarchies...' selected (highlighted with a yellow callout '2'). On the right, the 'Hierarchies: Medium' dialog box is shown, with the instruction 'Select the dimension hierarchies that you want to include in this widget'. It lists two options: 'Medium' (unchecked) and 'Medium (1)' (checked). The 'Medium (1)' option is expanded to show 'Medium > Visitors > Visits'. 'OK' and 'CANCEL' buttons are at the bottom right.



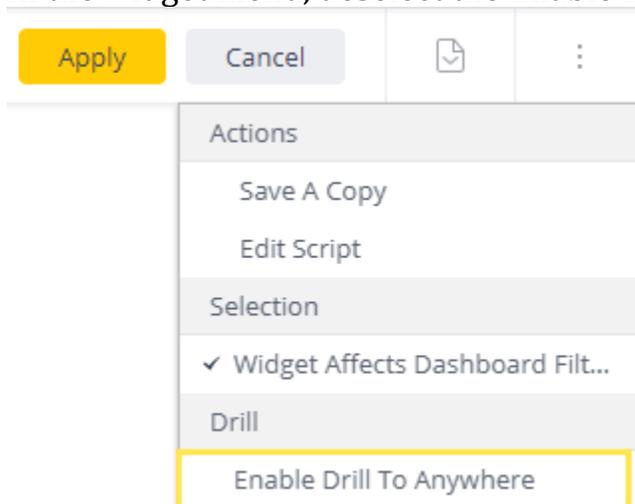
# Disabling the Drill to Anywhere Option

As a Designer, you can restrict viewers to predefined hierarchies only, by disabling the option to drill down freely to any field.

The drill-to-anywhere option is enabled by default for all users and widgets. You can disable the drill-to-anywhere option in the Sisense web app per widget (see below), or by user role in the REST API.

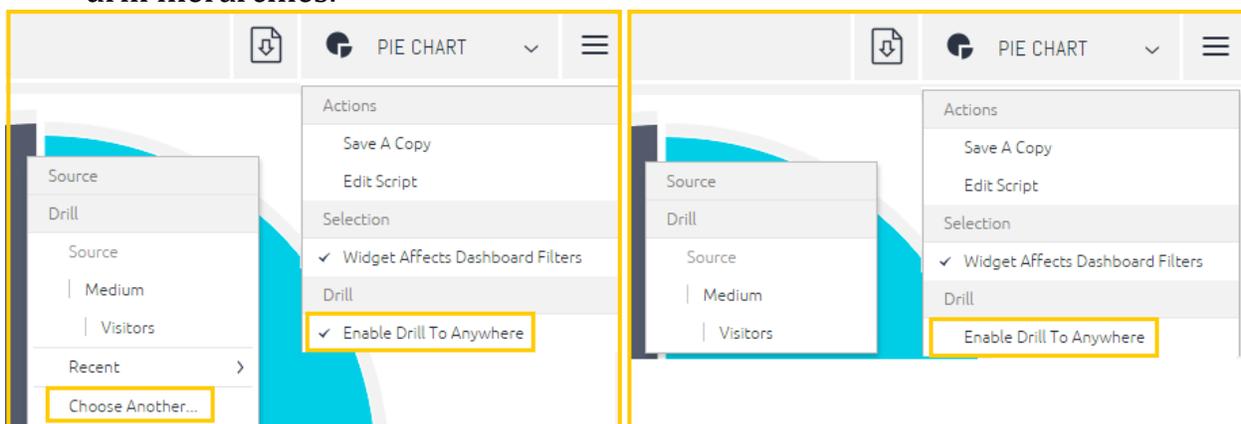
## To disable the drill-to-anywhere option:

- In the widget menu, deselect the Enable Drill to Anywhere option.



As a result, the Choose Another... option is removed from the widget's Drill options.

The left image below displays the widget with the option to drill to anywhere. The image on the right displays the widget with only predefined drill hierarchies.



## Managing Widgets

This topic describes how you can edit, copy, and delete your dashboard's widgets.

# Editing a Widget

When you first create a new dashboard (as described in [Creating a New Dashboard](#) ) or a new widget (as described in [Adding Widgets to a Dashboard](#)), the Widget Wizard automatically guides you through Sisense Web's best recommendations for visualizing the data that you select.

Then, after a widget is created, you can fine-tune/edit it in the [Widget Designer](#). The Widget Designer provides a variety of options for changing the widget's appearance and behavior.

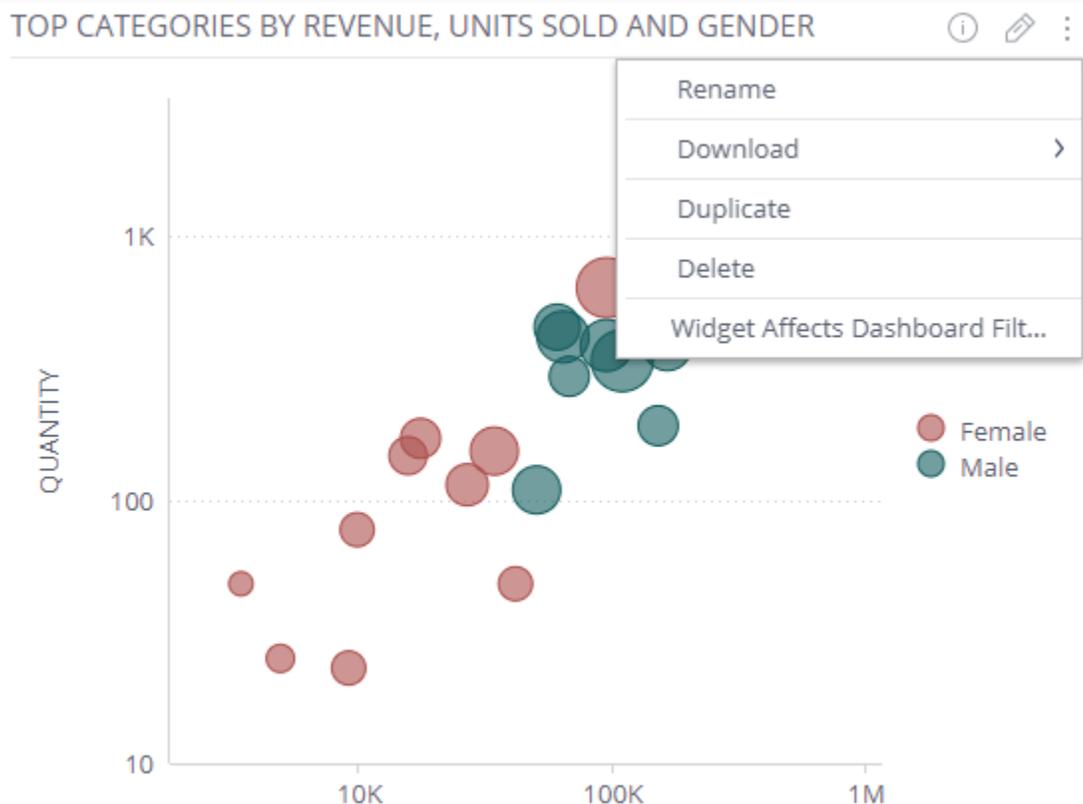
See [Fine-tuning a Widget](#) for more information.

# Copying a Widget

You can create a copy of a widget within the same dashboard or another dashboard in your list. This is useful for saving the original widget in the dashboard before making changes. The new widget is separate from the original so no changes to the new widget affect the original widget.

## To copy a Widget within the same dashboard:

- In Dashboard view, click the widget menu  button in the top-right corner of the widget to be copied and select **Duplicate**.



- If you are editing the widget, click the widget menu  button in the top-right corner of the widget to be copied and select Save A Copy.

## To copy a Widget to another dashboard:

- Select the Header of the widget to be copied and drag it to the dashboard you want to copy it to.

**Note:** When you copy a widget to another dashboard whose data source is another ElastiCube, the Dashboard filters of the new dashboard will not affect the widget.

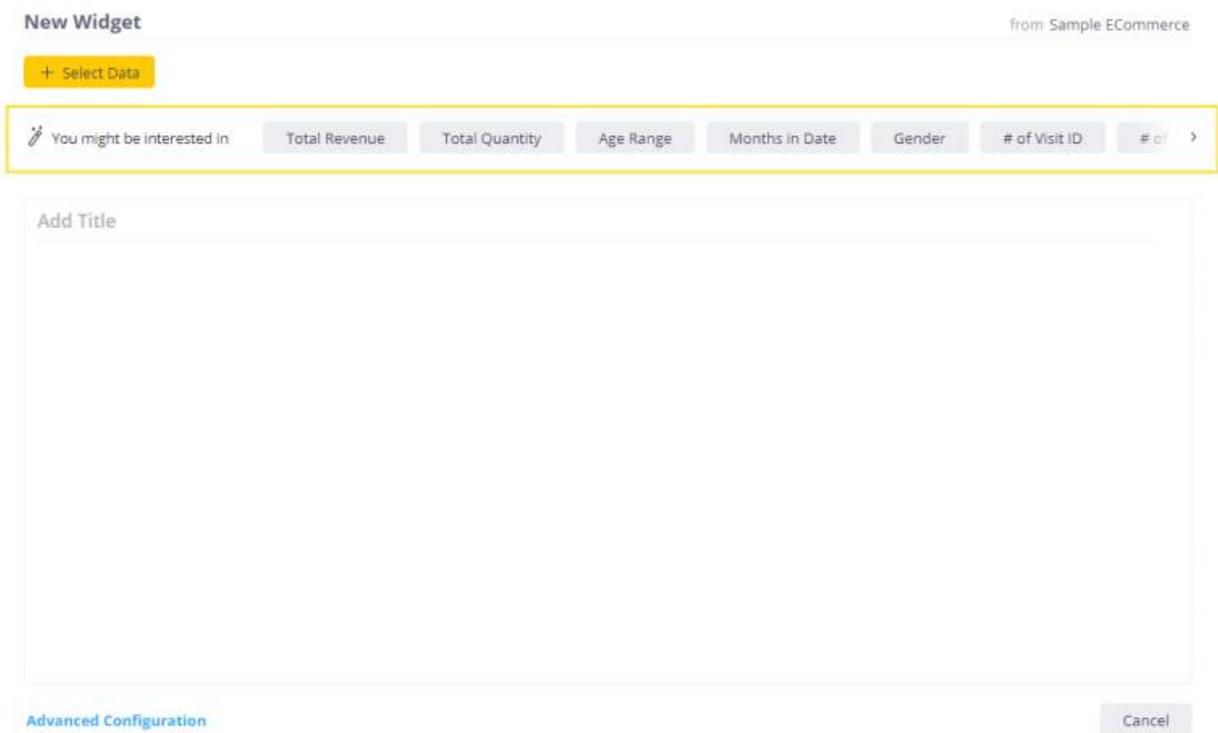
# Deleting a Widget

## To delete a Widget:

- Click the widget menu  in the top-right corner of the Widget and select **Delete**. Confirm the deletion.

## Field Suggestions

To quickly and easily create meaningful visualizations in your dashboard, Sisense displays field suggestions to Dashboard Designers when creating new widgets.



After creating your first dashboard, Sisense begins to display field suggestions the next time you create a widget.

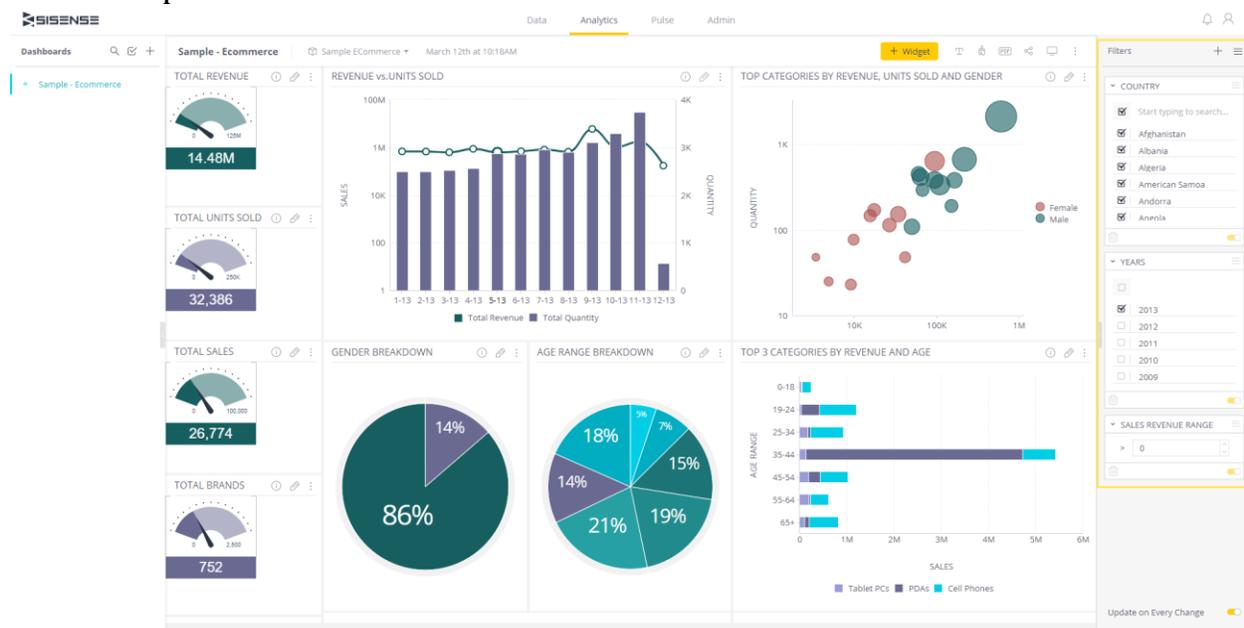
When determining what fields to suggest, Sisense looks at fields being used in other widgets on the same server, which fields are typically working together in other widgets, and what fields you have already selected.

Fields that are already being used in the widget are not suggested.

Sisense has exposed the POST /suggestions API endpoint in the Sisense 1.0 REST API that enables you to retrieve a list of suggestions based on Sisense's algorithm for calculating field suggestions. This is useful for displaying suggested fields in your own application. For more information, see the [Sisense REST API documentation](#).

## Introduction to Filters

The Filters panel on the right of the dashboard provides data filtering options that affect the data displayed in the widgets. You can change these filters to focus on more specific data.



Your Sisense user role determines how you can interact with filters. As a Designer you can create filters that are distributed to Viewers who interact with your dashboards. See [Interacting with Filters as a Designer](#).

Viewers can interact with filters and save their changes locally. See [Interacting with Filters as a Viewer](#) for more information.

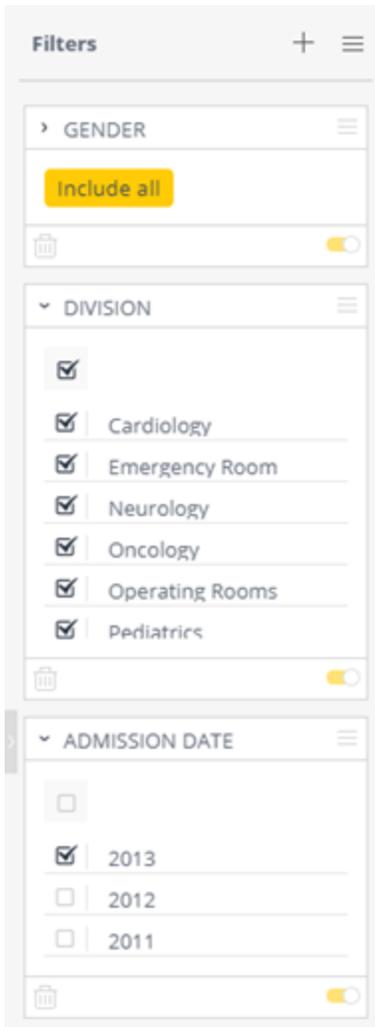
## Interacting with Filters as a Viewer

Viewers can interact with filters for analyzing data. Through the Filters pane, you can modify filters in a dashboard and change the way values are sorted in a visualization. This is useful for focusing on data displayed in dashboards that have been shared by the dashboard's owner.

As a Viewer, you can select and deselect filter visuals on a dashboard. Each time you interact with a filter, for example by selecting or entering a value, the filter is immediately applied to your dashboard. After you have configured the relevant filters, you can save it as your default filters view.

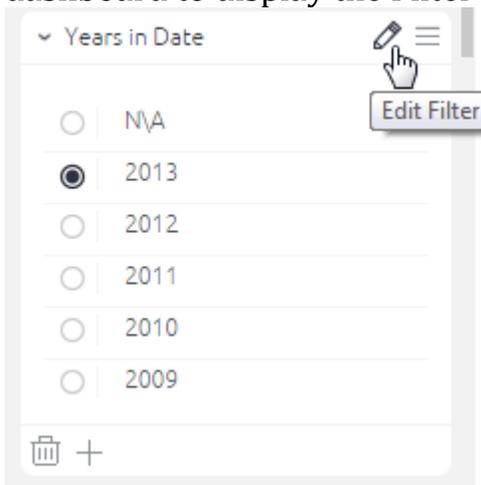
Viewers can interact with filters as follows:

- Make a different selection in the filter controls displayed in the Filter pane on the right side of the dashboard, as shown below:



OR

Click on the pencil icon next to the filter name (shown above) in the dashboard to display the Filter Definition window.



In addition, Viewers can perform the following procedures:

- [Switching Filters On and Off](#)
- [Saving your Default Filters View](#)

To create a filter you must be a [Designer](#).

[Click here](#) to learn more about filters.

#### Interacting with Filters as a Designer

If you are a Dashboard Designer, you can define both dashboard filters and widget filters, as follows:

- **Dashboard Filters:** This type of filter affects all the widgets in the dashboard (except the widgets that you configured as independent). Designers can create dashboard filters and Viewers can view and interact with them.
- **Widget Filters:** This type of filter affects only the specific widget it was applied to. A Designer can define which subset of data values is displayed in a widget. Viewers do not see these filters and cannot change them.

**Note:** Dashboard editing rights are assigned by Dashboard Designers and Designers.

The following topics include procedures for:

- [Creating Dashboard Filters](#)
- [Editing and Deleting a Dashboard Filter](#)
- [Creating Widget Filters](#)
- [Creating Dependent Filters](#)
- [Configuring how Filters Affect the Dashboard and Widgets](#)
- [Limiting Filters to Specific Values with Background Filters](#)
- [Locking Filters](#)
- [Restoring Filters](#)
- [Switching Filters On and Off](#)
- [Saving your Default Filters View](#)

#### Creating Dashboard Filters

This topic explains how to create dashboard filters using the different filtering options.

In this section you will learn how to:

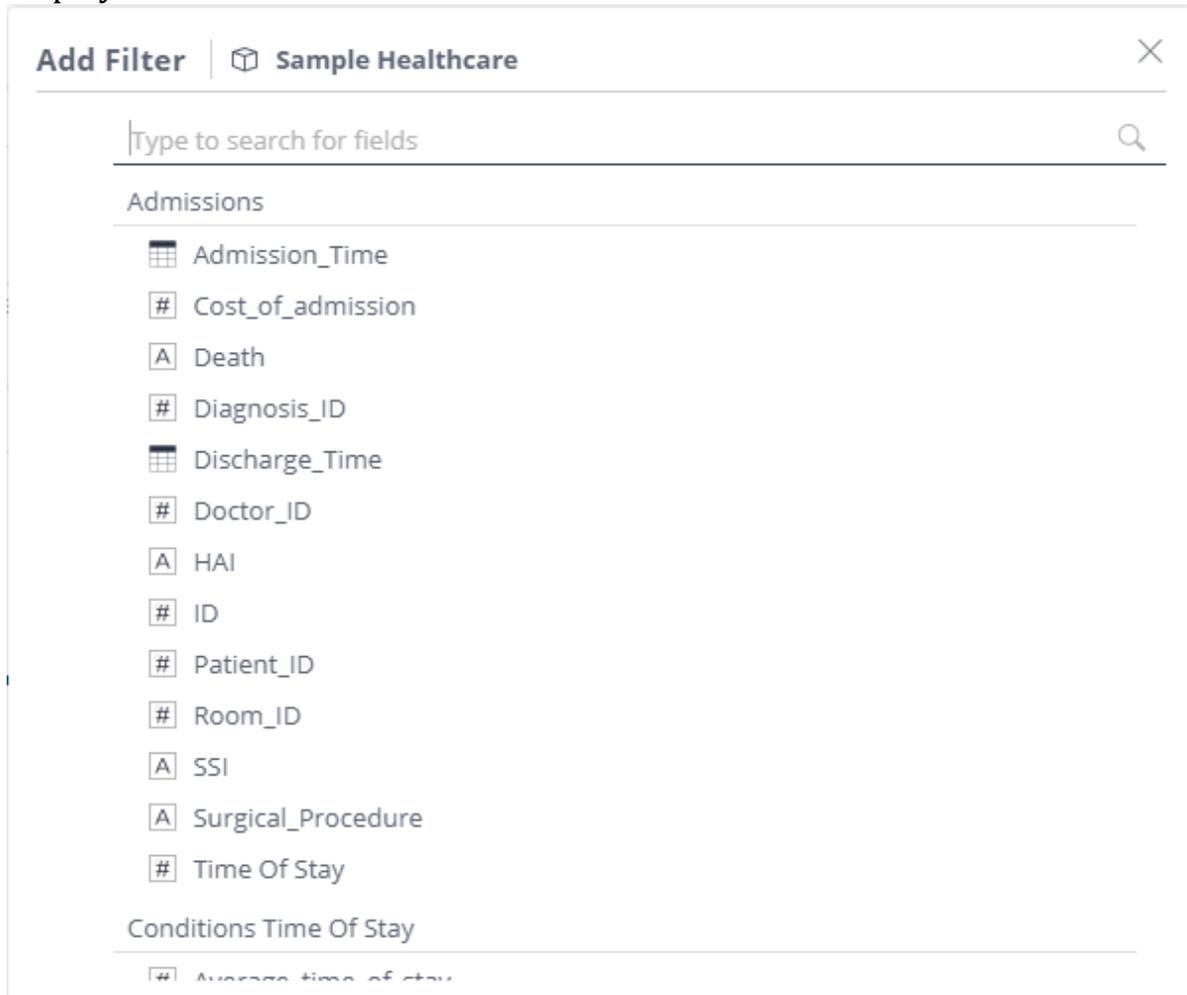
- [Create a Dashboard Filter](#)

- [Create a List Filter](#)
- [Create a Text Filter](#)
- [Create a Numeric Value Filter](#)
- [Create a Top/Bottom Ranking Filter](#)
- [Create a Time Filter](#)
- [Wildcards](#)

# Creating a Dashboard Filter

## To create a dashboard filter:

1. Click **Filter Your Dashboard** in the dashboard's right panel (if it's your first filter), or **+** if you are adding another filter. The Add Filter dialog box is displayed.



2. (Optional) If you have multiple data sources, select the data source that contains the fields you want to filter. Filters applied to fields from one data source do not affect fields from another data source even on the same dashboard.
3. Select the field by which to filter.
4. Select the type of filter from the pane on the left, such as **List**, **Criteria**, **Calendar** or **Ranking**. The options provided in this window may vary according to the selected field type (Date, Numeric or Descriptive).

5. Select the required filtering details in the panel on the right to create one of the following types of filters:

- Creating an Include/Exclude Filter
- Creating a Text/Numeric/Date Filter
- Creating a Top/Bottom Ranking Filter
- Creating a Calendar Filter

These are explained in depth below.

6. Click the **Star Favorite** button to save this filter. You can give your starred filter a unique name by clicking in the text field and editing the given name. This will be useful when you need to quickly add a new filter (with same field type) or update a filter to reflect the unique parameters that you defined in your starred filter.

Note: Your favorite filters are visible to shared users.

7. Click **OK** to filter the data in the Dashboard and add this filter to the Filters panel on the right of the Dashboard.

After a filter has been added to the Dashboard, it stays displayed on the Dashboard's Filters panel. Any changes that you make to the filter are immediately reflected in the Dashboard.

Tip: You can rearrange the filters in the Filters panel by dragging a filter above or below another filter.

# Creating an Include/Exclude Filter (List Filter)

This type of filter enables you to define whether selected field values are included or excluded.

## To define a List filter:

1. Follow Steps 1 – 3 of [Creating Dashboard Filters](#).
2. Select the **List** option in the left panel.

The screenshot shows the 'Add Filter' dialog for 'Sample Healthcare'. The left panel has 'List' selected. The main area shows a table with columns for a checkbox, a field name, a dropdown, and a grid icon. The table contains three rows: 'Years' with a checked checkbox and a dropdown arrow, '2013' with a checked checkbox, and '2012' with a checked checkbox. Below the table is a '2011' row with a checked checkbox. At the bottom, there is a star icon, an 'Include all' button, and an 'OK' button.

	Years		
<input checked="" type="checkbox"/>	2013		
<input checked="" type="checkbox"/>	2012		
<input checked="" type="checkbox"/>	2011		

3. Select/clear the field values to be included/excluded in the Dashboard.
4. Click **OK** to filter the data in the Dashboard and add a List filter to the Filter panel of the Dashboard.

# Selecting Single Selection or Multi Selection

By default, list filters are set to multi-selection. To toggle between single and multi-selection, click the button on the upper right of the list filter:

## Multi Selection:

The screenshot shows a filter dialog box titled "Add Filter" for "Sample Healthcare". On the left, there is a sidebar with filter categories: "List", "Values", "Ranking", "Starred", and "Advanced". The "List" category is selected. The main area displays a list of 11 items, each with a checkbox and a number (1 through 11). The checkbox for item 1 is checked. In the top right corner of the list area, there is a button with a 3x3 grid icon, which is highlighted with a yellow border, indicating it is the multi-selection toggle. At the bottom of the dialog, there is a star icon, a yellow button with the number "1", and a yellow "OK" button.

## Single Selection:

**Add Filter** |  **Sample Healthcare** ✕

---

**List** 

Values	<input checked="" type="radio"/>   1
Ranking	<input type="radio"/>   2
Starred	<input type="radio"/>   3
	<input type="radio"/>   4
	<input type="radio"/>   5
	<input type="radio"/>   6
	<input type="radio"/>   7
	<input type="radio"/>   8
	<input type="radio"/>   9
	<input type="radio"/>   10
Advanced	<input type="radio"/>   11

---

 **1** OK

# Creating a Text Filter

This type of filter enables you to filter according to text matching.

## To define a Text filter:

1. Select the Text tab.
2. Follow steps 1 – 3 of [Creating Dashboard Filters](#). Select a Textual field.
3. Select the operator, for example Starts with or Containing.
4. Type in the value by which to filter.
5. If you require additional conditions, click **+ Add condition**. Select **AND** or **OR** to define how filter operators are combined.

# Creating a Numeric Value Filter

This type of filter enables you to limit your dashboards to specific value ranges.

For example:

- Keep only sales above 100\$.
- Keep only product IDs between 1000020 and 1000030.

## To define a Numeric filter:

1. Select the **Values** tab.
2. Follow steps 1 – 3 of [Creating Dashboard Filters](#).
3. Select the operator, for example Equals or Greater than.
4. Type in the value by which to filter.
5. If you require additional conditions, click **+ Add condition**. Select **AND** or **OR** to define how filter operators are combined.

# Creating a Top/Bottom Ranking Filter

This type of filter enables you to select whether to include only the top/bottom ranking fields.

## To define a Ranking filter:

1. Follow steps 1 – 3 of [Creating Dashboard Filters](#).
2. Select the **Ranking** filter tab.
3. Select **Top** or **Bottom** and the number of items to include.
4. In the **By** field, select a field by which to rank the fields. A list of the fields that match these definitions is displayed so that you can verify that the results are as you expect.
5. Click **OK** to filter the data in the Dashboard and add a Ranking filter to the Filter panel of the Dashboard.

## Creating a Time Filter

There are two types of time filtering options, Calendar and Dynamic Time filters.

**Note:** Unlike other filters that you define, viewers will be able to change the type of date filter, for example, from Time Frame to Calendar, etc., providing more flexibility for viewing date ranges.

## Calendar Filter

This filter enables you to select custom date ranges from a calendar.

### To define a Calendar filter:

1. Follow steps 1 – 3 of [Creating Dashboard Filters](#). Select a date field.
2. Select the **Calendar** filter tab.
3. Use the quick navigation menu to jump to a point in time.
4. Select a date range using any of the following methods:
  - Use the calendar to select a date range to filter by. Use the left calendar to select the starting date of the range, and the right calendar to select the end date.
  - Type in the dates.
  - Use a shortcut such as Earliest Date or Today.

**Note:** Earliest Date and Latest Date refer to the earliest and latest dates with data.

The screenshot displays the 'FILTER DATE' dialog box for 'SAMPLE LEAD GENERATION'. It features a sidebar with options like 'List', 'Calendar', 'Time Frame', 'Ranking', and 'Starred'. The main area shows a calendar for Q2 2013, with callouts indicating 'Click to select a date' and 'Quick Navigation'. Below the calendar, there are input fields for 'From' (04/06/2013) and 'To' (05/31/2013), along with 'Today' and 'Earliest Date' shortcuts. A summary bar at the bottom shows 'From 4/6/13 - To 5/31/13' and an 'OK' button.

5. Click **OK** to filter the data in the Dashboard and add a date filter to the Filter panel of the Dashboard.

## Dynamic Time Filter

This filter enables you filter dashboards to preset dynamic time frames, such as 'Last Year', 'Last 2 Years', '2 Year ago', etc.

### **To define a dynamic time filter:**

1. Follow steps 1 – 3 of [Creating Dashboard Filters](#). (Select a date field)
2. Select the **Time Frame** filter tab.
3. Use the desired option.
4. Click **OK** to filter the data in the Dashboard and add a date filter to the Filter panel of the Dashboard.

## Advanced Filtering Criteria

If you require advanced filtering options that are not included in the provided settings (for dashboard or widget), you can add or edit filters in the **Advanced** section.

### To add or edit filtering criteria:

1. Click **Advanced** in the left panel.
2. Add or edit filtering criteria. Click **Test** to see the results of your new/edited filter.
3. Click **OK** to add the filter.

### Example 1 – Adding an OR statement

The example below shows an OR statement being used to display multiple names. After clicking **Test**, the matches are displayed to the right of the script.

The screenshot shows a dialog box titled "FILTER DIRECTOR1" with a close button (X) in the top right corner. On the left side, there is a vertical menu with four options: "List", "Text", "Ranking", and "Starred". The "Advanced" option at the bottom of this menu is highlighted with a yellow border. Below the menu is a dark blue bar with the word "Test" in white. The main area of the dialog is split into two panels. The left panel contains a JSON script: 

```
{  
  "or": [  
    "contains": "martin"  
  ],  
  "contains": "alfred"  
}]
```

 The right panel displays the results: "Showing 2 matches:" followed by "Alfred Hitchcock (dir.)" and "Martin Scorsese (dir.)". At the bottom left of the dialog is a star icon, and at the bottom right is a green button with the text "OK".

## Examples using JAQL Queries

### Example 2 – Filtering by Measures

This option lets you filter dimensions by measures, returning only the filtered members and associated values.

In this example, all products with a total price equal or greater than 1 million will be returned.

The query:

```
"datasource": "laptopsales",
  "metadata": [
    {
      "dim": "product"
    },
    {
      "dim": "price",
      "agg": "sum",
      "filter": {
        ">=": 1000000
      }
    }
  ]
}
```

Results include the relevant product names and price, for example:

- Product A, 1000000
- Product B, 1551246

### **Example 3 – Filtering by Dimension Attributes**

This option lets you filter dimensions by filtered attributes of other dimensions, returning only the filtered values.

In this example, all customers that purchased Product A and Product B will be returned.

```
{
  "datasource": "laptopsales",
  "metadata": [
    {
      "dim": "customers",
      "filter": {
        "attributes": [
          {
            "dim": "products",
            "filter": {
              "members": ["Product A",
                "Product B" ]
            }
          }
        ]
      }
    }
  ]
}
```

Note that you can add as many nested filters as needed.

## Updating on Every Change

When this option is on, any changes you make to the dashboard filters are updated immediately, and the dashboard is refreshed to reflect the changes. If you are working with large or complex data sets that result in longer refresh times, then switch off this option. Make all your changes, and click the Update button when you are ready to update your dashboard.

**Note:** This option is available for Designers only. Sisense viewers can make changes that will update automatically on every change.

# Wildcards

To filter lists or text quickly, you can use wildcards such as '\_' or '%', which will be interpreted as literal characters or an empty string.

Sisense supports the following wildcards:

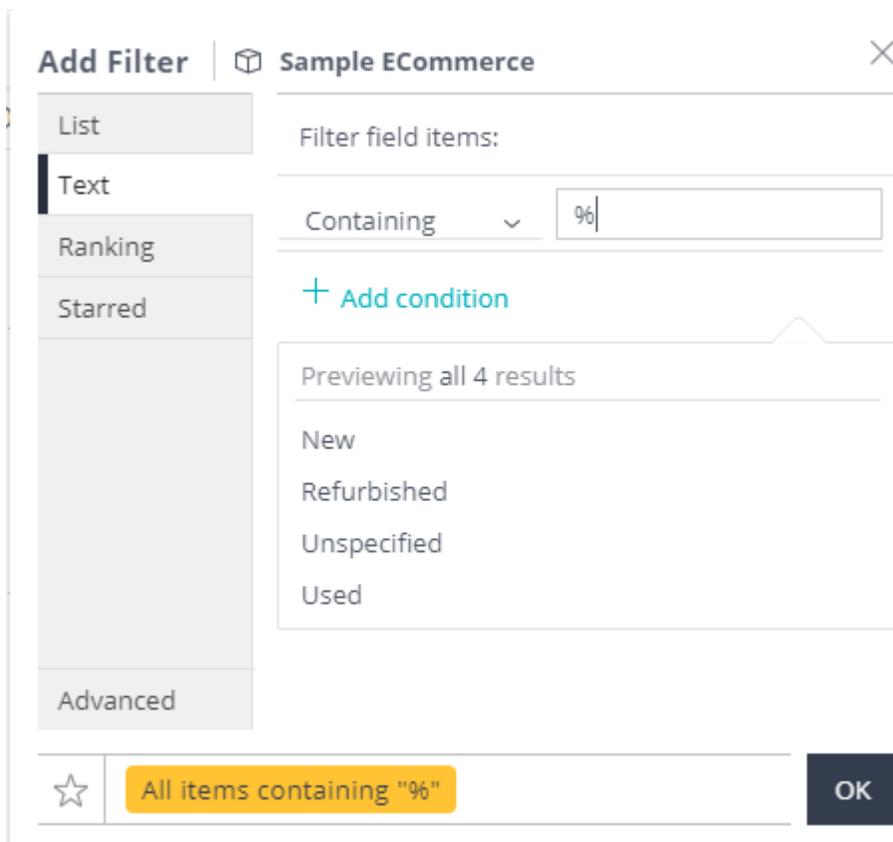
## Single Characters

You can enter an underscore '\_' as a placeholder for a single unknown character.

The screenshot shows a 'Sample ECommerce' dialog box with a sidebar on the left containing filter categories: List, Text (selected), Ranking, Starred, and Advanced. The main area is titled 'Add Filter' and contains a 'Filter field items:' section with a dropdown menu set to 'Containing' and a text input field containing 'n\_w'. Below this is a '+ Add condition' button. A preview section shows 'Previewing all 1 results' and a list item 'New'. At the bottom, a star icon is next to a yellow bar that reads 'All items containing "n\_w"', and an 'OK' button is on the right.

## Multiple Characters

You can use a percent % symbol to search for an undefined length of string.



## Escape Characters

If you are searching for a string with an underscore or percent symbol, add a backslash (\) in front of the special character. Backslashes are treated as an escape character by Sisense.

For example, to locate results for the string “\_te”, use a backslash before the string:

**Edit A Filter** |  **escape2** ✕

List

**Text**

Ranking

Starred

Advanced

Filter field items:

Containing v

[+ Add condition](#)

Previewing all 1 results

my\_text

---

 All items containing "\\_te" OK

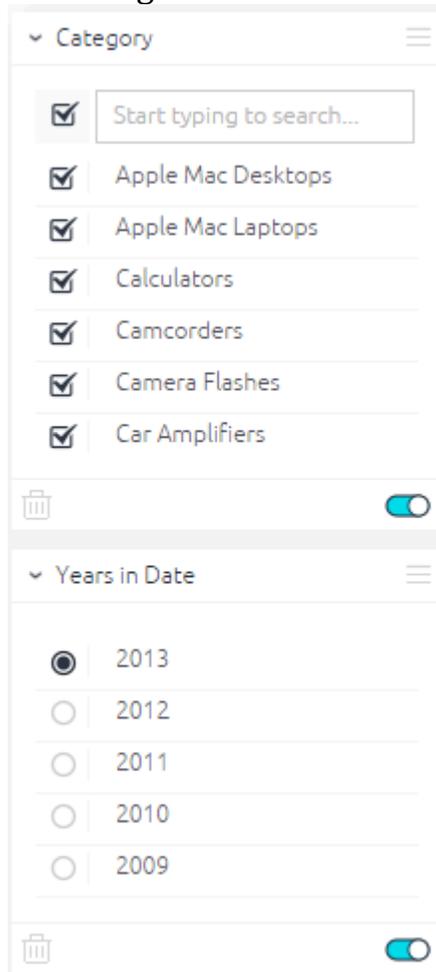
### Editing and Deleting a Dashboard Filter

This topic explains how to edit a dashboard or delete a dashboard filter.

# Editing a Dashboard Filter

## To edit a dashboard filter:

- Make a different selection in the filter controls displayed in the Filter panel on the right side of the dashboard, as shown below:



OR

Click on the pencil icon next to the filter name (shown above) in the dashboard to display the Filter Definition window, as described in [Creating](#)

## Dashboard Filters.

▼ Years in Date

N/A

2013

2012

2011

2010

2009

**Edit Filter**

# Deleting a Dashboard Filter

## To delete a filter:

- Click the bin  icon. To temporarily disable a widget's filter, you can toggle the filter on or off using the toggle switch. Delete the widget only if you do not need it any longer.

## Creating Widget Filters

This topic describes how you can use the Widget Designer to create widget filters and add widget filters to an existing field.

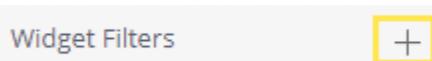
In the Widget Designer, you can define a filter for the data in a widget. or example, you can create a filter that only shows the fields of relevant countries. The filters are not visible in the dashboard, nor can they be edited directly from the dashboard.

Note about duplicate filtering: When creating a widget filter for a field that is already filtered in a dashboard filter, the widget filter will override the dashboard filter. Example: You have a dashboard filter, filtering the months field, with the values January and February selected. At the same time you create a filter for a widget, also for months, but this time March and April are selected. When both filters are applied in the dashboard, you will see data for March and April in the widget, according to the widget's filter. If your widget is filtered using measured values, then the measured value will override any other widget or dashboard filters you have for the same fields.

# Creating a Widget Filter

## To add a widget filter:

1. Open (edit) this Widget in the Widget Designer, as described in [Adding Widgets to a Dashboard](#).
2. Click the **Filters** tab in the Widget Designer.
3. The Filters panel is displayed showing the filters defined for this Widget. Click on the **Add Filter** button to display the Data Browser, which lists the fields in the ElastiCube.



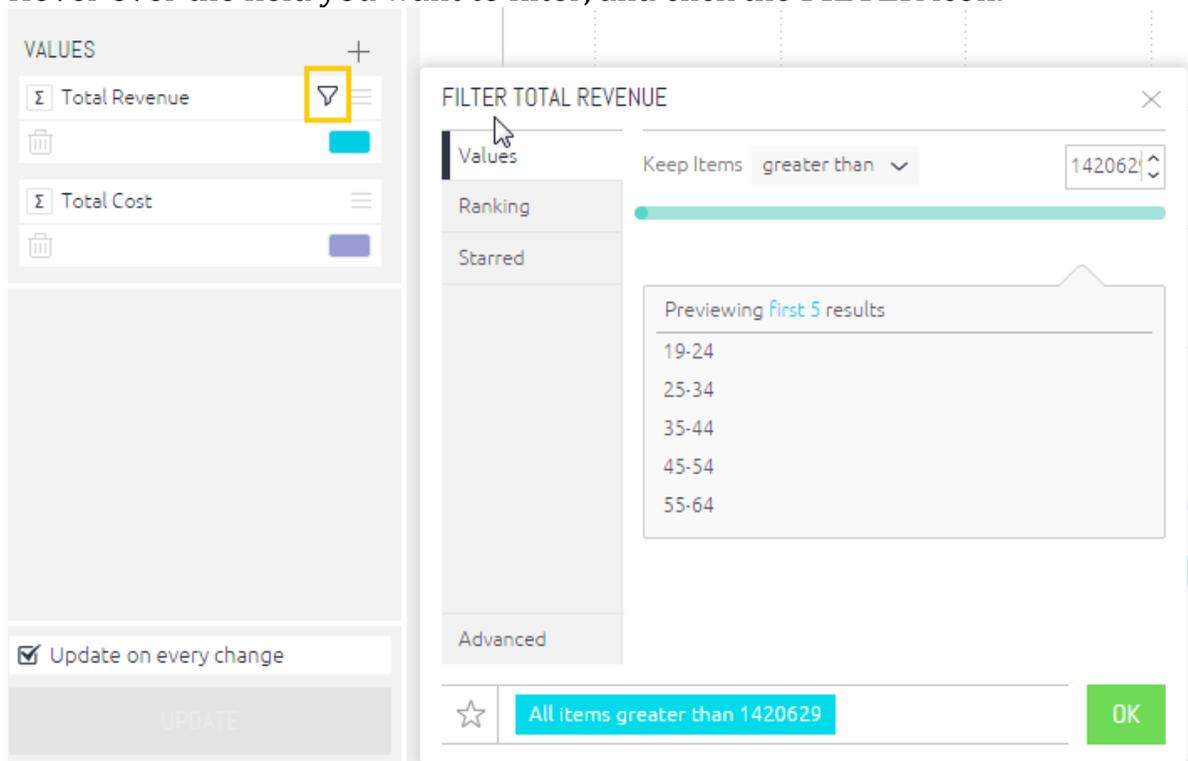
4. Click on a field in the Data Browser to add it as a filter. You can define multiple filters, each for a different field. These filters have an **AND** relationship between them. Each filter that you define appears on the **Widget Filters** panel on the right, enabling you to easily edit it if needed. The procedure for defining a Widget filter is similar to the procedure for defining a Dashboard filter, as described in [Creating Dashboard Filters](#). The difference is that the widget filter only affects its hosting widget, and is accessible only via the widget designer, not directly from the dashboard.

# Adding a Widget Filter to an Existing Field

Adding a new widget filter, as described above, lets you use any field as a widget filter. Sometimes you want to directly filter a field that is already part of the widget and visible in the data panel on the left.

## To add a Widget Filter to an existing field:

- Hover over the field you want to filter, and click the **FILTER** icon.



## Creating Dependent Filters

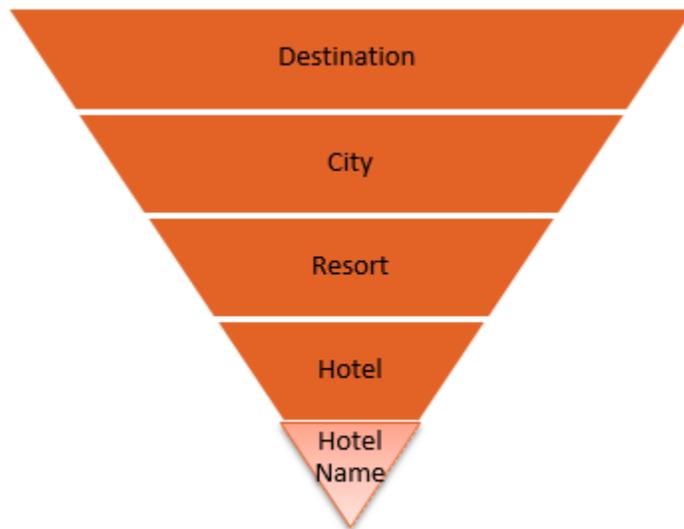
This topic describes dependent filters.

Dependent Filters combine multiple dashboard filters into a hierarchy of dependent filters that affect each other from parent, to child, to grandchild and so on. When you filter the parent filter, fewer options will be available in the child filter.

## Example Use Case

The following example shows how using a dependent filter helps you easily focus information in a dashboard.

This example describes an online travel agency with 250,000 hotels in its database. An agent wants to find a specific hotel, but does not remember its name. To narrow the results, the agent can define a dependent filter. This dependent filter is comprised of four sub-dependent filters defined with the following hierarchy:

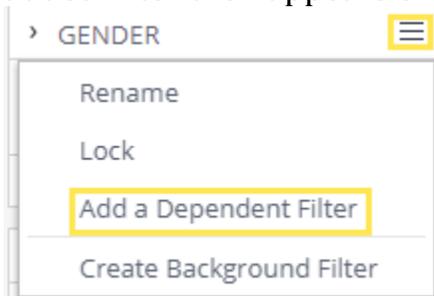


- **Destination:** Displays a list of States and Countries. In this example, the agent will select the Destination Florida.
- **City:** Because Florida was selected, only Destinations in Florida are displayed in the child filter, such as: Miami, Orlando and Tampa. In this example, the agent will select the City Orlando.
- **Resort:** Because Orlando was selected, only resorts in Orlando are displayed in the grandchild filter, such as Floridays, Buena Vista, Walt Disney World. In this example, the agent will select the Walt Disney World resort.
- **Hotel:** Because Walt Disney World was selected, only the Hotels in Walt Disney World are displayed in the great-grandchild filter.

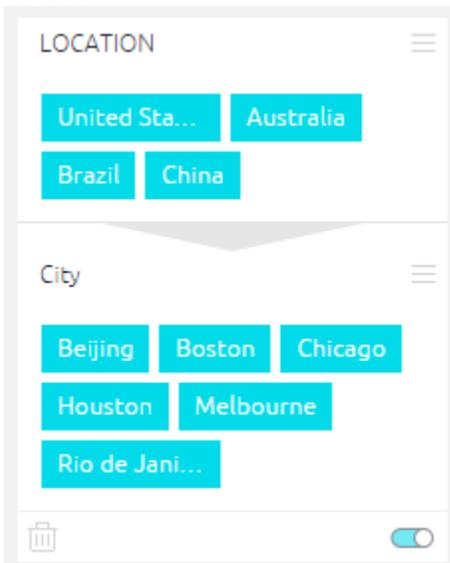
**Note:** By default, filters are not dependent. Dependent Filters can be defined only from selection type filters.

**To define a Dashboard Dependent Filter:**

1. The top level (most inclusive) filter of the Dependent Filter must first be defined in the usual manner (see [Creating Dashboard Filters](#)). This newly added filter then appears on the dashboard.



2. On the parent filter, click + or click → **Add a dependent filter**.
3. Define the child filter in the usual manner (see [Creating Dashboard Filters](#)). The number of available options for selection in the child filter are less because of the selections in the parent filter. The child filter appears on the dashboard with a down arrow indicating the dependent filter and its parent above.



4. Repeat this step for each new child filter to create as many sub-levels of Dependent Filters as you need.

#### Configuring How Filters Affect the Dashboard and Widgets

This topic explains how dashboard filters work together with widget filters, and how you can configure different filtering behaviors and their effects on the dashboard.

The following procedures are explained below:

- [Configuring how Dashboard Filters Affect a Widget](#)

- [Defining how a Widget Behaves when Filtered](#)
- [Defining how a Selection on the Widget Affects the Dashboard](#)

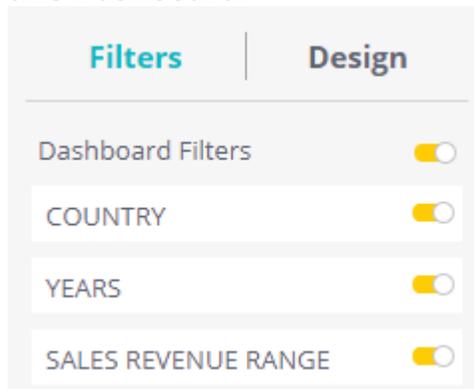
# Configuring how Dashboard Filters Affect a Widget

Dashboard filters affect all the widgets in the dashboard, except widgets that you have defined to be independent. An independent widget is not affected by dashboard filters.

**Note:** Dashboard filters are case-sensitive. For example, if you have two columns with the same name, but one is uppercase and the other is lowercase, a dashboard filter applied to one will not be applied to the other.

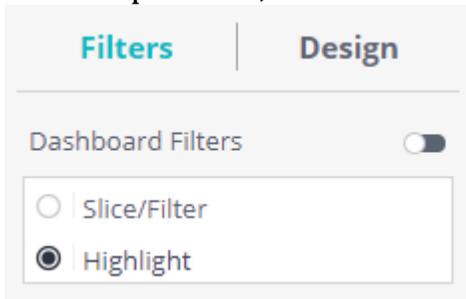
## To make a widget filter-independent:

1. Open (edit) this Widget in the Widget Designer, as described in [Adding Widgets to a Dashboard](#). The top part of the Widget Filters panel on the right of the Widget Designer lists the Dashboard Filters that are defined for this Dashboard.



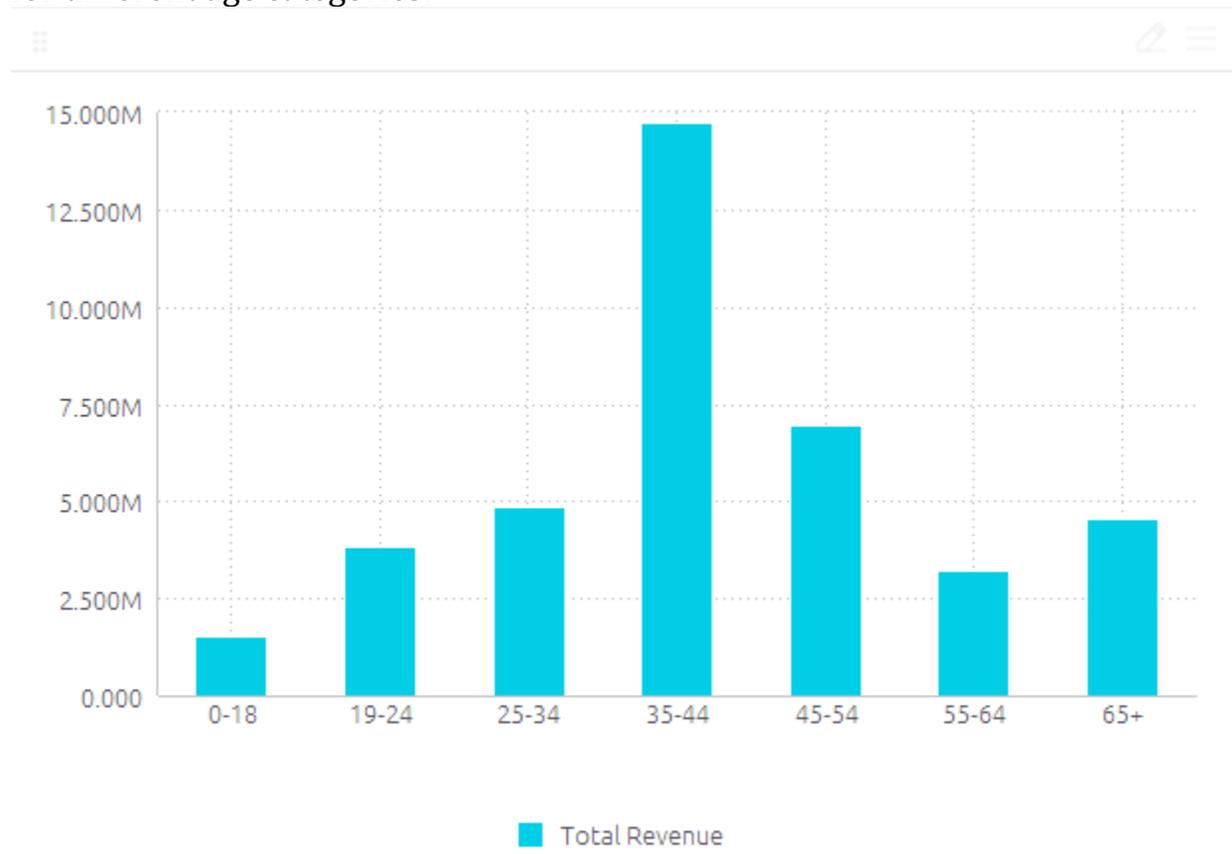
2. Turn off the Dashboard filters that you do not want to affect this Widget: The toggle on icon  (default) indicates that the Widget is affected by this Dashboard filter. The toggle off  icon indicates that the widget is not affected by this Dashboard filter. You also have the option to make this Widget independent of all current and future Dashboard Widgets by turning the Dashboard

Filters option off, as shown below:

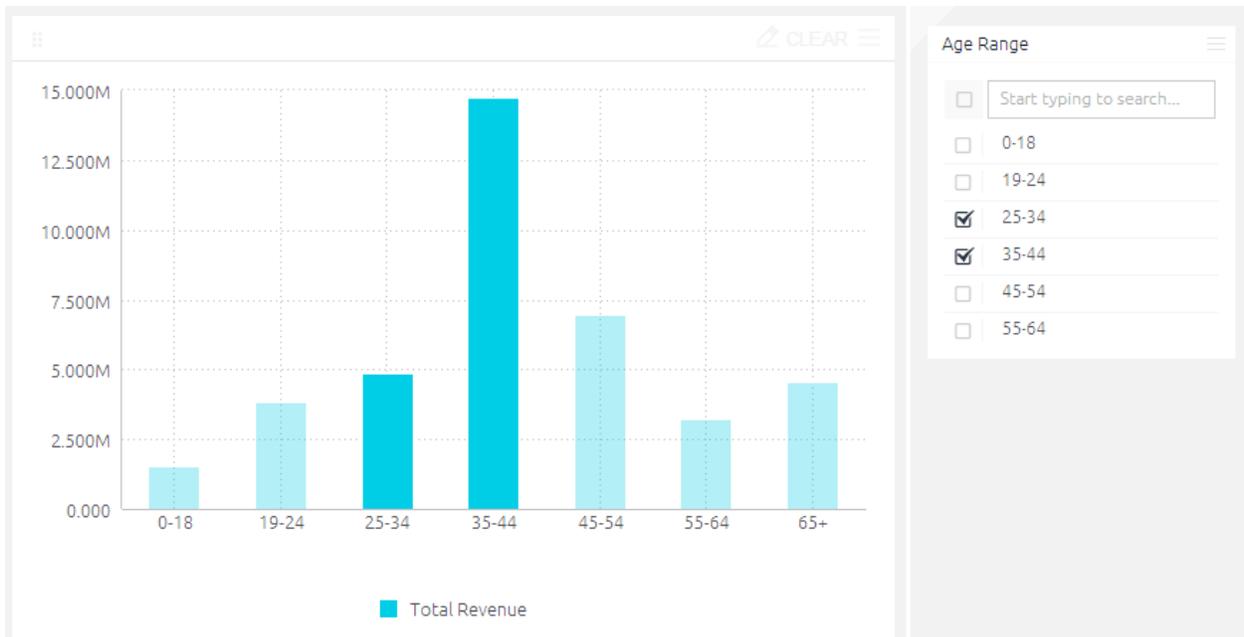


# Defining how a Widget Behaves when Filtered

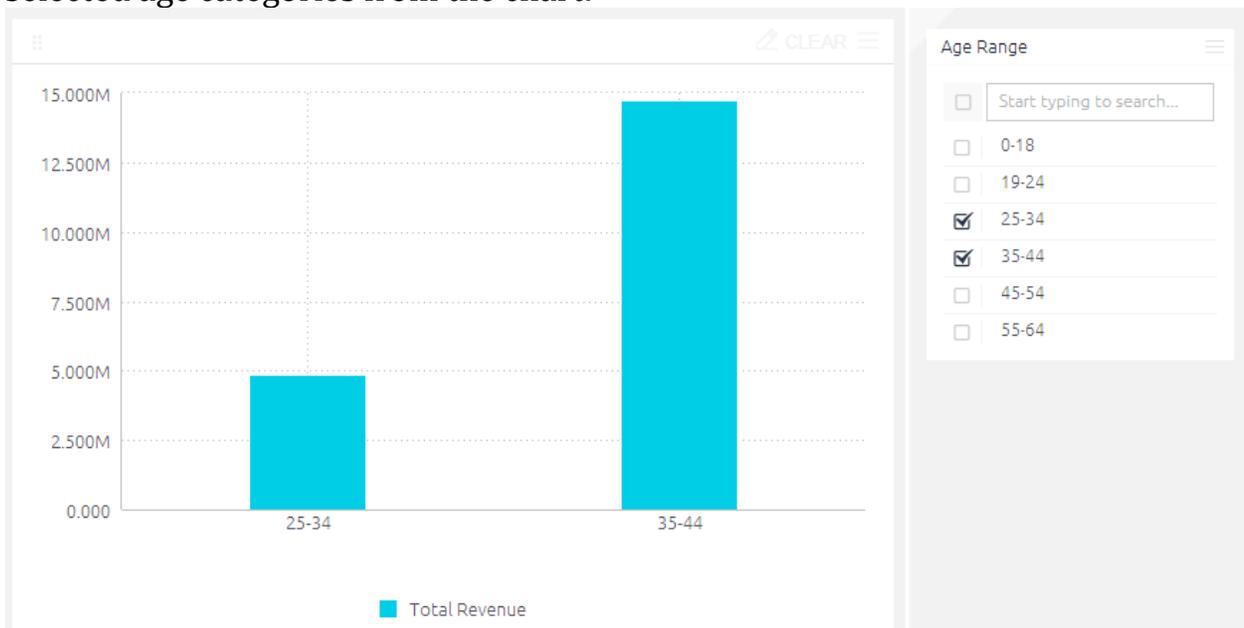
When fields that are visible in a widget are filtered, it can be displayed in two ways – Slice/Filter or Highlight. For example, this column chart shows revenue for different age categories:



If the chart is set to Highlight mode, filtering a subset of age categories will highlight them in the chart:

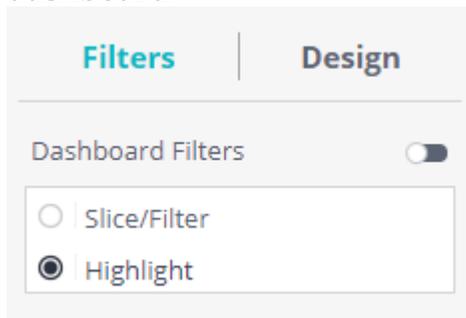


If the chart is set to **Slice/Filter** mode, the same filter will remove all non-selected age categories from the chart:



**To define the filter behavior mode:**

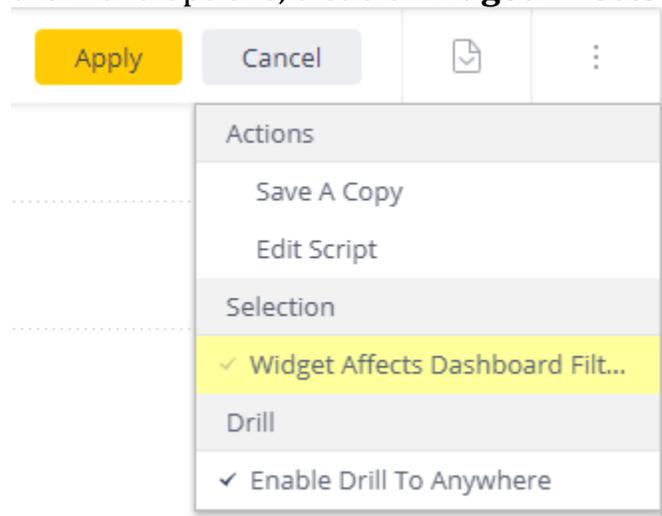
- Select **Slice/Filter** or **Highlight** in the Filters panel on the right side of the dashboard.



# Defining how a Selection on the Widget Affects the Dashboard

When making a selection on a widget, a filter is added to the dashboard with the selected part of the chart as the applied filter.

To disable this option for a widget, open the widget in editing mode, and from the menu options, disable **Widget Affects Dashboard Filters**.



## Limiting Filters to Specific Values with Background Filters

This topic describes how you can leverage background filters to limit what data is exposed to your Viewers or to reduce the amount of filtering a Viewer needs to do.

As a Designer, you can define which of the fields in a filter will be visible to Viewers. After defining which fields will be included in the filter, viewers will be able to use the updated and restricted filter to further filter the dashboard or widget using the available fields.

This can be useful in cases where Viewers only require selected data, and then they do not have to search through a long list of fields, or for security purposes in cases where you do not want to expose all the data.

**Note:** You must have access to edit a dashboard to view this option.

### **Example 1 – Simple List Filter**

A filter includes a list of 10 countries. As a designer, you can limit the filter to include only five specified countries. Viewers will see the filter as having only five countries to include or exclude in the filter.

### **Example 2 – Conditional Filter**

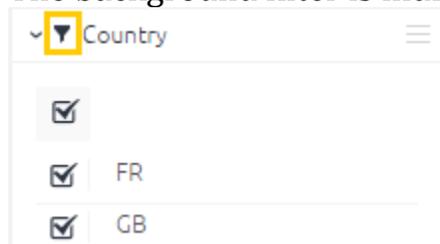
Another example requires having only the top 5 countries by population included in the available list of countries. In this case, Viewers will only see five countries, however, the list of countries will depend on the population sizes and any changes that may affect which countries are in the top 5 at the given time.

#### **To create a background filter:**

- Create a background filter using one of the following methods:
  - In a filter without selected values, open the filter's menu and click **Create background filter**. Define your filter settings, and click **OK**.  
OR

If you made changes to your filter (either by editing the filter or by selecting values directly in the filter), you can open the filter's menu, and click **Set as background filter**.

The background filter is indicated by a filter ▼ icon.



**Note:** The icon is not visible to viewers.

#### **To edit a background filter:**

1. In the filter's menu, click **Edit background filter**.
2. Update the filter settings, and click **OK**.

#### **To remove a background filter:**

- In the filter's menu, click **Remove background filter**.

## Locking Filters

As a Designer, you can lock a filter, preventing Viewers from making any changes. The Viewer will be able to see the defined filter settings, but the filter will be disabled for editing.

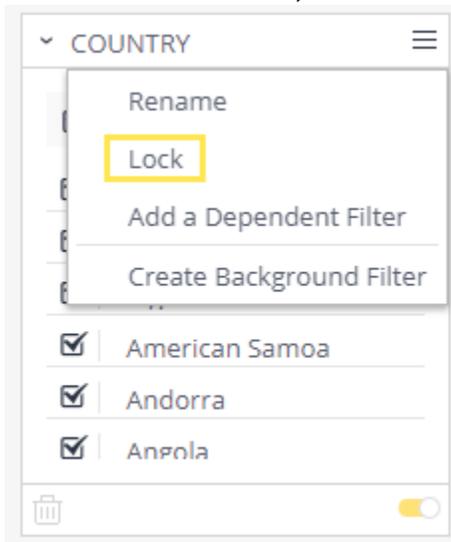
**Note:** You must have access to edit a dashboard to view this option.

### To lock a filter:

1. Click the Filter menu.
2. Click **Lock**.

A lock icon appears to indicate that the filter is locked.

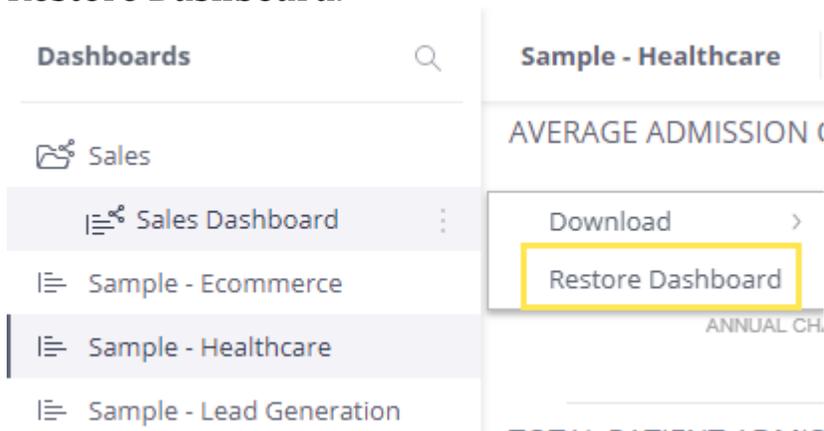
To unlock the filter, click on the menu, and select **Unlock**.



## Restoring Filters

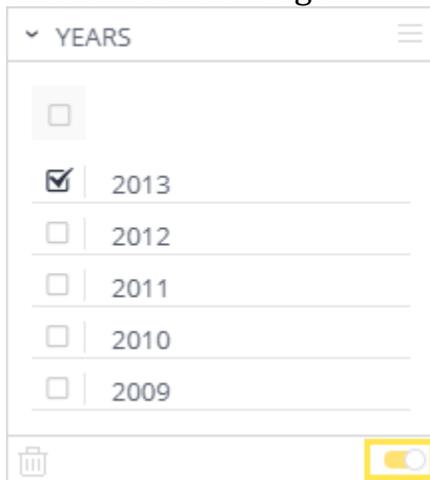
As a Designer, if you changed the dashboard's filters, you can revert to the latest copy shared by the dashboard's owner at any time. When you restore a dashboard, local changes to a dashboard by Viewers are overridden and the default dashboard is restored for all users.

- Under the Dashboards list, click on the menu of the dashboard and select **Restore Dashboard**.



### Switching Filters On and Off

You can easily toggle filters on and off using the toggle switch. Use this option to compare states (with and without the filter), or to temporarily disable a filter, rather than deleting it.



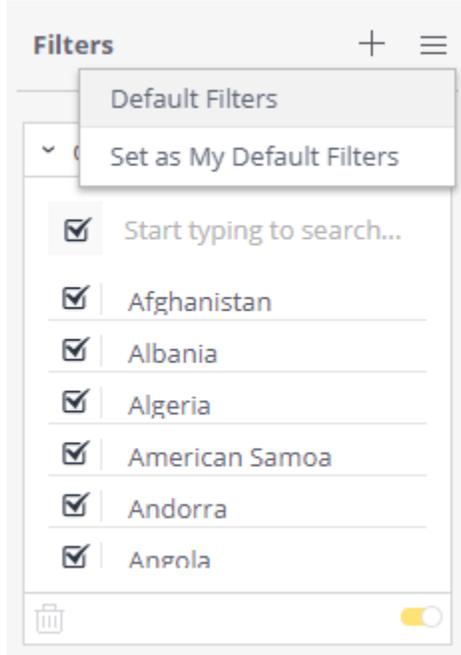
### Saving Your Default Filters View

You can save the current state of your filters and their settings at any time. The current state includes the existing filters, their configuration, and the order in which they appear in the filters panel.

After making changes to any of the above settings, you will be able to restore your filters to their previously saved state.

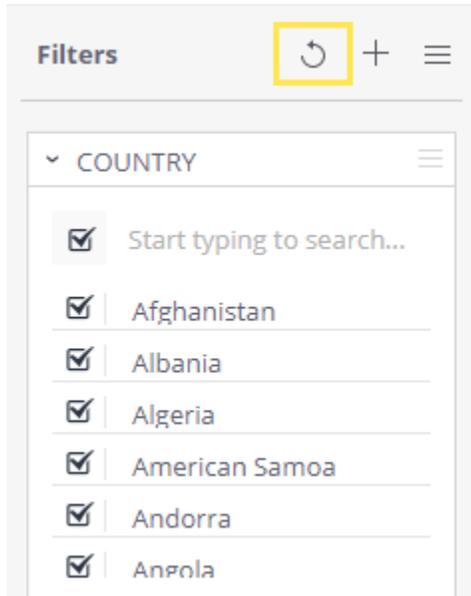
### To save your current set of filters:

- In the Filters menu, click **Set as My Default Filters**.



### To restore a saved filters set:

- Click the restore icon next to the Filters menu.



This topic provides an overview to working with formulas in Sisense along with important tips and examples.

Formulas are custom calculations performed on one or more fields in your data.

They offer an important way to analyze results and express business logic.

Sisense's formula capabilities are designed around several principles;

- Create complex business calculations without IT or technical knowledge.
- Easily combine fields from different data sources together.
- Customize formulas to reflect specific criteria and conditions.
- Work with raw data without the need to summarize data sets before creating formulas.
- Instantly recalculate formulas based on any filter, variable or level of granularity.

The table below provides a reference to the main formula functions available in Sisense. [Click here](#) to see a complete reference.

Goal	Function	Types and Syntax
Perform calculation based on criteria	Measured value	<p><b>Value Filters:</b> <math>\neq</math>, <math>=</math>, <math>&gt;</math>, <math>&lt;</math>, between</p> <p><b>Text Filters:</b> Contains, Doesn't Contain, Doesn't End With, Doesn't Start With, Ends With, Start With, Equals, Not Equal</p> <p><b>List Filter:</b> Include, Exclude</p> <p><b>Ranking Filters:</b> Top, Bottom Ranking</p> <p><b>Time Filter:</b> Date and Calendar</p>
Combine data/apply simple mathematics	Aggregate functions	<p><b>Operator:</b> <math>+</math>, <math>-</math>, <math>*</math>, <math>/</math></p> <p><b>Aggregate:</b> Sum()</p> <p><b>Average:</b> Avg()</p> <p><b>Count:</b> Count(), DupCount()</p> <p><b>Range:</b> Max()/Min()</p>
Summarize data	Statistical function	<p><b>Central Tendency:</b> Median(), Model, Largest()</p> <p><b>Std Deviation and Variance:</b> Stdev(), Stdevp(),</p>

Goal	Function	Types and Syntax
		Varp(), Var() <b>Quartile and Percentile:</b> Quartile(), Percentile()
Accumulate data	Rolling sum/average	<b>Sum to Date:</b> YTDSum(), QTDSum, MTDSum() <b>Avg to Date:</b> YTDAvg, QTDAvg, MTDAvg()
Compare Time or Trends	Time functions	<b>Past Periods:</b> PastYear(), PastQuarter(), PastMonth(), Next(), Prev() <b>Growth Trend:</b> Growth(), GrowthRate() <b>Time Difference:</b> YDiff(), QDiff(), MDiff(), DDiff(), HDiff(), MnDiff(), SDiff()

The following topics explain how to create formulas, and describe the features of the formula editor.

- [Using the Formula Editor](#)
- [Creating and Editing a Formula](#)
- [Reusing Formulas](#)
- [Using Quick Functions](#)
- [Creating Formulas Based on Criteria and Conditions \(Filters\)](#)
- [Building Formulas with Functions](#)
- [Function Reference](#)

### Formula Editor

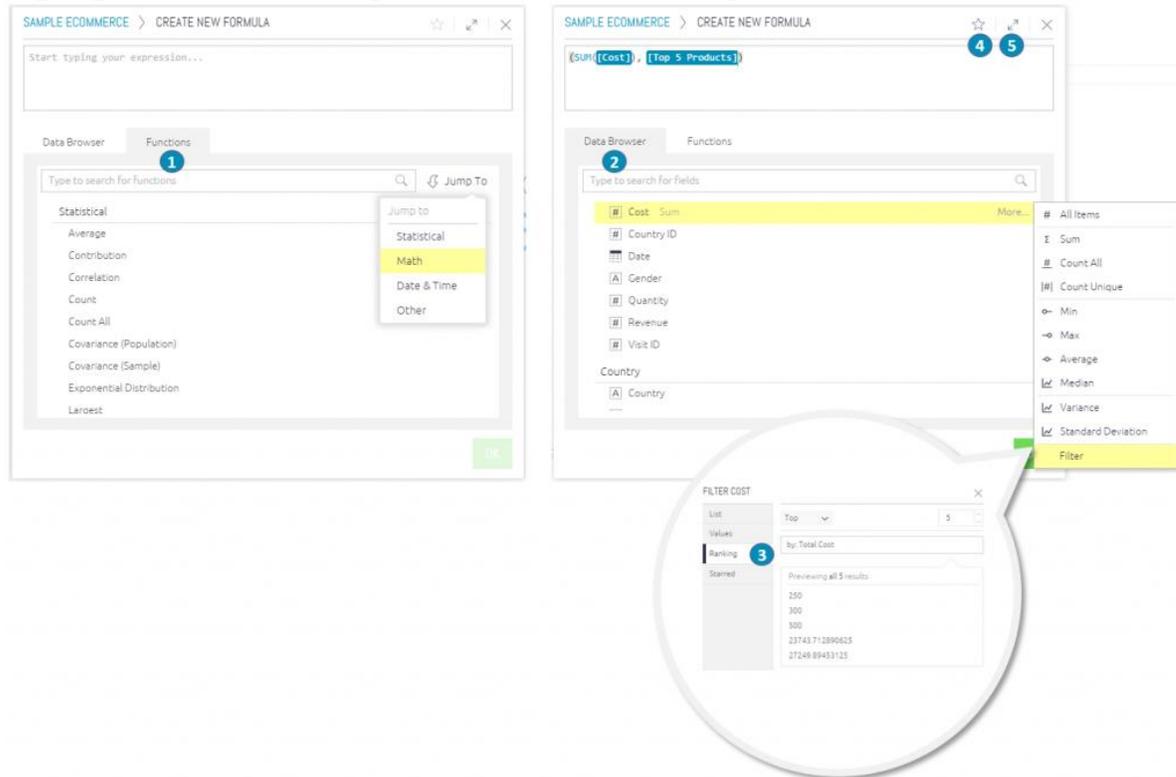
The Formula Editor is where Designers define formulas for a dashboard's widgets.

If you are familiar with the Formula Editor, see the [Quick Reference Guide](#). For a detailed guide with examples, continue reading below.

### To open the Formula Editor:

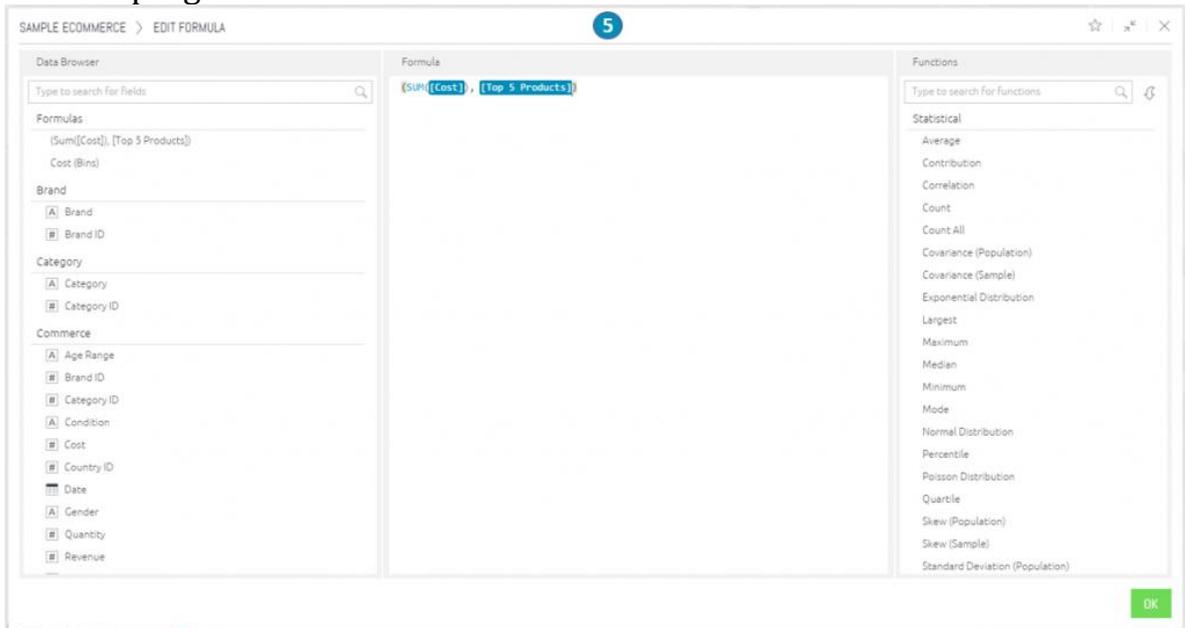
- Do one of the following:
  - For a new widget, click **Select Data**, and then  .
  - For an existing widget, click on the edit formula button  .

The Formula Editor has two tabs, the Data Browser to select fields and the Functions tab to select formula operations. You can create a formula combining one or more function, field and filters. The diagram below highlights the main components of the formula panel.



1. **Functions** are operations which perform different calculations, for example a sum. Use the 'Jump To' menu or the search box to quickly find the formula you need.
2. Fields in the **Data Browser** are variables contained in the data set (ElastiCube). Clicking on a field in the data browser will include it as part of the formula.
3. [Filters](#) can be applied to restrict formulas based on criteria.
4. [Starring](#) is a way to save a formula for later use.

5. The Formula Editor window can be expanded by clicking the expand button at the top right.



### Creating and Editing a Formula

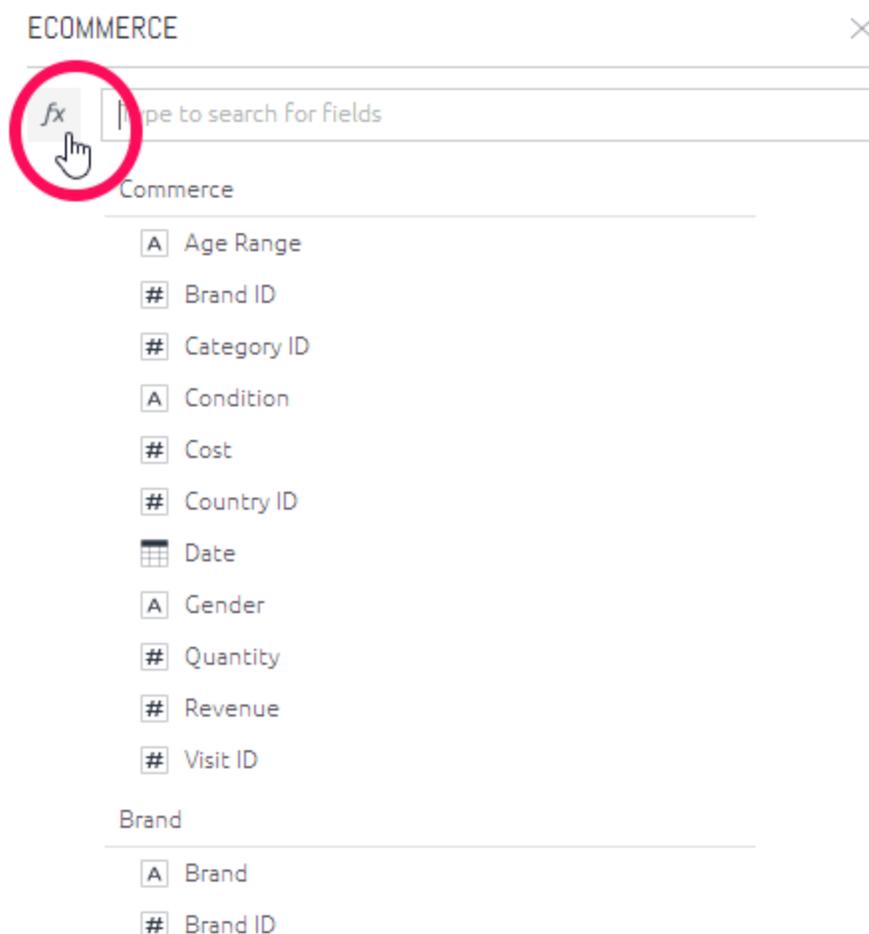
The Data Browser enables you to define formulas (freeform expressions) that define the values and filters of a widget.

A rich variety of functions are provided for you to use in the formula that you define.

#### To define a formula:

1. Open the formula editor in the Data Browser:
  - For a new widget, click Select Data, and then  .

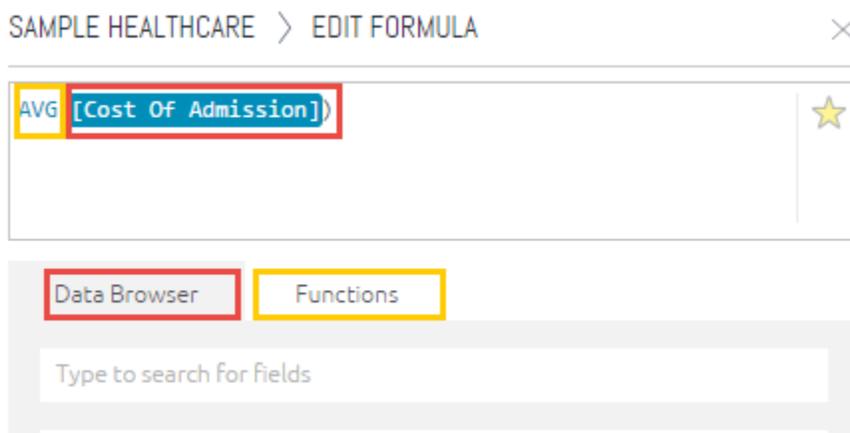
- For an existing widget, click on the edit formula button  .



The Data Browser then changes to display the Formula Editor, which has two tabs: Data Browser and Functions.

- The Data Browser tab provides fields to choose from.
  - The Functions tab lists the functions that you can include in your formula by selecting them. You can read a description of each function in a tooltip by hovering over it.
2. Define the formula as follows:
- From the **Data Browser** tab, select one or more fields.

- From the **Functions** tab, select the required functions.



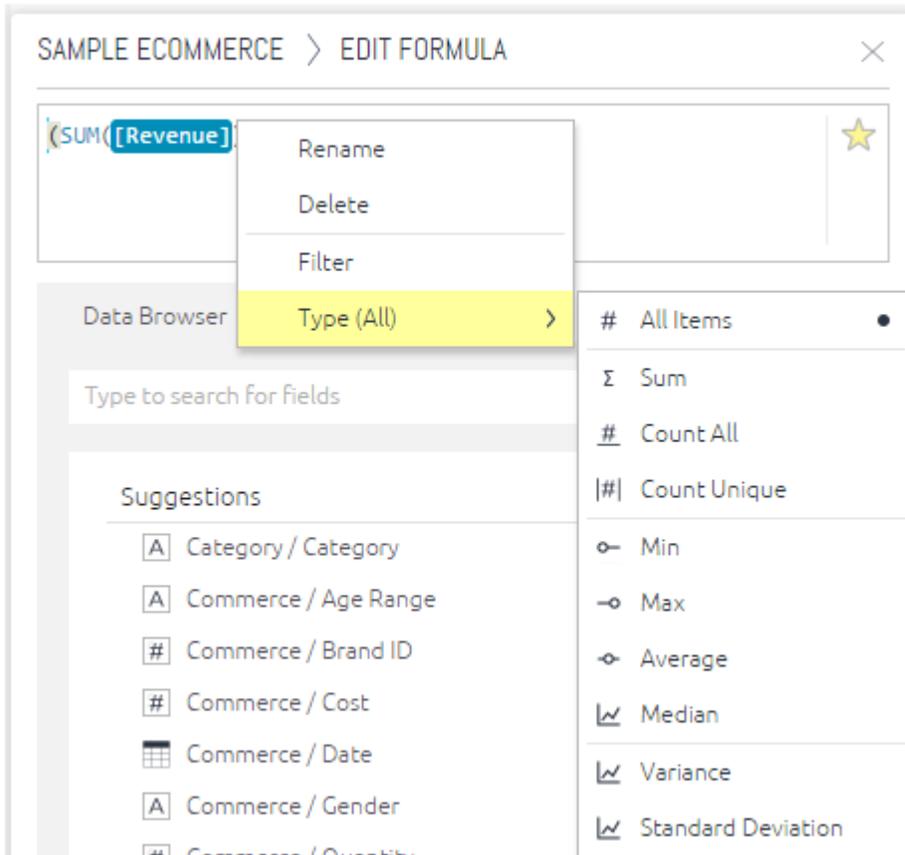
3. Type in the required parts of the formula. To see examples, see [Formulas Based on Criteria and Conditions](#), and [Functions to Build Formulas](#).
4. Click **OK**.

### To edit a formula:

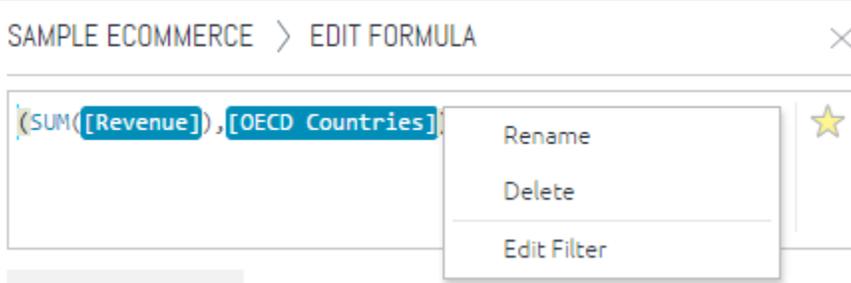
You can easily edit formulas using the right-click options. They include:

- **Rename:** Rename the formula, for example, give a name that represents a real-life task or expected result from the formula, or include in the name filters that you have added to the formula.
- **Filter:** Add filters to the formula.
- **Type:** Change the default aggregation method, for example, from Sum to Average.

The following image shows the right-click options.



The following image shows right-click options for a filtered formula. Read more [below](#).



### Reusing Formulas

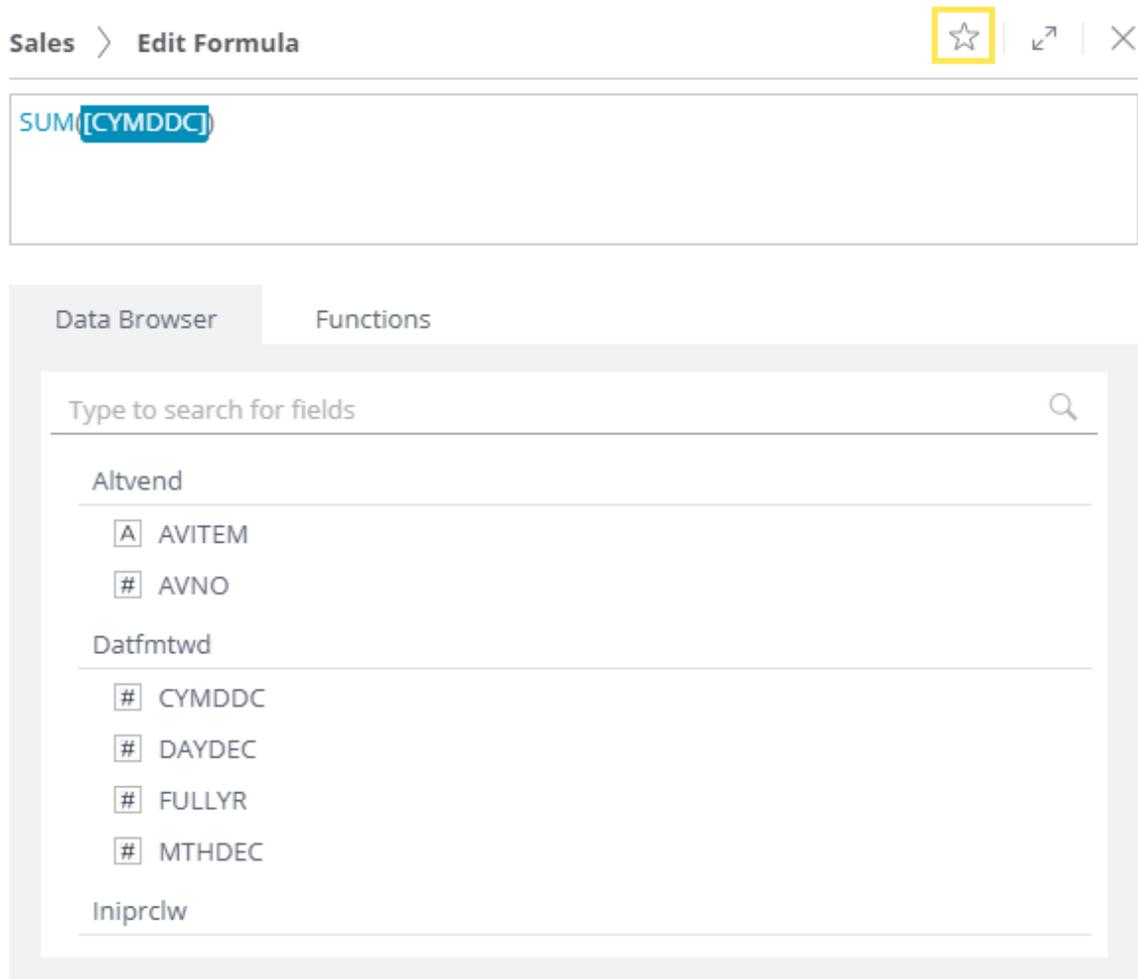
This topic describes how you can reuse formulas that you have previously marked as a favorite (starred).

Formulas are saved per ElastiCube.

**Important:** Changing a starred formula **does not** affect previous uses of that formula. These will continue to use the old version of the formula. Only future uses of the starred formula will implement your latest formula.

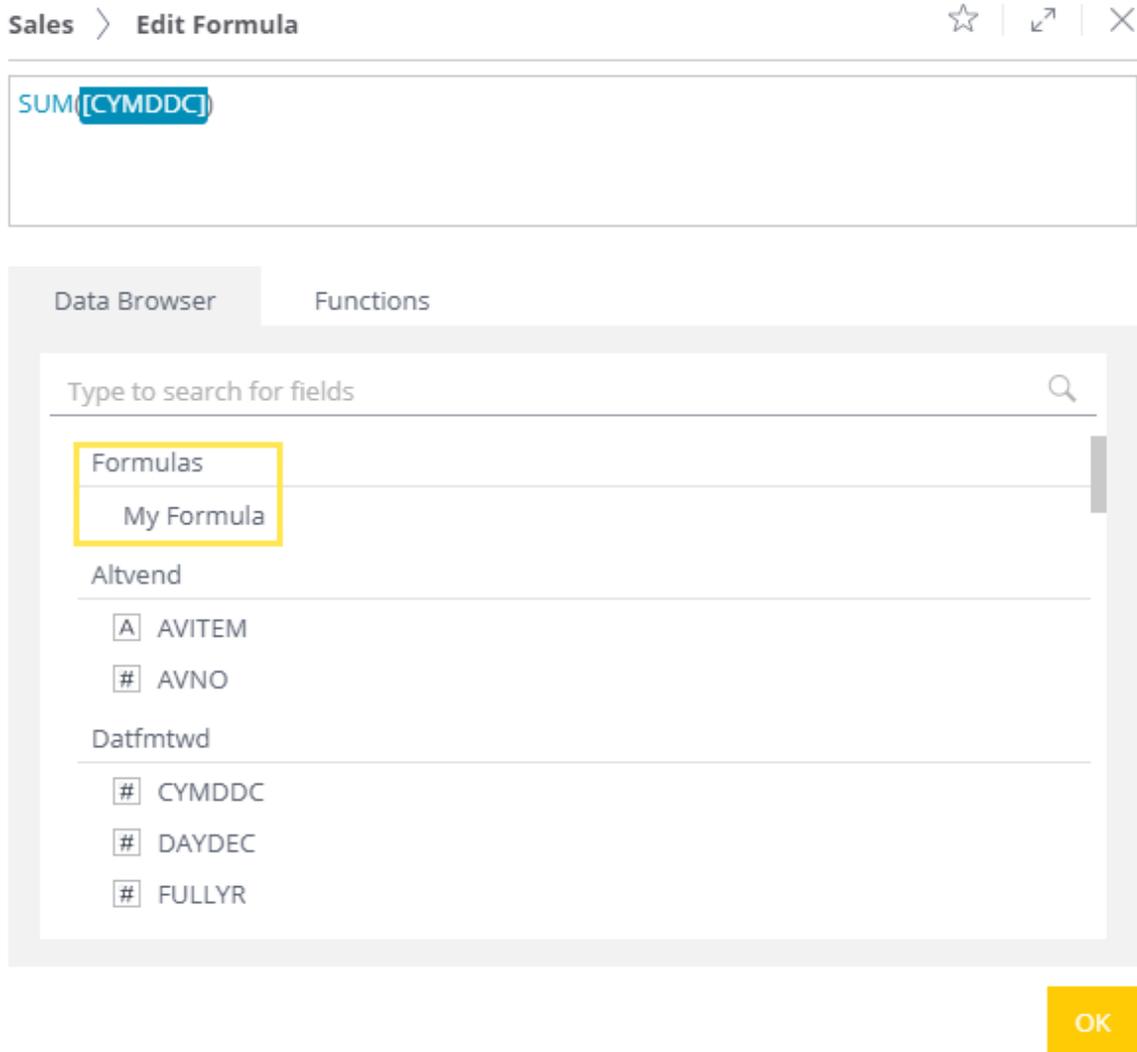
**To mark a formula as a Favorite:**

1. While defining a formula, click the Favorite (Star) button.
2. Enter a name for this Formula.
3. Click **OK**.



**To reuse a favorite formula:**

- Favorite Formulas appear in the Data Browser under the title Formulas, as shown below. Simply select it to use it.



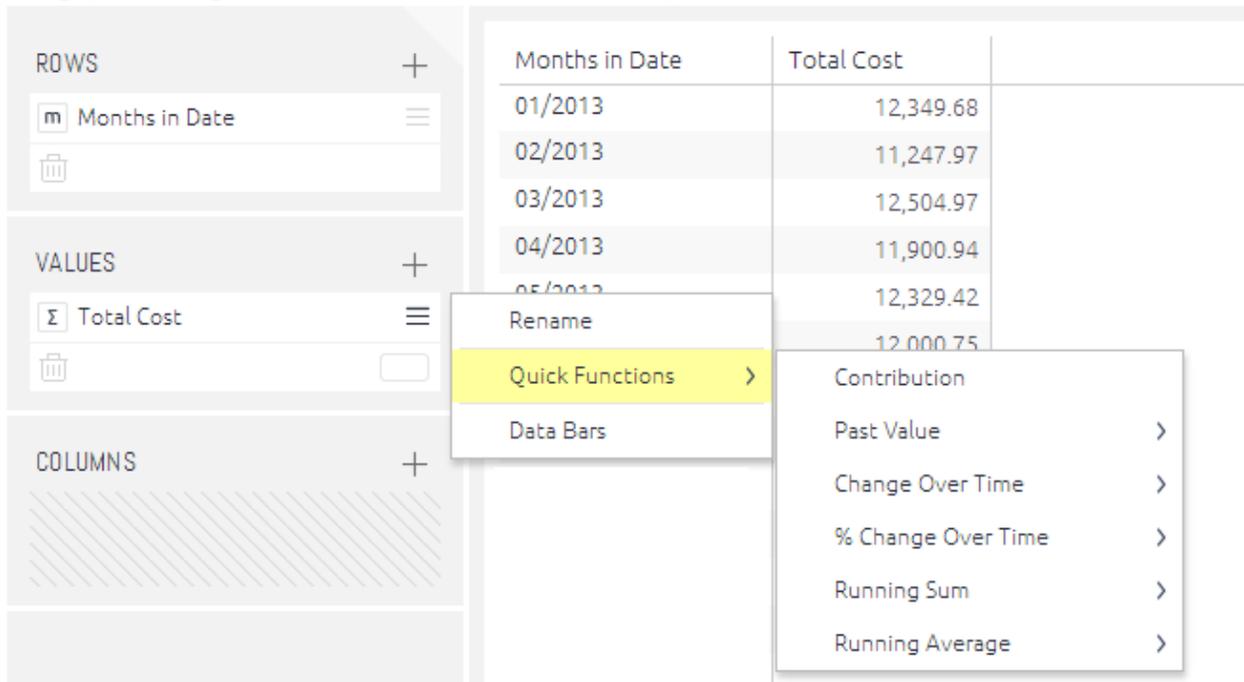
### Quick Functions

Just like [reusing formulas](#), quick functions is another feature to make working with formulas easier. The Widget Designer provides a variety of predefined commonly used functions that you can easily apply in the Data Browser when selecting a Value to be included in a widget.

Quick Functions instantly add a time dimension to any existing value and formula. These functions include calculations for past values, change over time, contribution and running totals. Quick Functions include all the Time Functions

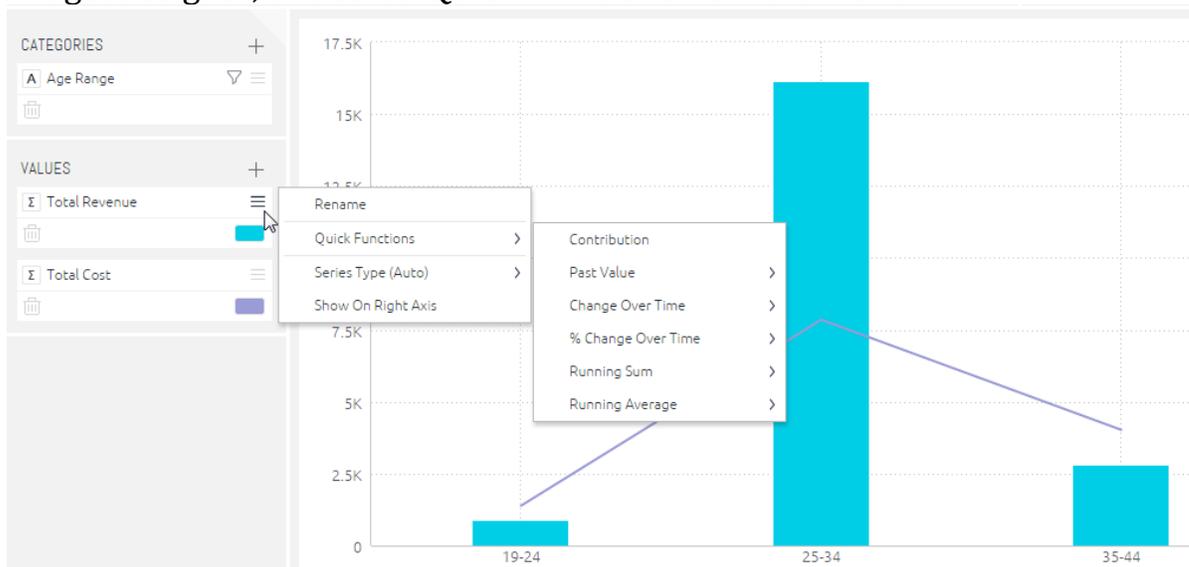
previously discussed but they can only be accessed by clicking on a formula that is already present in a widget.

A simple example of a Quick Function is a finance manager who is reviewing total costs per month but needs to track the accumulated annual costs. They can simply use a quick function to calculate the year to date total for costs.



### To use a quick function:

1. Hover and click on the menu icon of a numeric field in the data panel of the widget designer, and select Quick Functions from the menu:

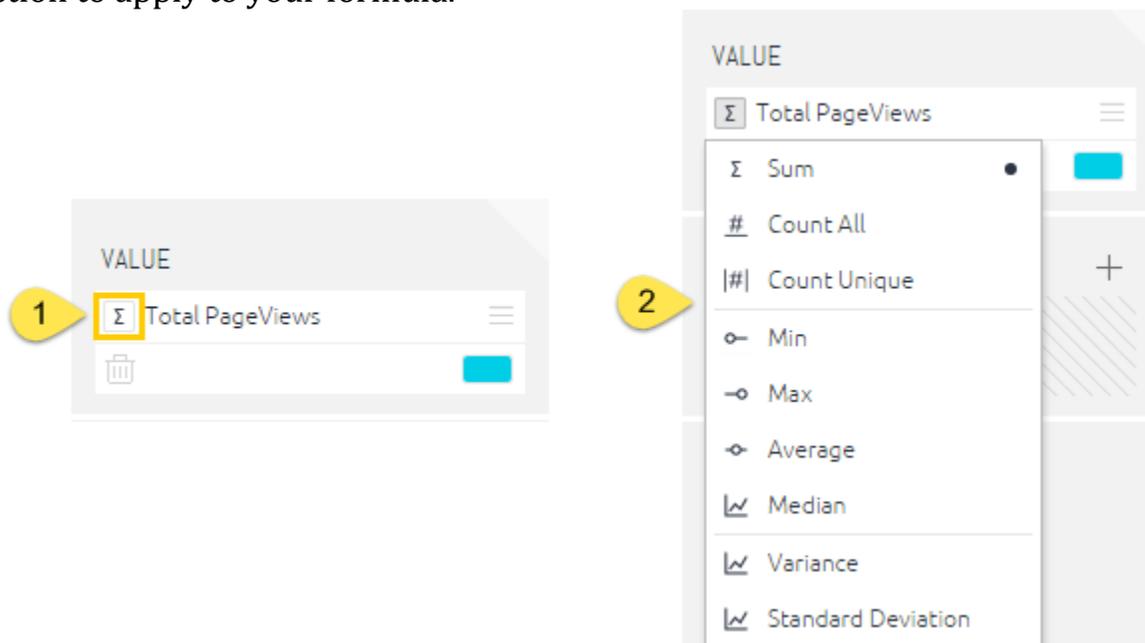


- A list of commonly used functions is displayed.
2. Select a function. The widget will be updated immediately.

# Adding Aggregate Functions

You can also add aggregate functions to your formula without opening the formula editor.

Click on the Value icon to open a list of aggregate functions, and select the function to apply to your formula.



## Starring Formulas with Quick functions

A more complex example uses starring with multiple quick functions. Let's say a finance manager also wants to compare year to date costs to the same period for the previous year. They can first apply the year to date function to total cost and then save it to the formula repository. They can then add the saved year to date formula but apply another quick formula for past values which will perform the same calculation but on data from the previous year.

**Note:** Starred (favorite) formulas will be shared with other users.

### Creating Formulas Based on Criteria and Conditions (Filters)

Often formulas must take into account specific criteria. To do this Sisense provides a feature called **Measured Value**, which like the SUMIF function in Excel, only performs a calculation when the values meet a set of criteria.

Criteria for Measured Values may be based on any logical operators in a filter.

#### *Measure Value Syntax:*

(Measure, Scope1 , Scope2...ScopeN)

#### *Parameters*

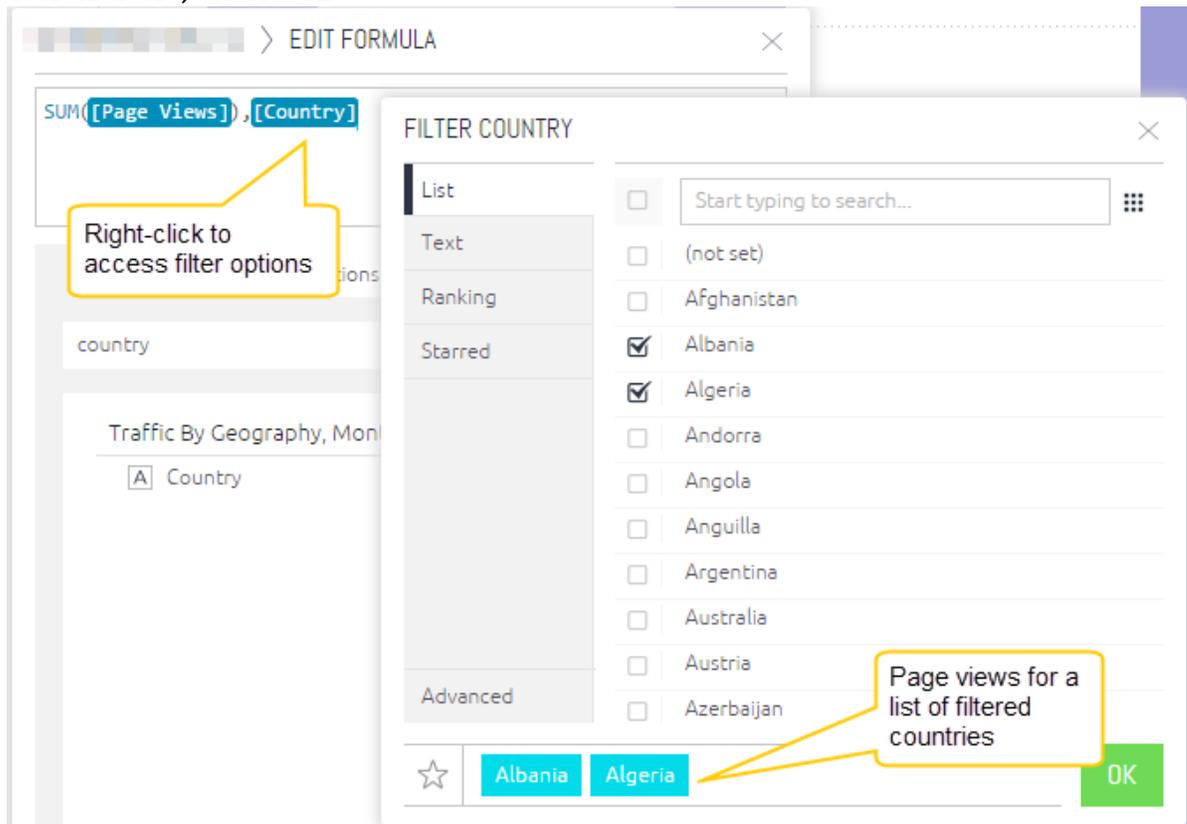
Measure: A field measure or formula.

Scope: A filter including *Value, Text, List, Ranking and Time filters*.

### **To filter the formula:**

1. In the Data Browser, create your formula from the Data Browser and Functions, as explained in [Formula Editor](#).
2. Add the field (criteria) by which you want to filter the formula. Right-click the field and select **Filter**.

- You can then filter the formula by listed items, text options, ranking, etc. When done, click **OK**.



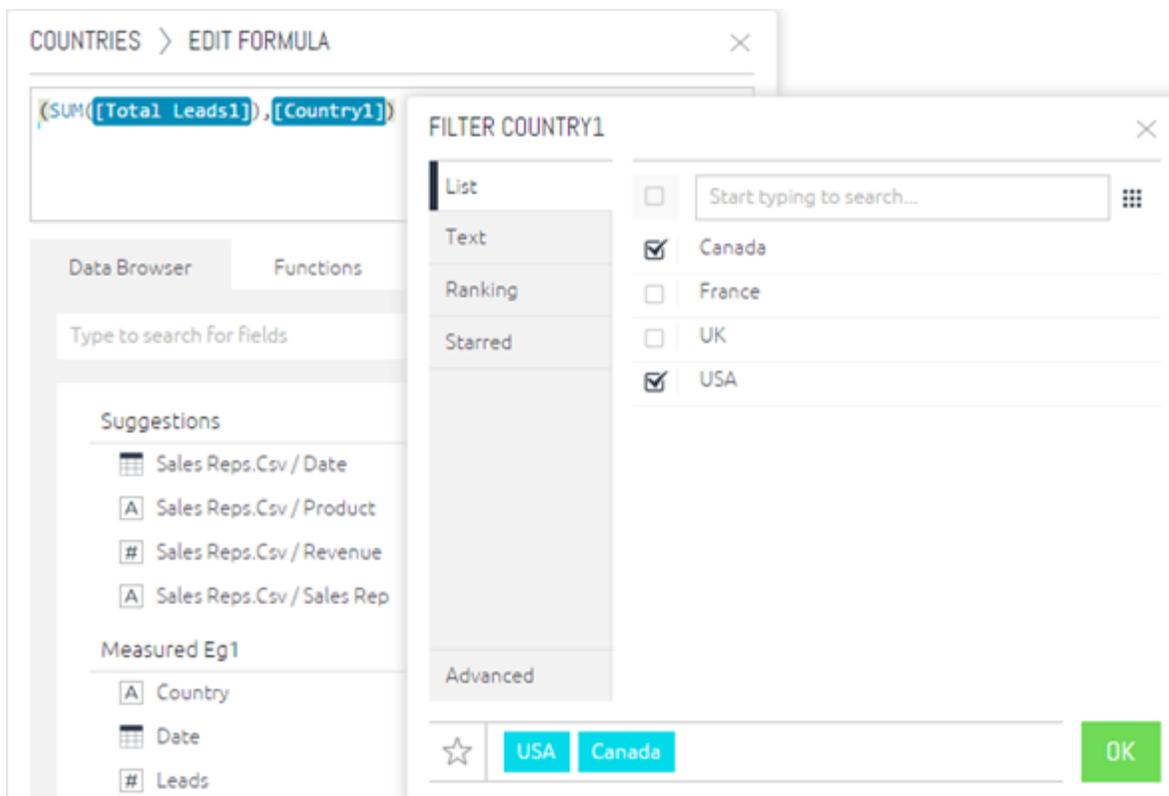
A simple example of Measured Value is the use of a list filter. A marketing team may need to count leads generated for a specific region such as North America. Even if leads come from many different countries, the measured value calculates leads generated only when the lead originates from the United States or Canada.

Country	# Leads
USA	15
France	20
Canada	5
UK	30

*Example 1: Measured Value with List Filter*  
 (Sum(Leads), (List Filter: Country = United States, Canada))

*Result*  
 15+5 = 20

The above example as defined in the Formula Editor.



A more sophisticated case is the **use of a ranking filter**, for example a sales team may want to track the contribution of best-selling products to total revenue. However, what constitutes a popular product may change over time. A measured value can be created for sales which includes a condition that only shows sales for the top products for any month. This simultaneously filters the data but also takes into account changes in what classifies as a top product over time.

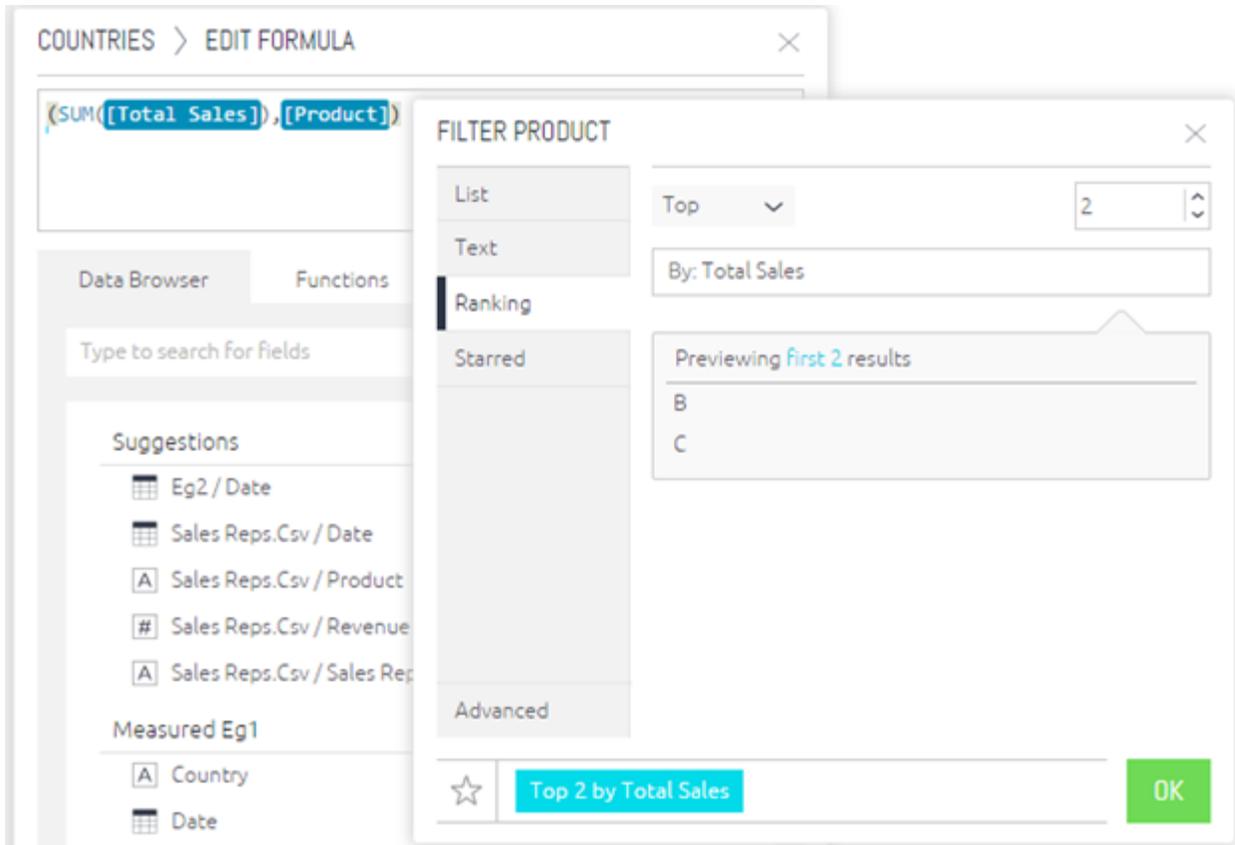
Month	Product	Sales
Jan	A	10
Jan	B	5
Jan	C	15
Feb	A	1
Feb	B	25
Feb	C	5

*Example 2: Measured Value with Top Ranking Filter*  
 (Sum(Sales), (Top Ranking Filter: Top 2 Products by Sales))

*Results*

Month	Total Sales Top 2 Products
Jan	(A/C) 10 + 15 = 25
Feb	(B/C) 25 + 5 = 30

The above example as defined in the Formula Editor.



Measured Values are a powerful feature to take into account business logic and quickly perform calculations only when a specific set of criteria is met.

**Note:** If your widget is filtered using measured values, then the measured value will override any other widget or dashboard filters you have for the same fields.

# Calculating Contributions Using the ALL Function

The All() function returns the total amount for a dimension, and can be used for various use cases. In the following example, we will use the All function to calculate how much each country contributed towards the total cost of a campaign.

Our final widget includes the following information:

Country	Total Cost	Total Cost per Countries	Contribution
Australia	9,643.09	283,755.53	3.40%
Brazil	9,638.80	283,755.53	3.40%
China	9,607.4	283,755.53	3.39%
England	19,492.8	283,755.53	6.87%
France	19,440.59	283,755.53	6.85%
Germany	19,377.61	283,755.53	6.83%
Greece	9,842.84	283,755.53	3.47%
India	9,603.16	283,755.53	3.38%
Kazakhstan	9,899.33	283,755.53	3.49%
Nigeria	9,764.44	283,755.53	3.44%
Norway	9,681.69	283,755.53	3.41%
Portugal	9,919.33	283,755.53	3.50%
South Africa	19,211.15	283,755.53	6.77%
Ukraine	9,693.03	283,755.53	3.42%
United States	99,241.60	283,755.53	34.97%
Vietnam	9,698.67	283,755.53	3.42%
Grand Total	283,755.53	283,755.53	100%

**Step 1:** The second column above represents a formula that sums up the total cost for all countries and does not represent the breakdown per country. The formula includes the calculation (total cost) followed by the all function (filter), followed by the dimension (country) in parenthesis. It looks like this:

```
sum([Total Cost], all([Country]))
```

We can save (star) the above formula and call it Total cost for Countries, which will be used in the next step.

**Step 2:** We can now use the above formula in another formula to calculate the contribution, like this:

```
SUM([Cost])/[Total Cost for Countries]
```

The result is the third column above (plus formatting the results as percentages).

### Function Reference

This page contains a list of all the functions you can use in Sisense's formula editor.

# Statistical Functions

## Average

`Avg(<numeric Field>)`

Calculates the mean average of the given values.

For example - `AVG(Score)` will calculate the mean average of the given scores.

`Avg(<group by field>, <aggregation>)`

Calculates the average of the given aggregation grouped by another field.

For example - `Avg( Product, Total Sales)` will calculate the average of the total sales per product.

## Contribution

Contribution (<numeric field>)

Calculates the percentage of total.

For example – Contribution( Total Sales ) will calculate the percentage of total sales per group (for example per day or per product) out of total sales (for all days or all products).

## Correlation

`CORREL(<Numeric Field a>, <Numeric Field b>)`

Returns the correlation coefficient of two numeric fields.

For example – `CORREL(Revenue, Cost)` will return the correlation between revenue and cost.

`CORREL(<group by field>, <aggregation a>, <aggregation b>)`

Returns the correlation coefficient of two fields aggregations grouped by another field.

For example – `CORREL(Products, AVG(Revenue), AVG(Cost))` will return the correlation between the average of revenue and cost per product.

## Count

Count(<Numeric Field>)

Counts the number of unique values withing the given values.

## Count All

DupCount (<Numeric Field>)

Returns the actual item count of the given list of items, including duplicates.

## Covariance (Population)

`COVARP(<Numeric Field a>, <Numeric Field b>)`

Returns the population covariance of <Numeric Field a> and <Numeric Field b>.

For example – `COVARP(Revenue, Cost)` will return the population covariance of revenue and cost.

`COVARP(<group by field>, <aggregation a>, <aggregation b>)`

Returns the population covariance of two fields aggregations grouped by another field.

For example – `COVARP(Products, AVG(Revenue), AVG(Cost))` will return the population covariance of the average revenue and the average cost per product.

## Covariance (Sample)

`COVAR(<Numeric Field a>, <Numeric Field b>)`

Returns the sample covariance of <Numeric Field a> and <Numeric Field b>.

For example – `COVAR(Revenue, Cost)` will return the sample covariance of revenue and cost.

`COVAR(<group by field>, <aggregation a>, <aggregation b>)`

Returns the sample covariance of two fields aggregations grouped by another field.

For example – `COVAR(Products, AVG(Revenue), AVG(Cost))` will return the sample covariance of the average revenue and the average cost per product.

## Exponential Distribution

`EXPONDIST(<numeric value>, <lambda>, <Cumulative  
(true/false)>)`

Returns the exponential distribution for a given value and a supplied distribution parameter lambda. Cumulative: TRUE = Cumulative distribution function, FALSE = Probability density function. For example – `EXPONDIST( Count(Leads), 2, False )` will return the exponential distribution density of the number of leads per country where lambda is 2.

## Intercept

`INTERCEPT(<field>, <numeric value>)`

Returns the intercept of the linear regression line through a supplied series of x- and y- values.

For example – `INTERCEPT(Date.Quarter, Total Sales)` will return the intercept of the regression line that represents the trend over quarter of the sum of sales.

## Largest

LARGEST(<Numeric Field>, <k>)

Returns the k-th largest value in a field.

## Maximum

Max(<Numeric Field>)

Returns the maximum value among the given values.

## Median

`MEDIAN( <Numeric Field> )`

Calculates the median of the given values. The median of a set of data is the middlemost number in the set. The median is also the number that is halfway into the set.

## Minimum

`Min(<Numeric Field>)`

Returns the minimum value among the given values.

## Mode

`MODE (<Numeric Field>)`

Returns the most frequently occurring value from the column.

## Normal Distribution

`NORMDIST(<Numeric Field>, <Mean>, <Standard Deviation>, <Cumulative (true/false)>)`

Returns the standard normal distribution for a given value, a supplied distribution mean and standard deviation. Cumulative: TRUE = Cumulative Normal Distribution Function, FALSE = Normal Probability Density Function. For example – `NORMDIST(Score, ( Mean(Score), All(Score)), ( STDEV(Score), All(Score) ), False )` will return the normal probability density of a given score.

## Percentile

`PERCENTILE (<Numeric Field>, <k>)`

Returns the k-th percentile value from the given field.

k is any number between 0..1 (inclusive).

## Poisson Distribution

POISSONDIST( <numeric value>, <mean>, <Cumulative  
(true/false)>)

Returns the poisson distribution for a given value and a supplied distribution mean. Cumulative: TRUE = Cumulative distribution function, FALSE = Probability mass function.

For example – POISSONDIST( Score, ( Mean(Score), All(Score) ), ( STDEV(Score), All(Score) ), False ) will return the poisson probability density of a given number of sales

# Quartile

QUARTILE(<Numeric Field>, <k>)

Returns the k-th quartile for the given field.

- k = 0 returns the Minimum value
- k = 1 returns the first quartile (25th percentile)
- k = 2 returns the Median value (50th percentile)
- k = 3 returns the third quartile (75th percentile)
- k = 4 returns the Maximum value

## Rank

RANK(<numeric value>, [DESC/ASC], [Rank Type], [<group by field 1>, ... , <group by field n>])

Returns the rank of a value in a list of values.

[DESC/ASC] – Optional. By default sort order is descending.

[Rank Type] – Optional. By default the type is standard competition ranking (“1224” ranking). Support also modified competition ranking (“1334” ranking), dense ranking (“1223” ranking) and ordinal ranking (“1234” ranking).

[<Group by field 1>, ... , <Group by field n>] – Optional. Rank partitions fields.

For example – RANK(Total Cost, “ASC”, “1224”, Product, Years) will return the rank of the total annual cost per each product were sorted in ascending order.

## Running Sum (RSUM)

RSUM ( <numeric value> ), RSUM ( <numeric value> ,  
<continuous> )

Returns the running total of the measure by the defined dimension according to the current sorting order in the widget.

By default, RSUM accumulates a measure by the sorting order of the dimension. To accumulate by another order, the relevant measure should be added as an additional column and sorted.

<continuous> is a boolean value that that accumulates the sum continuously when there are two or more dimensions. The default value is False.

**Note:** Filtering the RSUM column by Values, filters the dimensions and recalculates the RSUM from the first filtered value.

## Skewness (Population)

`SKEWP(<numeric value>)`

Returns the skewness of the distribution of a given value in the population.

For example – `SKEWP(Revenue)` will return the skewness of the distribution of revenue in the population.

## Skewness (Sample)

`SKEW(<numeric value>)`

Returns the skewness of the distribution of a given value.

For example – `SKEW(Revenue)` will return the skewness of the distribution of revenue.

## Slope

`SLOPE(<field>, <numeric value>)`

Returns the slope of the linear regression line through a supplied series of x- and y- values.

For example – `SLOPE(Date.Quarter, Total Sales)` will return the slope of the regression line that represent the trend over quarter of the sum of sales.

## Standard Deviation (Population)

STDEVP( <Numeric Value> )

Returns the Standard Deviation of the given values (Population). Standard deviation is the square root of the average squared deviation from the mean. The standard deviation of a population gives researchers the amount of dispersion of data for an entire population of survey respondents.

## Standard Deviation (Sample)

STDEV( <Numeric Value> )

Returns the Standard Deviation of the given values (Sample). Standard deviation is the square root of the average squared deviation from the mean. A standard deviation of a sample estimates the amount of dispersion in a given data set, based on a random sample.

## T Distribution

`TDIST( <numeric value x>, <degrees_freedom>, <Cumulative (true/false)> )`

Returns the student's T-distribution for a given value and a supplied number of degrees of freedom (must be  $\geq 1$ ). Cumulative: TRUE = Cumulative Distribution Function, FALSE = Probability Density Function.

For example - `TDIST( Score, 3, TRUE )` will return the student's T-distribution of a given score, with 3 degrees of freedom.

## Variance (Population)

`VARP( <Numeric Value> )`

Returns the Variance of the given values (Population). Variance (Sample) is the average squared deviation from the mean, based on an entire population of survey respondents.

## Variance (Sample)

`VAR( <Numeric Value> )`

Returns the Variance of the given values (Sample). Variance (Sample) is the average squared deviation from the mean, based on a random sample of the population.

# Mathematical Functions

# Absolute

Abs (<Numeric value>)

Returns the absolute value of the given value.

For example - ABS(Cost), where the absolute result for the value '2' or '-2' is '2'.

## Acos

`ACOS(<numeric value>)`

Returns the angle, in radians, whose cosine is the given numeric expression. Also referred to as arccosine.

For example - `ACOS(Total Revenue)` will return the angle, in radians, whose cosine is the given total revenue.

## Asin

`ASIN(<numeric value>)`

Returns the angle, in radians, whose sine is the given numeric expression. Also referred to as arcsine.

For example - `ASIN(Total Revenue)` will return the angle, in radians, whose sine is the given total revenue.

## Atan

`ATAN(<numeric value>)`

Returns the angle in radians whose tangent is the given numeric expression. Also referred to as arctangent.

For example – `ATAN(Total Revenue)` will return the angle in radians whose tangent is the given total revenue.

## Ceiling

`CEILING(<numeric value>)`

Returns number rounded up, away from zero, to the nearest multiple of significance.

For example – `CEILING(Cost)`, where the result of '83.2' rounded up is '84'.

## Cos

`COS (<numeric value>)`

Returns the trigonometric cosine of the given angle (in radians).

For example – `COS(Average Angle)` will return the trigonometric cosine of the average angle.

## Cosh

`COSH(<numeric value>)`

Returns the hyperbolic cosine of the given value.

For example – `COSH(Total Revenue)` will return the hyperbolic cosine of the total revenue.

## Cot

`COT(<numeric value>)`

Returns the trigonometric cotangent of the given angle (in radians).

For example – `COT(Average Angle)` will return the trigonometric cotangent of the average angle.

## Exp

`EXP(<numeric value>)`

Returns the exponential value of the given value.

For example – `EXP(Sales)` will return the exponential value of sales.

## Floor

`FLOOR(<numeric value>)`

Returns number rounded down, toward zero, to the nearest multiple of '1'.

For example – `FLOOR(Revenue)`, where the result of '88.6' rounded down is '88'.

## Ln

`LN(<numeric value>)`

Returns the base-e logarithm of the given value.

For example – `LN(Cost)` will return the base-e logarithm of cost.

## Log10

`LOG10(<numeric value>)`

Returns the base-10 logarithm of the given value.

For example – `LOG10(Revenue)` will return the base-10 logarithm of revenue.

## Mod

`MOD(<numeric value>, divisor)`

Returns the remainder after a number is divided by a divisor.

For example – `MOD(Cost, 10)`, where the remainder of '255' divided by '10' is '5'.

## Power

`Power(value, power)`

Returns the results of the given value raised to a supplied power.

For example – `POWER(Revenue, 2)` will return revenue raised by the power of 2.

## Quotient

`QUOTIENT(<numeric value>, divisor)`

Returns the integer portion of a division.

For example – `QUOTIENT(Cost, 2)`, where the integer portion of '5' divided by '2' is '2'.

## Round

`ROUND(<numeric value>, num_digits)`

Returns number rounded to a specified number of digits.

For example – `ROUND(Revenue, 2)` will return the revenue rounded to two decimal places.

## Sin

`SIN(<numeric value>)`

Returns the trigonometric sine of the given angle (in radians).

For example – `SIN(Average Angle)` will return the trigonometric sine of the average angle.

## Sinh

`SINH(<numeric value>)`

Returns the hyperbolic sine of the given value.

For example – `SINH(Total Revenue)` will return the hyperbolic sine of the total revenue.

## Square root

`SQRT(<Numeric value>)`

Returns the square root of the given value.

For example – `SQRT(Cost)` will return the square root of cost.

## Sum

Sum(<Numeric Field>)

Calculates the total of the given values.

## Tan

`TAN(<numeric value>)`

Returns the trigonometric tangent of the given angle (in radians).

For example – `TAN(Average Angle)` will return the trigonometric tangent of the average angle.

# Tanh

TANH(<numeric value>)

Returns the hyperbolic tangent of the given value.

For example – TANH(Total Revenue) will return the hyperbolic tangent of the total revenue.

# **Time Related Functions**

## Day Difference

DDiff( <Start Time>, <End Time> )

Returns the difference between <Start Time> and <End Time> in days.

## Past Month Difference

```
DiffPastMonth( <numeric value> )
```

Returns the difference between this month's data and the data from the previous month. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

```
DiffPastMonth( <Total Sales> )
```

Returns the difference between this month's sales and previous month's sales, for the displayed time resolution. For example, for day resolution: (sales in current day - sales in same day one month back).

## Past Period Difference

`DiffPastPeriod( <numeric value> )`

Returns the difference between this period's data and the data from the previous period. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`DiffPastPeriod( <Total Sales> )`

Returns the difference between this period's sales and previous period's sales, for the displayed time resolution. Formula: (current value - compared value).

## Past Quarter Difference

`DiffPastQuarter( <numeric value> )`

Returns the difference between this quarter's data and the data from the previous quarter. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`DiffPastQuarter( <Total Sales> )`

Returns the difference between this quarter's sales and previous quarter's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one quarter back).

## Past Week Difference

`DiffPastWeek( <numeric value> )`

Returns the difference between this week's data and the data from the previous week. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`DiffPastWeek( <Total Sales> )`

Returns the difference between this week's sales and previous week's sales, for the displayed time resolution. For example, for day resolution: (sales in current day - sales in same day one week back).

## Past Year Difference

```
DiffPastYear( <numeric value> )
```

Returns the difference between this year's data and the data from the previous year. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

```
DiffPastYear( <Total Sales> )
```

Returns the difference between this year's sales and previous year's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one year back).

## Growth

Growth( <Numeric Value> )

Calculates growth over time. The time dimension to be used is determined by the time resolution in the widget/dashboard.

Formula: (current value - compared value) / compared value.

For example:

- If this month your value is 12, and last month it was 10, your Growth for this month is 20% (0.2).  
Calculation:  $(12 - 10) / 10 = 0.2$
- If this year your value is 80, and last year it was 100, your Growth for this year is -20% (-0.2).  
Calculation:  $(80 - 100) / 100 = -0.2$

## Growth Rate

`GrowthRate( <Numeric Value> )`

Calculates growth rate over time. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

- If this month your value is 12, and last month it was 10, your Growth Rate for this month is  $12/10 = 120\%$  (1.2).  
Calculation:  $12 / 10 = 1.2$
- If this year your value is 80, and last year it was 100, your Growth for this year is  $80/100 = 80\%$  ( 0.8).  
Calculation:  $80 / 100 = 0.8$

## Growth Past Month

`GrowthPastMonth( <Numeric Value> )`

Calculates the growth from the past month to the current month. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`GrowthPastMonth([Total Sales])`

Calculates the difference between this month's sales and previous month's sales, for the displayed time resolution. For example, for day resolution: (sales in current day - sales in same day one month back) / sales in same day one month back.

## Growth Past Quarter

`GrowthPastQuarter( <Numeric Value> )`

Calculates the growth from the past quarter to the current quarter. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`GrowthPastQuarter([Total Sales])`

Calculates the difference between this quarter's sales and previous quarter's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one quarter back) / sales in same month one quarter back.

## Growth Past Week

`GrowthPastWeek( <Numeric Value> )`

Calculates the growth from the past week to the current week. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`GrowthPastWeek([Total Sales])`

Calculates the difference between this week's sales and previous week's sales, for the displayed time resolution. For example, for day resolution: (sales in current day - sales in same day one week back) / sales in same day one week back.

## Growth Past Year

`GrowthPastYear( <Numeric Value> )`

Calculates the growth from the past year to the current year. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`GrowthPastYear([Total Sales])`

Calculates the difference between this year's sales and previous year's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one year back) / sales in same month one year back.

## Hour Difference

`HDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in hours.

## Prev

Prev( <Time Field> [, <N>] )

Returns the Time period Member in <Time Field> which is N periods back from the current Member. This function only works as a scope function and not by itself.

For example – This formula will return the numeric value 2 months ago:

(<Numeric Value>, Prev(<Month Field>, 2))

## Minute Difference

`MnDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in minutes.

## Month Difference

`MDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in months.

Returns whole numbers.

## Month to Date Average

MTDAvg( <Numeric Value> )

Returns the running average starting from the beginning of the month up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is quarters or years.

## Month to Date Sum

MTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the month up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is quarters or years.

## Next

`Next( <Time Field> [, <N>] )`

Returns the Time period Member in <Time Field> which is N periods after the current Member. This function only works as a scope function and not by itself.

For example - This formula will return the numeric value 2 months ahead of now:

`(<Numeric Value>, Next(<Month Field>, 2))`

## Now

Now(<Day from Date field>)

Returns the value for the current time period. Supports day, month, quarter or year. The Now function receives a date dimension and its level and returns all the members in that dimension which match the current query execution time.

**Note:** This function only works as a scope function and not by itself.

The following example will return the value for the current day.

```
([Total Sales], Now([Days in Datefield]))
```

## Past Year

`PastYear( <Numeric Value> )`

Calculates the value for the same period in the past (previous) year.

For example:

- If you're looking at a specific day, you will see the value of the same day one year back.
- If you're looking at a specific month, you will see the value of the same month one year back.

**Note:** When using the Past Year function in a weeks table and using a week filter, no results are returned.

## Past Quarter

`PastQuarter( <Numeric Value> )`

Calculates the value for the same period in the past (previous) quarter.

For example:

- If you're looking at a specific day, you will see the value of the same day one quarter back.
- If you're looking at a specific month, you will see the value of the same month one quarter back.

## Past Month

PastMonth( <Numeric Value> )

Calculates the value for the same period in the past (previous) month.

For example:

- If you're looking at a specific day, you will see the value of the same day one month back.

## Past Week

`PastWeek( <Numeric Value> )`

Calculates the value for the same period in the past (previous) week.

For example:

- If you're looking at a specific day, you will see the value of the same day one week back.

## Past Day

PastDay( <Numeric Value> )

Calculates the value for the same period in the past (previous) day.

For example:

- If you're looking at a specific day, you will see the value of the same day one day back.

## Quarter Difference

`QDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in quarters.

Returns whole numbers.

## Quarter to Date Average

QTD Avg( <Numeric Value> )

Returns the running average starting from the beginning of the quarter up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is years.

## Quarter to Date Sum

QTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the quarter up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is years.

## Range

`range( <Field1>, <Field2> )`

Returns a graphical range selector for a data set where two members of the same dimension and level define the minimum and maximum values of the range.

## Second Difference

`SDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in seconds.

## Week to Date Average

WTDAvg( <Numeric Value> )

Returns the running average starting from the beginning of the week up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is years, quarters, or months.

## Week to Date Sum

WTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the week up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is years, quarters, or months.

## Year Difference

`YDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in years. Returns whole numbers.

## Year to Date Average

YTDAvg( <Numeric Value> )

Returns the running average starting from the beginning of the year up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

## Year to Date Sum

YTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the year up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

**Other**

# All

All (<Field>)

Ignores the scope set on the dimension.

## CASE

```
(WHEN <condition> THEN <result_expression> [...] [ELSE  
<result_expression>] END)
```

Returns the result\_expression of the first condition evaluated as true. When no condition is true, else\_expression is returned, if one is defined.

For example, the below function will return '1' when the Total Sales value is between 100 and 1000. It will return '2' if the Total Sales value is above 1000. It will return '3' in any other case (meaning, when Total Sales are below 100).

```
CASE  
WHEN Sum(Sales) < 100 THEN 1  
WHEN Sum(Sales) < 1000 THEN 2  
ELSE 3  
END
```

## IF

IF (<condition>, <numeric expression 1>, <numeric expression 2>)

Returns numeric expression '1' when the condition is true, and expression '2' when the condition is false. Nested conditional statements are supported.

For example, if the number of unique values within the Sales values is larger than 100, the below function will return the Total Sales x 1.1 (sales increase of 10%).

Otherwise - if the number of unique values within the Sales values is lower than 100, the function will return only the Total Sales, without an increase.

```
IF(Count(Sales)>100, Sum(Sales)*1.1, sum(Sales))
```

## IsNull

(<numeric value>)

Returns true if the expression doesn't contain data (Null). Can be used as a condition when writing conditional statements.

## Ordering

`ORDERING (<expression1>, <expression2>)`

Returns the numeric order position of rows sorted into ascending or descending order, breaking ties with further arguments.

The expressions must be aggregated by applying the MIN/MAX functions as in the example below:

```
ORDERING(MIN([Sales Person Name]), MIN([Days in Transaction_Date]), -  
1*Sum([Sales]))
```

## Rdouble

```
RDOUBLE(<R expression>, <numeric value 1>, [<numeric value 2>, ..., <numeric value n>] )
```

Returns a numeric result for a given R expression and a list of numeric values (use 'args[[i]]' in the R expression to reference numeric values parameters).

The R expression is passed to the running Rserve.

The optional ordering expression determines the order in which the rows are sent to R. The argument of the Ordering parameter can be an index in your data source or you can use the ORDERING() function to determine the order of the field. For more information about the ORDERING() function, [click here](#).

For example – RDOUBLE("m <- log(matrix(unlist(args), ncol=2)); kmeans(m,3)\$cluster", [Total Cost], [Total Revenue]) will return the k-means cluster (R expression) of the args: [Total Cost] and [Total Revenue].

```
RDOUBLE(<recycle>, <R expression>, <numeric value 1>, [<numeric value 2>, ..., <numeric value n>] )
```

recycle = TRUE (default) – Results will be cached for unchanged functions and data.

recycle = FALSE – Results will not be cached. Use this option if your R code contains randomness.

## Rint

```
RINT(<R expression>, <numeric value 1>, [<numeric value 2>, ..., <numeric value n>] )
```

Returns an integer result for a given R expression and a list of numeric values (use 'args[[i]]' in the R expression to reference numeric values parameters).

The R expression is passed to the running Rserve.

The optional ordering expression determines the order in which the rows are sent to R. The argument of the Ordering parameter can be an index in your data source or you can use the ORDERING() function to determine the order of the field. For more information about the ORDERING() function, [click here](#).

For example – RINT("m <- log(matrix(unlist(args), ncol=2)); kmeans(m,3)\$cluster", [Total Cost], [Total Revenue]) will return the k-means cluster (R expression) of the args: [Total Cost] and [Total Revenue].

```
RINT(<recycle>, <R expression>, <numeric value 1>, [<numeric value 2>, ..., <numeric value n>] )
```

recycle = TRUE (default) – Results will be cached for unchanged functions and data.

recycle = FALSE – Results will not be cached. Use this option if your R code contains randomness.

## RSUM

RSUM (<numeric value>)

Returns the running total of the measure by the defined dimension, according to the current sorting order in the widget.

RSUM (<numeric value>, <continuous>)

Continuous: TRUE = Accumulates the sum continuously in the case of two dimensions or more. False by default.

### **Building Formulas with Functions**

Functions are operations that perform common types of calculations, and can be used to build formulas. In this topic, you can read about four types of functions. The functions' syntax is explained and examples are provided.

# Combine Data: Aggregate Functions

Aggregations are used to perform mathematical calculations on data. Although this is an essential function Sisense offers an advantage in the ability to run multiple aggregations on several fields simultaneously – this makes it easy to summarize data based on multiple factors.

## *Aggregate Syntax:*

Function(Numeric Field)

## *Parameters*

Function: Sum(), Avg(), Count(), DupCount(), Max(), Min()

Numeric Field: A numeric field or formula.

An example **simple aggregation** is a sales manager who wants to calculate the average sales revenue for each sales rep. They can create a pivot table which shows the sales rep and their average sales revenue.

A more complex example is a **multi-pass aggregation** (or grouping) that is an aggregation that performs multiple calculations simultaneously. Following our first example let's assume the sales manager wants to also see average sales per day for each sales rep. Instead of having to add an additional column for day in the pivot table the manager can create a multi-pass aggregation that first performs a sum of sales per day and then averages the results for each rep. This requires two fields – a day from a date field and the revenue field, as well as two aggregations, sum of sales and average. The result is the sales manager does not need to add a column for days in the pivot.

*Data Set*

Day	Sales Rep	Revenue
1 Jan	John	10
1 Jan	Jane	20
1 Jan	Jane	40
2 Feb	John	30
2 Feb	John	10
2 Feb	Jane	5

*Results Example 1: Avg(Revenue)*

Sales Rep	Avg Revenue
John	$(10+30+10)/3 = 16.67$
Jane	$(20+40+5)/3 = 21.67$

*Results Example 2: Avg(Days, Sum(Revenue))*

Sales Rep	Avg Daily Revenue
John	$(10+(30+10))/2 = 25$
Jane	$((20+40)+5)/2 = 32.5$

The above example as shown in the formula editor.

Sales Rep	Average Revenue	Avg( $([Days in Date]),([Total Revenue])$ )
Jane	21.67	32.5
John	16.67	25

COUNTRIES > EDIT FORMULA

`AVG( $([Days in Date]),([Total Revenue])$ )`

COUNTRIES > EDIT FORMULA

`AVG( $([Days in Date]),([Total Revenue])$ )`

# Summarize Data: Statistical Functions

Descriptive statistics provide meaningful summaries of data and help make more informed decisions. This is particularly important for large data sets where descriptive statistics can help to focus analysis.

*Statistical Syntax:* Function(Numeric Field)

*Parameters*

Function: Median(), Mode(), Largest(), Stdev(), Stdevp(), Varp(), Var(), Quartile(), Percentile()

Numeric Field: A numeric field or formula.

An example of statistical functions is a marketing team that has a large data set on leads generated from various channels and want to understand where to focus their budget. Descriptive statistics can be used to summarize valuable insight about each channel such as the central tendency or median leads generated along with standard deviations to assess typical lead volume.

*Statistical Syntax:* Function(Numeric Field)

*Parameters*

Function: Median(), Mode(), Largest(), Stdev(), Stdevp(), Varp(), Var(), Quartile(), Percentile()

Numeric Field: A numeric field or formula.

# Accumulate Data: Running Total and Average

Often, to measure performance, data must be viewed in a continuous and accumulative format over extended periods such as years, quarters or months. Sisense provides functions to create running totals and averages over standard or custom time periods.

*Accumulate Syntax: Function(Numeric Field)*

*Parameters*

Function: YTD Avg(), QTD Avg(), MTD Avg(), YTD Sum(), QTD Sum(), MTD Sum(), RPSum()/ RPAvg()

Numeric Field: A numeric field or formula.

For example, a support team has a goal to reduce the average monthly cost to resolve open issues. A Year to Date Average can be used to track progress towards reducing the average cost of support.

*Data*

<i>Month</i>	<i>Cost</i>
Jan	30
Feb	35
Mar	40
Apr	20
May	25
Jun	10

*Results Example 1: YTD Avg(Cost)*

<i>Month</i>	<i>YTD Avg Cost</i>
Jan	30
Feb	32.5
Mar	35
Apr	31.25
May	30
Jun	26.67

# Change over Time: Time Functions

Time is critical for business decisions, time functions make it easy to compare outcomes at different points in time, determine growth rates and calculate the time differences. Functions can be set for common time periods such as year, month or day as well as for custom periods.

*Time Syntax: Function(Numeric Field) \**

*Parameters*

Function:

PastYear(), PastQuarter(), PastMonth(), Growth(), GrowthRate(), Contribution(), YDiff(), QDiff(), MDiff(), DDiff(), HDiff(), MnDiff(), SDiff(), Next(), Prev(), DiffPastYear(), DiffPastMonth(), DiffPastQuarter(), DiffPastDay()

Numeric Field: A numeric field or formula.

*\* To calculate time functions require a time filter applied on the dashboard or widget.*

For example, an executive team wants to compare growth in revenue to the same period in the previous year. A Difference in Past Year function can be used to compare past values based on the current month to the same month in the previous year.

<i>Data</i>		<i>Results Example 1: DiffPastYear(Revenue)</i>	
<i>Date</i>	<i>Revenue</i>	<i>Month</i>	<i>Difference to Past Year</i>
01-2013	5	01-2013	
02-2013	10	02-2013	
03-2013	15	03-2013	
01-2014	25	01-2014	25-5 =20
02-2014	5	02-2014	5-10= -5
03-2014	10	03-2014	10-15 =-5

## **Conditional Statements**

Conditional statements are formulas that enable you to create additional categories in your data set. When the condition is met, the new category is added to your visualization without modifying the data source itself.

For example, if you want to calculate only certain values according to conditions you define, you can use a conditional statement to include only the values of interest.

Sisense provides several conditional statements that enable you to evaluate criteria and display only the relevant results in your dashboard.

**Note:** Conditional states work on measures and aggregations only and return numeric values.

# IF

IF (Boolean\_expression, true value, false value)

Evaluates a boolean expression and returns the first value when evaluated as true, and returns the second value when evaluated as false.

For example:

```
(if(Sum(Sales) < 10, 0, Sum(Sales) )
```

Nested IF statements are also supported.

For example:

```
if (Sum(Sales) < 100 , 1 , (if (Sum(Sales) < 1000 , 2, 3)
)
```

# isNull

ISNULL (<numeric value>)

Returns true if the expression doesn't contain data (Null). Can be used as a condition when writing conditional statements.

For example:

```
If (ISNULL(sum(deals)), 0, sum(deals) )
```

# Case

```
CASE  
WHEN Boolean_expression THEN result_expression  
[ ... ]  
[ ELSE else_result_expression ]  
END
```

The CASE function evaluates boolean expressions, and when evaluated as true, returns the corresponding result\_expression. If no match is found, the else\_result\_expression is returned. If there is no default returned and no values match, then Null is returned.

For example:

```
CASE  
WHEN Sum(Sales) < 100 THEN 1  
WHEN Sum(Sales) < 1000 THEN 2  
ELSE 3  
END
```

## Using R in Formulas

[R](#) is a software environment for statistical computing and graphics. Sisense supports the integration of R functions in your formulas.

You can write R code directly in the formula editor, and send fields as parameters. You can also combine R functionality with the regular functions to create advanced and tailored formulas. (For general help with the formula editor and functions, click [here](#).)

Before you can integrate R into your formulas, you must have an R server set up in your organization.

This document will cover the following:

- Understanding How R works in Sisense
- Connecting Sisense to Your R Server
- Using R functions in Formulas
- Guidelines for using R code in Sisense
- Example 1: Simple R Function – Kmeans Clustering
- Example 2: R Logistic Regression via Sisense
- Example 3: Loading an Existing R Model
- Example 4: Building and Saving a R Model using Data from Sisense

# Understanding How R works in Sisense

Within a widget, R code is used and assigned fields as parameters, the following happens:

1. The selected fields are sent fully to R as a list of arrays.
2. The R code is executed on the R server, referencing the fields as needed.
3. The numeric results from R are sent back to the widget for visualization.

Each request from the R server is done in a self-contained namespace, thus R variables and results cannot be reused across requests unless they are loaded from a saved location.

R calculations are computed in the R server, and not in the ElastiCube, this means that for larger data sets results will not return as fast as regular widgets.

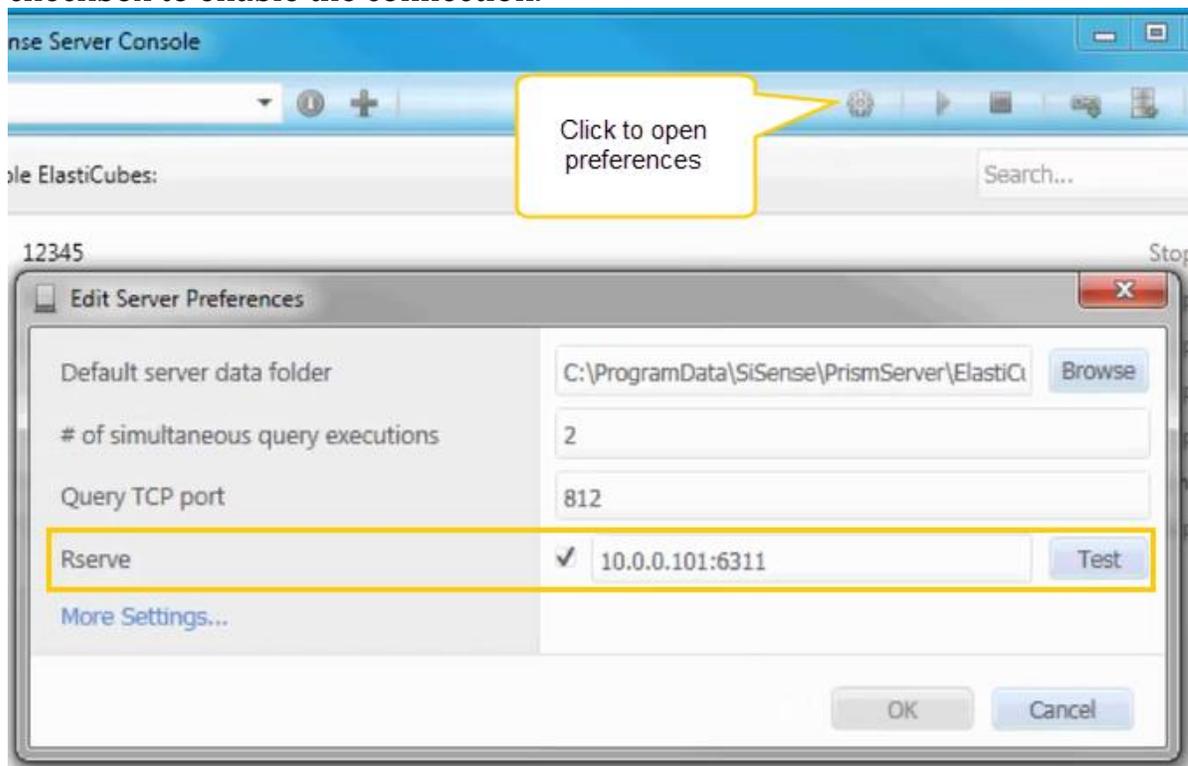
## Running R on Windows

If your R server is running on Windows you will be bound to the concurrency limitations the R distribution has on Windows. This means that concurrent requests to R may return wrong results. Sisense recommends using R on a Linux machine for production environment with concurrent user activity.

# Connecting Sisense to Your R Server

**To connect your Sisense instance to your R server:**

1. Open the Sisense Server Console from the start menu.
2. Click on the Server Preferences icon to open the server settings.
3. In the **Rserve** field, enter the IP address of your R server, and select the checkbox to enable the connection.



You can test the connection by clicking on the **Test** button.

## Installing Rserve

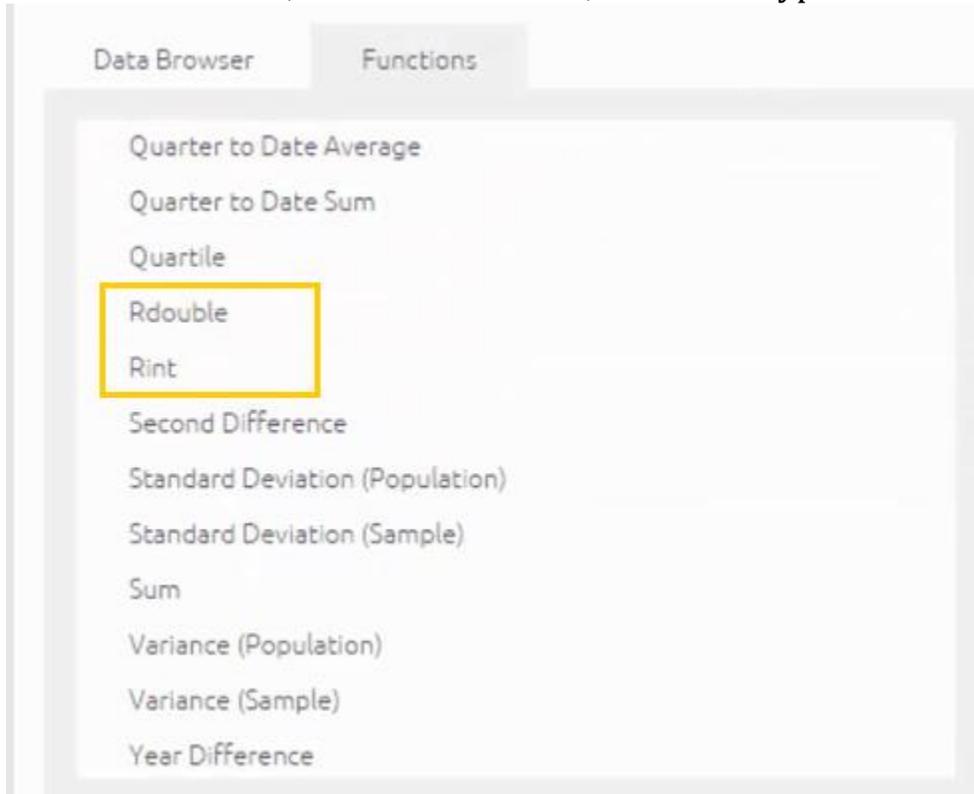
Sisense can only connect to a running Rserve instance. For more information about installing and running Rserve [follow this link](#).

Please note that if you're using RStudio, Rserve still has to be installed.

# Using R functions in Formulas

Using R code in Sisense is achieved by using two dedicated formulas – RINT & RDOUBLE. Depending on the result type of your R code, use the appropriate function within the [formula editor](#).

For either formula, RINT or RDOUBLE, the return type has to be an exact match.



Multiple numeric field values can be passed as parameters to the R functions, which within your R code can be accessed via the 'args' argument (See K-means example below).

## Syntax:

```
RINT([recycle (true)], [<Ordering>], <R expression>,
[<numeric value 1>, ..., <numeric value n>] )
```

```
RDOUBLE([recycle (true)], [<Ordering>], <R expression>,
[<numeric value 1>, ..., <numeric value n>] )
```

Sisense does not support the use of a final ';' in R statements. If you add a semi-colon at the end of your R statement, the code will fail.

### Parameters:

- **Recycle true/false (default = true)** – This is an optional parameter that controls whether the results from R should be recycled (cached), so that consequent queries will not have to be recalculated unless they or the data have changed. Generally, this behavior is automatically managed by the ElastiCube automatically for all functions. However, since R code might have non-deterministic components to it (such as randomness functions or date-specific functions), the ElastiCube cannot rely on a data-set and function that hasn't changed not to return a different result in multiple executions.

By default, the Recycle value is set to true. Use 'off' if your R code contains randomness or other non-deterministic content.

- **Ordering** – This is an optional parameter that defines the sort order in which numeric data is sent to R. The argument of the Ordering parameter can be an index in your data source or you can use the ORDERING() function to determine the order of your fields. This function arranges the values of the arguments into ascending or descending order, breaking ties by further arguments.

For example:

```
ORDERING([Total Sales], -1*[COUNT Salesman],  
MIN(<Office Name>))
```

For more information about the ORDERING() function, [click here](#).

- **R expression** – Your R code needs to be passed here, wrapped in double quotes. R expects the return type to be an array with the same size as widget's row count. Nulls will be used to make up for shorter arrays, and longer arrays will be trimmed.

Use single quotes to wrap strings within your R code when using the Rint/Rdouble functions, so that there will be no double-quote collision with the quotes wrapping your R code within the Rint/Rdouble function.

- **Numeric Value Arguments** – Numeric values can be passed as arguments to your R code.
- All arguments are passed to R as a 1-based list named "args". Each item in the list contains an array that represents the field.

For example:

- `args[[1]]` will return an array which represents the first field that was used as an argument.

- `args[[2]][3]` will return the 3rd data value within the 2nd field that was used as an argument.

# Guidelines for using R code in Sisense

In order for your R code to work well in Sisense you need to make sure of the following:

1. The resultset returned from R into Sisense has to be equal in size to the dataset that is sent to R as parameters.
2. The order of the resultset return from R into Sisense has to match the incoming order. No assumptions can be made about the incoming order – This means that you cannot assume that the incoming order matches the order you visually see in the widget.

# Example 1: Simple R Function – Kmeans Clustering

**Description:** In the example below the Total Cost and Total Revenue fields from Sisense are being used to cluster data via a Kmeans function. The result will cluster the data based on the kmeans settings, in this case 4 clusters.

**Sisense Syntax:**

```
RINT(TRUE, "m<- log(matrix(unlist(args), ncol = 2));  
kmeans(m,4)$cluster" , [Total Cost], [Total Revenue])
```

**Result:** The R expression will return a result from 1 to 4. The widgets color settings can be configured to color corresponding values based on if they equal 1, 2, 3 or 4.

**Tip:** You can [save \(star\)](#) the complex formula above, and use it again in additional formulas and charts.

In the chart's BREAK BY/COLOR panel, you can change the color of the clusters as well as define the number of clusters and their break points.

POINT

Brand

BREAK BY / COLOR

R Cluster Formula

SIZE

R CLUSTER FORMULA > COLOR

Single Color

Range

Conditional

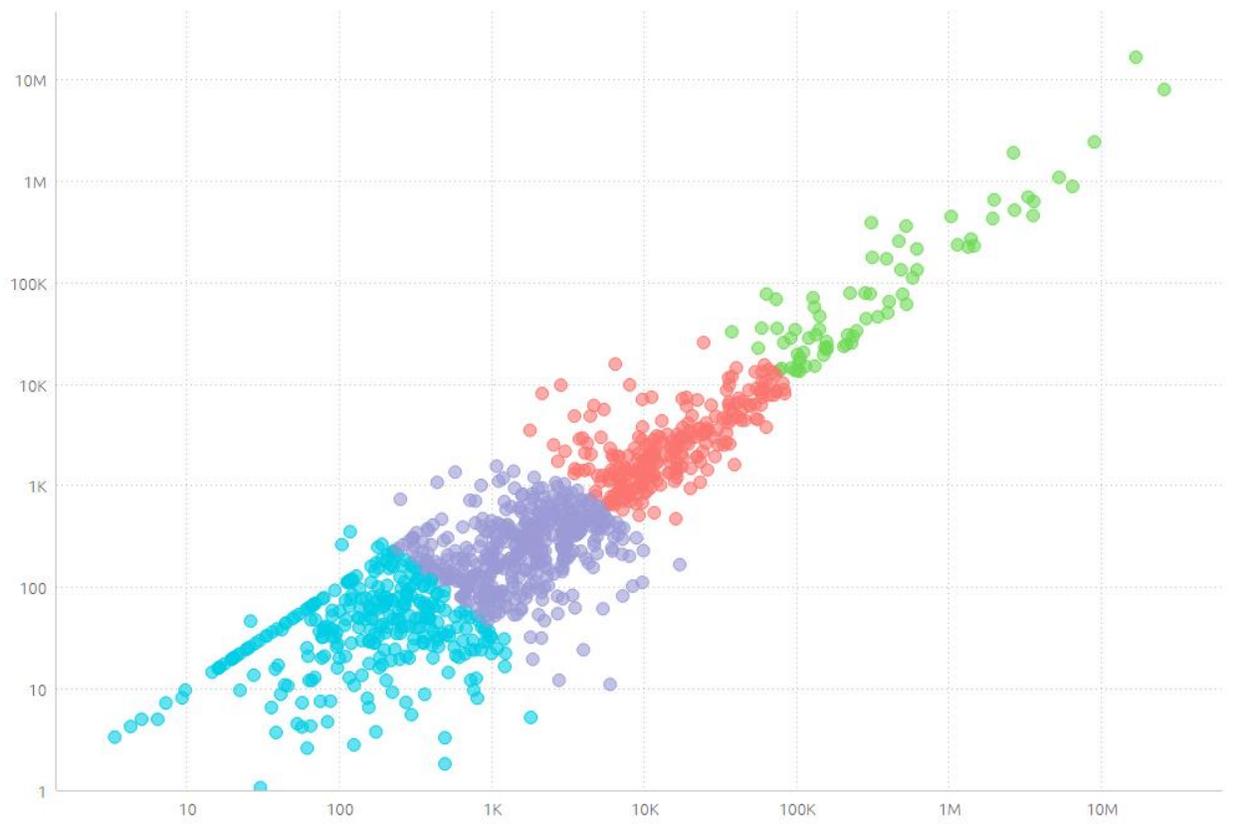
+ Add condition

=	fx 1		
=	fx 2		
=	fx 3		
=	fx 4		

reset

OK

Update on Every Change



## Example 2: R Logistic Regression via Sisense

**Description:** In the example below website traffic data is used to predict if a conversion is likely to occur based on the source of web traffic. The following fields contained in the ElastiCube are passed to the R logistic model: Average Outcome (1=conversion, 0= no conversion), Average Time on Page, Average Pages (viewed) and Average Bounce Rate. This data is passed to a model to predict the outcome and likelihood the traffic will convert.

[For the sake of illustration both the training and test data set are the same in the example]

### Sisense Syntax:

```
RDOUBLE(TRUE, "mydata<-data.frame(convert=args[[1]],
time=args[[2]], pages=args[[3]],
bounce=args[[4]]);lrmodel <- glm(convert ~ time + pages +
bounce, data = mydata, family = 'binomial');prob <-
predict(lrmodel, newdata = mydata, type = 'response'",
[Average Outcome], [Average TimeonPage], [Average Pages],
[Average BounceRate])
```

**Result:** The R expression will return a result from 0 to 1 showing the likelihood of a conversion occurring. Note a similar process can be applied to create a linear regression.

ROWS	+	Ref	Predicted Outcome
#	☰	(direct) / (none)	0.60
🗑️	🔘	t.co / referral	0.57
		social / twitter	0.56
		bing / organic	0.54
		amddatawiz.com / referral	0.54
		bing.com / referral	0.53
		answers.yahoo.com / referral	0.50
		bing.com / referral	0.45
		bingsandbox.com / referral	0.43
		buy-cheap-online.info / referral	0.41
		checkpagerank.net / referral	0.40
		getpocket.com / referral	0.39
		criticue.com / referral	0.39

Predicted outcome of website conversion based on logistic regression.

## Example 3: Loading an Existing R Model

**Description:** In the example below a saved logistic regression built on previous training data is called and used with new website traffic data to predict if a conversion occurs. The following fields contained in the ElastiCube are passed to the saved R logistic model: Average Outcome (1=conversion, 0= no conversion), Average Time on Page, Average Pages (viewed) and Average Bounce Rate. This data is passed to a model to predict the outcome and likelihood the traffic will convert.

**Sisense Syntax:**

```
RDOUBLE(TRUE, "mydata<-data.frame(convert=args[[1]],
time=args[[2]], pages=args[[3]],
bounce=args[[4]]);load('C:\rdata');prob <-
predict(lrmodel, newdata = mydata, type = 'response')",
[Average Outcome], [Average TimeonPage], [Average Pages],
[Average BounceRate])
```

**Result:** The R expression will return a result from 0 to 1 showing the likelihood of a conversion occurring based on the loaded R model in this case stored in 'rdata'.

## Example 4: Building and Saving a R Model using Data from Sisense

**Description:** In the example below a logistic regression to predict if a website conversion will occur is built using data from Sisense and then saved as an R model. The following fields contained in the ElastiCube are passed to create the R logistic model: Average Outcome (1=conversion, 0= no conversion), Average Time on Page, Average Pages (viewed) and Average Bounce Rate. This data is saved as an rdata file and can be used on other data sets to predict outcomes.

### Sisense Syntax:

```
RDOUBLE(TRUE, "mydata<-data.frame(convert=args[[1]],
time=args[[2]], pages=args[[3]],
bounce=args[[4]]);lrmodel <- glm(convert ~ time + pages +
bounce, data = mydata, family = 'binomial');
save(lrmodel, file = 'C:\rdata' );1", [Average Outcome],
[Average TimeonPage], [Average Pages], [Average
BounceRate])
```

**Result:** The R expression will use Sisense data to build and save a logistic model in the specified file in this case 'rdata'.

### Introduction to Formulas

This topic provides an overview to working with formulas in Sisense along with important tips and examples.

Formulas are custom calculations performed on one or more fields in your data.

They offer an important way to analyze results and express business logic.

Sisense's formula capabilities are designed around several principles;

- Create complex business calculations without IT or technical knowledge.
- Easily combine fields from different data sources together.
- Customize formulas to reflect specific criteria and conditions.
- Work with raw data without the need to summarize data sets before creating formulas.
- Instantly recalculate formulas based on any filter, variable or level of granularity.

The table below provides a reference to the main formula functions available in Sisense. [Click here](#) to see a complete reference.

Goal	Function	Types and Syntax
Perform calculation based on criteria	Measured value	<p><b>Value Filters:</b> <math>\neq</math>, <math>=</math>, <math>&gt;</math>, <math>&lt;</math>, between</p> <p><b>Text Filters:</b> Contains, Doesn't Contain, Doesn't End With, Doesn't Start With, Ends With, Start With, Equals, Not Equal</p> <p><b>List Filter:</b> Include, Exclude</p> <p><b>Ranking Filters:</b> Top, Bottom Ranking</p> <p><b>Time Filter:</b> Date and Calendar</p>
Combine data/apply simple mathematics	Aggregate functions	<p><b>Operator:</b> +, -, *, /</p> <p><b>Aggregate:</b> Sum()</p> <p><b>Average:</b> Avg()</p> <p><b>Count:</b> Count(), DupCount()</p> <p><b>Range:</b> Max()/Min()</p>
Summarize data	Statistical function	<p><b>Central Tendency:</b> Median(), Model, Largest()</p> <p><b>Std Deviation and Variance:</b> Stdev(), Stdevp(), Varp(), Var()</p> <p><b>Quartile and Percentile:</b> Quartile(), Percentile()</p>
Accumulate data	Rolling sum/average	<p><b>Sum to Date:</b> YTDSum(), QTDSum, MTDSum()</p> <p><b>Avg to Date:</b> YTDAvg, QTDAvg, MTDAvg()</p>
Compare Time or	Time functions	<p><b>Past Periods:</b> PastYear(), PastQuarter(),</p>

Goal	Function	Types and Syntax
Trends		PastMonth(), Next(), Prev() <b>Growth Trend:</b> Growth(), GrowthRate() <b>Time Difference:</b> YDiff(), QDiff(), MDiff(), DDiff(), HDiff(), MnDiff(), SDiff()

The following topics explain how to create formulas, and describe the features of the formula editor.

- [Using the Formula Editor](#)
- [Creating and Editing a Formula](#)
- [Reusing Formulas](#)
- [Using Quick Functions](#)
- [Creating Formulas Based on Criteria and Conditions \(Filters\)](#)
- [Building Formulas with Functions](#)
- [Function Reference](#)

## Formula Editor

The Formula Editor is where Designers define formulas for a dashboard's widgets.

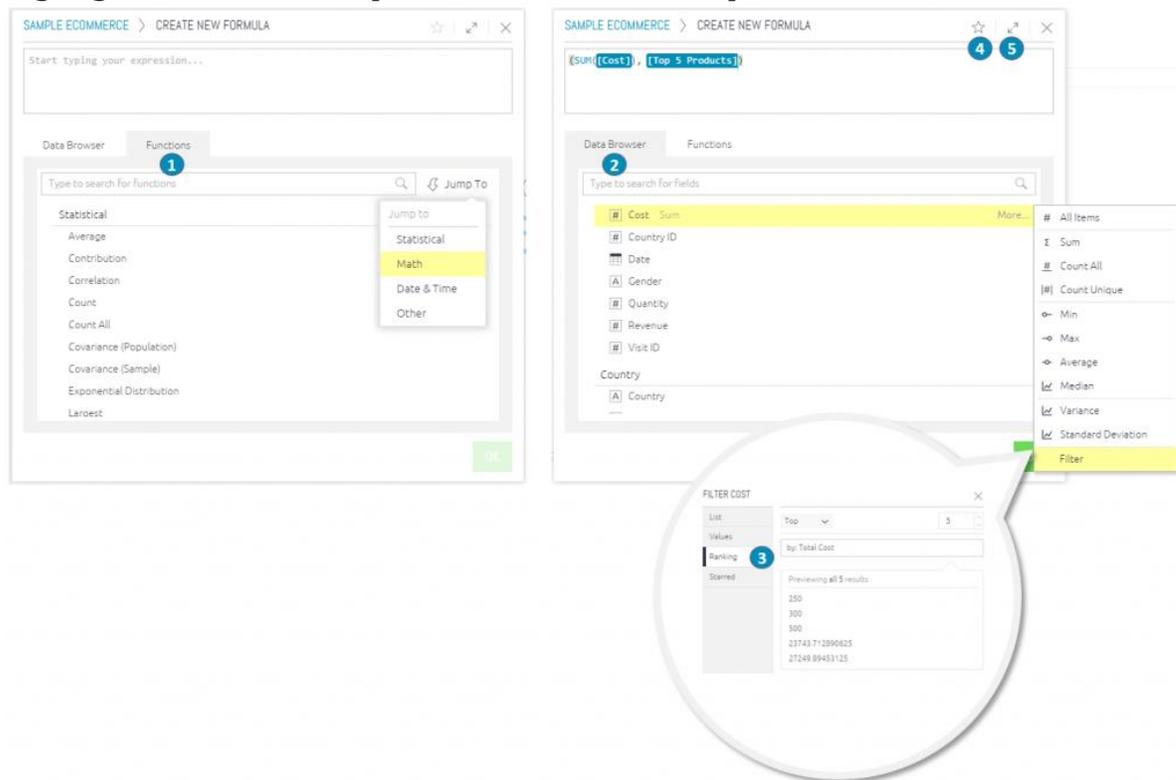
If you are familiar with the Formula Editor, see the [Quick Reference Guide](#). For a detailed guide with examples, continue reading below.

### To open the Formula Editor:

- Do one of the following:
  - For a new widget, click **Select Data**, and then  .
  - For an existing widget, click on the edit formula button  .

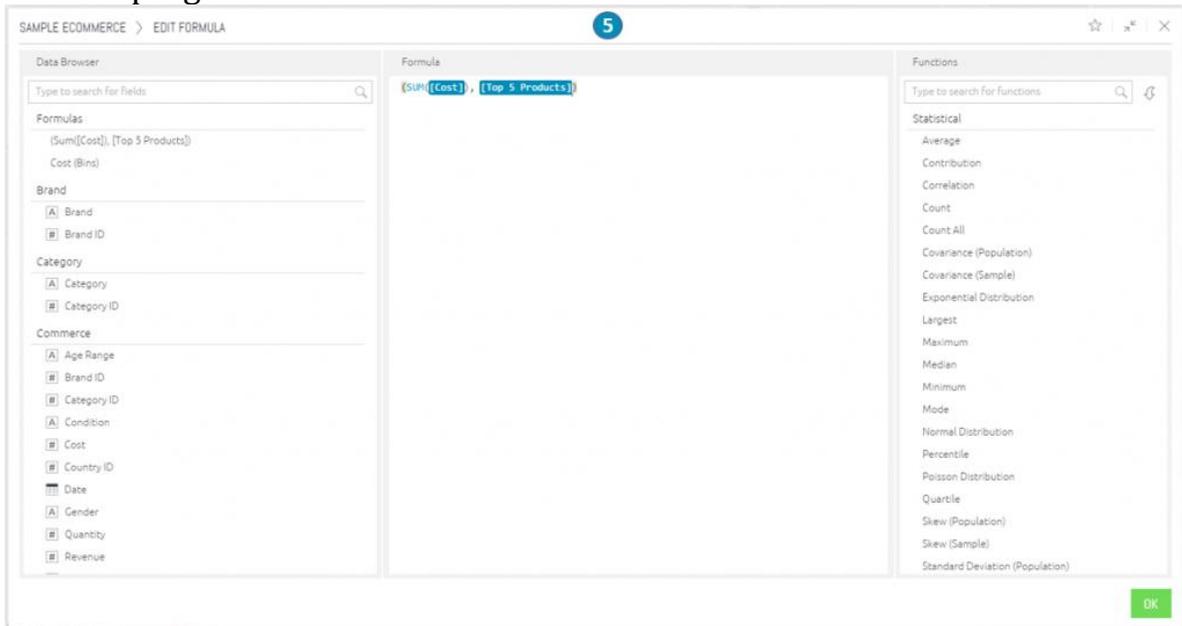
The Formula Editor has two tabs, the Data Browser to select fields and the Functions tab to select formula operations. You can create a formula combining one or more function, field and filters. The diagram below

highlights the main components of the formula panel.



1. **Functions** are operations which perform different calculations, for example a sum. Use the 'Jump To' menu or the search box to quickly find the formula you need.
2. Fields in the **Data Browser** are variables contained in the data set (ElastiCube). Clicking on a field in the data browser will include it as part of the formula.
3. [Filters](#) can be applied to restrict formulas based on criteria.
4. [Starring](#) is a way to save a formula for later use.

5. The Formula Editor window can be expanded by clicking the expand button at the top right.



## Creating and Editing a Formula

The Data Browser enables you to define formulas (freeform expressions) that define the values and filters of a widget.

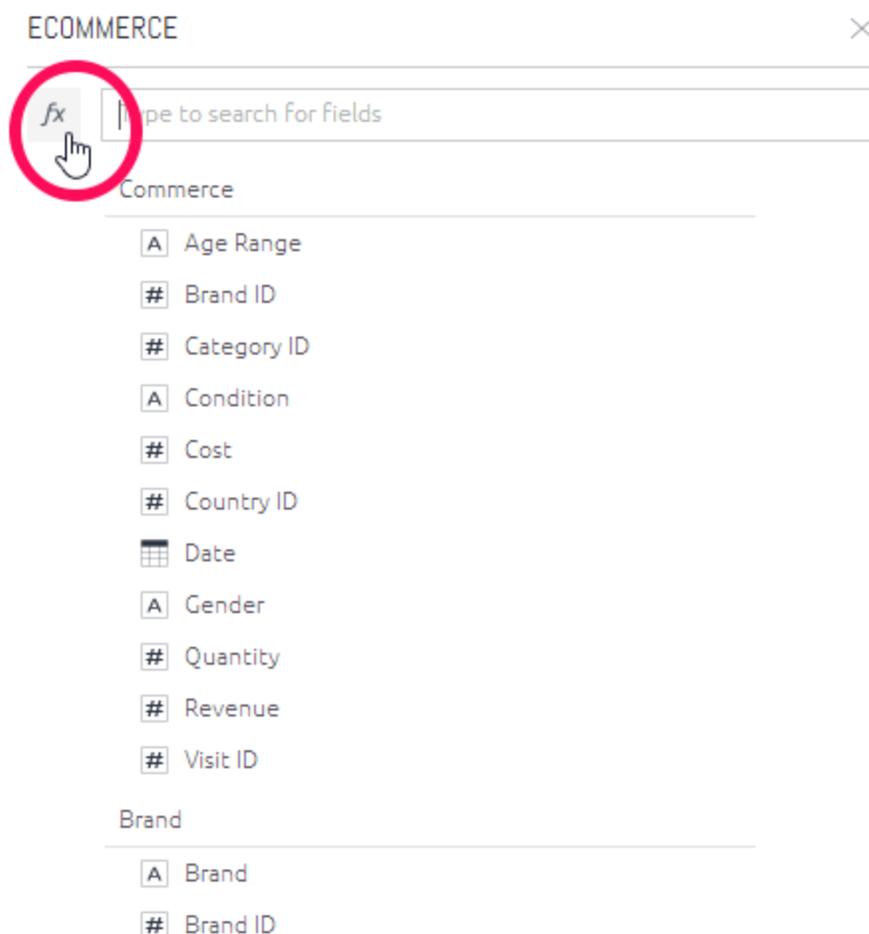
A rich variety of functions are provided for you to use in the formula that you define.

### To define a formula:

1. Open the formula editor in the Data Browser:

- For a new widget, click Select Data, and then  .

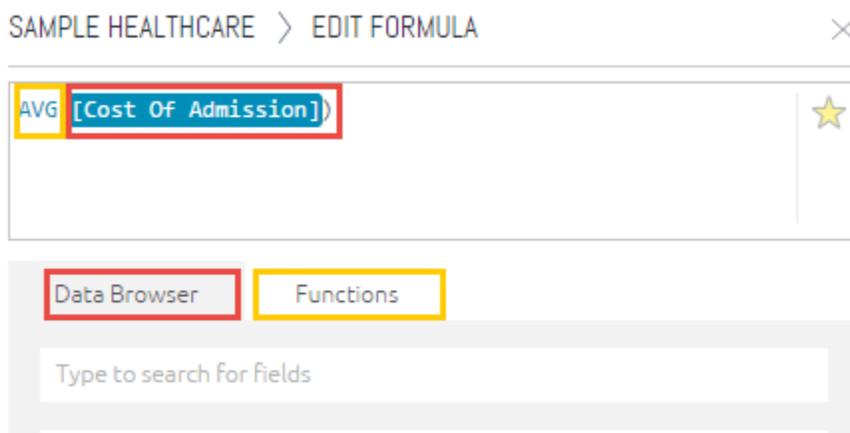
- For an existing widget, click on the edit formula button  .



The Data Browser then changes to display the Formula Editor, which has two tabs: Data Browser and Functions.

- The Data Browser tab provides fields to choose from.
  - The Functions tab lists the functions that you can include in your formula by selecting them. You can read a description of each function in a tooltip by hovering over it.
2. Define the formula as follows:
- From the **Data Browser** tab, select one or more fields.

- From the **Functions** tab, select the required functions.



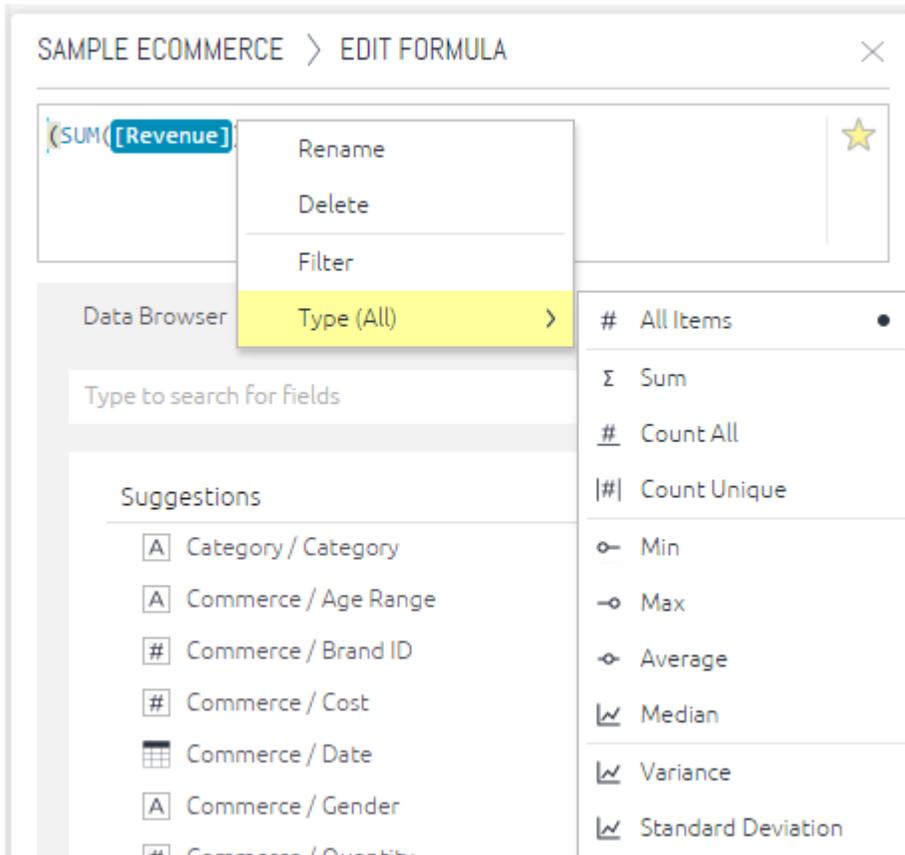
3. Type in the required parts of the formula. To see examples, see [Formulas Based on Criteria and Conditions](#), and [Functions to Build Formulas](#).
4. Click **OK**.

### To edit a formula:

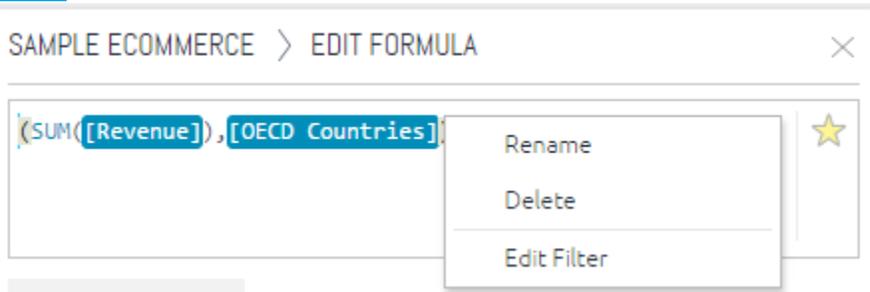
You can easily edit formulas using the right-click options. They include:

- **Rename:** Rename the formula, for example, give a name that represents a real-life task or expected result from the formula, or include in the name filters that you have added to the formula.
- **Filter:** Add filters to the formula.
- **Type:** Change the default aggregation method, for example, from Sum to Average.

The following image shows the right-click options.



The following image shows right-click options for a filtered formula. Read more [below](#).



## Reusing Formulas

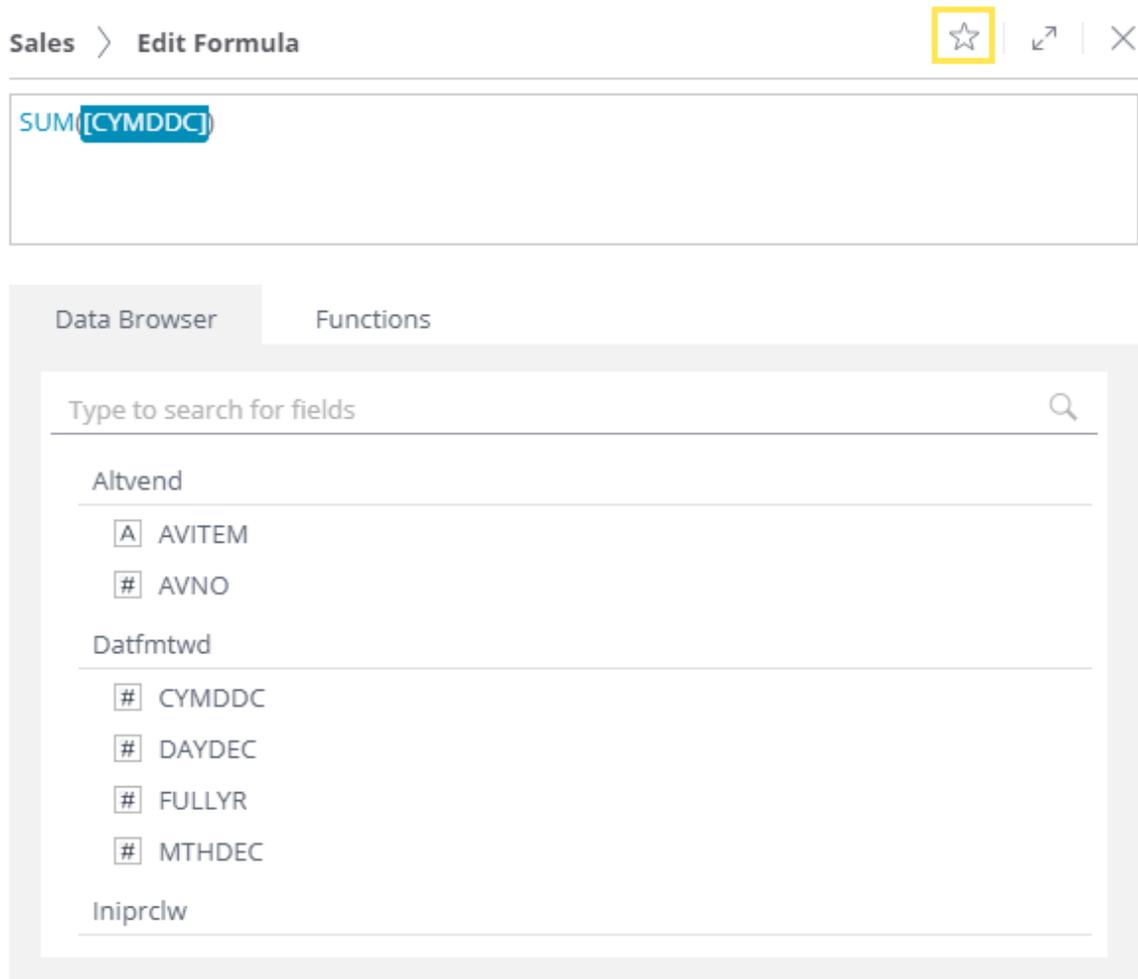
This topic describes how you can reuse formulas that you have previously marked as a favorite (starred).

Formulas are saved per ElastiCube.

**Important:** Changing a starred formula **does not** affect previous uses of that formula. These will continue to use the old version of the formula. Only future uses of the starred formula will implement your latest formula.

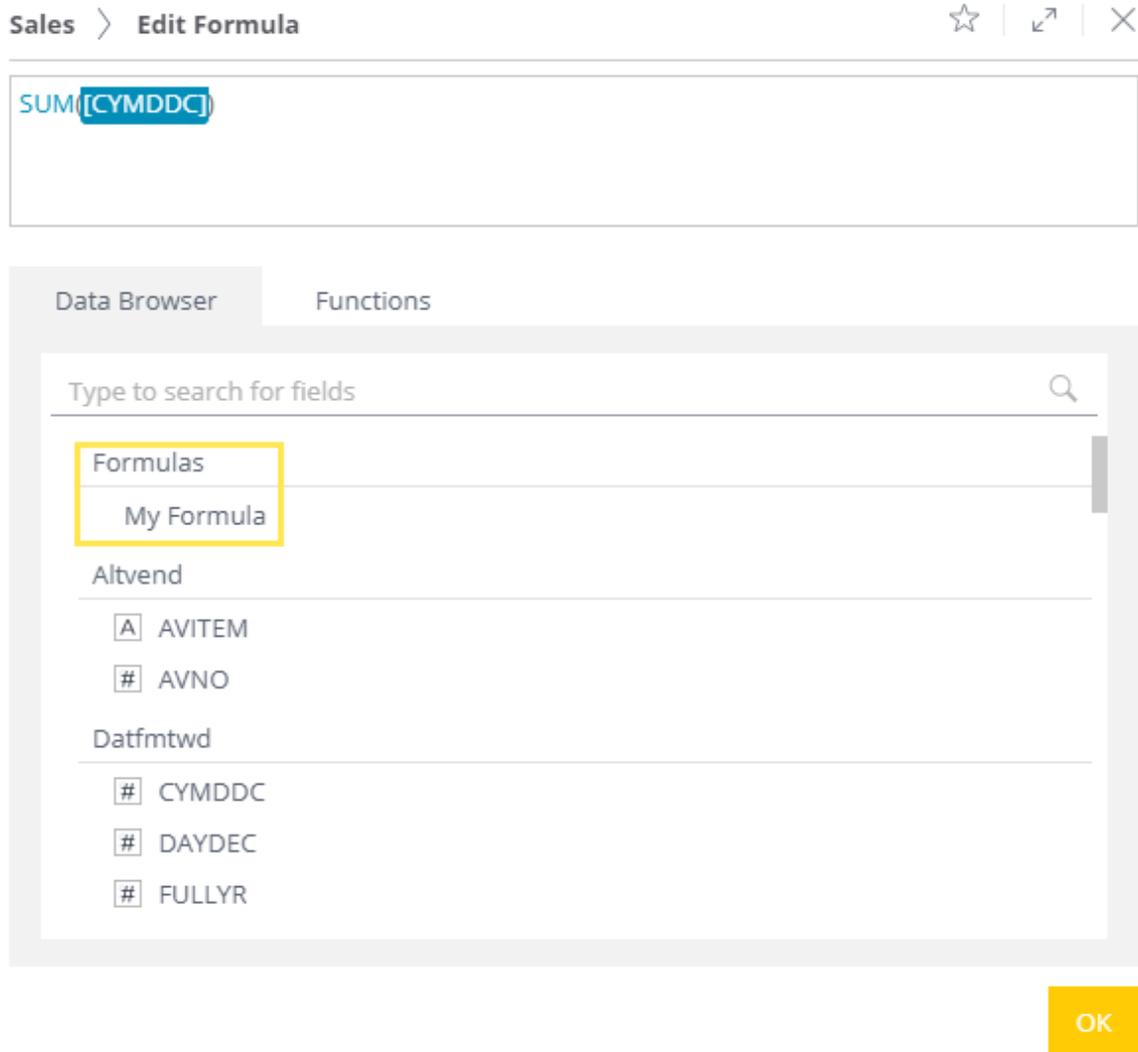
**To mark a formula as a Favorite:**

1. While defining a formula, click the Favorite (Star) button.
2. Enter a name for this Formula.
3. Click **OK**.



**To reuse a favorite formula:**

- Favorite Formulas appear in the Data Browser under the title Formulas, as shown below. Simply select it to use it.



## Quick Functions

Just like [reusing formulas](#), quick functions is another feature to make working with formulas easier. The Widget Designer provides a variety of predefined commonly used functions that you can easily apply in the Data Browser when selecting a Value to be included in a widget.

Quick Functions instantly add a time dimension to any existing value and formula. These functions include calculations for past values, change over time,

contribution and running totals. Quick Functions include all the Time Functions previously discussed but they can only be accessed by clicking on a formula that is already present in a widget.

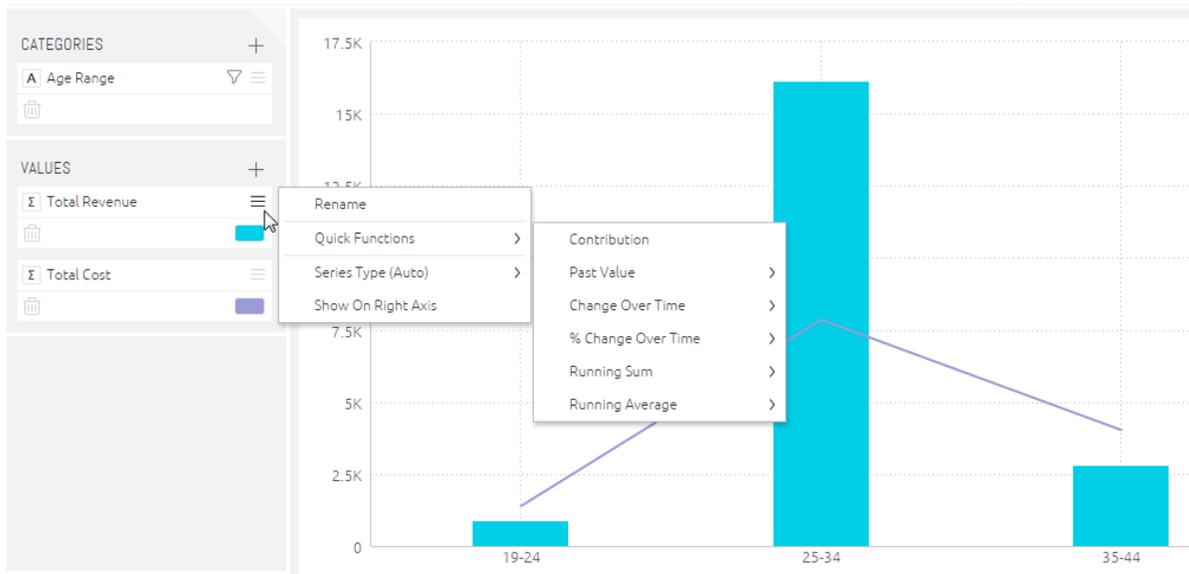
A simple example of a Quick Function is a finance manager who is reviewing total costs per month but needs to track the accumulated annual costs. They can simply use a quick function to calculate the year to date total for costs.

The screenshot shows a widget designer interface with three main sections: ROWS, VALUES, and COLUMNS. The ROWS section contains a widget named 'Months in Date'. The VALUES section contains a widget named 'Total Cost'. The COLUMNS section is currently empty. A data table is displayed, showing the relationship between 'Months in Date' and 'Total Cost'. A context menu is open over the 'Total Cost' column, listing various functions including 'Quick Functions', which is highlighted. The 'Quick Functions' sub-menu is also open, showing options like 'Contribution', 'Past Value', 'Change Over Time', '% Change Over Time', 'Running Sum', and 'Running Average'.

Months in Date	Total Cost
01/2013	12,349.68
02/2013	11,247.97
03/2013	12,504.97
04/2013	11,900.94
05/2013	12,329.42
06/2013	12,000.75

**To use a quick function:**

1. Hover and click on the menu icon of a numeric field in the data panel of the widget designer, and select Quick Functions from the menu:

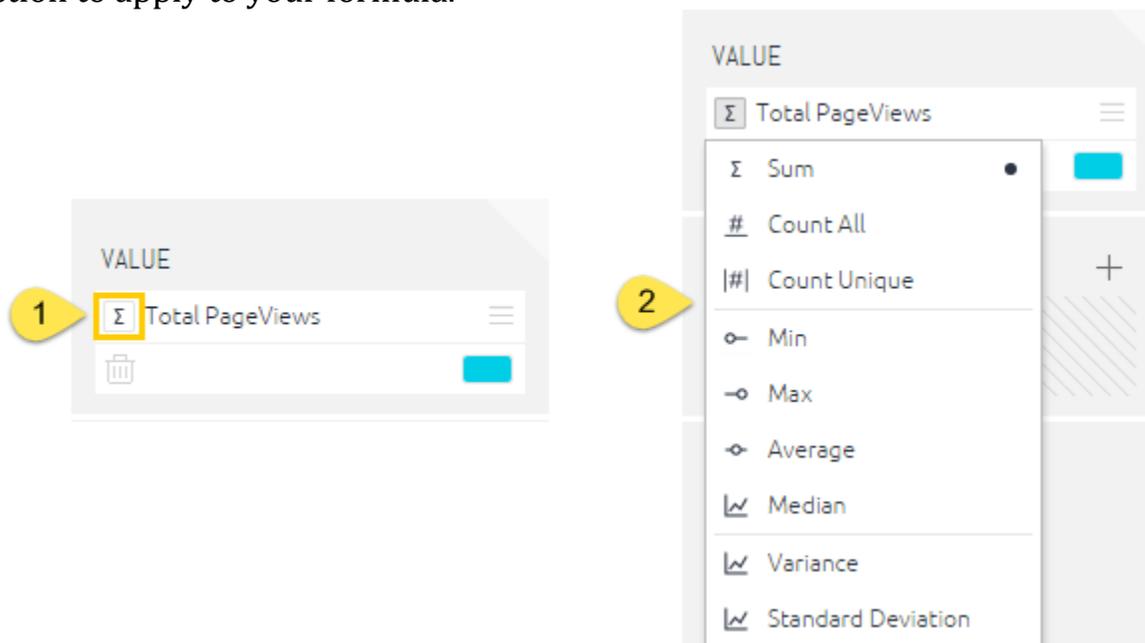


- A list of commonly used functions is displayed.
2. Select a function. The widget will be updated immediately.

# Adding Aggregate Functions

You can also add aggregate functions to your formula without opening the formula editor.

Click on the Value icon to open a list of aggregate functions, and select the function to apply to your formula.



## Starring Formulas with Quick functions

A more complex example uses starring with multiple quick functions. Let's say a finance manager also wants to compare year to date costs to the same period for the previous year. They can first apply the year to date function to total cost and then save it to the formula repository. They can then add the saved year to date formula but apply another quick formula for past values which will perform the same calculation but on data from the previous year.

**Note:** Starred (favorite) formulas will be shared with other users.

### Using R in Formulas

[R](#) is a software environment for statistical computing and graphics. Sisense supports the integration of R functions in your formulas.

You can write R code directly in the formula editor, and send fields as parameters. You can also combine R functionality with the regular functions to create advanced and tailored formulas. (For general help with the formula editor and functions, click [here](#).)

Before you can integrate R into your formulas, you must have an R server set up in your organization.

This document will cover the following:

- Understanding How R works in Sisense
- Connecting Sisense to Your R Server
- Using R functions in Formulas
- Guidelines for using R code in Sisense
- Example 1: Simple R Function – Kmeans Clustering
- Example 2: R Logistic Regression via Sisense
- Example 3: Loading an Existing R Model
- Example 4: Building and Saving a R Model using Data from Sisense

# Understanding How R works in Sisense

Within a widget, R code is used and assigned fields as parameters, the following happens:

1. The selected fields are sent fully to R as a list of arrays.
2. The R code is executed on the R server, referencing the fields as needed.
3. The numeric results from R are sent back to the widget for visualization.

Each request from the R server is done in a self-contained namespace, thus R variables and results cannot be reused across requests unless they are loaded from a saved location.

R calculations are computed in the R server, and not in the ElastiCube, this means that for larger data sets results will not return as fast as regular widgets.

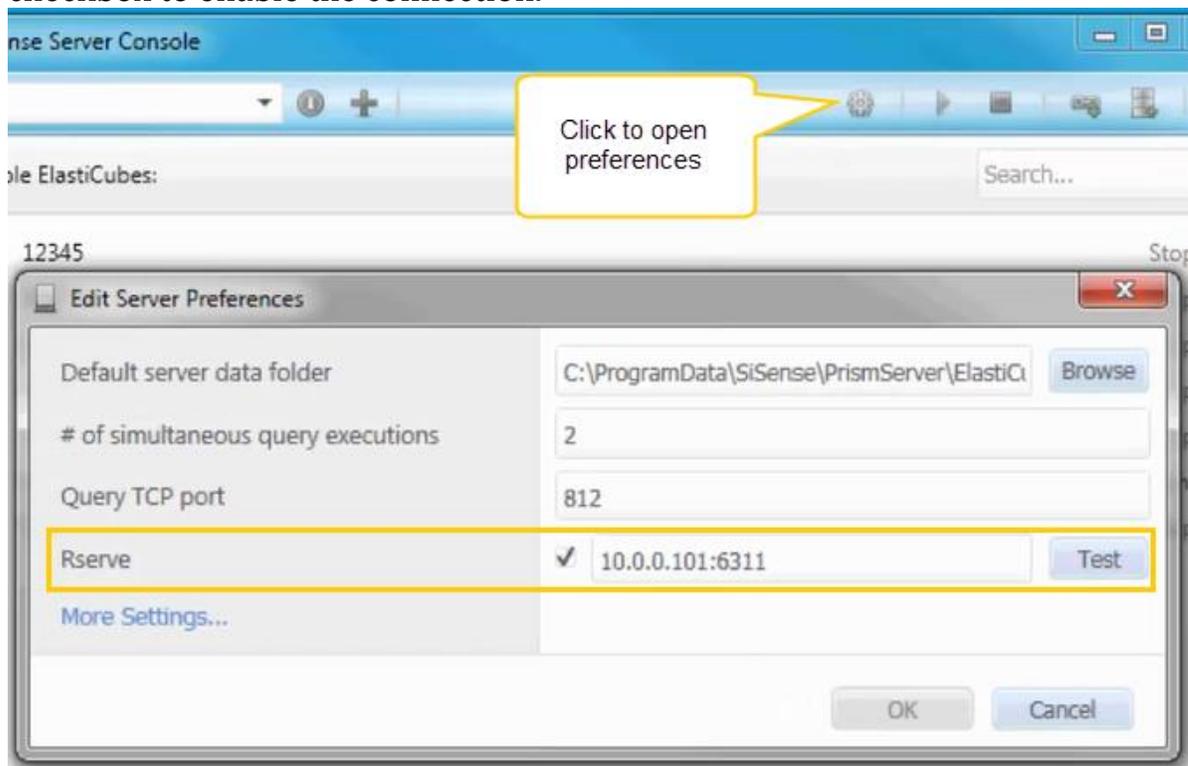
## Running R on Windows

If your R server is running on Windows you will be bound to the concurrency limitations the R distribution has on Windows. This means that concurrent requests to R may return wrong results. Sisense recommends using R on a Linux machine for production environment with concurrent user activity.

# Connecting Sisense to Your R Server

**To connect your Sisense instance to your R server:**

1. Open the Sisense Server Console from the start menu.
2. Click on the Server Preferences icon to open the server settings.
3. In the **Rserve** field, enter the IP address of your R server, and select the checkbox to enable the connection.



You can test the connection by clicking on the **Test** button.

## Installing Rserve

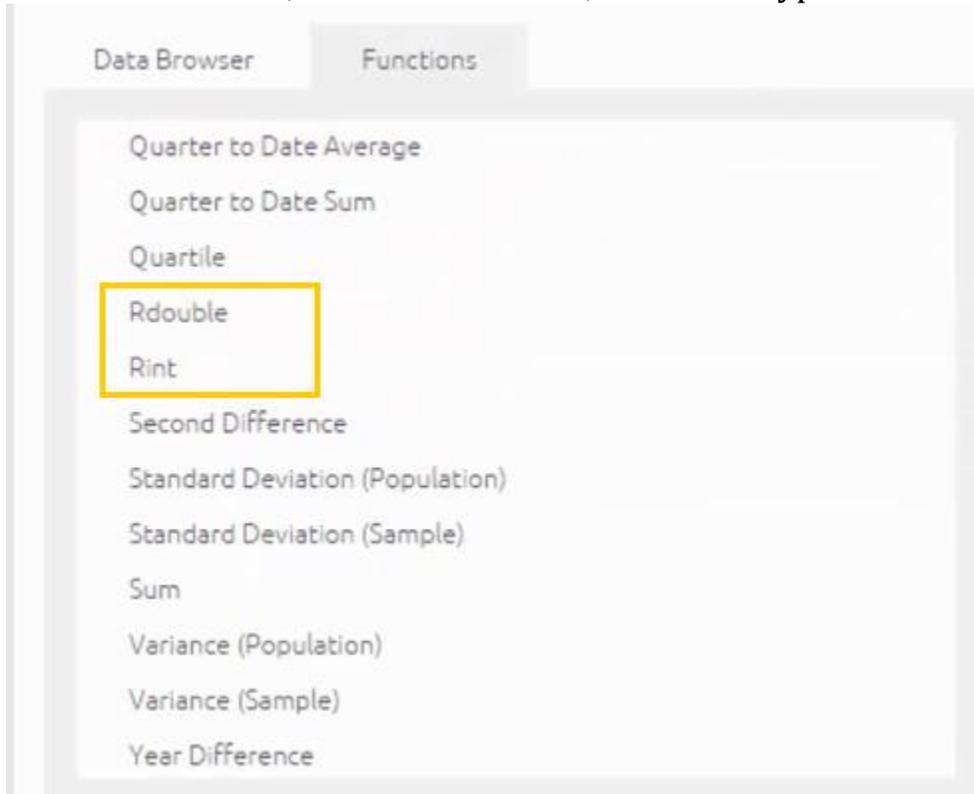
Sisense can only connect to a running Rserve instance. For more information about installing and running Rserve [follow this link](#).

Please note that if you're using RStudio, Rserve still has to be installed.

# Using R functions in Formulas

Using R code in Sisense is achieved by using two dedicated formulas – RINT & RDOUBLE. Depending on the result type of your R code, use the appropriate function within the [formula editor](#).

For either formula, RINT or RDOUBLE, the return type has to be an exact match.



Multiple numeric field values can be passed as parameters to the R functions, which within your R code can be accessed via the 'args' argument (See K-means example below).

## Syntax:

```
RINT([recycle (true)], [<Ordering>], <R expression>,
[<numeric value 1>, ..., <numeric value n>] )
```

```
RDOUBLE([recycle (true)], [<Ordering>], <R expression>,
[<numeric value 1>, ..., <numeric value n>] )
```

Sisense does not support the use of a final ';' in R statements. If you add a semi-colon at the end of your R statement, the code will fail.

### Parameters:

- **Recycle true/false (default = true)** – This is an optional parameter that controls whether the results from R should be recycled (cached), so that consequent queries will not have to be recalculated unless they or the data have changed. Generally, this behavior is automatically managed by the ElastiCube automatically for all functions. However, since R code might have non-deterministic components to it (such as randomness functions or date-specific functions), the ElastiCube cannot rely on a data-set and function that hasn't changed not to return a different result in multiple executions.

By default, the Recycle value is set to true. Use 'off' if your R code contains randomness or other non-deterministic content.

- **Ordering** – This is an optional parameter that defines the sort order in which numeric data is sent to R. The argument of the Ordering parameter can be an index in your data source or you can use the ORDERING() function to determine the order of your fields. This function arranges the values of the arguments into ascending or descending order, breaking ties by further arguments.

For example:

```
ORDERING([Total Sales], -1*[COUNT Salesman],  
MIN(<Office Name>))
```

For more information about the ORDERING() function, [click here](#).

- **R expression** – Your R code needs to be passed here, wrapped in double quotes. R expects the return type to be an array with the same size as widget's row count. Nulls will be used to make up for shorter arrays, and longer arrays will be trimmed.

Use single quotes to wrap strings within your R code when using the Rint/Rdouble functions, so that there will be no double-quote collision with the quotes wrapping your R code within the Rint/Rdouble function.

- **Numeric Value Arguments** – Numeric values can be passed as arguments to your R code.
- All arguments are passed to R as a 1-based list named "args". Each item in the list contains an array that represents the field.

For example:

- `args[[1]]` will return an array which represents the first field that was used as an argument.

- `args[[2]][3]` will return the 3rd data value within the 2nd field that was used as an argument.

# Guidelines for using R code in Sisense

In order for your R code to work well in Sisense you need to make sure of the following:

1. The resultset returned from R into Sisense has to be equal in size to the dataset that is sent to R as parameters.
2. The order of the resultset return from R into Sisense has to match the incoming order. No assumptions can be made about the incoming order – This means that you cannot assume that the incoming order matches the order you visually see in the widget.

# Example 1: Simple R Function – Kmeans Clustering

**Description:** In the example below the Total Cost and Total Revenue fields from Sisense are being used to cluster data via a Kmeans function. The result will cluster the data based on the kmeans settings, in this case 4 clusters.

**Sisense Syntax:**

```
RINT(TRUE, "m<- log(matrix(unlist(args), ncol = 2));  
kmeans(m,4)$cluster" , [Total Cost], [Total Revenue])
```

**Result:** The R expression will return a result from 1 to 4. The widgets color settings can be configured to color corresponding values based on if they equal 1, 2, 3 or 4.

**Tip:** You can [save \(star\)](#) the complex formula above, and use it again in additional formulas and charts.

In the chart's BREAK BY/COLOR panel, you can change the color of the clusters as well as define the number of clusters and their break points.

POINT

Brand

BREAK BY / COLOR

R Cluster Formula

SIZE

Update on Every Change

R CLUSTER FORMULA > COLOR

Single Color

Range

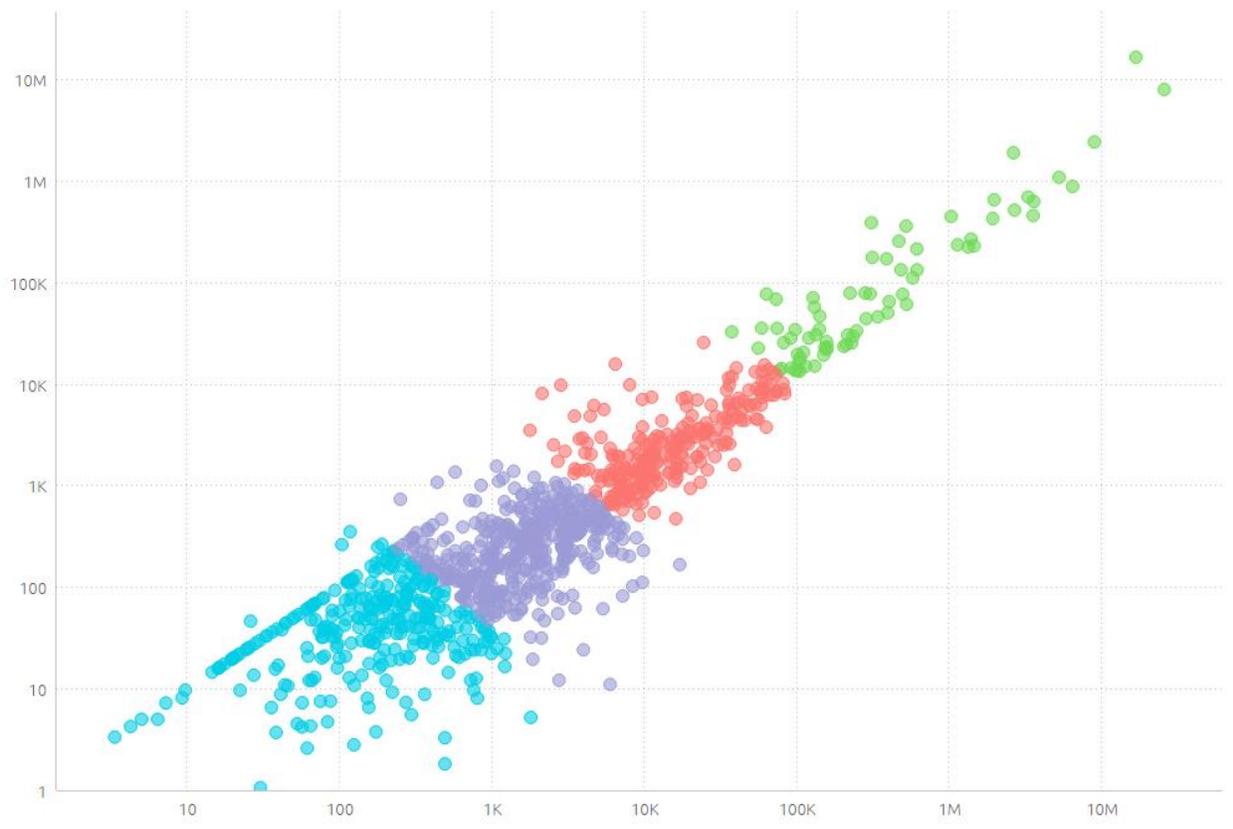
Conditional

+ Add condition

=	fx 1		
=	fx 2		
=	fx 3		
=	fx 4		

reset

OK



## Example 2: R Logistic Regression via Sisense

**Description:** In the example below website traffic data is used to predict if a conversion is likely to occur based on the source of web traffic. The following fields contained in the ElastiCube are passed to the R logistic model: Average Outcome (1=conversion, 0= no conversion), Average Time on Page, Average Pages (viewed) and Average Bounce Rate. This data is passed to a model to predict the outcome and likelihood the traffic will convert.

[For the sake of illustration both the training and test data set are the same in the example]

### Sisense Syntax:

```
RDOUBLE(TRUE, "mydata<-data.frame(convert=args[[1]],
time=args[[2]], pages=args[[3]],
bounce=args[[4]]);lrmodel <- glm(convert ~ time + pages +
bounce, data = mydata, family = 'binomial');prob <-
predict(lrmodel, newdata = mydata, type = 'response')",
[Average Outcome], [Average TimeonPage], [Average Pages],
[Average BounceRate])
```

**Result:** The R expression will return a result from 0 to 1 showing the likelihood of a conversion occurring. Note a similar process can be applied to create a linear regression.

Ref	Predicted Outcome
(direct) / (none)	0.60
t.co / referral	0.57
social / twitter	0.56
bing / organic	0.54
amddatawiz.com / referral	0.54
bing.com / referral	0.53
answers.yahoo.com / referral	0.50
bing.com / referral	0.45
bingsandbox.com / referral	0.43
buy-cheap-online.info / referral	0.41
checkpagerank.net / referral	0.40
getpocket.com / referral	0.39
criticue.com / referral	0.39

Predicted outcome of website conversion based on logistic regression.

## Example 3: Loading an Existing R Model

**Description:** In the example below a saved logistic regression built on previous training data is called and used with new website traffic data to predict if a conversion occurs. The following fields contained in the ElastiCube are passed to the saved R logistic model: Average Outcome (1=conversion, 0= no conversion), Average Time on Page, Average Pages (viewed) and Average Bounce Rate. This data is passed to a model to predict the outcome and likelihood the traffic will convert.

**Sisense Syntax:**

```
RDOUBLE(TRUE, "mydata<-data.frame(convert=args[[1]],
time=args[[2]], pages=args[[3]],
bounce=args[[4]]);load('C:\rdata');prob <-
predict(lrmodel, newdata = mydata, type = 'response')",
[Average Outcome], [Average TimeonPage], [Average Pages],
[Average BounceRate])
```

**Result:** The R expression will return a result from 0 to 1 showing the likelihood of a conversion occurring based on the loaded R model in this case stored in 'rdata'.

## Example 4: Building and Saving a R Model using Data from Sisense

**Description:** In the example below a logistic regression to predict if a website conversion will occur is built using data from Sisense and then saved as an R model. The following fields contained in the ElastiCube are passed to create the R logistic model: Average Outcome (1=conversion, 0= no conversion), Average Time on Page, Average Pages (viewed) and Average Bounce Rate. This data is saved as an rdata file and can be used on other data sets to predict outcomes.

### Sisense Syntax:

```
RDOUBLE(TRUE, "mydata<-data.frame(convert=args[[1]],
time=args[[2]], pages=args[[3]],
bounce=args[[4]]);lrmodel <- glm(convert ~ time + pages +
bounce, data = mydata, family = 'binomial');
save(lrmodel, file = 'C:\rdata' );1", [Average Outcome],
[Average TimeonPage], [Average Pages], [Average
BounceRate])
```

**Result:** The R expression will use Sisense data to build and save a logistic model in the specified file in this case 'rdata'.

### Creating Formulas Based on Criteria and Conditions (Filters)

Often formulas must take into account specific criteria. To do this Sisense provides a feature called **Measured Value**, which like the SUMIF function in Excel, only performs a calculation when the values meet a set of criteria. Criteria for Measured Values may be based on any logical operators in a filter.

### Measure Value Syntax:

(Measure, Scope1 , Scope2...ScopeN)

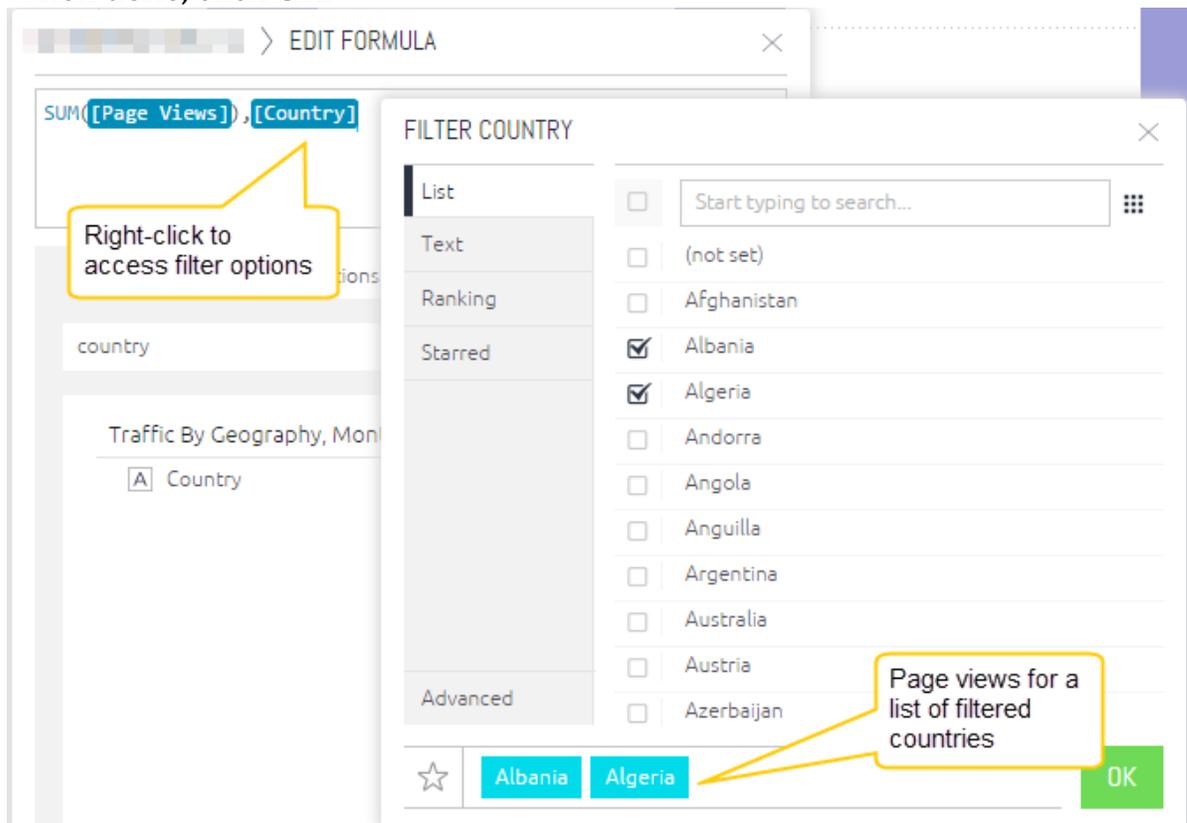
### Parameters

Measure: A field measure or formula.

Scope: A filter including *Value, Text, List, Ranking and Time filters.*

## To filter the formula:

1. In the Data Browser, create your formula from the Data Browser and Functions, as explained in [Formula Editor](#).
2. Add the field (criteria) by which you want to filter the formula. Right-click the field and select **Filter**.
3. You can then filter the formula by listed items, text options, ranking, etc. When done, click **OK**.



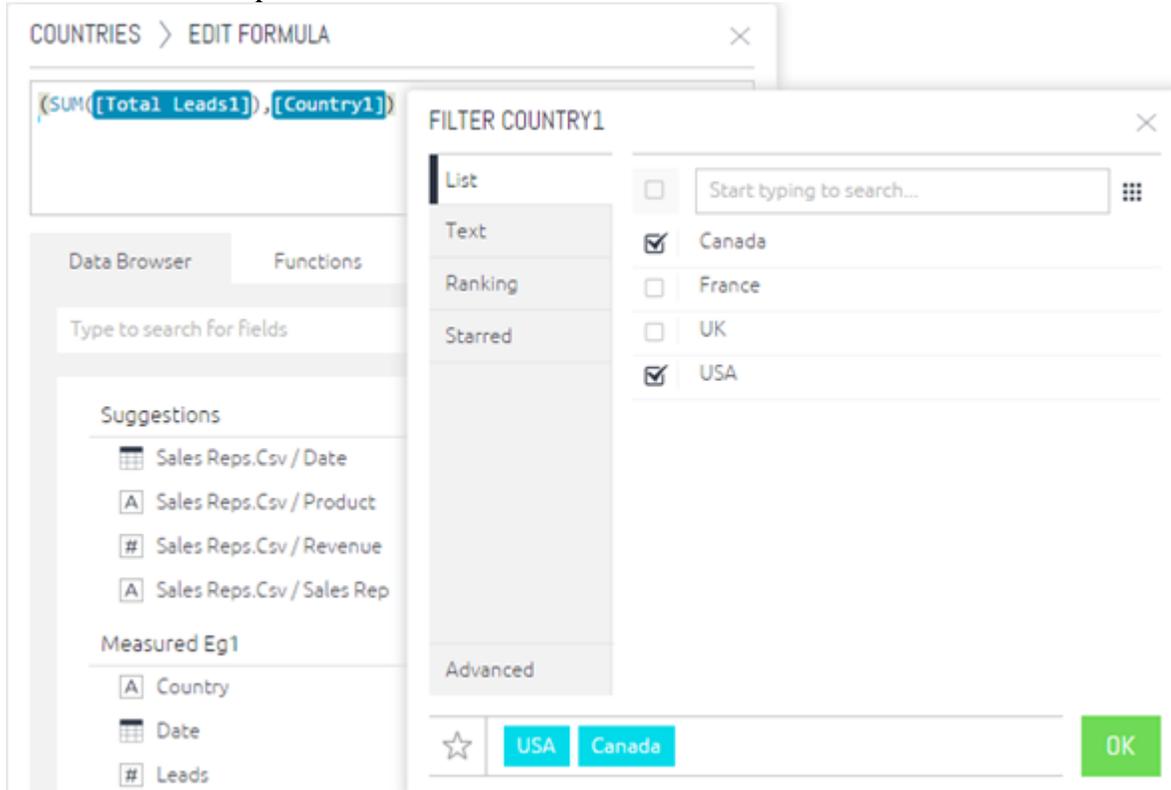
A simple example of Measured Value is the use of a list filter. A marketing team may need to count leads generated for a specific region such as North America. Even if leads come from many different countries, the measured value calculates leads generated only when the lead originates from the United States or Canada.

Country	# Leads
USA	15
France	20
Canada	5
UK	30

*Example 1: Measured Value with List Filter*  
 (Sum(Leads), (List Filter: Country = United States, Canada))

*Result*  
 15+5 = 20

The above example as defined in the Formula Editor.



A more sophisticated case is the **use of a ranking filter**, for example a sales team may want to track the contribution of best-selling products to total revenue. However, what constitutes a popular product may change over time. A measured value can be created for sales which includes a condition that only shows sales for the top products for any month. This simultaneously filters the data but also takes into account changes in what classifies as a top product over time.

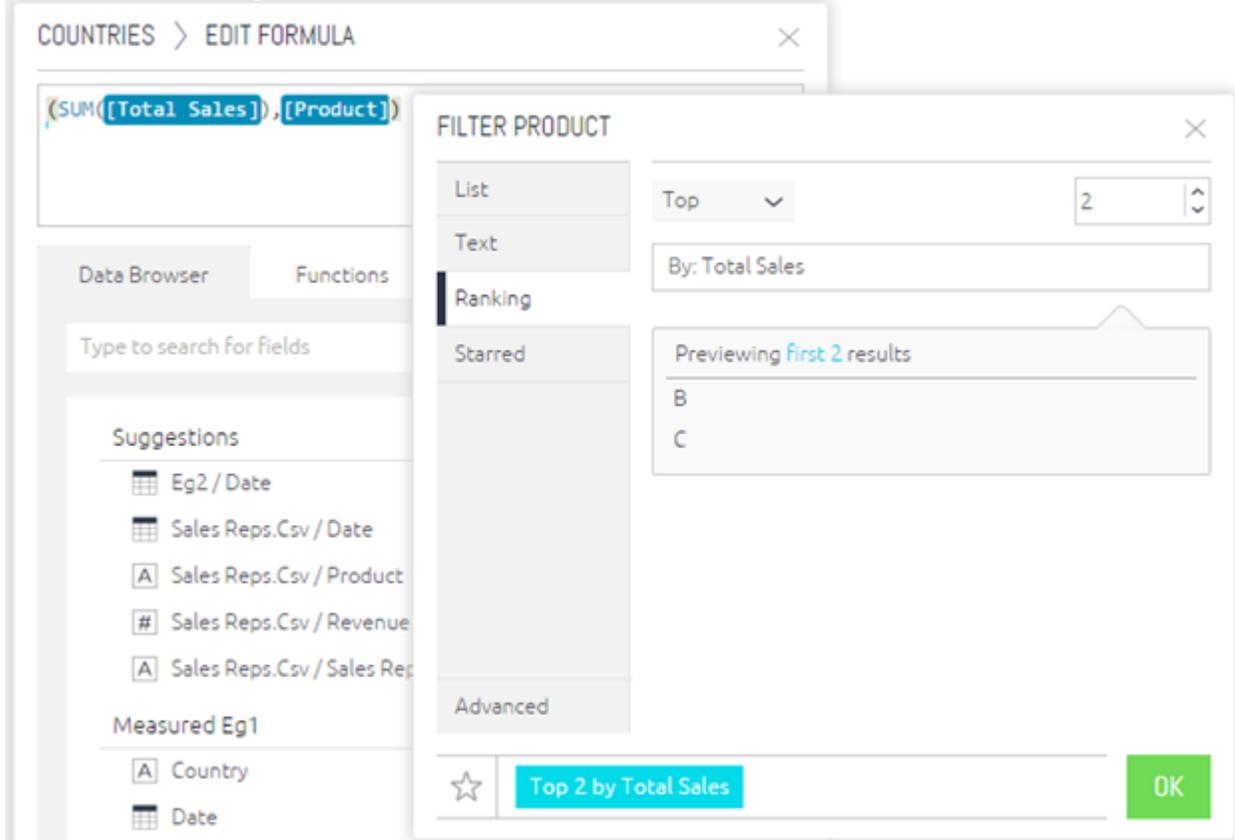
Month	Product	Sales
Jan	A	10
Jan	B	5
Jan	C	15
Feb	A	1
Feb	B	25
Feb	C	5

*Example 2: Measured Value with Top Ranking Filter*  
 (Sum(Sales), (Top Ranking Filter: Top 2 Products by Sales))

*Results*

Month	Total Sales Top 2 Products
Jan	(A/C) 10 + 15 = 25
Feb	(B/C) 25 + 5 = 30

The above example as defined in the Formula Editor.



Measured Values are a powerful feature to take into account business logic and quickly perform calculations only when a specific set of criteria is met.

**Note:** If your widget is filtered using measured values, then the measured value will override any other widget or dashboard filters you have for the same fields.

# Calculating Contributions Using the ALL Function

The All() function returns the total amount for a dimension, and can be used for various use cases. In the following example, we will use the All function to calculate how much each country contributed towards the total cost of a campaign.

Our final widget includes the following information:

Country	Total Cost	Total Cost per Countries	Contribution
Australia	9,643.09	283,755.53	3.40%
Brazil	9,638.80	283,755.53	3.40%
China	9,607.4	283,755.53	3.39%
England	19,492.8	283,755.53	6.87%
France	19,440.59	283,755.53	6.85%
Germany	19,377.61	283,755.53	6.83%
Greece	9,842.84	283,755.53	3.47%
India	9,603.16	283,755.53	3.38%
Kazakhstan	9,899.33	283,755.53	3.49%
Nigeria	9,764.44	283,755.53	3.44%
Norway	9,681.69	283,755.53	3.41%
Portugal	9,919.33	283,755.53	3.50%
South Africa	19,211.15	283,755.53	6.77%
Ukraine	9,693.03	283,755.53	3.42%
United States	99,241.60	283,755.53	34.97%
Vietnam	9,698.67	283,755.53	3.42%
Grand Total	283,755.53	283,755.53	100%

**Step 1:** The second column above represents a formula that sums up the total cost for all countries and does not represent the breakdown per country. The formula includes the calculation (total cost) followed by the all function (filter), followed by the dimension (country) in parenthesis. It looks like this:

```
sum([Total Cost], all([Country]))
```

We can save (star) the above formula and call it Total cost for Countries, which will be used in the next step.

**Step 2:** We can now use the above formula in another formula to calculate the contribution, like this:

```
SUM([Cost])/[Total Cost for Countries]
```

The result is the third column above (plus formatting the results as percentages).

## Building Formulas with Functions

Functions are operations that perform common types of calculations, and can be used to build formulas. In this topic, you can read about four types of functions. The functions' syntax is explained and examples are provided.

# Combine Data: Aggregate Functions

Aggregations are used to perform mathematical calculations on data. Although this is an essential function Sisense offers an advantage in the ability to run multiple aggregations on several fields simultaneously – this makes it easy to summarize data based on multiple factors.

## *Aggregate Syntax:*

Function(Numeric Field)

## *Parameters*

Function: Sum(), Avg(), Count(), DupCount(), Max(), Min()

Numeric Field: A numeric field or formula.

An example **simple aggregation** is a sales manager who wants to calculate the average sales revenue for each sales rep. They can create a pivot table which shows the sales rep and their average sales revenue.

A more complex example is a **multi-pass aggregation** (or grouping) that is an aggregation that performs multiple calculations simultaneously. Following our first example let's assume the sales manager wants to also see average sales per day for each sales rep. Instead of having to add an additional column for day in the pivot table the manager can create a multi-pass aggregation that first performs a sum of sales per day and then averages the results for each rep. This requires two fields – a day from a date field and the revenue field, as well as two aggregations, sum of sales and average. The result is the sales manager does not need to add a column for days in the pivot.

*Data Set*

Day	Sales Rep	Revenue
1 Jan	John	10
1 Jan	Jane	20
1 Jan	Jane	40
2 Feb	John	30
2 Feb	John	10
2 Feb	Jane	5

*Results Example 1: Avg(Revenue)*

Sales Rep	Avg Revenue
John	$(10+30+10)/3 = 16.67$
Jane	$(20+40+5)/3 = 21.67$

*Results Example 2: Avg(Days, Sum(Revenue))*

Sales Rep	Avg Daily Revenue
John	$(10+(30+10))/2 = 25$
Jane	$((20+40)+5)/2 = 32.5$

The above example as shown in the formula editor.

Sales Rep	Average Revenue	Avg(([Days in Date]),([Total Revenue]))
Jane	21.67	32.5
John	16.67	25

COUNTRIES > EDIT FORMULA

AVG(([Days in Date]),([Total Revenue]))

COUNTRIES > EDIT FORMULA

AVG(([Days in Date]),([Total Revenue]))

# Summarize Data: Statistical Functions

Descriptive statistics provide meaningful summaries of data and help make more informed decisions. This is particularly important for large data sets where descriptive statistics can help to focus analysis.

*Statistical Syntax:* Function(Numeric Field)

*Parameters*

Function: Median(), Mode(), Largest(), Stdev(), Stdevp(), Varp(), Var(), Quartile(), Percentile()

Numeric Field: A numeric field or formula.

An example of statistical functions is a marketing team that has a large data set on leads generated from various channels and want to understand where to focus their budget. Descriptive statistics can be used to summarize valuable insight about each channel such as the central tendency or median leads generated along with standard deviations to assess typical lead volume.

*Statistical Syntax:* Function(Numeric Field)

*Parameters*

Function: Median(), Mode(), Largest(), Stdev(), Stdevp(), Varp(), Var(), Quartile(), Percentile()

Numeric Field: A numeric field or formula.

# Accumulate Data: Running Total and Average

Often, to measure performance, data must be viewed in a continuous and accumulative format over extended periods such as years, quarters or months. Sisense provides functions to create running totals and averages over standard or custom time periods.

*Accumulate Syntax:* Function(Numeric Field)

*Parameters*

Function: YTD Avg(), QTD Avg(), MTD Avg(), YTD Sum(), QTD Sum(), MTD Sum(), RPSum()/ RPAvg()

Numeric Field: A numeric field or formula.

For example, a support team has a goal to reduce the average monthly cost to resolve open issues. A Year to Date Average can be used to track progress towards reducing the average cost of support.

*Data*

<i>Month</i>	<i>Cost</i>
Jan	30
Feb	35
Mar	40
Apr	20
May	25
Jun	10

*Results Example 1: YTD Avg(Cost)*

<i>Month</i>	<i>YTD Avg Cost</i>
Jan	30
Feb	32.5
Mar	35
Apr	31.25
May	30
Jun	26.67

# Change over Time: Time Functions

Time is critical for business decisions, time functions make it easy to compare outcomes at different points in time, determine growth rates and calculate the time differences. Functions can be set for common time periods such as year, month or day as well as for custom periods.

*Time Syntax: Function(Numeric Field) \**

*Parameters*

Function:

PastYear(), PastQuarter(), PastMonth(), Growth(), GrowthRate(), Contribution(), YDiff(), QDiff(), MDiff(), DDiff(), HDiff(), MnDiff(), SDiff(), Next(), Prev(), DiffPastYear(), DiffPastMonth(), DiffPastQuarter(), DiffPastDay()

Numeric Field: A numeric field or formula.

*\* To calculate time functions require a time filter applied on the dashboard or widget.*

For example, an executive team wants to compare growth in revenue to the same period in the previous year. A Difference in Past Year function can be used to compare past values based on the current month to the same month in the previous year.

<i>Data</i>		<i>Results Example 1: DiffPastYear(Revenue)</i>	
<i>Date</i>	<i>Revenue</i>	<i>Month</i>	<i>Difference to Past Year</i>
01-2013	5	01-2013	
02-2013	10	02-2013	
03-2013	15	03-2013	
01-2014	25	01-2014	25-5 =20
02-2014	5	02-2014	5-10= -5
03-2014	10	03-2014	10-15 =-5

## Conditional Statements

Conditional statements are formulas that enable you to create additional categories in your data set. When the condition is met, the new category is added to your visualization without modifying the data source itself.

For example, if you want to calculate only certain values according to conditions you define, you can use a conditional statement to include only the values of interest.

Sisense provides several conditional statements that enable you to evaluate criteria and display only the relevant results in your dashboard.

**Note:** Conditional states work on measures and aggregations only and return numeric values.

# IF

IF (Boolean\_expression, true value, false value)

Evaluates a boolean expression and returns the first value when evaluated as true, and returns the second value when evaluated as false.

For example:

```
(if(Sum(Sales) < 10, 0, Sum(Sales) )
```

Nested IF statements are also supported.

For example:

```
if (Sum(Sales) < 100 , 1 , (if (Sum(Sales) < 1000 , 2, 3)
)
```

# isNull

ISNULL (<numeric value>)

Returns true if the expression doesn't contain data (Null). Can be used as a condition when writing conditional statements.

For example:

```
If (ISNULL(sum(deals)), 0, sum(deals) )
```

# Case

```
CASE
WHEN Boolean_expression THEN result_expression
[ ... ]
[ ELSE else_result_expression ]
END
```

The CASE function evaluates boolean expressions, and when evaluated as true, returns the corresponding result\_expression. If no match is found, the else\_result\_expression is returned. If there is no default returned and no values match, then Null is returned.

For example:

```
CASE
WHEN Sum(Sales) < 100 THEN 1
WHEN Sum(Sales) < 1000 THEN 2
ELSE 3
END
```

## Function Reference

This page contains a list of all the functions you can use in Sisense's formula editor.

# Statistical Functions

## Average

`Avg(<numeric Field>)`

Calculates the mean average of the given values.

For example - `AVG(Score)` will calculate the mean average of the given scores.

`Avg(<group by field>, <aggregation>)`

Calculates the average of the given aggregation grouped by another field.

For example - `Avg( Product, Total Sales)` will calculate the average of the total sales per product.

## Contribution

Contribution (<numeric field>)

Calculates the percentage of total.

For example – Contribution( Total Sales ) will calculate the percentage of total sales per group (for example per day or per product) out of total sales (for all days or all products).

## Correlation

`CORREL(<Numeric Field a>, <Numeric Field b>)`

Returns the correlation coefficient of two numeric fields.

For example – `CORREL(Revenue, Cost)` will return the correlation between revenue and cost.

`CORREL(<group by field>, <aggregation a>, <aggregation b>)`

Returns the correlation coefficient of two fields aggregations grouped by another field.

For example – `CORREL(Products, AVG(Revenue), AVG(Cost))` will return the correlation between the average of revenue and cost per product.

## Count

Count(<Numeric Field>)

Counts the number of unique values withing the given values.

## Count All

DupCount (<Numeric Field>)

Returns the actual item count of the given list of items, including duplicates.

## Covariance (Population)

`COVARP(<Numeric Field a>, <Numeric Field b>)`

Returns the population covariance of <Numeric Field a> and <Numeric Field b>.

For example – `COVARP(Revenue, Cost)` will return the population covariance of revenue and cost.

`COVARP(<group by field>, <aggregation a>, <aggregation b>)`

Returns the population covariance of two fields aggregations grouped by another field.

For example – `COVARP(Products, AVG(Revenue), AVG(Cost))` will return the population covariance of the average revenue and the average cost per product.

## Covariance (Sample)

`COVAR(<Numeric Field a>, <Numeric Field b>)`

Returns the sample covariance of <Numeric Field a> and <Numeric Field b>.

For example – `COVAR(Revenue, Cost)` will return the sample covariance of revenue and cost.

`COVAR(<group by field>, <aggregation a>, <aggregation b>)`

Returns the sample covariance of two fields aggregations grouped by another field.

For example – `COVAR(Products, AVG(Revenue), AVG(Cost))` will return the sample covariance of the average revenue and the average cost per product.

## Exponential Distribution

`EXPONDIST(<numeric value>, <lambda>, <Cumulative  
(true/false)>)`

Returns the exponential distribution for a given value and a supplied distribution parameter lambda. Cumulative: TRUE = Cumulative distribution function, FALSE = Probability density function. For example – `EXPONDIST( Count(Leads), 2, False )` will return the exponential distribution density of the number of leads per country where lambda is 2.

## Intercept

`INTERCEPT(<field>, <numeric value>)`

Returns the intercept of the linear regression line through a supplied series of x- and y- values.

For example – `INTERCEPT(Date.Quarter, Total Sales)` will return the intercept of the regression line that represents the trend over quarter of the sum of sales.

## Largest

`LARGEST(<Numeric Field>, <k>)`

Returns the k-th largest value in a field.

## Maximum

Max(<Numeric Field>)

Returns the maximum value among the given values.

## Median

`MEDIAN( <Numeric Field> )`

Calculates the median of the given values. The median of a set of data is the middlemost number in the set. The median is also the number that is halfway into the set.

## Minimum

`Min(<Numeric Field>)`

Returns the minimum value among the given values.

## Mode

`MODE (<Numeric Field>)`

Returns the most frequently occurring value from the column.

## Normal Distribution

`NORMDIST(<Numeric Field>, <Mean>, <Standard Deviation>, <Cumulative (true/false)>)`

Returns the standard normal distribution for a given value, a supplied distribution mean and standard deviation. Cumulative: TRUE = Cumulative Normal Distribution Function, FALSE = Normal Probability Density Function. For example – `NORMDIST(Score, ( Mean(Score), All(Score)), ( STDEV(Score), All(Score) ), False )` will return the normal probability density of a given score.

## Percentile

`PERCENTILE (<Numeric Field>, <k>)`

Returns the k-th percentile value from the given field.

k is any number between 0..1 (inclusive).

## Poisson Distribution

POISSONDIST( <numeric value>, <mean>, <Cumulative  
(true/false)>)

Returns the poisson distribution for a given value and a supplied distribution mean. Cumulative: TRUE = Cumulative distribution function, FALSE = Probability mass function.

For example – POISSONDIST( Score, ( Mean(Score), All(Score) ), ( STDEV(Score), All(Score) ), False ) will return the poisson probability density of a given number of sales

# Quartile

QUARTILE(<Numeric Field>, <k>)

Returns the k-th quartile for the given field.

- k = 0 returns the Minimum value
- k = 1 returns the first quartile (25th percentile)
- k = 2 returns the Median value (50th percentile)
- k = 3 returns the third quartile (75th percentile)
- k = 4 returns the Maximum value

## Rank

RANK(<numeric value>, [DESC/ASC], [Rank Type], [<group by field 1>, ... , <group by field n>])

Returns the rank of a value in a list of values.

[DESC/ASC] – Optional. By default sort order is descending.

[Rank Type] – Optional. By default the type is standard competition ranking (“1224” ranking). Support also modified competition ranking (“1334” ranking), dense ranking (“1223” ranking) and ordinal ranking (“1234” ranking).

[<Group by field 1>, ... , <Group by field n>] – Optional. Rank partitions fields.

For example – RANK(Total Cost, “ASC”, “1224”, Product, Years) will return the rank of the total annual cost per each product were sorted in ascending order.

## Running Sum (RSUM)

RSUM ( <numeric value> ), RSUM ( <numeric value> ,  
<continuous> )

Returns the running total of the measure by the defined dimension according to the current sorting order in the widget.

By default, RSUM accumulates a measure by the sorting order of the dimension. To accumulate by another order, the relevant measure should be added as an additional column and sorted.

<continuous> is a boolean value that that accumulates the sum continuously when there are two or more dimensions. The default value is False.

**Note:** Filtering the RSUM column by Values, filters the dimensions and recalculates the RSUM from the first filtered value.

## Skewness (Population)

`SKEWP(<numeric value>)`

Returns the skewness of the distribution of a given value in the population.

For example – `SKEWP(Revenue)` will return the skewness of the distribution of revenue in the population.

## Skewness (Sample)

`SKEW(<numeric value>)`

Returns the skewness of the distribution of a given value.

For example – `SKEW(Revenue)` will return the skewness of the distribution of revenue.

## Slope

`SLOPE(<field>, <numeric value>)`

Returns the slope of the linear regression line through a supplied series of x- and y- values.

For example – `SLOPE(Date.Quarter, Total Sales)` will return the slope of the regression line that represent the trend over quarter of the sum of sales.

## Standard Deviation (Population)

STDEVP( <Numeric Value> )

Returns the Standard Deviation of the given values (Population). Standard deviation is the square root of the average squared deviation from the mean. The standard deviation of a population gives researchers the amount of dispersion of data for an entire population of survey respondents.

## Standard Deviation (Sample)

STDEV( <Numeric Value> )

Returns the Standard Deviation of the given values (Sample). Standard deviation is the square root of the average squared deviation from the mean. A standard deviation of a sample estimates the amount of dispersion in a given data set, based on a random sample.

## T Distribution

`TDIST( <numeric value x>, <degrees_freedom>, <Cumulative (true/false)> )`

Returns the student's T-distribution for a given value and a supplied number of degrees of freedom (must be  $\geq 1$ ). Cumulative: TRUE = Cumulative Distribution Function, FALSE = Probability Density Function.

For example - `TDIST( Score, 3, TRUE )` will return the student's T-distribution of a given score, with 3 degrees of freedom.

## Variance (Population)

`VARP( <Numeric Value> )`

Returns the Variance of the given values (Population). Variance (Sample) is the average squared deviation from the mean, based on an entire population of survey respondents.

## Variance (Sample)

`VAR( <Numeric Value> )`

Returns the Variance of the given values (Sample). Variance (Sample) is the average squared deviation from the mean, based on a random sample of the population.

# Mathematical Functions

# Absolute

Abs (<Numeric value>)

Returns the absolute value of the given value.

For example - ABS(Cost), where the absolute result for the value '2' or '-2' is '2'.

## Acos

`ACOS(<numeric value>)`

Returns the angle, in radians, whose cosine is the given numeric expression. Also referred to as arccosine.

For example - `ACOS(Total Revenue)` will return the angle, in radians, whose cosine is the given total revenue.

## Asin

`ASIN(<numeric value>)`

Returns the angle, in radians, whose sine is the given numeric expression. Also referred to as arcsine.

For example - `ASIN(Total Revenue)` will return the angle, in radians, whose sine is the given total revenue.

## Atan

`ATAN(<numeric value>)`

Returns the angle in radians whose tangent is the given numeric expression. Also referred to as arctangent.

For example – `ATAN(Total Revenue)` will return the angle in radians whose tangent is the given total revenue.

## Ceiling

`CEILING(<numeric value>)`

Returns number rounded up, away from zero, to the nearest multiple of significance.

For example – `CEILING(Cost)`, where the result of '83.2' rounded up is '84'.

## Cos

`COS (<numeric value>)`

Returns the trigonometric cosine of the given angle (in radians).

For example – `COS(Average Angle)` will return the trigonometric cosine of the average angle.

## Cosh

`COSH(<numeric value>)`

Returns the hyperbolic cosine of the given value.

For example – `COSH(Total Revenue)` will return the hyperbolic cosine of the total revenue.

## Cot

`COT(<numeric value>)`

Returns the trigonometric cotangent of the given angle (in radians).

For example – `COT(Average Angle)` will return the trigonometric cotangent of the average angle.

## Exp

`EXP(<numeric value>)`

Returns the exponential value of the given value.

For example – `EXP(Sales)` will return the exponential value of sales.

## Floor

`FLOOR(<numeric value>)`

Returns number rounded down, toward zero, to the nearest multiple of '1'.

For example – `FLOOR(Revenue)`, where the result of '88.6' rounded down is '88'.

## Ln

`LN(<numeric value>)`

Returns the base-e logarithm of the given value.

For example – `LN(Cost)` will return the base-e logarithm of cost.

## Log10

`LOG10(<numeric value>)`

Returns the base-10 logarithm of the given value.

For example – `LOG10(Revenue)` will return the base-10 logarithm of revenue.

## Mod

`MOD(<numeric value>, divisor)`

Returns the remainder after a number is divided by a divisor.

For example – `MOD(Cost, 10)`, where the remainder of '255' divided by '10' is '5'.

## Power

`Power(value, power)`

Returns the results of the given value raised to a supplied power.

For example – `POWER(Revenue, 2)` will return revenue raised by the power of 2.

## Quotient

`QUOTIENT(<numeric value>, divisor)`

Returns the integer portion of a division.

For example – `QUOTIENT(Cost, 2)`, where the integer portion of '5' divided by '2' is '2'.

## Round

`ROUND(<numeric value>, num_digits)`

Returns number rounded to a specified number of digits.

For example – `ROUND(Revenue, 2)` will return the revenue rounded to two decimal places.

## Sin

`SIN(<numeric value>)`

Returns the trigonometric sine of the given angle (in radians).

For example – `SIN(Average Angle)` will return the trigonometric sine of the average angle.

## Sinh

`SINH(<numeric value>)`

Returns the hyperbolic sine of the given value.

For example – `SINH(Total Revenue)` will return the hyperbolic sine of the total revenue.

## Square root

`SQRT(<Numeric value>)`

Returns the square root of the given value.

For example – `SQRT(Cost)` will return the square root of cost.

## Sum

Sum(<Numeric Field>)

Calculates the total of the given values.

## Tan

`TAN(<numeric value>)`

Returns the trigonometric tangent of the given angle (in radians).

For example – `TAN(Average Angle)` will return the trigonometric tangent of the average angle.

# Tanh

TANH(<numeric value>)

Returns the hyperbolic tangent of the given value.

For example – TANH(Total Revenue) will return the hyperbolic tangent of the total revenue.

# **Time Related Functions**

## Day Difference

DDiff( <Start Time>, <End Time> )

Returns the difference between <Start Time> and <End Time> in days.

## Past Month Difference

```
DiffPastMonth( <numeric value> )
```

Returns the difference between this month's data and the data from the previous month. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

```
DiffPastMonth( <Total Sales> )
```

Returns the difference between this month's sales and previous month's sales, for the displayed time resolution. For example, for day resolution: (sales in current day - sales in same day one month back).

## Past Period Difference

`DiffPastPeriod( <numeric value> )`

Returns the difference between this period's data and the data from the previous period. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`DiffPastPeriod( <Total Sales> )`

Returns the difference between this period's sales and previous period's sales, for the displayed time resolution. Formula: (current value - compared value).

## Past Quarter Difference

`DiffPastQuarter( <numeric value> )`

Returns the difference between this quarter's data and the data from the previous quarter. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`DiffPastQuarter( <Total Sales> )`

Returns the difference between this quarter's sales and previous quarter's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one quarter back).

## Past Week Difference

`DiffPastWeek( <numeric value> )`

Returns the difference between this week's data and the data from the previous week. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`DiffPastWeek( <Total Sales> )`

Returns the difference between this week's sales and previous week's sales, for the displayed time resolution. For example, for day resolution: (sales in current day - sales in same day one week back).

## Past Year Difference

```
DiffPastYear( <numeric value> )
```

Returns the difference between this year's data and the data from the previous year. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

```
DiffPastYear( <Total Sales> )
```

Returns the difference between this year's sales and previous year's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one year back).

## Growth

Growth( <Numeric Value> )

Calculates growth over time. The time dimension to be used is determined by the time resolution in the widget/dashboard.

Formula: (current value - compared value) / compared value.

For example:

- If this month your value is 12, and last month it was 10, your Growth for this month is 20% (0.2).  
Calculation:  $(12 - 10) / 10 = 0.2$
- If this year your value is 80, and last year it was 100, your Growth for this year is -20% (-0.2).  
Calculation:  $(80 - 100) / 100 = -0.2$

## Growth Rate

`GrowthRate( <Numeric Value> )`

Calculates growth rate over time. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

- If this month your value is 12, and last month it was 10, your Growth Rate for this month is  $12/10 = 120\%$  (1.2).  
Calculation:  $12 / 10 = 1.2$
- If this year your value is 80, and last year it was 100, your Growth for this year is  $80/100 = 80\%$  ( 0.8).  
Calculation:  $80 / 100 = 0.8$

## Growth Past Month

`GrowthPastMonth( <Numeric Value> )`

Calculates the growth from the past month to the current month. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`GrowthPastMonth([Total Sales])`

Calculates the difference between this month's sales and previous month's sales, for the displayed time resolution. For example, for day resolution: (sales in current day - sales in same day one month back) / sales in same day one month back.

## Growth Past Quarter

`GrowthPastQuarter( <Numeric Value> )`

Calculates the growth from the past quarter to the current quarter. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`GrowthPastQuarter([Total Sales])`

Calculates the difference between this quarter's sales and previous quarter's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one quarter back) / sales in same month one quarter back.

## Growth Past Week

`GrowthPastWeek( <Numeric Value> )`

Calculates the growth from the past week to the current week. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`GrowthPastWeek([Total Sales])`

Calculates the difference between this week's sales and previous week's sales, for the displayed time resolution. For example, for day resolution: (sales in current day - sales in same day one week back) / sales in same day one week back.

## Growth Past Year

`GrowthPastYear( <Numeric Value> )`

Calculates the growth from the past year to the current year. The time dimension to be used is determined by the time resolution in the widget/dashboard.

For example:

`GrowthPastYear([Total Sales])`

Calculates the difference between this year's sales and previous year's sales, for the displayed time resolution. For example, for month resolution: (sales in current month - sales in same month one year back) / sales in same month one year back.

## Hour Difference

`HDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in hours.

## Prev

Prev( <Time Field> [, <N>] )

Returns the Time period Member in <Time Field> which is N periods back from the current Member. This function only works as a scope function and not by itself.

For example – This formula will return the numeric value 2 months ago:

(<Numeric Value>, Prev(<Month Field>, 2))

## Minute Difference

`MnDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in minutes.

## Month Difference

`MDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in months.

Returns whole numbers.

## Month to Date Average

MTDAvg( <Numeric Value> )

Returns the running average starting from the beginning of the month up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is quarters or years.

## Month to Date Sum

MTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the month up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is quarters or years.

## Next

`Next( <Time Field> [, <N>] )`

Returns the Time period Member in <Time Field> which is N periods after the current Member. This function only works as a scope function and not by itself.

For example - This formula will return the numeric value 2 months ahead of

now:

`(<Numeric Value>, Next(<Month Field>, 2))`

## Now

`Now(<Day from Date field>)`

Returns the value for the current time period. Supports day, month, quarter or year. The Now function receives a date dimension and its level and returns all the members in that dimension which match the current query execution time.

**Note:** This function only works as a scope function and not by itself.

The following example will return the value for the current day.

```
([Total Sales], Now([Days in Datefield]))
```

## Past Year

`PastYear( <Numeric Value> )`

Calculates the value for the same period in the past (previous) year.

For example:

- If you're looking at a specific day, you will see the value of the same day one year back.
- If you're looking at a specific month, you will see the value of the same month one year back.

**Note:** When using the Past Year function in a weeks table and using a week filter, no results are returned.

## Past Quarter

`PastQuarter( <Numeric Value> )`

Calculates the value for the same period in the past (previous) quarter.

For example:

- If you're looking at a specific day, you will see the value of the same day one quarter back.
- If you're looking at a specific month, you will see the value of the same month one quarter back.

## Past Month

PastMonth( <Numeric Value> )

Calculates the value for the same period in the past (previous) month.

For example:

- If you're looking at a specific day, you will see the value of the same day one month back.

## Past Week

`PastWeek( <Numeric Value> )`

Calculates the value for the same period in the past (previous) week.

For example:

- If you're looking at a specific day, you will see the value of the same day one week back.

## Past Day

PastDay( <Numeric Value> )

Calculates the value for the same period in the past (previous) day.

For example:

- If you're looking at a specific day, you will see the value of the same day one day back.

## Quarter Difference

`QDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in quarters.

Returns whole numbers.

## Quarter to Date Average

QTD Avg( <Numeric Value> )

Returns the running average starting from the beginning of the quarter up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is years.

## Quarter to Date Sum

QTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the quarter up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is years.

## Range

`range( <Field1>, <Field2> )`

Returns a graphical range selector for a data set where two members of the same dimension and level define the minimum and maximum values of the range.

## Second Difference

`SDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in seconds.

## Week to Date Average

WTDAvg( <Numeric Value> )

Returns the running average starting from the beginning of the week up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is years, quarters, or months.

## Week to Date Sum

WTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the week up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

Returns 0 if the active time resolution is years, quarters, or months.

## Year Difference

`YDiff( <Start Time>, <End Time> )`

Returns the difference between <Start Time> and <End Time> in years. Returns whole numbers.

## Year to Date Average

YTDAvg( <Numeric Value> )

Returns the running average starting from the beginning of the year up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

## Year to Date Sum

YTDSum( <Numeric Value> )

Returns the running total starting from the beginning of the year up to the current time period member.

The time dimension to be used is determined by the time resolution in the widget/dashboard.

**Other**

# All

All (<Field>)

Ignores the scope set on the dimension.

## CASE

```
(WHEN <condition> THEN <result_expression> [...] [ELSE  
<result_expression>] END)
```

Returns the result\_expression of the first condition evaluated as true. When no condition is true, else\_expression is returned, if one is defined.

For example, the below function will return '1' when the Total Sales value is between 100 and 1000. It will return '2' if the Total Sales value is above 1000. It will return '3' in any other case (meaning, when Total Sales are below 100).

```
CASE  
WHEN Sum(Sales) < 100 THEN 1  
WHEN Sum(Sales) < 1000 THEN 2  
ELSE 3  
END
```

## IF

IF (<condition>, <numeric expression 1>, <numeric expression 2>)

Returns numeric expression '1' when the condition is true, and expression '2' when the condition is false. Nested conditional statements are supported.

For example, if the number of unique values within the Sales values is larger than 100, the below function will return the Total Sales x 1.1 (sales increase of 10%).

Otherwise - if the number of unique values within the Sales values is lower than 100, the function will return only the Total Sales, without an increase.

```
IF(Count(Sales)>100, Sum(Sales)*1.1, sum(Sales))
```

## IsNull

(<numeric value>)

Returns true if the expression doesn't contain data (Null). Can be used as a condition when writing conditional statements.

## Ordering

`ORDERING (<expression1>, <expression2>)`

Returns the numeric order position of rows sorted into ascending or descending order, breaking ties with further arguments.

The expressions must be aggregated by applying the MIN/MAX functions as in the example below:

```
ORDERING(MIN([Sales Person Name]), MIN([Days in Transaction_Date]), -  
1*Sum([Sales]))
```

## Rdouble

```
RDOUBLE(<R expression>, <numeric value 1>, [<numeric value 2>, ..., <numeric value n>] )
```

Returns a numeric result for a given R expression and a list of numeric values (use 'args[[i]]' in the R expression to reference numeric values parameters).

The R expression is passed to the running Rserve.

The optional ordering expression determines the order in which the rows are sent to R. The argument of the Ordering parameter can be an index in your data source or you can use the ORDERING() function to determine the order of the field. For more information about the ORDERING() function, [click here](#).

For example – RDOUBLE("m <- log(matrix(unlist(args), ncol=2)); kmeans(m,3)\$cluster", [Total Cost], [Total Revenue]) will return the k-means cluster (R expression) of the args: [Total Cost] and [Total Revenue].

```
RDOUBLE(<recycle>, <R expression>, <numeric value 1>, [<numeric value 2>, ..., <numeric value n>] )
```

recycle = TRUE (default) – Results will be cached for unchanged functions and data.

recycle = FALSE – Results will not be cached. Use this option if your R code contains randomness.

## Rint

```
RINT(<R expression>, <numeric value 1>, [<numeric value 2>, ..., <numeric value n>] )
```

Returns an integer result for a given R expression and a list of numeric values (use 'args[[i]]' in the R expression to reference numeric values parameters).

The R expression is passed to the running Rserve.

The optional ordering expression determines the order in which the rows are sent to R. The argument of the Ordering parameter can be an index in your data source or you can use the ORDERING() function to determine the order of the field. For more information about the ORDERING() function, [click here](#).

For example – RINT("m <- log(matrix(unlist(args), ncol=2));

kmeans(m,3)\$cluster", [Total Cost], [Total Revenue]) will return the k-means cluster (R expression) of the args: [Total Cost] and [Total Revenue].

```
RINT(<recycle>, <R expression>, <numeric value 1>, [<numeric value 2>, ..., <numeric value n>] )
```

recycle = TRUE (default) – Results will be cached for unchanged functions and data.

recycle = FALSE – Results will not be cached. Use this option if your R code contains randomness.

## RSUM

RSUM (<numeric value>)

Returns the running total of the measure by the defined dimension, according to the current sorting order in the widget.

RSUM (<numeric value>, <continuous>)

Continuous: TRUE = Accumulates the sum continuously in the case of two dimensions or more. False by default.

### Fiscal Years

By default, Sisense calculates the beginning of the fiscal year as January 1st. In some cases, you might need to express your company's date fields according to another fiscal month, such as April 1st. Sisense allows you to easily shift your fiscal calendar to start at a different month according to your company's requirements.

This can be defined at the system level, by the Administrator, and also at the ElastiCube level, by the Data Designer. This means that you can support one fiscal calendar month for your system, such as January 1st, while a specific ElastiCube can be set according to the US tax year which begins October 1st.

When you change the fiscal calendar, this affects how time is calculated in your dashboard and you are also getting an indication, for example when filtering your widgets and dashboards by time, through the FY tag.

Filter Quarters (FY) in Date / Filter a Field ×

List

Calendar

Time Frame

Ranking

Starred

Advanced

Quarters

- This Quarter (FY)
- Last Quarter (FY)
- Last 2 Quarters (FY)
- 4 Quarters (FY) Ago
- This & Last Quarter (FY)
- This & Last 3 Quarters (FY)
- This & Next Quarter (FY)
- This & Next 3 Quarters (FY)
- Next Quarter (FY)
- Next 2 Quarters (FY)

☆ This Quarter (FY) OK

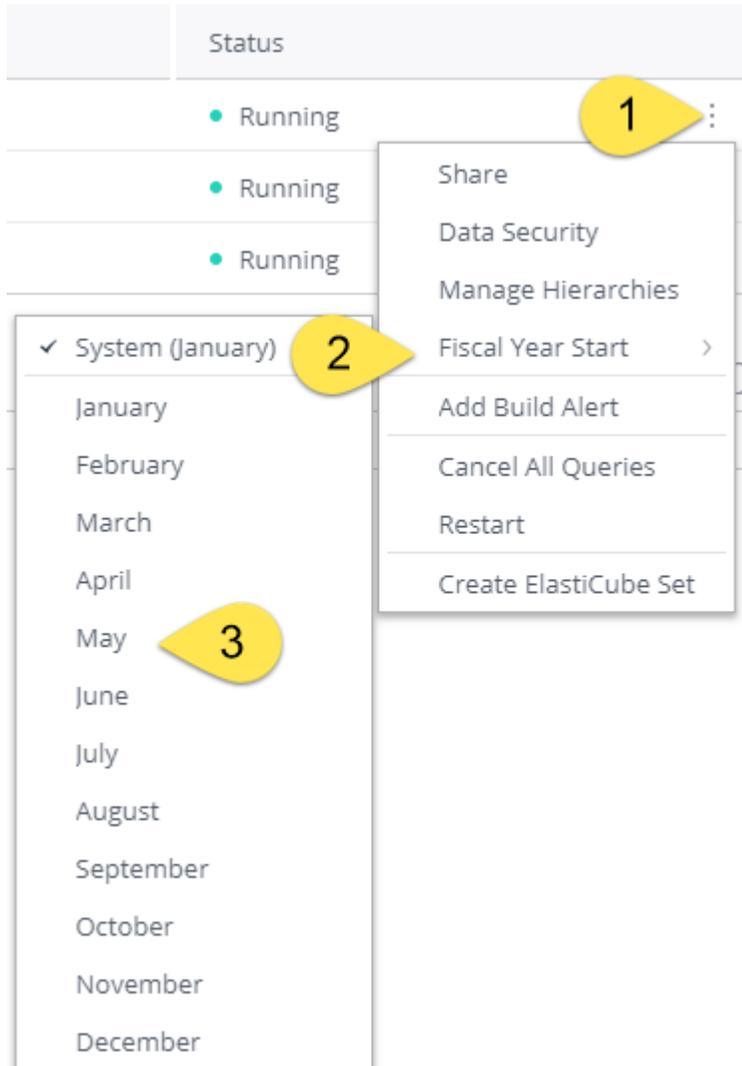
Date Level	Effects
Year	Changes the calendar year to fiscal. By default, the name reflects the year in which it ends. For example, if the fiscal year starts in April, the year for the date June 1, 2014 would be shown as FY 2015.
Quarter	Changes the calendar quarter to that of the quarter in the fiscal year. For example, if the fiscal year starts in April, the quarter for the date June 1, 2014 would be Q1.
Month	Not affected by fiscal year settings. The calendar month is the same as the fiscal month.
Week	Changes the calendar week to the week of the fiscal year. For example, if your fiscal year begins April 1st, April 3rd would fall under Week 1.
Day/Minute/Hour	Not affected by fiscal year settings.

To set your fiscal year on the system level, see [System Settings](#).

Administrators and Data Designers can set the fiscal year on the ElastiCube level.

**To set your fiscal year on the ElastiCube level:**

1. In the Sisense Web Application, select **Admin > Data Sources**.
2. For the relevant ElastiCube, select the ElastiCube menu > **Fiscal Year Start**, and then select the month.



# Formatting Fiscal Dates

When you apply fiscal years, the default format for your dates is 2018 FY for years or Q1 2018 FY for quarters.

You can customize the format of the date through placeholders such as “y” and “p” where “y” represents the year and “p” the previous year.

The examples below illustrate how you can format your dates:

- 2017/18: yyyyp/yy
- 17/18: yp/yy
- FY17/18: FY yp/yy
- Q1 2018 FY: Q yyyy FY

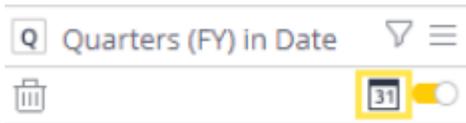
By default, Sisense applies FY at the end of the year when fiscal years are applied. However, if you make any changes, Sisense will no longer add the FY automatically, so you must define this with the placeholder FY in the location you want as shown above.

## To format your fiscal dates:

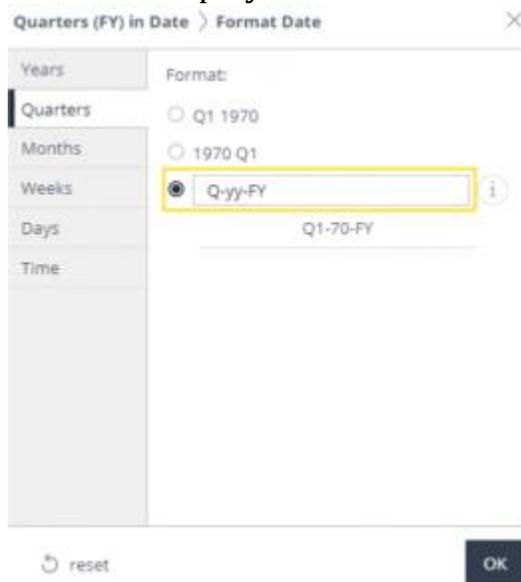
1. On the Dashboard, click the **Pencil (Edit)** button  that appears in the top-right corner of a widget with a date dimension.



2. Select the Calendar icon to define your formatting.



3. Select the relevant time period and enter the relevant placeholders in the format you want to be displayed in your widget. A preview of the custom format is displayed below.



4. Click **OK**. The format of your dates is updated in the widget.

# Formatting Fiscal Dates through the REST API

In addition to formatting Fiscal dates through Sisense, you can also define the format through the Sisense REST API.

The settings/system endpoint contains the Fiscal object that has two keys, **month** and **format**.

```
{
  "_id": "string",
  "alias": "string",
  "firstday": "string",
  "fiscal": {
    "month": "jan",
    "format": "string"
  },
  "dashboardAdministration": {
    "enabled": true
  },
  "email": {
    "enabled": true,
    "senderEmail": "string",
    "senderName": "string"
  },
  "webServer": {
    "enableSSL": true,
    "cors": {
      "enabled": true,
      "allowedOrigins": [
        "string"
      ]
    }
  }
}
```

The **month** key determines the first month of your fiscal year. The **format** key determines how your fiscal dates are displayed.

For example:

**years:** fiscalFormatRestAPI + ' yyyy';

**quarters:** fiscalFormatRestAPI + ' yyyy Q';

**months:** 'MM/yyyy ' fiscalFormatRestAPI;

**weeks:** 'ww ' + fiscalFormatRestAPI + ' yyyy';

**days:** 'MM/dd/yyyy ' + fiscalFormatRestAPI;

If you were to set the format to the following, 'EEE d/MMM//QQ//yp-yy', the format would be displayed in Sisense as follows:

FY Mon 1/Apr/Quarter 2/96-97 1997 Q2

In addition, you can add strings to the format as long as they are in single quotes.

To modify your fiscal date formats through the REST API, you can send a POST request to the settings/system API.

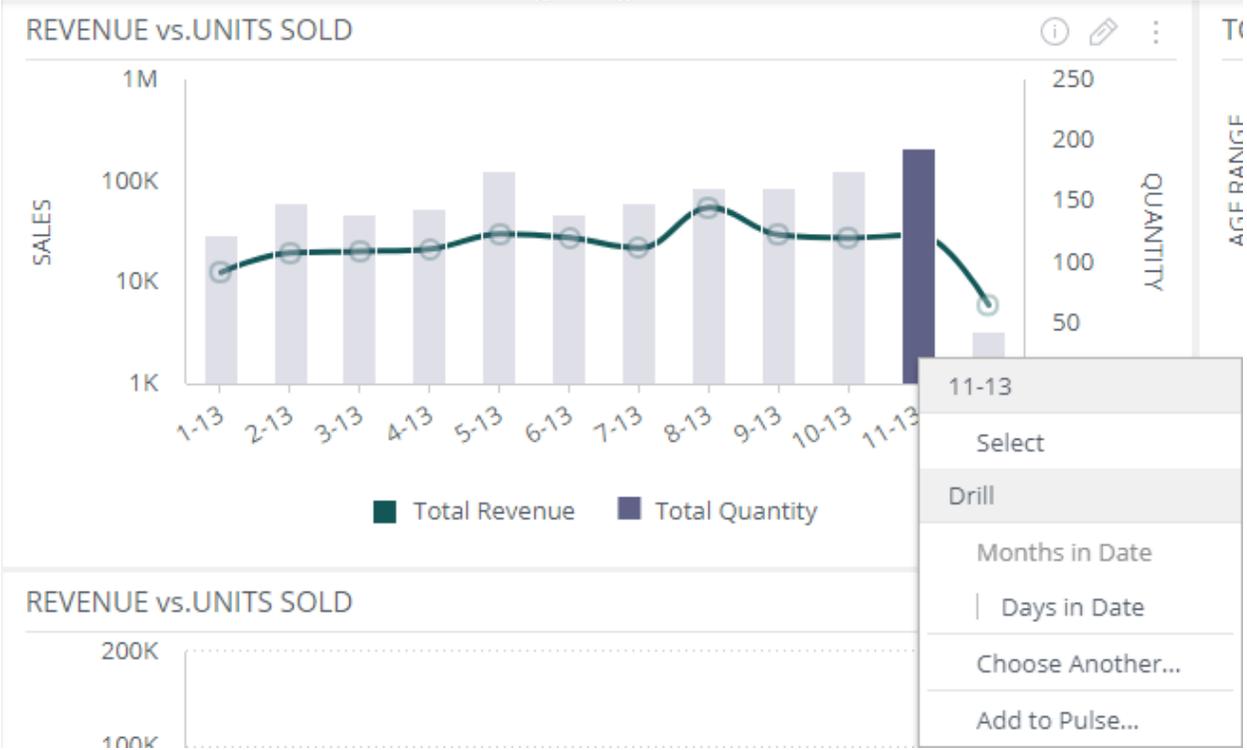
## Date and Time Fields

Many widgets include data that is relative to date and time fields. Sisense enables you to apply date and time filters so you can easily identify trends in your data.

When you create a widget in your dashboard that includes a date field, Sisense automatically breaks the data down by year, however, you can change this to a shorter resolution. For example, a Date field included in a widget can be separated into Years, Quarters, Months, Weeks, Days, and Time from your widget's filter.

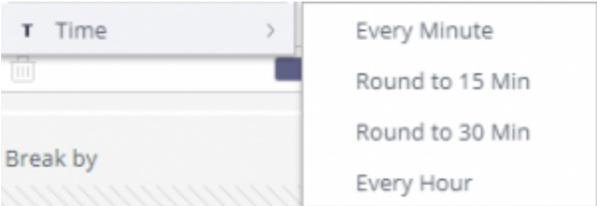


Dashboard Viewers have the option to drill down the widget according to the Date level as shown in the following image:



Viewers can drill down into the lowest date or time resolution available in your data.

The highest resolution available is by Year and the lowest resolution is Time, which includes Hour and Minute intervals.



For more information about filters, see [Creating Dashboards Filters](#).

## Customizing the Dashboard Layout

This topic describes how to rearrange your dashboard layout by moving your widgets around and resizing them.

By default, widgets are arranged in a dashboard one underneath the other. When you add a new widget, it is added at the bottom of the dashboard.

You can drag widgets one on top of another to split them horizontally by selecting the widget's title and moving the widget to the relevant location.

The dashboard can also be organized in columns. By default, a dashboard is created with a single column, but you can add up to four columns. Each column can contain multiple widgets, but a widget cannot span more than one column.

You can rearrange the order of the widgets, reorganize the widgets in columns, add/remove widgets from columns, and resize columns/widgets.

# Layout Mode and View Mode

A dashboard can be viewed in either Layout mode or in View mode.

## Layout Mode

Layout mode is the default mode in which you can add new widgets and rearrange them on the dashboard, as described throughout this section.

 is displayed in the top-right corner of the screen when the dashboard is in Layout mode. Click this button to display View mode.

## View Mode

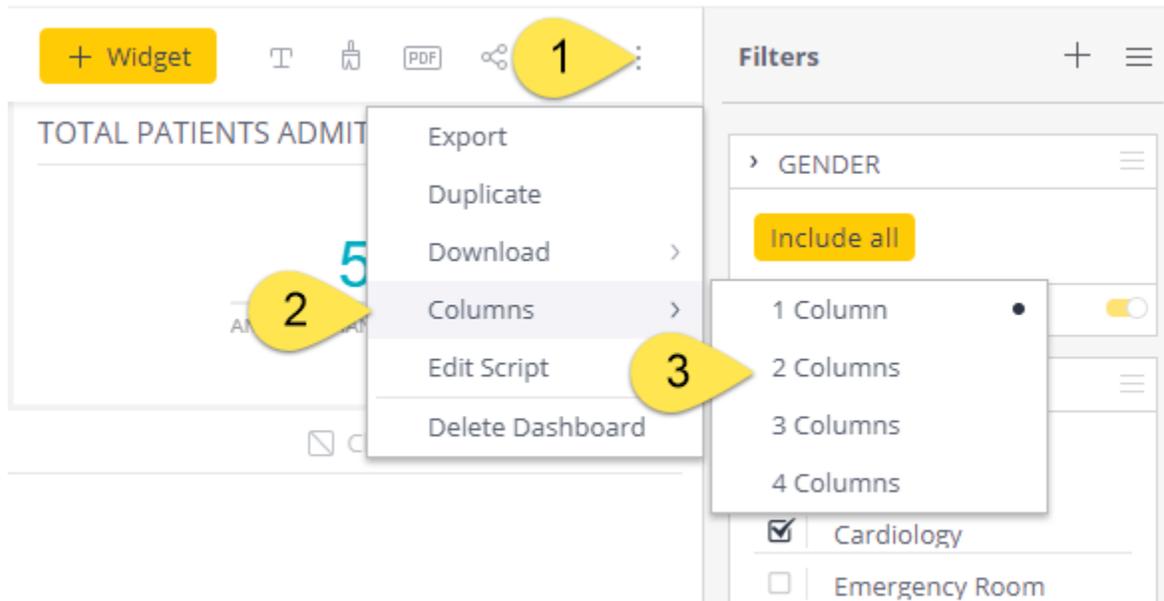
View mode enables you to see what the dashboard looks like when a user is only viewing the dashboard. In View mode, the dashboard cannot be rearranged or edited.

 is displayed when the dashboard is in View mode. Click this button to display Layout mode.

# Adding Dashboard Columns

To change the number of columns:

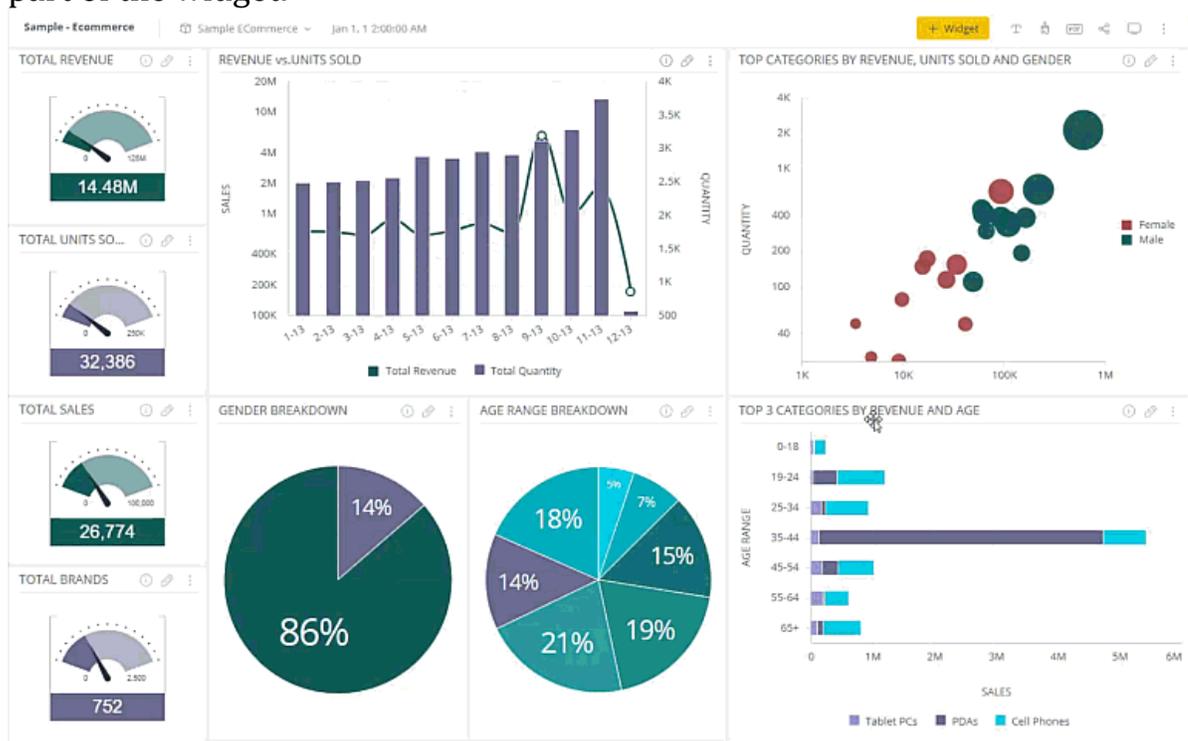
- Click on the dashboard's menu, and select Columns and the number of columns.



# Moving a Widget on the Dashboard

**To move a widget to a different column or position:**

1. Make sure that you are in Layout mode (default mode). The Layout mode indicator  is displayed in the top-right corner of the dashboard. If not, click  to go from View Mode to Layout mode.
2. Drag and drop the widget into the desired position by dragging it by the top part of the widget.



The position of a widget can be split to contain multiple widgets within a single column by dragging one widget on top of another.

# Placing Widgets Side-by-Side

You can place widgets side-by-side using one of the following methods:

- By putting each widget in a different column.

OR

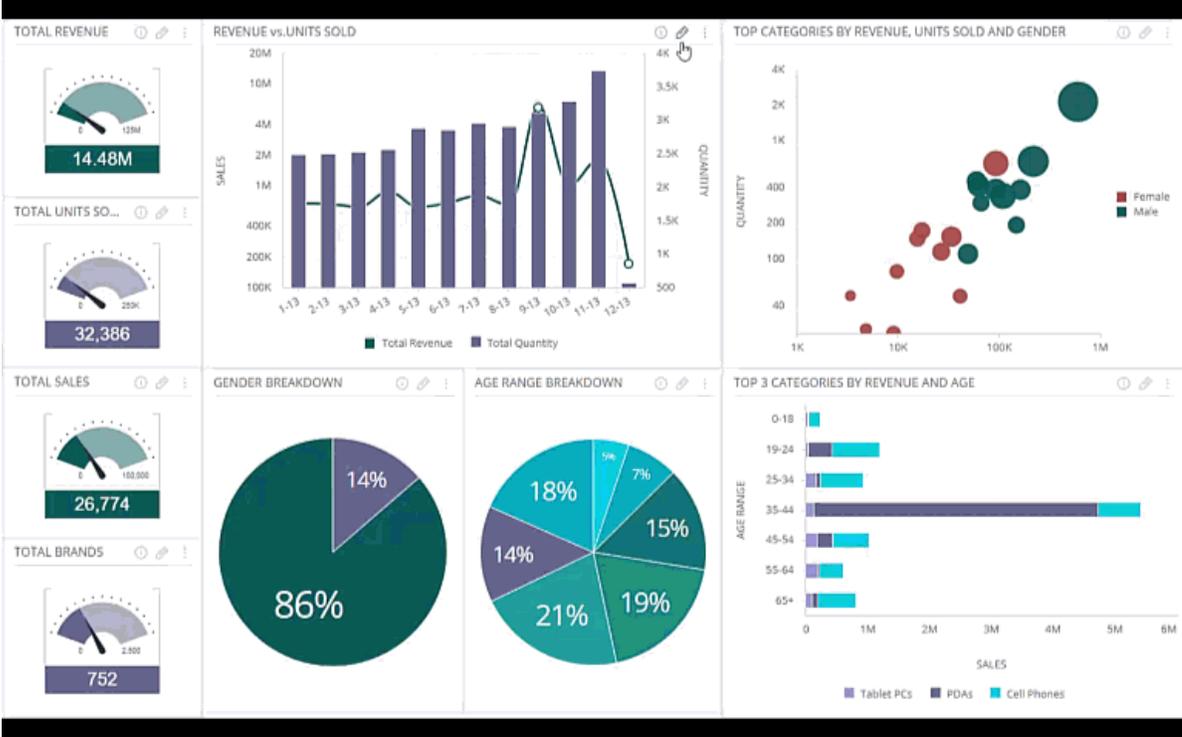
By dragging one widget on top of the other. Both these widgets will then be in the same column.

# Resizing Widgets

Resizing the column height or width automatically resizes the widgets in the best possible way.

## To resize a widget:

- Resize the dashboard column in which the widget is located by dragging the edge of the column right or left. All the widgets in this same column are resized accordingly in an optimal manner.
- Resize the widget by dragging its edges right, left, up or down.



Refer to Ticker widgets below for a description of how an Indicator Widget turns into a ticker type widget when you reduce its height.

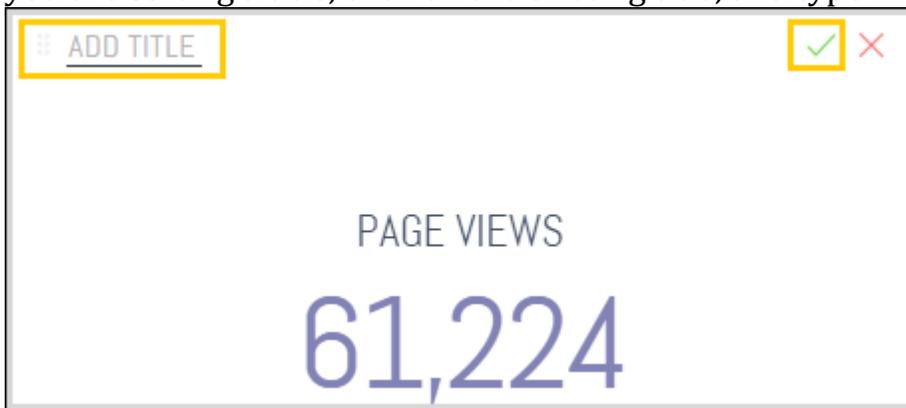
# Adding Widget Titles

You can add titles to your widgets directly in the dashboard view.

If you already added a title when you created the widget in the Widget Designer, you can edit the title directly in the dashboard.

## To add a widget title from the dashboard:

1. To add a title, click **Add Title** at the top of the widget, and type in the title. If you are editing a title, click on the existing title, and type in the new title.

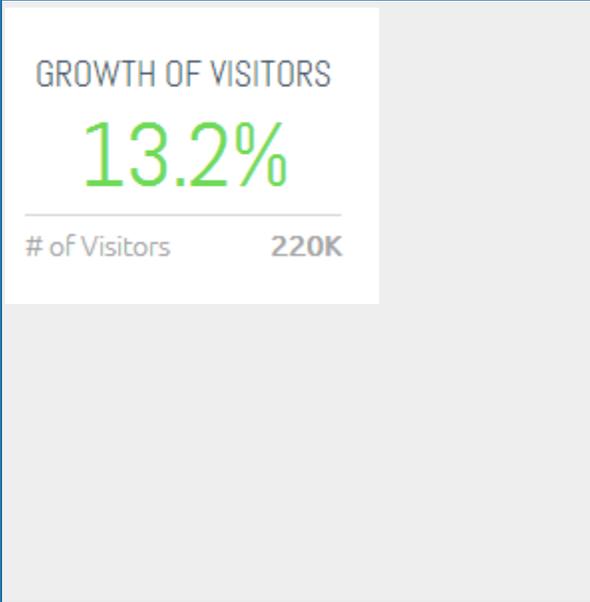
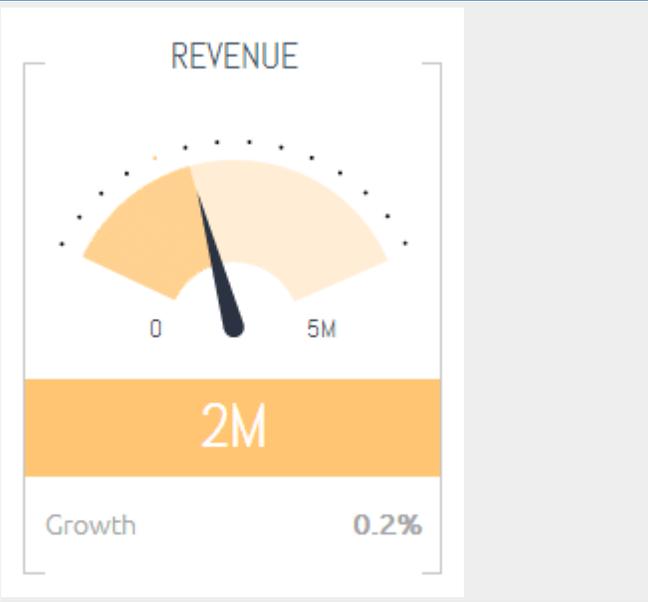


2. Click on the green  icon to save the title.

# Working with Ticker Widgets

If you resize the height of an Indicator widget, it automatically turns into a Ticker-type widget. An Indicator widget that was a numeric Indicator appears differently to an Indicator widget that was a gauge.

## Regular Indicators

	
Full size Indicator widget	Full size Gauge widget

## Ticker Indicators

	
Indicator widget after resizing down	Gauge widget after resizing down

## Changing the Dashboard's Color Palette

You can easily change the color scheme of the dashboard to suit your professional or brand preferences.

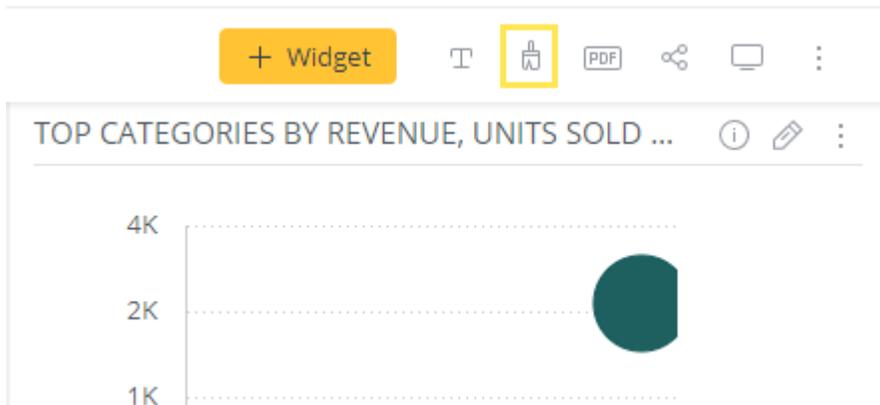
**Note:** The following procedure explains how to change the existing palette by way of selecting a different predefined palette. If you want to create a custom

palette, it is possible using the REST API. [Click here](#) to learn more about using the REST API. To learn more about creating a custom palette using the REST API, see this [support article](#).

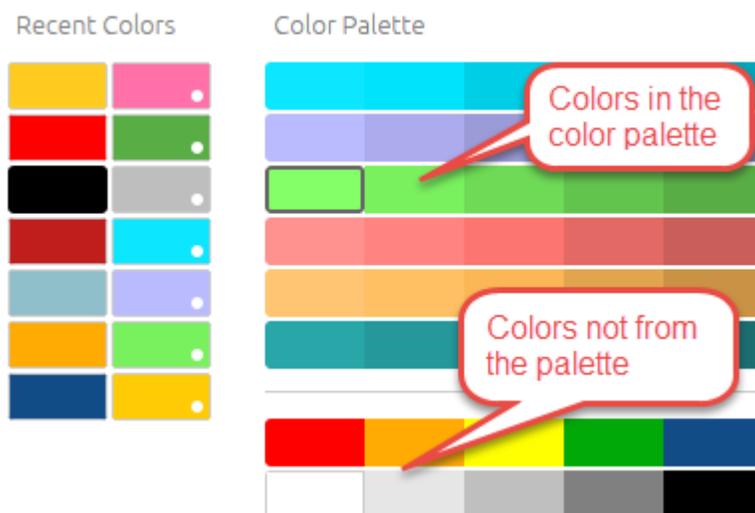
**To change the color palette:**

- Click  in the top menu, and select the preferred color palette. The default palette is Vivid.

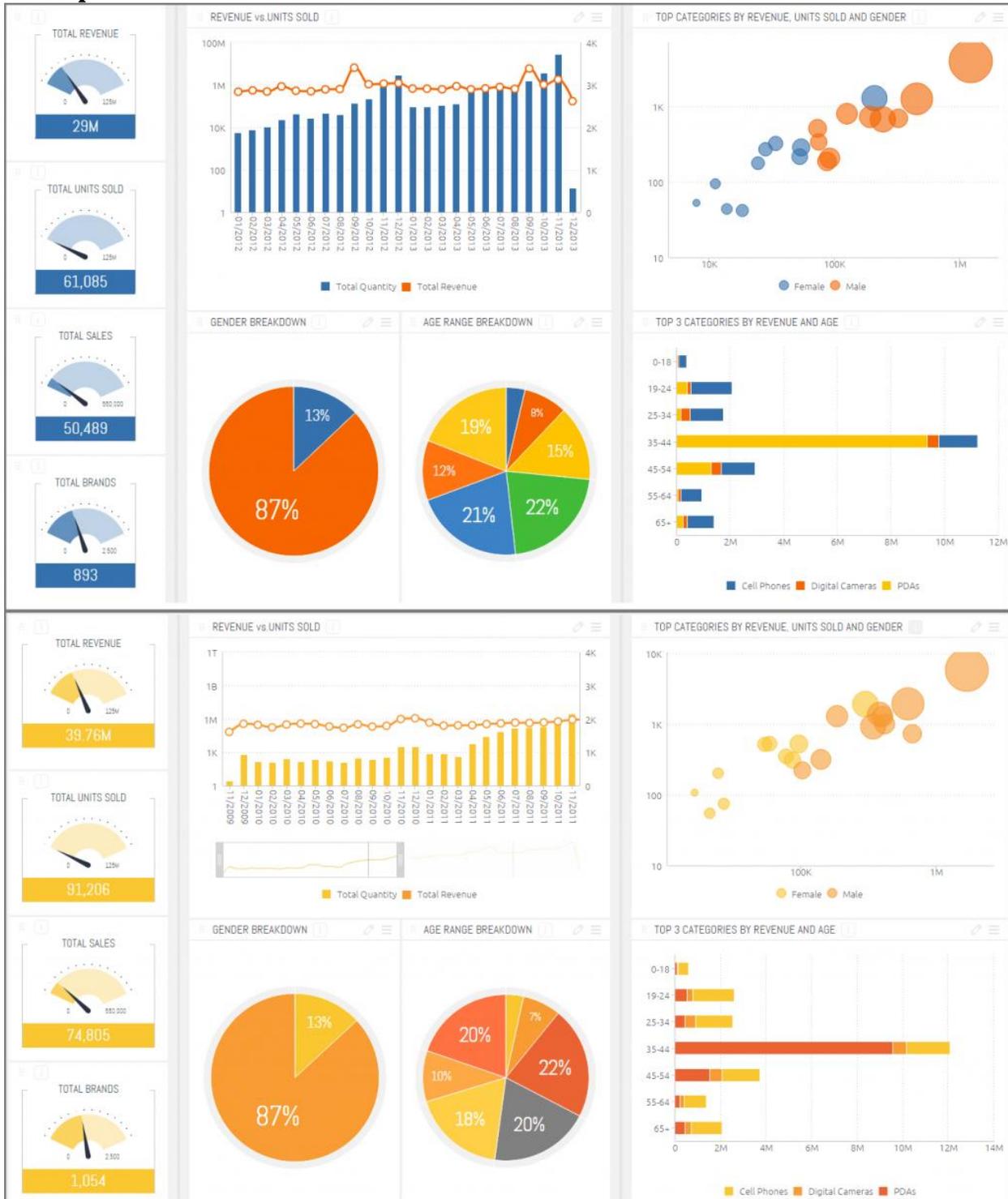
**Note:** Three of the palettes are suitable for color blindness: ColorBlind 1, 2, and 3.



**Note:** When you change the color of an individual widget to a new color that is not one of the palette colors, the new color will be applied to the widget regardless of the applied palette. If you select a new color from one of the palette colors, then when changing the dashboard's palette, the color will change according to the new palette.



# Examples

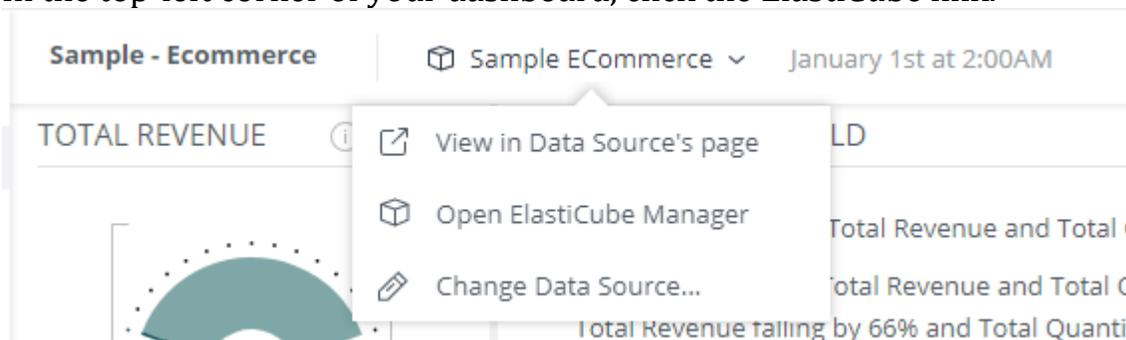


## Changing a Dashboard's Data Source

Changing a dashboard's data source is useful when you have recently changed your server and you need to reassign your dashboard's data source or in cases where you have deleted a data source and you want to reassign the dashboards. Dashboards can also support multiple data sources including ElastiCube and Live data models, which means you can have widgets built on several data sources within a single dashboard. For example, if you have widgets from an ElastiCube on a dashboard, you can add a data set to the dashboard and create widgets built on that data set's live data source. The widgets from both data sources continue to function independently of each other in the same dashboard. In addition, filters applied to fields from each data source, do not affect the fields from another data source. For more information about filtering, click [here](#).

### To change your dashboard's data source:

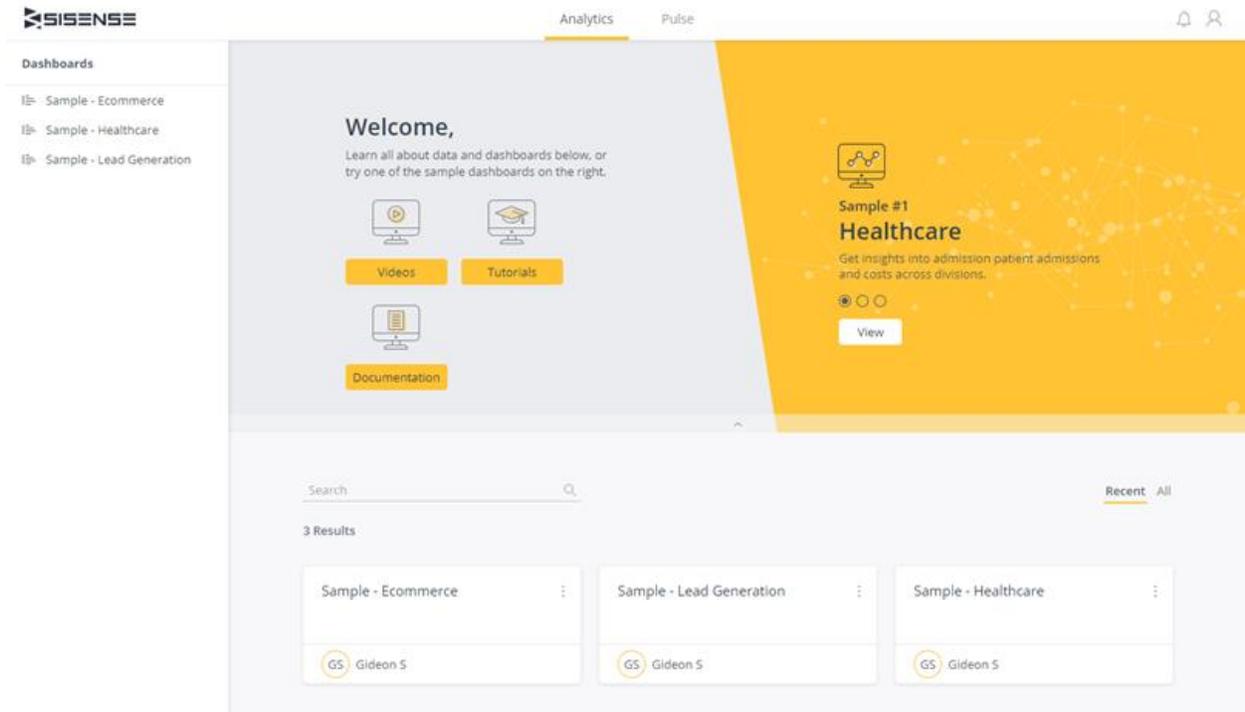
1. In the top-left corner of your dashboard, click the ElastiCube link.



2. Hover over Change Data Source and select the new data source. The source is added to your list. You now add new widgets from any of your data sources to the dashboard.

## Exploring Dashboards

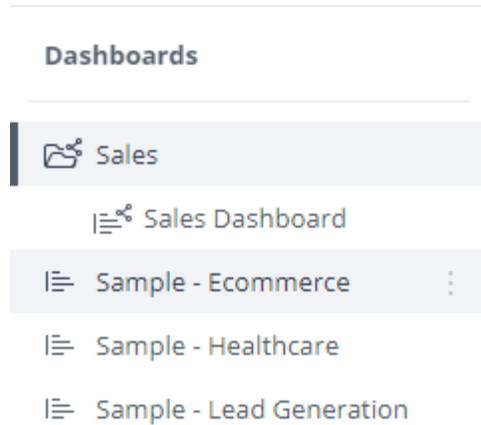
In the **Analytics** page, you can see dashboards that you created or were shared with you on the left or the main dashboard area below.



To open a dashboard, you can click the title of the dashboard on the left side or click the dashboard tile below. This opens up the dashboard inside the **Analytics** page.

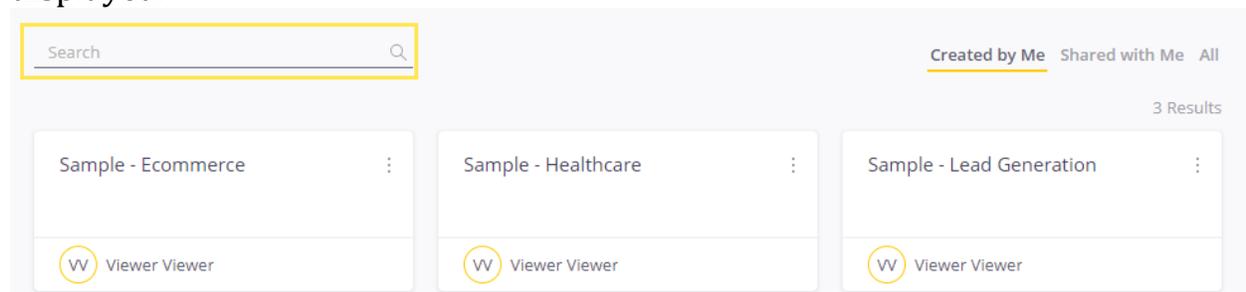


A dashboard that was shared with you appears with this  icon in the Dashboards list.



To open a dashboard, you can click the title of the dashboard on the left side or click the dashboard tile below.

Alternatively, if you have a lot of dashboards, you can search for dashboards by typing the title in the **Search** field. As you begin typing, any relevant results are displayed.



To locate a dashboard by name, source, or owner, you can use search operators within the **Search** field. Sisense supports the following operators:

What You Can Search By	Search Operator and Examples
Dashboard name	Name: <b>Example:</b> Name:Sales
Data source	Source: <b>Example:</b> Source:Sample eCommerce
Dashboard Owner	Owner: <b>Example:</b> Owner:John

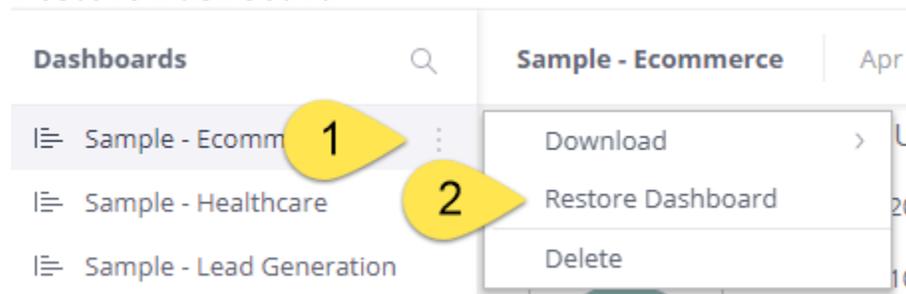
## Interacting with Filters as a Viewer

Once you've found the dashboard you're looking for, you can start exploring your data.

Sisense dashboards usually have one or more filters that affect the dashboard widgets.

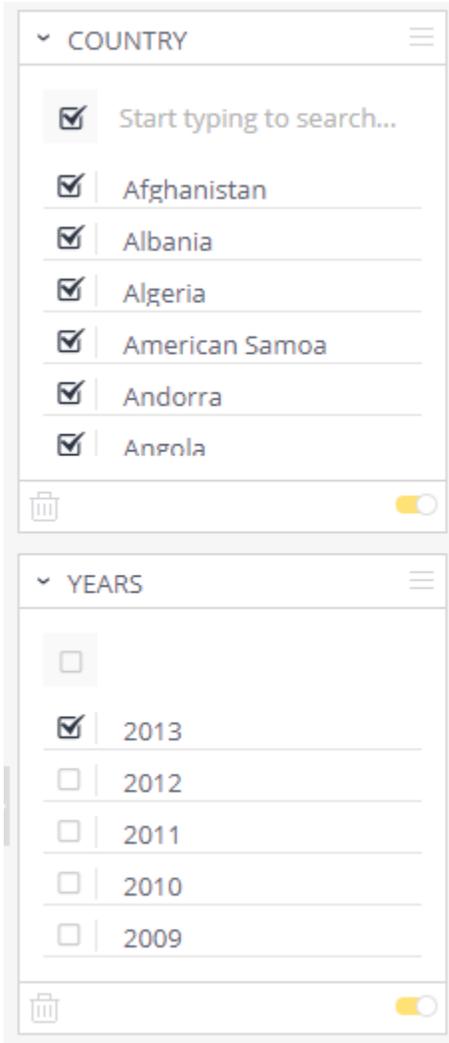
This topic describes how Viewers can interact with filters for analyzing data.

You can interact with filters for analyzing data, either through the Filters pane or by simply selecting values by left-clicking the widget visualization. Each time you interact with a filter, for example by selecting or entering a value, the filter is immediately applied to your dashboard. You will not affect anyone else by changing the filters. In addition, you can restore the original state of the dashboard at any given point by selecting the dashboard menu item and clicking **Restore Dashboard**.



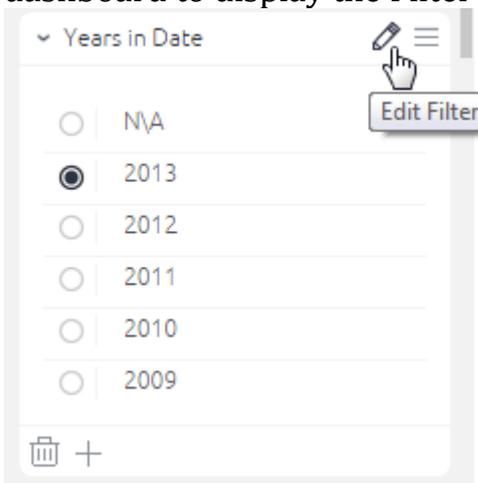
As a Viewer, you can interact with filters as follows:

- Make a different selection in the filter controls displayed in the Filters pane on the right side of the dashboard, as shown below:



OR

- Click on the pencil icon next to the filter name (shown below) in the dashboard to display the Filter Definition window.



In addition, you can perform the following procedures:

- [Switching Filters On and Off](#)
  - [Saving your Default Filters View](#)
- To create a filter, you must be a [Designer](#).  
[Click here](#) to learn more about filters.

## Changing Sisense's Language

This topic discusses how Sisense Administrators can change the language of the Sisense Web Application for all of their users.

Administrators can set the default language for a Sisense server from the Admin section. If you are an Administrator and want to set the language for your system, users groups, or users, see Changing the Sisense Web Application's Language.

If your users would like to set another language for the Sisense Web Application other than the language defined by the Administrator, they can select a language from their User Profiles or from the Login page.

Sisense supports the following languages:

- English
- Chinese
- Dutch
- French
- German
- Italian
- Japanese
- Portuguese
- Russian
- Spanish (LA)
- Spanish (Spain)

In addition, any languages that your company have translated will appear in the list according to the name defined in the translated system files.

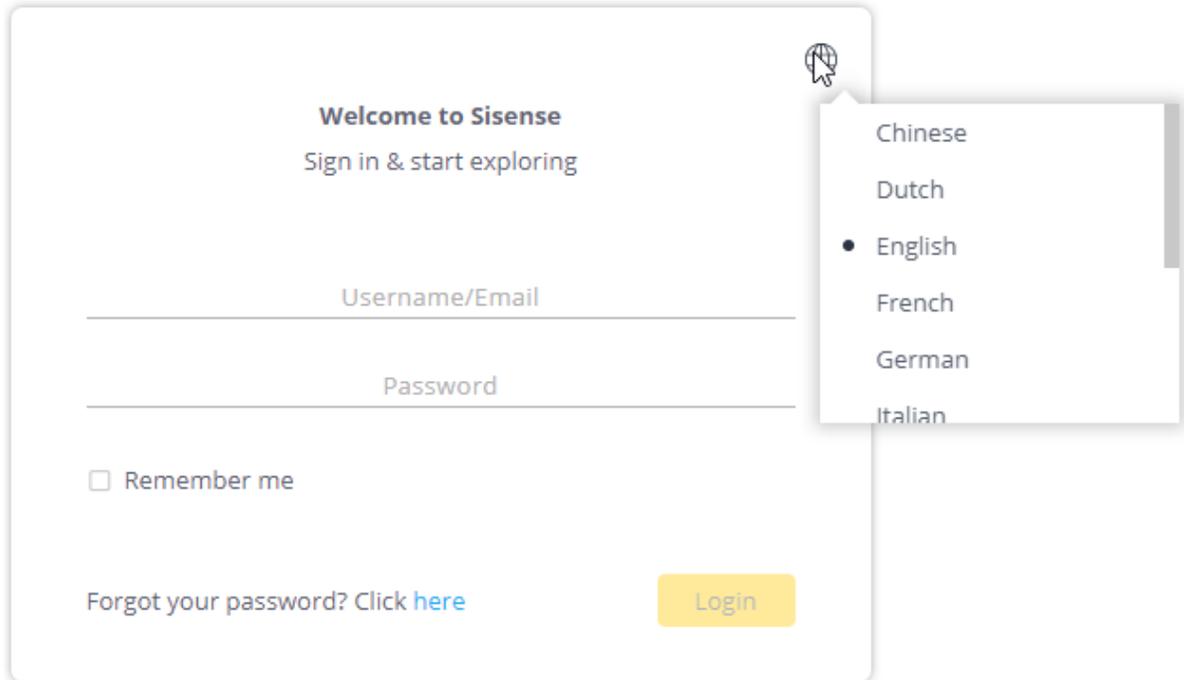
Note, the following content is not being translated:

- Built-in Javascript editor
- ElastiCube Manager and Server Console
- Sisense documentation and online help
- REST API and documentation

- Sisense automated emails

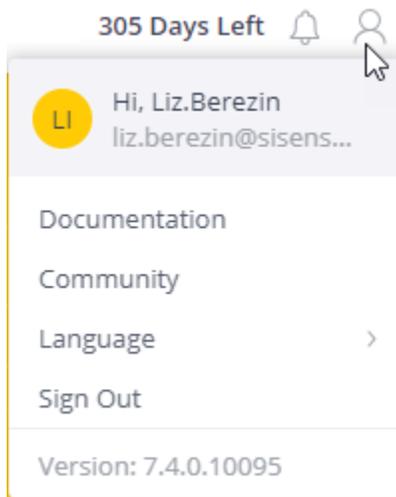
**To change your system's language:**

- Select the language from the Login page of the Sisense Web Application.



OR

In the Sisense User Options, open the languages list and select the language you want to display.



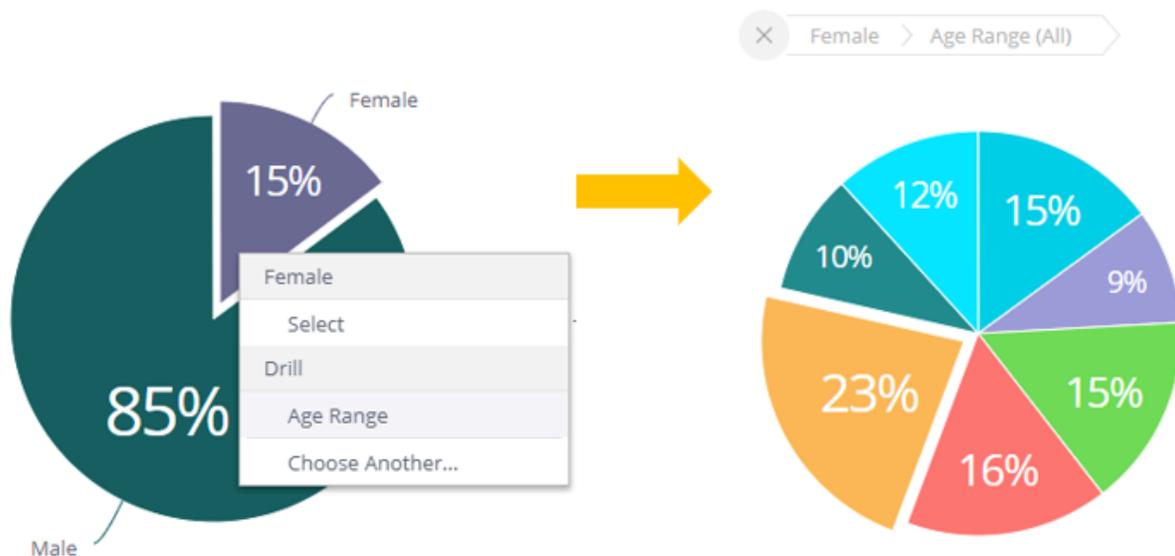
After setting the user's language, Sisense will automatically be displayed in the selected language in future sessions.

## Drilling Down in a Widget

This topic describes how you can drill down into your data in most widgets to get an in-depth view of a selected value.

In the following example, the pie chart on the left shows a breakdown by gender. This is the original chart as created by the dashboard's Designer. On the right side, the chart shows a breakdown or drill down showing age groups of the 'female' segment from the original pie chart.

GENDER BREAKDOWN



With Sisense you can drill down from any field to any field, unless disabled by the dashboard's Designer.

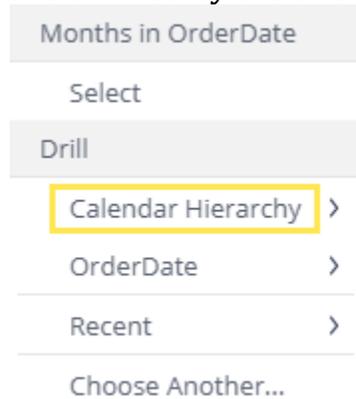
For information about enabling or disabling the drilling feature, see [Drill Hierarchies](#).

### To drill down into a chart:

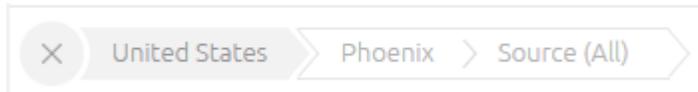
1. Right-click on the item in the widget into which you want to drill down.
2. To manually select the drill hierarchy path, select **Drill Into**, and then in the Data Browser, select the field into which to drill down. If you have already drilled into this chart, then you will have shortcuts to previously selected fields, or select **Choose Another Field** to select a different field for the first time.

OR

Select a predefined drill hierarchy (if available). Predefined drill hierarchies are defined by the dashboard's owner.



To drill up to a higher level, click on a breadcrumb. To drill all the way up, click on the X icon.



## Making Selections in a Widget

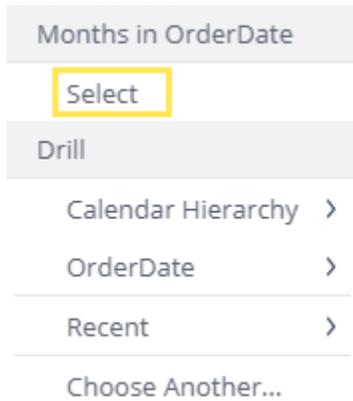
You can click on a specific portion of a widget to select it. This filters the dashboard according to the selected data by adding a filter to the dashboard's Filters panel.

You can also use the selection to drill down in the widget (see [Drilling Down in a Widget](#)).

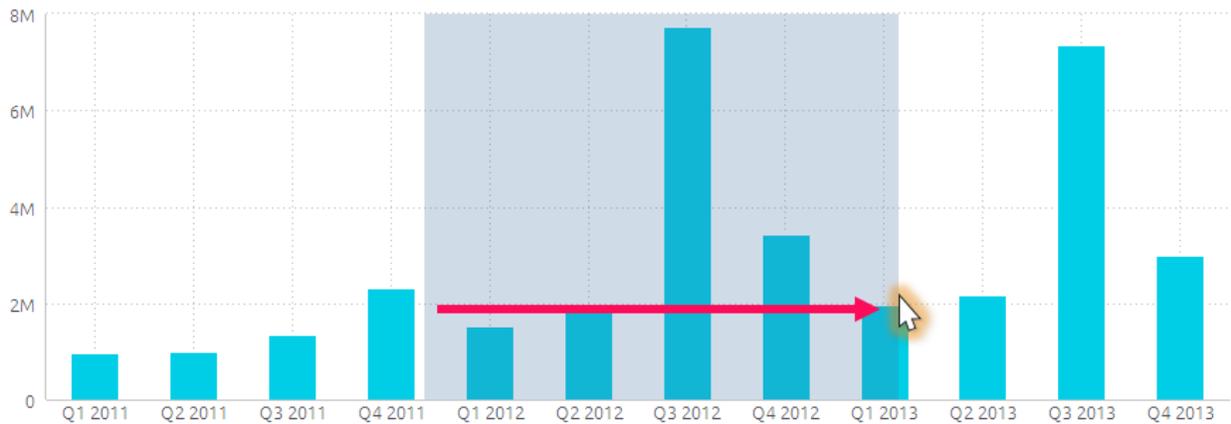
### To select an item in a widget for filtering:

- Left-click an item in a widget to automatically select it and add a dashboard filter according to the selection.

- For multiple selection, use the **Ctrl** key. When released, a menu will appear. Click **Select**.



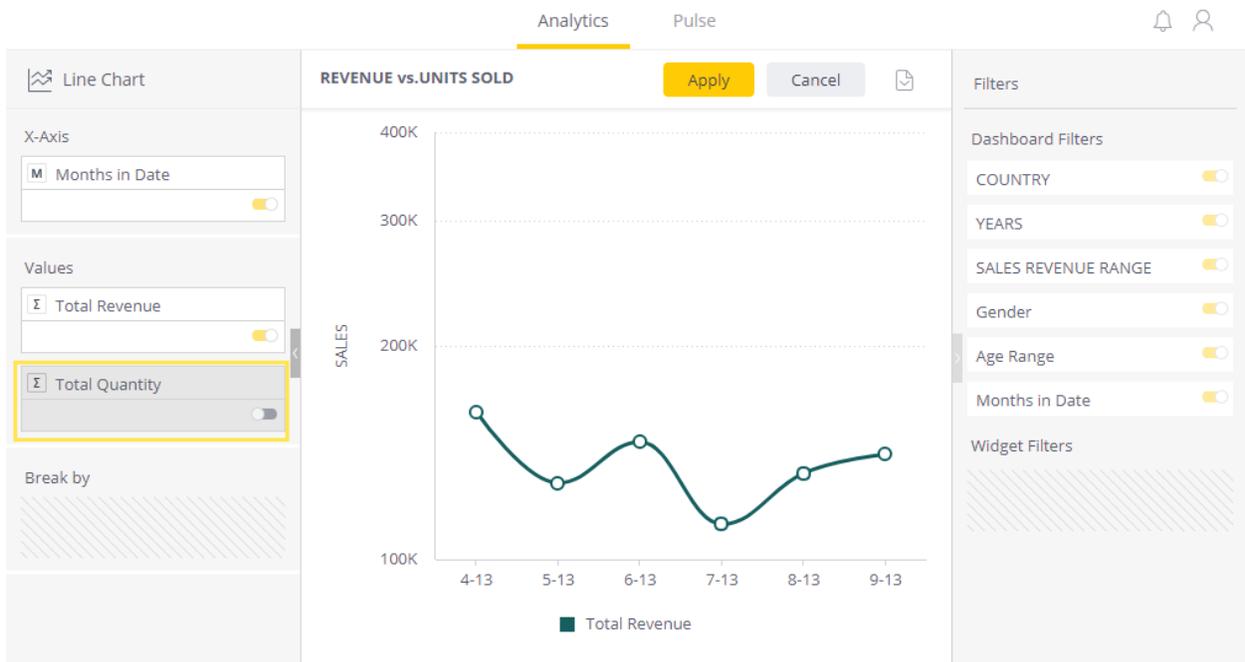
- In chart widgets, you can drag and draw the area to be selected, as shown below:



When you select a portion of the widget,  appears in the widget's menu bar. Click  to expand your view of the selection.



You can refine your data further by toggling the switches on the left side as shown in the example below.



## Reporting in Sisense

After you have viewed your dashboards, you may have discovered some insights you want to share. Sisense supports a variety of ways you can share your data with other Sisense users.

Downloading widgets and dashboards as various file types is useful for sharing your data with users or including in your reports and presentations, however, Sisense makes it easy to share dashboards directly with other Sisense users by allowing you to share the dashboard's URL from the Sisense Web Application or sending your insights via email.

In addition, you can schedule automatic reports to be sent to yourself or other Sisense users. See the [Sending Email Dashboard](#) reports for more information.

The topics below describe how you can share widgets and dashboards with other Sisense users:

- [Exporting Widgets](#)
- [Exporting and Sharing Dashboards](#)
- [Downloading Dashboards as an Image](#)
- [Creating PDF Reports](#)
- [Customizing PDF Reports](#)

## Exporting Widgets

You can download widgets in one of the following formats depending on the widget type:

- **CSV:** Download the data described by the widget in a CSV file. If you have applied any filters, the filtered data is the data included in the file.
- **Excel:** Download the data described by the widget in an Excel file. When you export a pivot table to Excel, the data that is exported is the data that is displayed in your widget. This means that any active filters, layout structure, and masks that you have applied in Sisense to your data are exported as well. For example, if you have modified how currency, percentages, or numbers are formatted in the widget in Sisense, these will be reflected in your exported Excel file. Exporting to Excel maintains your pivot's layout and structure, and values such as sub-totals, which are not maintained when you export a pivot table to CSV.
- **Image:** You can download a widget as an image in PNG format. The image size will reflect the size of the widget on the screen when you download it. To create a larger image, enlarge the widget space in the dashboard, or open the widget in edit view, to get a bigger picture.

- **PDF:** Download the widget as a PDF. For Table widgets, you can customize how your table is displayed in your PDF including the table's orientation and page size. In the PDF Report Settings, the first 14 pages of your Pivot table are displayed in the preview window, however, when you export your Pivot table to PDF, the entire table is included, up to 10,000 rows on multiple pages. For more information, see [Customizing PDF Reports](#).

### To download a widget:

- In dashboard view, click on the widget's menu, and select **Download** and select the relevant file type.
- In edit widget view, click on the download icon, and select **Download** and select the relevant file type.

## Exporting and Sharing Dashboards

You can share your Sisense dashboards and widgets with other Sisense users to distribute your Sisense insights.

The topics below describe how to share your dashboards in a variety of formats:

- [Exporting Pivot Tables to PDF](#)
- [Exporting Pivot Tables to Excel](#)
- [Export Tables to PDF](#)
- [Exporting CSVs](#)
- [Downloading Dashboards](#)
- [Downloading Widgets as Images](#)
- [Sharing Dashboards](#)
- [Sending Email Dashboard Reports](#)
- [Email Troubleshooting](#)

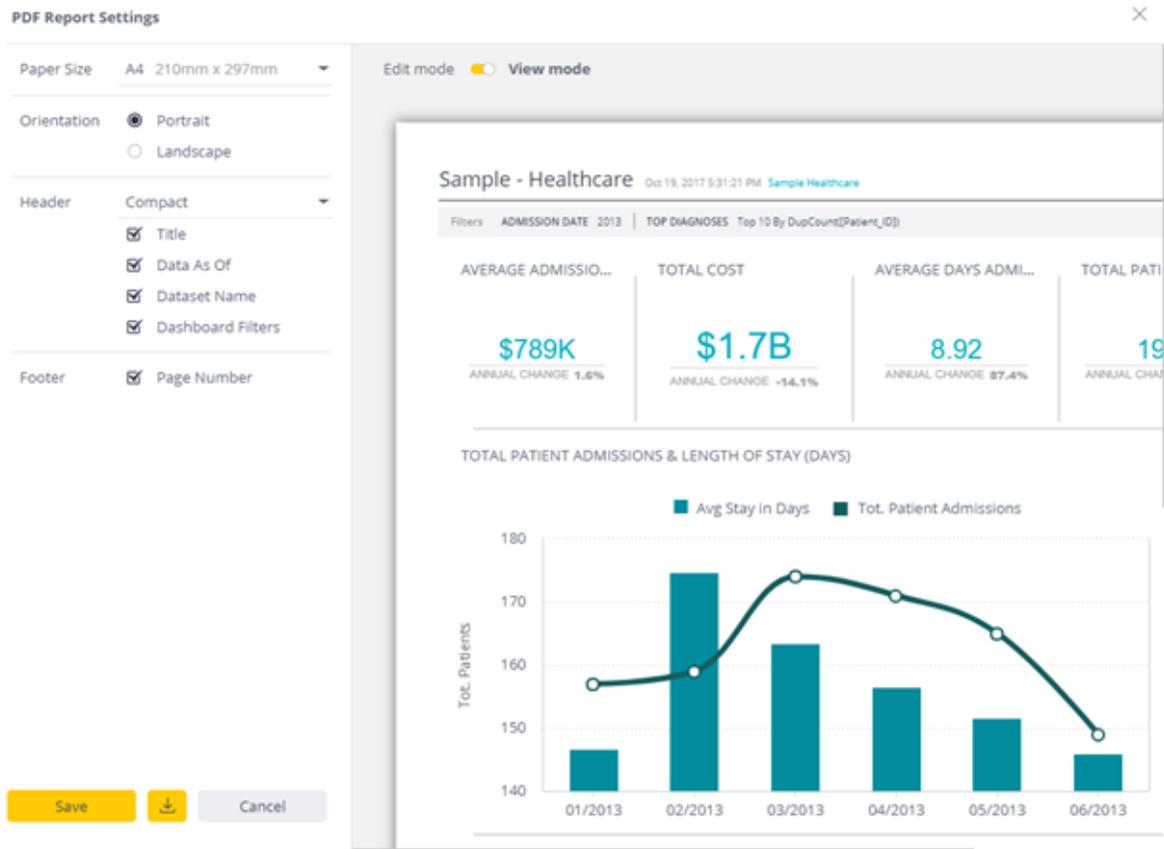
## Creating PDF Reports

When you need to take copies of your dashboards with you for meetings or sharing with others, you can generate a PDF report of your dashboard.

By clicking the  icon on your dashboard, you can customize your dashboard's appearance through the PDF Report Settings page. After you have defined how your dashboard is to be displayed, you can download your dashboard locally as a PDF.

## To create a PDF report:

1. From your dashboard menu, click the PDF icon . The PDF Report Settings page is displayed.
2. Customize how your dashboard is displayed in the PDF. For more information, see [Customizing PDF Reports](#).



3. Click . The dashboard is downloaded locally as a PDF file.

## Customizing PDF Reports

If you need to share dashboards with other users, or create a hardcopy for yourself, Sisense allows you to send an email report to your users or export your dashboard to PDF. Dashboards in email reports or PDFs though look different compared to online dashboard as they serve different purposes. Through the Sisense PDF Report Settings, you can customize and create formatted email reports and PDFs according to your requirements. The Sisense PDF Report

Settings allow you to define both the content and design of your report quickly and easily.

The screenshot displays the 'PDF Report Settings' window. On the left, there are settings for Paper Size (A4 210mm x 297mm), Orientation (Portrait selected), Header (Compact), and Footer (Page Number). The main area shows a preview of a healthcare dashboard in 'View mode'. The dashboard includes a title 'Sample - Healthcare', filters for 'ADMISSION DATE 2013' and 'TOP DIAGNOSES Top 10 By DupCount(Patient\_ID)', and four key metrics: Average Admissions (\$789K, +1.6%), Total Cost (\$1.7B, -14.1%), Average Days Admitted (8.92, +7.4%), and Total Patients (19, -14.1%). Below these is a chart titled 'TOTAL PATIENT ADMISSIONS & LENGTH OF STAY (DAYS)' showing 'Avg Stay in Days' as bars and 'Tot. Patient Admissions' as a line graph from 01/2013 to 06/2013.

Metric	Value	Annual Change
AVERAGE ADMISSIO...	\$789K	+1.6%
TOTAL COST	\$1.7B	-14.1%
AVERAGE DAYS ADML...	8.92	+7.4%
TOTAL PATI	19	-14.1%

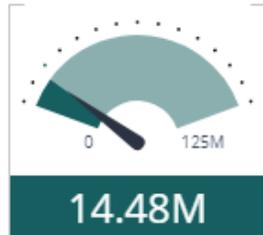
Month	Tot. Patient Admissions	Avg Stay in Days
01/2013	157	146
02/2013	159	175
03/2013	175	163
04/2013	171	156
05/2013	165	151
06/2013	149	146

From the PDF Report Settings page, you have two modes, Edit mode, for customizing your PDF, and View mode, for seeing how the dashboard will be displayed in the PDF. In Edit Mode, all the settings you can apply to your report are displayed in the left menu.

In Edit mode, you can set the size of your widgets by selecting and dragging the borders of each widget. These borders are displayed when Edit mode is toggled on.

Filters YEARS 2013 | SALES REVENUE RANGE All items Greater than 0

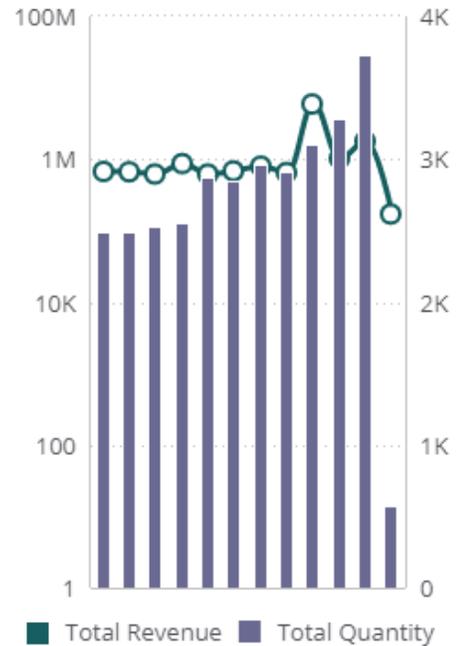
TOTAL REVENUE



TOTAL UNITS SOLD



REVENUE vs. UNITS SOLD



In View and Edit mode, you can define your dashboard’s orientation, its layout, and what is displayed in the header and footer on your report.

After you have customized your report, you can save the settings and use them when sharing reports, or download the report as a PDF.

**To customize a PDF report:**

1. In your dashboard’s menu, click . The PDF Report Settings page is displayed.
2. Edit the dashboard by selecting any of the following options:

Setting	Description
Paper Size	The size of the email report in the PDF.
Orientation	The orientation of the dashboard in the PDF, landscape or portrait. Landscape orientations display the dashboard horizontally while portrait

Setting	Description
	orientations display the dashboard vertically.
Header	<p>The design and content of your report’s header. The header is displayed on the top of each page or your PDF report.</p> <p>From the Header list, you can define how your header is to be displayed: Compact, Medium, Large, or No Header.</p> <p>After you have set the design, you can determine what information is included in the header.</p> <p><b>Title:</b> Select to display the dashboard name at the top of the PDF.</p> <p><b>Dashboard As Of:</b> Select to display the ElastiCube name and last build time at the top of the PDF.</p> <p><b>Dataset Name:</b> Select to display the name of the ElastiCube that contains the dashboard’s data.</p> <p><b>Dashboard Filters:</b> Select to display dashboard filter selections.</p>
Footer	<p>The content displayed in the footer section of each page of your report. You can display the following information:</p> <p><b>Page Number:</b> Select to display the page number on each page of the report.</p>

3. Click **Save** to save your settings when sharing reports or  to download your report.

## Managing Dashboards

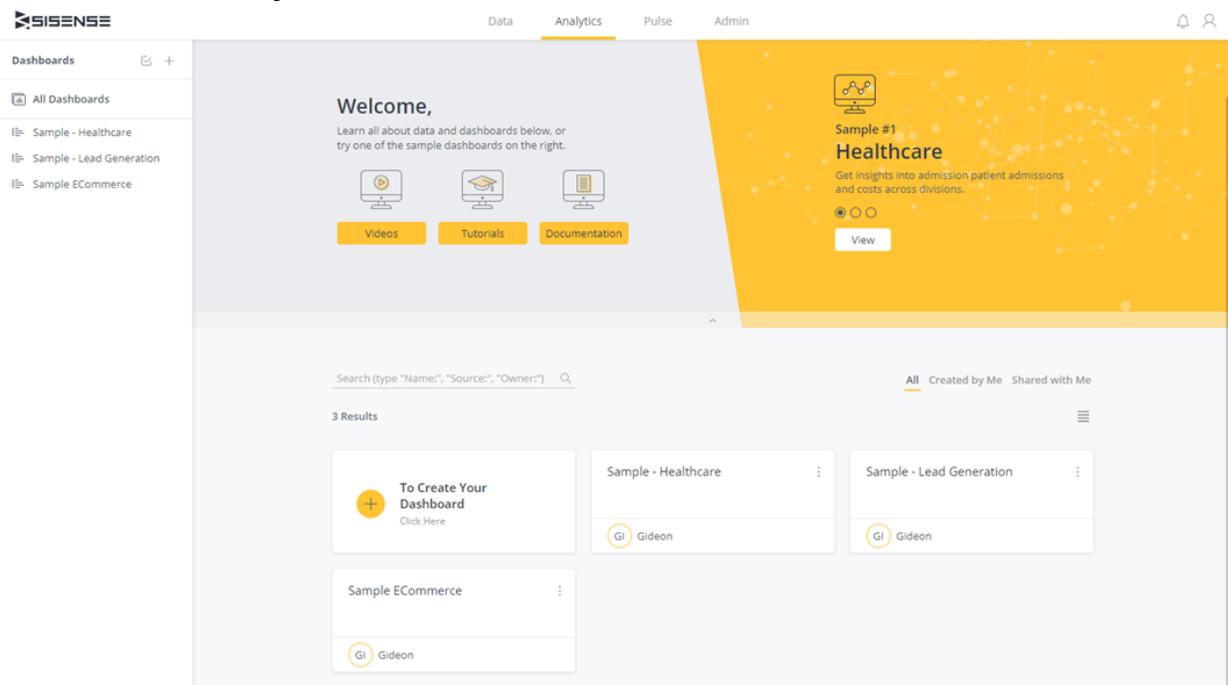
This section is for Sisense users who have dashboard editing rights (Designers). The topics in this section describe how you can manage your dashboards within Sisense:

- [Viewing Dashboards](#)
- [Locating Dashboards](#)
- [Duplicating Dashboards](#)

- [Organizing Dashboards](#)
- [Sharing Dashboards](#)
- [Deleting Dashboards](#)
- [Exporting and Importing Dashboards](#)
- For information on improving dashboard performance, see [Maximize Dashboard Performance](#).

## Viewing Dashboards

In the **Analytics** page, you can view all the dashboards that you own or that were shared with you.



There are two locations from where you can access your dashboards, on the left side in the Navigation Pane, and the Dashboard area below.

In the Dashboard area, your dashboards are displayed as clickable tiles. There are two dashboard views you can view your dashboards with.

Search

3 Results



**To Create Your Dashboard**  
Click Here

Sample - Healthcare



Gideon

**Tile View**

Search

3 Results

<input type="checkbox"/>	Name	Owner	Data Sources	Last Published	Last Modified	No. of
<input type="checkbox"/>	Sample - Ecommerce	gideon	Sample ECommerce, Sample ECommerce	Unpublished	Today at 12:49 PM	9
<input type="checkbox"/>	Sample - Healthcare	gideon	Sample Healthcare, Sample Healthcare	Unpublished	Today at 11:33 AM	7
<input type="checkbox"/>	Sample - Lead Generation	gideon	Sample Lead Generation, Sample Lead ...	Unpublished	Today at 12:46 PM	7

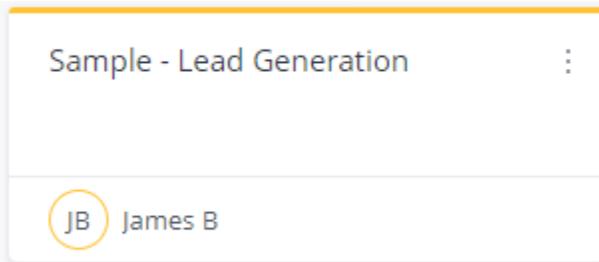
**List View**

# Tile View

The Tile view displays the name of the dashboard and its owner. You can click a tile to open the dashboard, or click its menu to see a variety of options. The options displayed are determined by your role and whether or not you own the dashboard.

When you see a yellow tag attached to the tile this indicates that the dashboard is in your system, but you are viewing the owners version, and not your version.

This tag is only visible for Administrators. For more information see [Managing Your Users Dashboards](#).



# List View

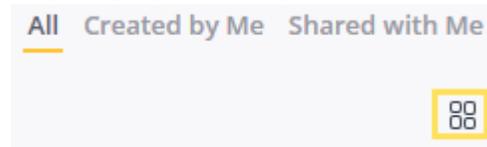
The Dashboard List provides information about your system adoption, such as the number of Viewers of each dashboard, and the last time a dashboard was modified.

The Dashboard List provides the following information:

- **Name:** The name of the dashboard.
- **Owner:** The name of the dashboard's owner.
- **Data Sources:** The data sources that are being queried by the dashboard.
- **Last Published:** The last time the dashboard was published.
- **Last Modified:** The last time the dashboard was modified. Any changes to the dashboard or to the dashboard metadata (share list, and reporting schedule) affect the “last modified” timestamp.
- **Total Viewers:** This column is only displayed to Administrators. The total number of Viewers who have opened the dashboard. This number may be lower than the number of users the dashboard is shared with if some of them have not opened the dashboard. This number does not include Administrators who opened a dashboard that was not owned by or shared with them.
- **No. of Widgets:** The number of widgets in the dashboard.

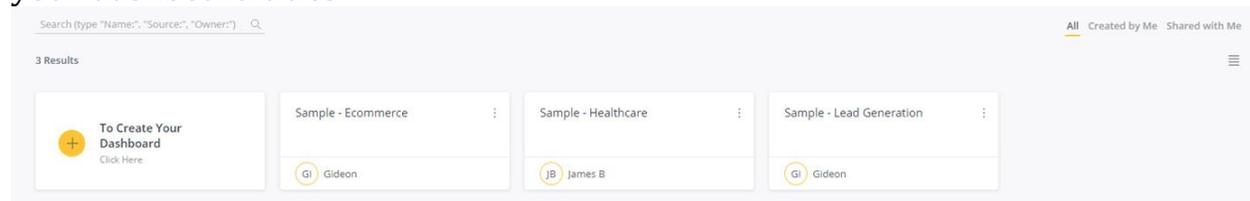
# Toggling between Views

You can toggle between these views by clicking the View icon of the right-side of the Dashboard area.



## Locating Dashboards

You can quickly and easily locate dashboards without having to navigate through your entire list of dashboards. When you enter any characters in the **Search** field, the list of dashboards is filtered according to the characters found within your dashboard titles.



To locate a dashboard by name, source, owner, or who the dashboard is shared with, you can use search operators within the **Search** field. Sisense supports the following operators:

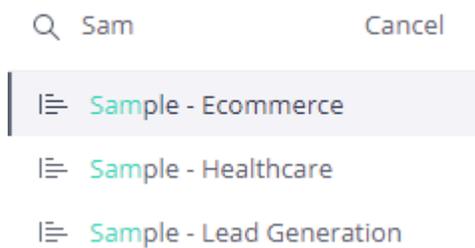
What You Can Search By	Search Operator with Example
Dashboard name	Name: Example: Name: Sales
Data source	Source: Example: Source: Sample eCommerce
Users the dashboard is Shared with	Shared with Example: "Shared with: John"

Dashboard Owner	Owner: Example: Owner: John
-----------------	--------------------------------

In addition, you can use a combination of search operators to refine your results further. For example, Name: Sales Source: Sample - eCommerce

**To locate a dashboard:**

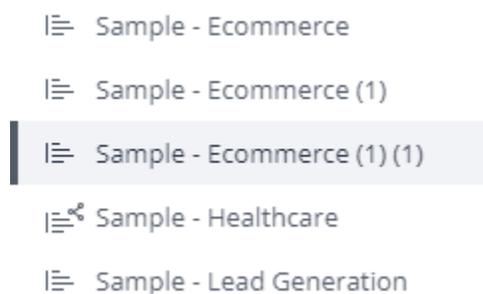
1. In the Navigation Pane, click  to display the **Search** field.
2. In the **Search** field, enter the title of your dashboard. Any dashboards that contain the characters you entered are displayed.



## Duplicating Dashboards

Copying a dashboard (even one that was shared with you) creates a new copy of the dashboard with a different name. This new dashboard is totally separate from the original and you are its owner, meaning that you have full editing rights. When you copy a dashboard, the new dashboard is added below the copied dashboard in your Navigation Pane with the same name plus a number in the order it was created.

For example, Sample Ecommerce(1) is a copy of Sample Ecommerce, while Sample Ecommerce(1)(1) is a copy of Sample Ecommerce(1):



**To copy a Dashboard:**

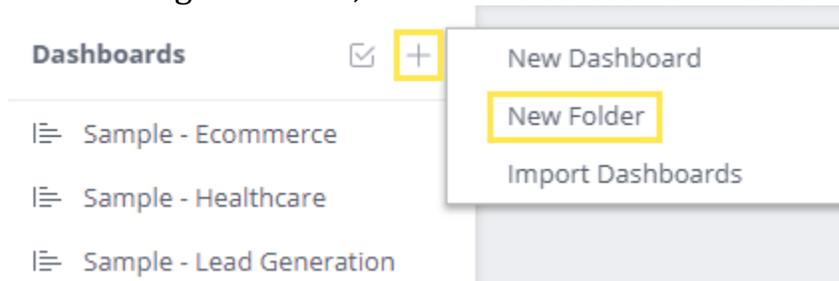
- Right-click the dashboard's name in the Navigation Pane and select **Duplicate**.

## Organizing Dashboards

You can organize your dashboards into folders and sub-folders within any of your folders to improve how you navigate the Navigation Pane.

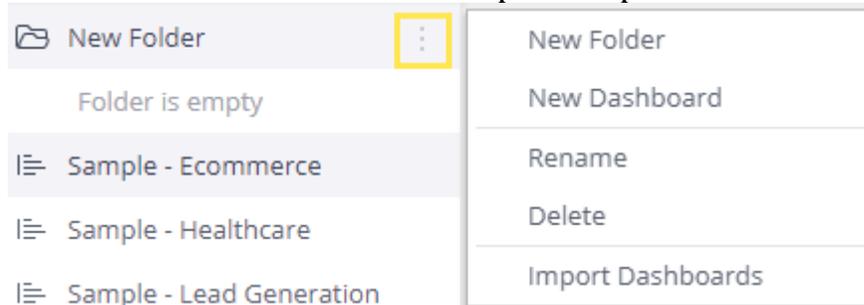
### To create a folder:

1. In the Navigation Pane, click **+** and select **New Folder**.



The new folder is added to the dashboards list.

2. Hover over the new folder and click **>** **Rename** to name the folder. To create additional sub-folders within your folders, hover over the folder and select **> New Folder** and repeat Step 2.



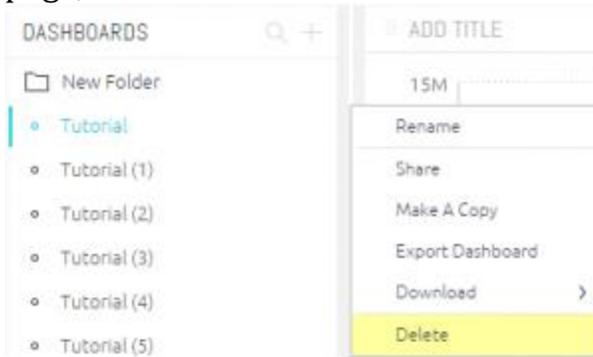
You can drag and drop your dashboards from other folders or the Navigation Pane to any of your folders.

For information on improving dashboard performance, see [Maximize Dashboard Performance](#).

## Deleting Dashboards

### To delete a dashboard:

- Click on the menu  of the Dashboard to be deleted in the list in the Home page, and select **Delete**.

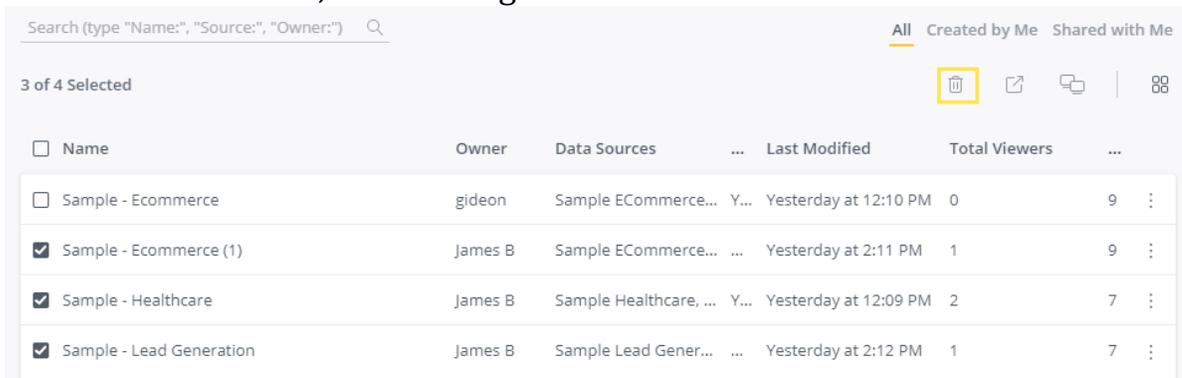


OR

Click the Dashboard Menu  and select **Delete Dashboard**.

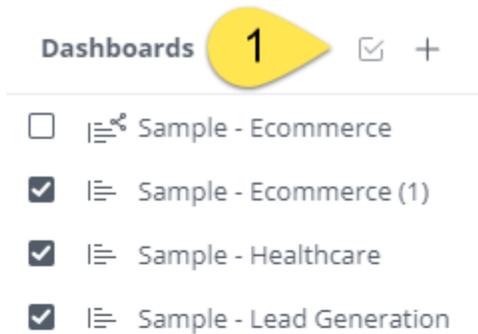
**To delete multiple dashboards:**

- You can delete multiple dashboards by opening list view, selecting the relevant dashboards, and clicking .



OR

- In your Navigation Pane, click the checkbox icon , select the relevant dashboards, and click .



## Exporting and Importing Dashboards

Sisense supports exporting and importing of dashboards. This is useful for copying dashboards between separate Sisense installations, and also for making backups.

Dashboards are exported as files with a .dash extension, which can then be imported.

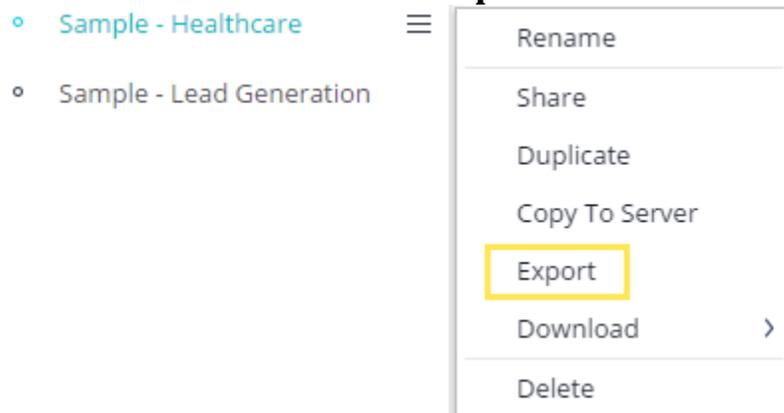
# Exporting a Dashboard

Administrators and Designers can export dashboards and import them into multiple instances of Sisense. When you export a dashboard or multiple dashboards, a single .dash file is created. Afterwards, when the .dash file is imported into another instance of Sisense, all the dashboards are added to the main **Analytics** page.

## To export dashboards to a .dash file:

### Option #1 (Single dashboard)

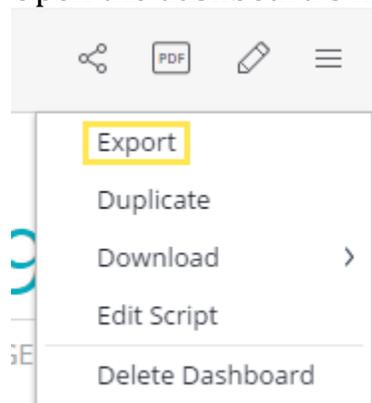
In the Navigation Pane on the left of the Sisense environment, click the dashboard's menu and click **Export Dashboard**.



OR

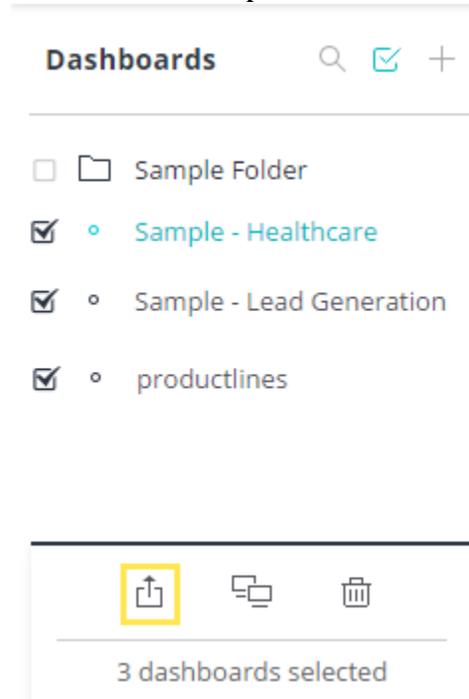
### Option #2 (Single dashboard)

Open the dashboard's menu and click **Export**.



### Option #3 (Multiple dashboards)

In the Navigation Pane on the left of Sisense, click . This displays checkboxes next to all of your dashboards and folders. Select the relevant dashboards and folders to be exported and click .



OR

In List view, select all the relevant dashboards and click .

Search (type "Name:", "Source:", "Owner:") 

**All** Created by Me Shared with Me

3 of 4 Selected    | 

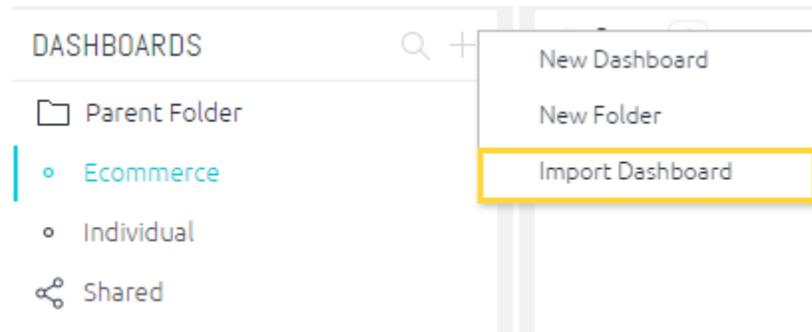
<input type="checkbox"/>	Name	Owner	Data Sources	...	Last Modified	Total Viewers	...
<input type="checkbox"/>	Sample - Ecommerce	gideon	Sample ECommerce...	Y...	Yesterday at 12:10 PM	0	9 ⋮
<input checked="" type="checkbox"/>	Sample - Ecommerce (1)	James B	Sample ECommerce...	...	Yesterday at 2:11 PM	1	9 ⋮
<input checked="" type="checkbox"/>	Sample - Healthcare	James B	Sample Healthcare, ...	Y...	Yesterday at 12:09 PM	2	7 ⋮
<input checked="" type="checkbox"/>	Sample - Lead Generation	James B	Sample Lead Gener...	...	Yesterday at 2:12 PM	1	7 ⋮

All of these methods prompt you to download the .dash file and save it to disk. After you have downloaded the exported file, you or your users can import it into their instance of Sisense. For more information, see [Importing Dashboards](#) below.

# Importing a Dashboard

## To import a dashboard:

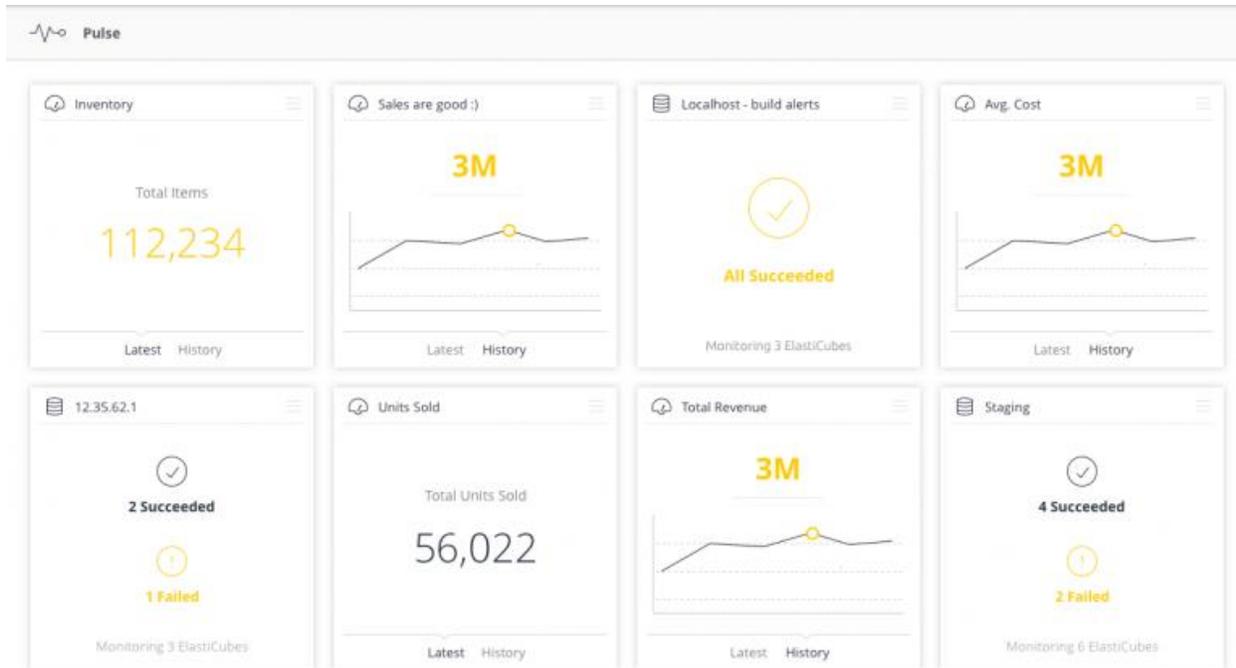
1. Hover over Dashboards list in the home page and click the '+' symbol. Select **Import Dashboard** from the menu. You can also right-click anywhere in the dashboards list, or click on the menu of one of your existing folders to import a dashboard into a folder.
2. Select your exported dash file. The dashboard will be added to your list of dashboards or folder. You can drag it from there to your desired location in the tree.



# Introduction Sisense Pulse

Sisense Pulse is a centralized location where you can stay on top of your most important KPIs across multiple dashboards or manage your data and build alerts. For example, you can consolidate and monitor important aspects of your business by adding important values from widgets from various dashboards to the Sisense **Pulse** page.

Then, you can create alerts to notify you when certain thresholds are met or anomalies in your data are detected. Sisense Pulse provides you with access to your most important data and notifies you when to take action.



The following sections describe Sisense alerts and managing alerts and KPIs in Sisense Pulse.

## Sisense Pulse

- [Overview](#)
- [Managing Pulse Tiles](#)

## Alerts

- [Overview](#)
- [Creating Data Alerts](#)
- [Creating Build Alerts](#)
- [Advanced Alert Settings](#)

## Sisense Alerts

This topic describes the two types of alerts you can define, data alerts and system alerts in Sisense.

Data alerts are triggered by changes to your data. Sisense provides several options for defining the type of changes that trigger data alerts including threshold based alerts, automatic anomaly detection, and Always which is triggered by any change to your data. For example, if you have an Indicator

widget that represents total sales, Sisense can notify you when a threshold you defined is met or exceeded.

Build alerts can be configured to be sent following build attempts.

When creating an alert, you can customize your notifications, the channel through which they are sent, and how often a notification is sent following a data or build.

After you create an alert, the alert is added to the Sisense **Pulse** page where you can manage the alert and its notifications. For example, you can delete alerts or enable and disable notifications.

If you want to make a similar alerts, for example, alerts with different notifications for different groups of users, you can duplicate the alert and modify its settings.

When an alert is triggered, you and your users are notified through the channel you defined when creating the alert and through the Pulse Feed. The Pulse Feed is displayed in the Sisense Web Application and shows a list of your most recent triggered alerts. From the Pulse Feed you can open the dashboard from where

the alert originated or open the Sisense **Pulse** page.

 Build Alert - Success The ElastiCube <b>Test</b> on <b>LocalHost</b> was built successfully.	11/1/16 3:30 PM
 Converted (%) Converted (%) = 7.29%	10/31/16 9:35 AM
 Average Cost Average Cost = \$15.65	10/31/16 9:35 AM
 Converted (%) Converted (%) = 7.29%	10/25/16 3:55 PM
 Average Cost Average Cost = \$15.65	10/25/16 3:55 PM
 Build Alert - Success The ElastiCube <b>Test</b> on <b>LocalHost</b> was built successfully.	10/25/16 10:12 AM
 Build Alert - Success The ElastiCube <b>Test</b> on <b>LocalHost</b> was built successfully.	10/13/16 9:20 AM
<a href="#">Clear all</a>	

## Related Topics

[Creating Data Alerts](#)

[Creating Build Alerts](#)

## Creating Data Alerts

This topic describes data alerts and how you can add them to Sisense Pulse. Data alerts are triggered by conditions you define for your widgets. For example, if you have an Indicator widget that represents your company's revenue, you can define an alert that notifies you when this revenue reaches or falls below a certain threshold.

Sisense supports alerts for the following widgets:

- Indicator Widgets
- Area Chart Widgets
- Bar Chart Widgets
- Column Chart
- Line Chart

For each type of widget you can add to Pulse, you can monitor a single value. For example, for Column charts you can monitor a single section (value) of a column. To monitor multiple values, you can add each value you want to monitor to Pulse separately.

When creating data alerts, you define the condition that triggers the alert.

Sisense provides three alert conditions that can trigger an alert:



Threshold

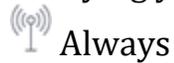
Sisense notifies you when a threshold is reached following a build. This condition is useful for monitoring metrics that follow a regular, cyclical schedule.



Automatic

Sisense automatically notifies you when the Sisense alert engine identifies an anomaly in your data based on machine learning algorithms. The Sisense algorithm actively learns from your data and eliminates noise to identify

anomalies in your data. Sisense's anomaly detection is based on a smoothing algorithm that takes the latest values into account. The more times an ElastiCube is built, the more accurate Sisense becomes at identifying any anomalies and notifying you.



Sisense notifies you after each build, regardless if there is a change in your data or not.

In addition, you can configure advanced settings that determine who receives the alert and what channel the alert is sent through. For more information see [Advanced Settings](#).

### To create a data alert:

1. From the widget's menu, select **Add To Pulse**.
2. The Add to Pulse window is displayed.

**Add to Pulse**

---

Name

Alert Condition

Threshold      Automatic      Always

---

The system will monitor this value and notify you when anomalies are detected.

[Advanced](#)      **Add**      Cancel

3. In the **Name** field, enter a meaningful name that represents your alert.
4. In the Alert Condition area, select the condition that triggers the alerts.  
**Threshold:** Sisense defines the threshold value as the current value displayed in the widget. You can increase or decrease this value and define

the equality and relational operators that determine when the threshold is met.

Alert Condition



Threshold



Automatic



Always

**Automatic:** Sisense automatically identifies anomalies in your data and alerts you.

Alert Condition



Threshold



Automatic



Always

**Always:** Sisense notifies you every time the data changes.

Alert Condition



Threshold



Automatic



Always

You will be notified about this measure  
whenever it updates.

5. Click **Add** to create the alert. The alert is added to Sisense Pulse.  
For additional advanced configuration options, see [Advanced Settings](#).

## Creating Build Alerts

Administrators can set build alerts to notify themselves or other users when a build fails or succeeds. Build alerts are created from Sisense and are sent following an ElastiCube build.

Build alerts allow you to monitor the status of ElastiCube builds.

Administrators can create build alerts from the **Admin** page. After a build alert is created, it is added to Sisense Pulse from where you can manage it.

### To create a build alert:

1. In Sisense, select **Admin > Data Sources > Add Build Alert**.

OR

From the ElastiCube menu, **Add Build Alert**.

The screenshot shows the Sisense Admin interface. The left sidebar has 'Data Sources' highlighted. The main area displays a table of ElastiCubes and Data Sets. A context menu is open over the 'Facebook Ads Demo' row, showing options like 'Share', 'Data Security', 'Manage Hierarchies', 'Add Build Alert', 'Cancel All Queries', 'Pulse!', 'Create ElastiCube Set', and 'Add To ElastiCube Set'. The 'Add Build Alert' option is highlighted with a yellow circle '3'. A yellow circle '2' is also present on the 'Data Sources' menu item.

Server	ElastiCube	Editors	Last Model Change	Last Build	Size On Disk	Status
LOCALHOST	Facebook Ads Demo	gideon@senses.com*1	5/5/17 1:09 PM	5/5/17 10:45 AM	4.09MB	Stopped
Local Machine						
ElastiCube	Editors	Last Model Change	Last Build	Size On Disk	Status	
Commerce Demo	gideon@senses.com	15/28/15 12:03 PM	9/9/17 2:29 PM	566.95MB	Running	
Facebook Ads (1)	gideon@senses.com	5/3/17 1:09 PM	5/3/17 2:12 PM	5.38MB	Stopped	
Facebook Ads Demo	gideon@senses.com*1	5/5/17 1:09 PM	5/5/17 10:45 AM	4.09MB	Stopped	
Facebook News 20175433	gideon@senses.com*1	9/3/17 12:45 PM	9/3/17 12:45 PM	13.07MB	Stopped	
Facebook Page	gideon@senses.com*1	4/25/17 2:12 PM	8/16/17 12:49 PM	18.99MB	Stopped	
news	gideon@senses.com	15/10/16 5:45 PM	6/6/17 11:56 AM	35.49MB	Stopped	
Sample Commerce	gideon@senses.com	2/19/14 2:21 PM	3/12/14 10:18 AM	58.06MB	Running	
Sample Healthcare	gideon@senses.com	7/3/13 1:07 AM	3/13/14 2:01 PM	1.32MB	Running	
Sample Lead Generation	gideon@senses.com	3/3/14 7:48 PM	3/12/14 11:35 AM	2.2MB	Running	
Training	gideon@senses.com	10/16/14 2:36 PM	10/16/14 2:39 PM	803.07KB	Stopped	

The Add Build Alert window is displayed.

2. In **Name**, enter a name for the alert.

**Add Build Alert**

Name

Server

ElastiCube

Notify when

- Build failed
- Build succeeded
- Build succeeded after failure

Add Cancel

3. From the **Server** list, select the server where the ElastiCube is hosted.
4. From the ElastiCube list, select the ElastiCube(s) that triggers an alert following a build.
5. Select when to notify you following a build. There are three conditions that when met, can trigger an alert:
  - Build failed:** The alert is sent after the ElastiCube build fails.
  - Build succeeded:** The alert is sent after the ElastiCube is built successfully.
  - Build succeeded after failure:** The alert is sent following a successful build after the ElastiCube previously failed to build.
6. Click **Add**. The alert is added to your Sisense **Pulse** page.  
For additional advanced configuration options, see [Advanced Settings](#).

## Sisense Pulse

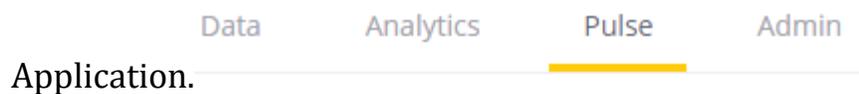
Sisense Pulse enables you can stay on top of your most important KPIs across all your dashboards and your latest ElastiCube builds. By adding important KPIs

from your dashboards to Sisense Pulse, you can get a comprehensive picture of your data from a single location.

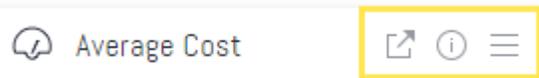
Sisense Pulse contains tiles that display information from your dashboards and the status of your ElastiCube builds.

Within Sisense Pulse, there are two types of tiles, data tiles that display your data's current status and recent history and build tiles that display the latest status of your ElastiCube builds.

You can access Sisense Pulse from the menu bar of the Sisense Web



Application. By hovering over the header of any tile in Sisense Pulse, you can view additional options for your tiles.



The  icon opens a new tab in your browser. For data tiles the dashboard where the widget originated is displayed and for build tiles the Data Sources section in the **Admin** page is displayed.

The  icon is a tooltip that describes the tile in more detail. For data tiles the tooltip describes who added the tile to Pulse and when it was added. Build tile tooltips describe when the tile was created, which ElastiCube is being monitored, and under what conditions an alert is triggered when alerts are enabled.

The  icon displays a menu for managing tiles in Sisense Pulse. For more information regarding managing tiles in Sisense Pulse, see [Managing Tiles in Sisense Pulse](#).

# Adding Tiles to Sisense Pulse

To keep track of your important KPIs and ElastiCube builds, you add the relevant data and build tiles to Sisense Pulse.

The flow for adding tiles to Sisense Pulse is the same for creating alerts for Indicator widgets and ElastiCube builds. For more information about these flows, see the following topics:

For data tiles, see [Creating Data Alerts](#).

For build tiles, see [Creating Build Alerts](#).

Each of these flows adds tiles to Sisense Pulse and creates an alert. If you want to monitor your data and ElastiCube builds without alerts, see [Deactivating Notifications](#).

## Managing Tiles in Sisense Pulse

This topic describes how to manage your Pulse tiles and notifications when an alert is triggered.

After you have created alerts or added tiles to Sisense Pulse, you can manage those tiles from Sisense Pulse. Through the Sisense **Pulse** page, you can duplicate alerts and modify them, disable notifications, and delete alerts.

# Duplicating Tiles in Sisense Pulse

You can create copies of alerts and modify them to quickly generate new alerts with different notification messages or notify various groups across separate channels.

For example, if you want to notify Group 1 via email and Group 2 via Slack when the same event occurs, you can duplicate an alert and modify the notification channel.

## To duplicate a tile:

1. In the Sisense Web Application, select **Pulse**.
2. In the relevant alert's menu, select **Duplicate**. A copy of the alert is created and the Add to Pulse window is displayed.

**Add to Pulse**   

Name	Value
Copy of TOTAL	TOTAL

Filters

5 active filters [Show](#)

Alert Condition - Threshold ▾

Greater than ▾ 90000  

[Add](#) [Cancel](#)

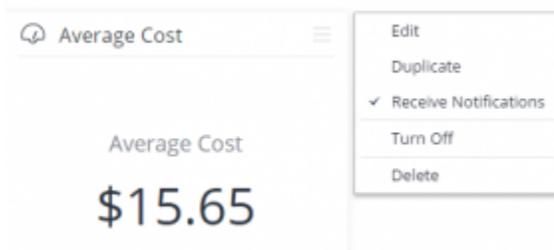
You can modify the alert's settings from the **Add to Pulse** window. For more information about these settings, click [here](#).

# Deactivating Your Notifications

Each time you add a tile to Sisense Pulse or create an alert, by default, you will receive notifications when the event is triggered. If you do not want to receive any notifications, you can disable notifications from being sent by toggling the notifications. When you toggle the notifications, you no longer receive notifications when an alert is triggered, however, other users with whom the alert was shared will continue to be notified. See [Deactivating Notifications for All Users](#) to turn off notifications for everyone associated with an alert.

## To deactivate notifications:

1. In Sisense, select **Pulse**.
2. From the Tile menu of the relevant alert, toggle **Receive Notifications**.



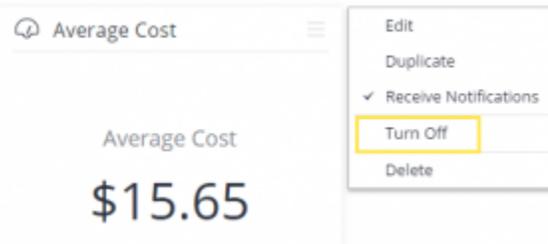
# Deactivating Notifications for All Users

If you have added an alert that is no longer relevant or you want to deactivate the alert for all users, you can turn off the alert.

When you turn off an alert, no users are notified when the event that triggers the alert occurs.

## To deactivate notifications for all users:

1. In the Sisense Web Application, select **Pulse**.
2. From the Tile menu of the relevant alert, select **Turn Off**. A confirmation dialog box is displayed.



3. Click **Yes** to confirm that you want to turn off all notifications for the alert. The alert is turned off.  
Alerts that have been turned off appear in the **Pulse** page as shaded tiles. You can reactivate the alert by selecting **Turn On** from the Tile menu.

# Deleting Tiles in Sisense Pulse

When a tile is no longer needed, you can delete the tile through the Sisense Pulse page.

Deleting an alert removes it from the Sisense **Pulse** page and users are no longer notified.

## **To delete an alert:**

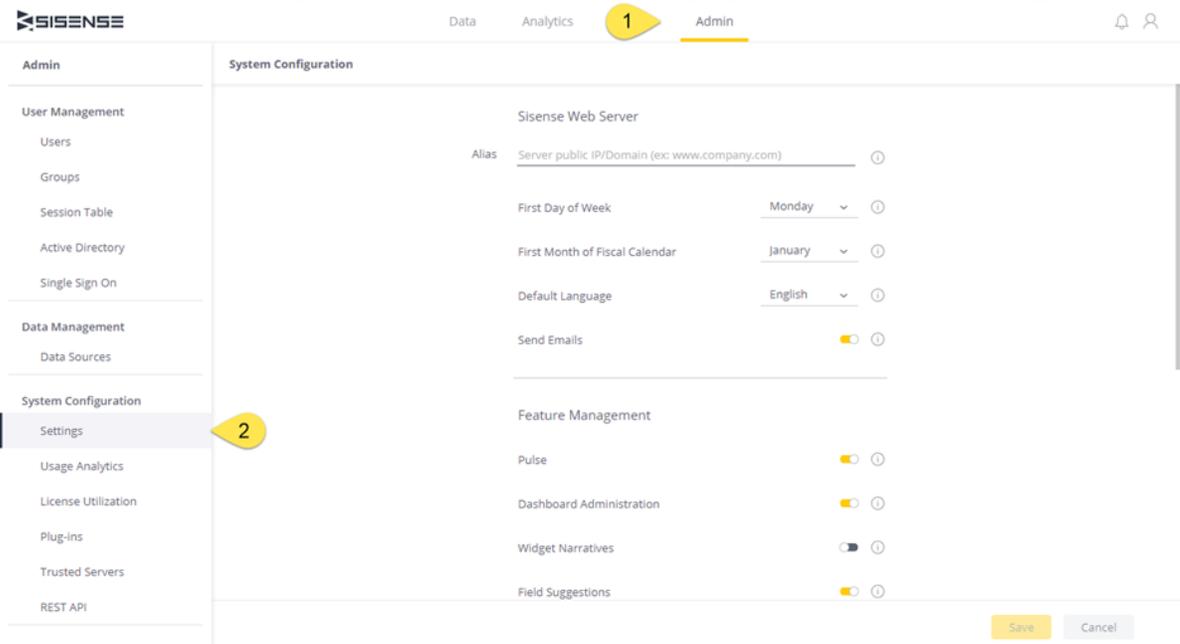
1. In Sisense, select **Pulse**.
2. In the relevant alert's menu, select **Delete**.

# Removing Sisense Pulse

You can remove the Pulse page from Sisense and deactivate all the alerts defined in Sisense Pulse from the **Admin** page. This is useful if you have embedded Sisense and your end users do not need to receive alerts.

## To remove Sisense Pulse:

1. In Sisense, in the top menu, open the **Admin** page and select **Settings**.



2. Toggle the **Pulse** switch to disable Sisense Pulse. The Sisense Pulse tab on the top menu is no longer displayed and no new notifications will be sent.

## Advanced Settings

By default, when an alert is triggered, you are notified via your email address registered with Sisense.

You can also send alerts using more advanced configurations that can let you notify additional users when an alert is triggered, notify them you across multiple channels, and allows you to customize the message displayed in the notification.

The following topics describe the functionality available in the Advanced Settings options:

- **Notifying Additional Users**
- [Setting Notification Channels](#)
- [Customizing Notification Messages](#)
- [Configuring Additional Options](#)

## Notifying Additional Users

When an alert is triggered, Administrators and Designers can notify additional Sisense users and groups by configuring the alert's advanced settings. Viewers cannot share alerts with additional users.

Users who are notified receive a notification through Sisense Pulse and any other additional channels you define.

**To notify additional users:**

1. For Build Alerts, select the  tab.

**Add Build Alert**

Name

Server  

ElastiCube  

Notify when

- Build failed
- Build succeeded
- Build succeeded after failure

For Data Alerts, select **Advanced Options** > 

**Add to Pulse**   

---

Name	Value
Total Revenue	Total Revenue

Filters

3 active filters [Show](#)

Alert Condition - Threshold ▾

Greater than ▾  

---

[Basic](#) [Add](#) [Cancel](#)

2. Enter the email address, Sisense username, or group name of the users you want to notify when an alert is triggered. Entering **Everyone** specifies that all Sisense users and user groups associated with your account are notified. The contact information of the people you add is listed in the center of the window and is displayed the next time you open this window.
3. Click **Add** to add the alert to Pulse.

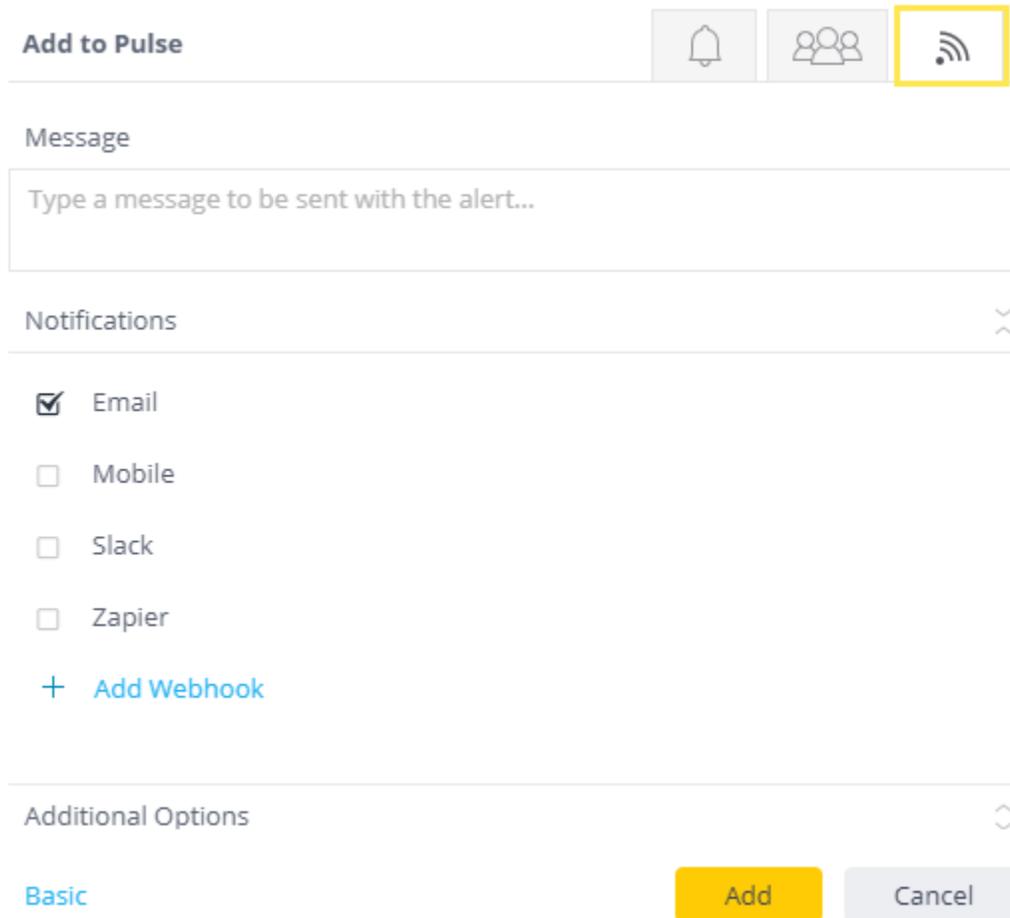
## Setting Notification Channels

By default, your users are notified via their email addresses registered with Sisense. In addition, Sisense supports a variety of workflows by sending notifications across multiple applications and services. For example, your users

can receive notifications through the Sisense mobile application, Zapier, Slack, and other 3rd services via webhooks.

**To define how users are notified:**

1. For Build Alerts, select the  tab.
2. For Data Alerts, select **Advanced Options** > .



**Add to Pulse**

Message

Type a message to be sent with the alert...

Notifications

- Email
- Mobile
- Slack
- Zapier

+ [Add Webhook](#)

Additional Options

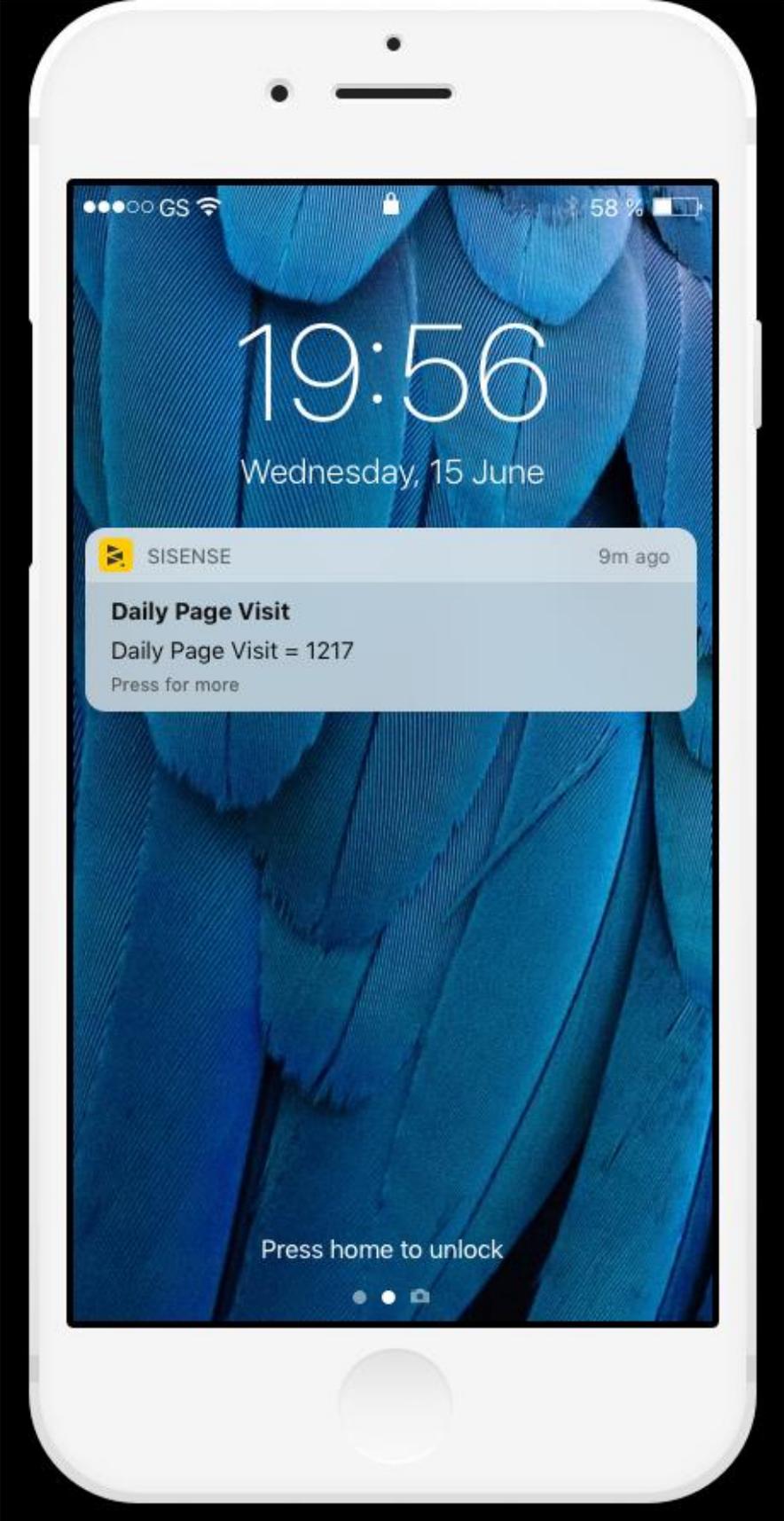
Basic Add Cancel

3. In the Notifications area, select the relevant channel for your notification. The default channel, Email, is already selected. You can select one or more of the following channels:
  - Email
  - Mobile
  - Slack
  - Zapier
  - Webhook
  - mobile
4. Click **Save**.

# Sisense Mobile

Sisense Mobile enables users to view Sisense dashboards through their mobile devices. You or your users can receive notifications through Sisense Mobile. After selecting the users you want to notify, any users with a Sisense user account and

Sisense Mobile will receive a notification when an alert is triggered.



GS 58%

19:56

Wednesday, 15 June



SISENSE

9m ago

**Daily Page Visit**

Daily Page Visit = 1217

Press for more

Press home to unlock

# Slack

Slack is a real-time team collaboration messaging platform that allows you to receive incoming Webhooks from external sources such as Sisense. To send notifications through Slack, you need to enter the URL and optionally, your Slack channel or username, where the notification is to be displayed.

You can ask your Administrator to provide you with the Slack URL and channel. You can obtain these details by adding the Incoming WebHooks app into your Slack configuration.

## **To obtain a Webhook URL and channel from Slack:**

1. Add the Incoming WebHooks app to Slack. You can find this app in Slack's App Directory.
2. In the Incoming WebHooks app page, click **Add Configuration**.
3. After you add the Incoming WebHooks app, you can select which channel to post the notifications to. The channel name is optional and displays the notification in the channel associated with your WebHook URL. In the Post to Channel area, select the channel where the notification is to be displayed

and click **Add Incoming WebHooks integration.**

[Browse apps](#) > [Custom Integrations](#) > [Incoming WebHooks](#) > New configuration



## Incoming WebHooks

Send data into Slack in real-time.

Incoming Webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details described later.

[Message Attachments](#) can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.



### New to Slack integrations?

Check out our [Getting Started](#) guide to familiarize yourself with the most common types of integrations, and tips to keep in mind while building your own. You can also [register as a developer](#) to let us know what you're working on, and to receive future updates to our APIs.

### Post to Channel

Start by choosing a channel where your Incoming Webhook will post messages to.

#general

[or create a new channel](#)

**Add Incoming WebHooks integration**

By creating an incoming webhook, you agree to the [Slack API Terms of Service](#).

4. In the Setup Instructions the Webhook URL is displayed. Copy the Webhook URL.



## Incoming WebHooks

[Disable](#) • [Remove](#)

Incoming Webhooks are a simple way to post messages from external sources into Slack. They make use of normal HTTP requests with a JSON payload, which includes the message and a few other optional details described later.

[Message Attachments](#) can also be used in Incoming Webhooks to display richly-formatted messages that stand out from regular chat messages.



### New to Slack integrations?

Check out our [Getting Started](#) guide to familiarize yourself with the most common types of integrations, and tips to keep in mind while building your own. You can also [register as a developer](#) to let us know what you're working on, and to receive future updates to our APIs.

### Setup Instructions

We'll guide you through the steps necessary to configure an Incoming Webhook so you can start sending data to Slack.

[close](#)

**Webhook URL**

<https://hooks.slack.com/services/T2ZT2KV1QNiLJ37kmv>

5. In Sisense when sending notifications through Slack, in the **URL** field, enter the URL copied in the previous step and in the #channel/@username field, enter the channel name if you have defined one.

# Zapier

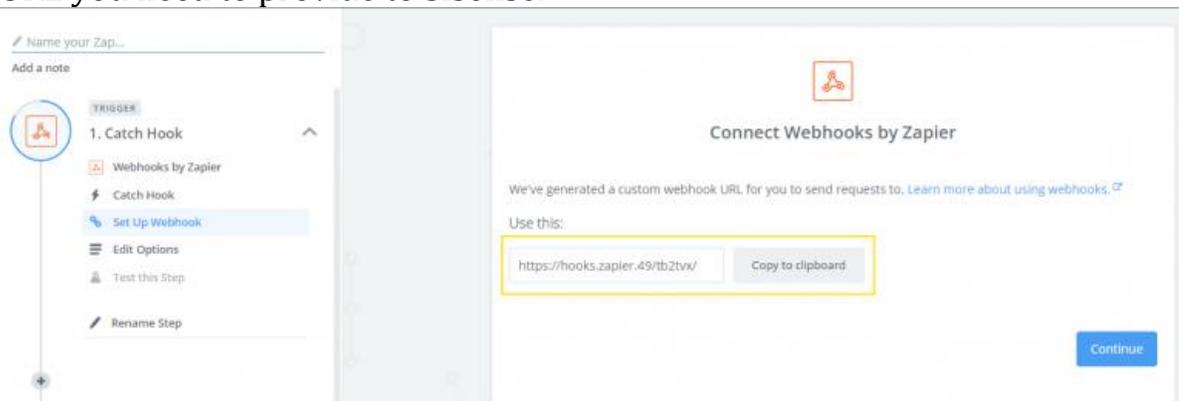
Zapier is an online service that lets you create automated actions connecting disparate business apps and services such as Sisense. To receive notifications through Zapier, you must create a Zap and provide Sisense with the relevant URL when configuring your alerts' notifications.

When an alert is triggered, Sisense sends the notification to the Webhook you define in Zapier.

You can ask your Administrator to provide you with the Zapier URL.

## To obtain a Webhook URL from Zapier:

1. Log in to Zapier and click **Make a Zap**. The Trigger App page is displayed.
2. From the Trigger App page select Webhooks. The Connect to Webhooks by Zapier page is displayed.
3. In Step 1 > Set Up Webhook, click **Copy to clipboard** to copy the Webhook URL you need to provide to Sisense.



4. In Sisense, when sending notifications through Zapier, in the **URL** field, enter the URL copied in the previous step.

# Webhooks

Webhooks are a system of automated notifications that indicate that an event occurred, in this case an alert being triggered. To send notifications through additional 3rd party channels, you can select **Webhook** and enter the Webhook name and URL.

Your Webhook server listens for incoming messages from Sisense across the URL specified in the URL field.

You can ask your system administrator to provide you with the Webhook URL.

When Sisense sends notifications to your custom Webhook, they are sent within a JSON payload. Your Webhook provider must interpret the payload and display the notification to your users. The following are examples of a notifications sent in a JSON payload and a description of the contents of the payload:

```
{ type: 'kpi', name: 'Total Sales', message: 'Hello World', originDashboardLink: '58128745677f52dc0f000019', measure: 'Revenue', value: 39759625.51502721, condition: 'none', filters: [ { asDimension: false, title: 'Category', value: 'Include all', jaql: [Object], panel: 'scope' }, { asDimension: false, title: 'Gender', value: 'Include all', jaql: [Object], panel: 'scope' } ] }
```

Data Events	Build Events
<pre>{ type: 'kpi',   name: 'Total Sales',   message: 'Hello World',   originDashboardLink: '58128745677f52dc0f000019',   measure: 'Revenue',</pre>	<pre>{ type: 'build', name: 'Success Event', measure: null, value: 1,</pre>

<pre> value: 39759625.51502721,   condition: 'none', filters:   [ { asDimension: false,       title: 'Category', value: 'Include all',       jaql: [Object],       panel: 'scope' },     { asDimension: false, title: 'Gender', value: 'Include all',       jaql: [Object],       panel: 'scope' } ] </pre>	<pre> condition: 'none', filters: null } </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------

Name	Type	Description
Type	String	Type of alert. 'kpi' for data events and 'build' for build events.
Name	String	Name of the alert.
Message	String	Notification message.
originDashboardLink	String	Dashboard ID provided by Sisense. You can find the dashboard ID from the URL of the dashboard. For example, the ID of the dashboard below if: <a href="http://localhost:8081/app/main#/dashboar">http://localhost:8081/app/main#/dashboar</a>

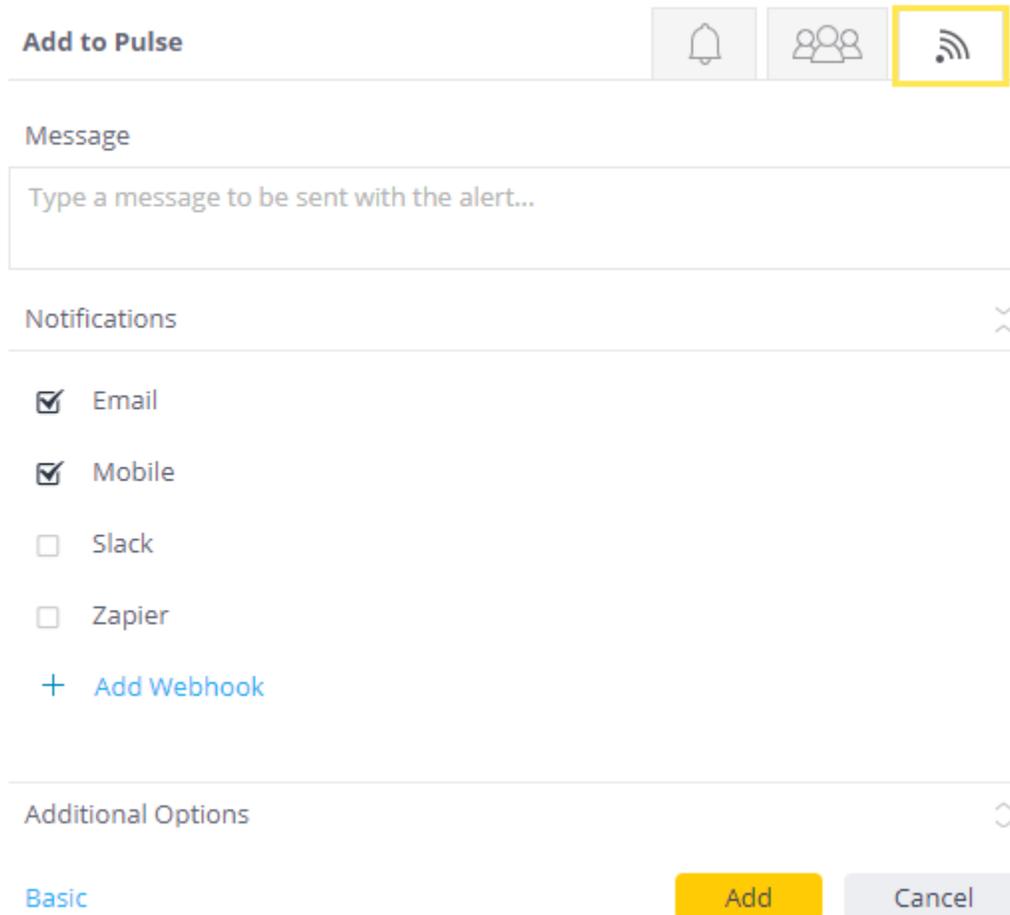
Name	Type	Description
		ds/ <b>573b4322048e5e382a000001</b>
value	Double	For data events when a threshold condition is applied, this is the value of the threshold. For build events, the value is 0 for failed builds and 1 for successful builds.
condition	String	Type of condition that generated the alert.
filters	Object	Object that defines filters applied to the dashboard.

## Configuring Additional Options

You can configure how often and when a notification is sent after an event is triggered. For example, you can limit the amount of notification sent to one when a data alert is triggered. This is useful if you build your ElastiCubes frequently, but do not need multiple notifications.

**To configure additional options:**

1. When adding or editing a widget to Pulse, in the Advanced Options window, select  > **Additional Options**.



**Add to Pulse**

Message

Type a message to be sent with the alert...

Notifications

- Email
- Mobile
- Slack
- Zapier
- + Add Webhook

Additional Options

Basic

Add Cancel

2. Select any of the following options:  
**For data alerts:**
  - Select **Notify only once after a condition is met** to limit how many notifications are sent after an event occurs.
  - Select **Notify when a condition is no longer met** to notify users after the event is no longer triggered.**For build alerts:**
  - Select **Notify once when there is a status change** to notify users only once when the latest build status is different than the previous build status.
3. Click **Add** to save the alert.

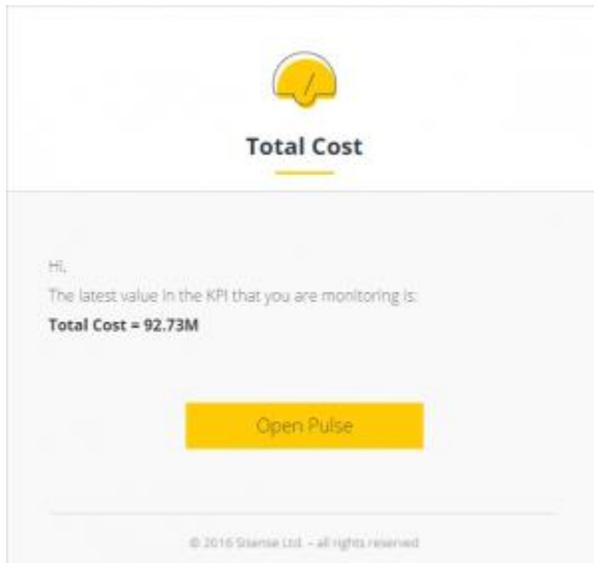
## Customizing Notification Messages

By default, the following message is provided in your email alert notifications:

*Hi,*

*The latest value in the KPI that you are monitoring is:*

**Value**



You can provide customized messages for your notifications when configuring your alert through the alert's Advanced Settings.

The message you provide replaces the string: The latest value in the KPI that you are monitoring is:

For example, if your message is "The last value of my KPI is:" the message in your alert's notification will be:

Hi,

The last value of my KPI is:

**Value**

**To customize your notification's text:**

1. For Build Alerts, select the  tab.

**Add to Pulse**

Message

Type a message to be sent with the alert...

Notifications 

Email

Mobile

Slack

Zapier

[+ Add Webhook](#)

Additional Options 

[Basic](#)  

2. For Data Alerts, select **Advanced** > .

#### Add to Pulse

Name

Total Quantity

Alert Condition



Threshold



Automatic



Always

You will be notified when the value is:

Greater than ▾

32386

**Advanced**

Add

Cancel

3. In the Message area, enter your custom text.
4. Click **Save**. The next time a notification is sent, your custom message is displayed in the notification.
5. You can modify your alert notifications further including customizing the text and images displayed. For more information, see [Rebranding Sisense Automated Emails](#).

## Administration

Sisense supports several [user roles](#) with Administrative privileges for customizing Sisense for your company. The following roles: System Administer, Data Administrator, and Administrator have varying levels of access to your system including servers, users, ElastiCubes and more. The **Admin** page is your dedicated location in your web application for managing your system.

This section describes how your Sisense Administrator can configure and manage users, groups, data sources, and system settings from the **Admin** page.

User Name	Email	Name	Groups	Roles	Origin
<input type="checkbox"/>	gideon@sisense.com	gideon@sisense.com	gideon	Sys. Admin	Sisense
<input type="checkbox"/>	sam@sisense.com	sam@sisense.com	sam Jones	Data Admin	Sisense
<input type="checkbox"/>	John@sisense.com	John@sisense.com	John Smith	Viewer	Sisense

**Note:** The **Admin** page is only displayed to [users](#) with the relevant privileges. In addition, certain settings are only available to certain user roles except for Administrators who have access to all settings.

- [User Management](#)
- [Dashboard Management](#)
- [Data Source Management](#)
- [System Settings](#)
- [Embedding and Rebranding](#)
- [Security](#)
- [Migration](#)
- [High Availability](#)

## Managing Sisense Users

Sisense Administrators have access to the User Management section of the Admin Page where users, groups, and advanced user settings can be configured. The topics below describe how to create users and groups, and allow users you authenticate to access your implementation of the Sisense Web Application.

- [Managing Dashboard Users](#)
- [Managing User Groups](#)
- [Managing User Sessions](#)
- [Customizing User Roles](#)
- [License Utilization](#)
- [Integrating Active Directory](#)
- [Introduction to SSO](#)

## Managing Dashboard Users

This topic describes how to add and manage Sisense users that can access Sisense dashboards.

Users are managed from the **Admin** page through the Users tab.

The screenshot shows the Sisense Admin interface. The 'Admin' menu is open, and the 'Users' tab is selected. The 'Users' table contains the following data:

Username	Email	Name	Groups	Role	Origin	Last A...	Days Since Last...	
<input type="checkbox"/> gideon@sisense...	gideon@sisense.com	gideon	Admins	Sys. Admin	Sisense	6/11/18	0	
<input type="checkbox"/> Jamesb@sisense...	Jamesb@sisense.com	James B		Data Designer	Sisense	6/11/18	0	
<input type="checkbox"/> sally@sisense.com	sally@sisense.com	Sally Jones		Viewer	Sisense			

Page 1 of 1      << < > >>      3 Users found

In the Users table, you can edit and delete users in your system. The table provides the following information about your users:

- **Username:** The user's username in Sisense.
- **Email:** The user's email address.
- **Name:** The name of the user.
- **Groups:** The groups, if any, that the user belongs to. By default, the System Administrator belongs to a group called Admins.
- **Role:** The user role assigned to the user.
- **Origin:** The origin of the user. This is typically your company's name.

- **Last Active:** The date of the last time the user was active. Activity includes opening any of the Sisense tabs (Data, Analytics, Pulse, and Admin), viewing dashboards, working with widgets, or exporting reports. This field is blank if the user has never been active in Sisense.
- **Days Since Last Active:** The number of days that have passed since the user last was active. This field is blank if the user has never been active in Sisense.

# Adding a New User

You can add users using the following methods:

- Add new users by entering them one by one or in bulk
- Add new users from your Active Directory database. [Click here](#) to learn how to configure your Active Directory in Sisense

## To add a new user:

1. Click **+ Users**. If a connection with Active Directory also exists, an additional Import Many option is available. Click it to select a user from Active Directory.

The screenshot shows the 'Add Users' form with the following fields and options:

- Tab: Add Users (selected), Single User, Import Many
- Email: [Text Input]
- First Name: [Text Input] Optional
- Last Name: [Text Input] Optional
- Language: [Dropdown Menu] Group/System Lang...
- Role: [Dropdown Menu] Viewer
- Define Password: OFF (toggle)
- \* The user will receive an email containing a password activation link.
- Buttons: Save, Cancel

2. Fill in the user's details in the window.
3. (Optional) The **Language** for your account is the language set by an Administrator for your group or server. From the Language list, you can override those settings and select the language you prefer to be displayed in the Sisense Web Application.
4. Select the **Role** of the user:
  - **Admin:** Admins can access the **Admin** page of the Sisense Web Application from where they can manage ElastiCubes, users, user groups and more. Sisense supports multiple Admin users.
  - **Data Admin:** Data Admins can manage all data sources in the system. They have access to the **Admin** page of the Sisense Web Application from where they can manage ElastiCubes and Live connections. Data Admins can also add and manage servers. This is useful for migrating

Sisense across environments. Data Admins do not have access to User Management and System Configuration.

- **Data Designers:** Data Designers can create and edit ElastiCubes they created or were shared with them. Data Designers can access the **Admin** page of the Sisense Web Application from where they can manage the ElastiCubes and Live Connections they have access to in the Data Sources section. In addition, Data Designers can see servers, but can not add new servers. Data Designers do not have access to User Management and System Configuration.
  - **Designers:** Designers can create, design, edit and share Dashboards. Sisense Designers determine whether the user with who they share a Dashboard has editing rights (is a Designer) or only viewing rights (Viewer).
  - **Viewers:** Viewers can view, explore, drill down, make selections and filter the Dashboards that Designers share with them. Viewers only need a standard web browser; no plugins or downloads. They cannot create new Dashboards or edit existing ones.
5. Click **Save**. The user then receives an email from Sisense that redirects the user to activate their account and set a password.

**To manually set a password for the new user:**

- Set the **Define Password** option to **ON** and enter a password before saving.



The image shows a portion of a web form. At the top, there is a label 'Define Password' followed by a toggle switch that is currently set to 'ON' (indicated by a green dot and the text 'ON'). Below this, there are three input fields: 'Set Password', 'Confirm Password', and another unlabeled field. The 'Set Password' field has a vertical cursor at the beginning. The 'Confirm Password' field is empty.

In this case, the user receives an invitation email. You must personally notify the user of the password that you set for them.

# Adding Multiple Users

## To add multiple users:

1. Click the **Import Many** option at the top right of the Add users window.
2. In the displayed text box, paste a comma separated list of emails and click **Next**.
3. The next screen enables you to assign User rights to the users that you entered and to verify that the user count and permissions match your license.
4. After you click **Add**, each user receives an activation email.

# Editing an Existing User

**To edit an existing user:**

1. Click **Admin** and select the **Users** tab on the left.
2. Click  on the right of the user's entry in the list.

# Deleting a User

## To delete an existing user:

1. Click **Admin** in the upper right corner and select the **Users** tab on the left.
2. Click  at the right of the user's entry in the list.

## Managing User Groups

This topic describes how to add and use user groups in Sisense.

### How Do User Groups Work?

1. When building and sharing dashboards, wherever you can select a user, you can also select a user group. For example, you can share a dashboard with an entire user group at once.
2. When you add an additional user to an existing user group, the user receives the role you assigned to them when you created that user
3. When users are added in Active Directory to a previously imported Active Directory user group, then those users are automatically added to the corresponding Sisense user group.
4. User groups can only contain users. They do not support nested user groups.
5. Additional users cannot be added to a user group that was imported from Active Directory.

You can add users to a user group using the following methods:

- Add existing Sisense users to a user group
- Add a user group from Active Directory

# Accessing User Group Management

You can add, edit and delete user groups, and assign users to them.

## To access the user group settings:

- Click **Admin** and select the **Groups** tab on the left. A list of the previously defined user groups is displayed. The **Origin** column displays **Sisense** or **Active Directory** to indicate whether the user group was defined in Sisense or imported from Active Directory.

If no user groups have yet been defined, the following message is displayed: No Groups Found

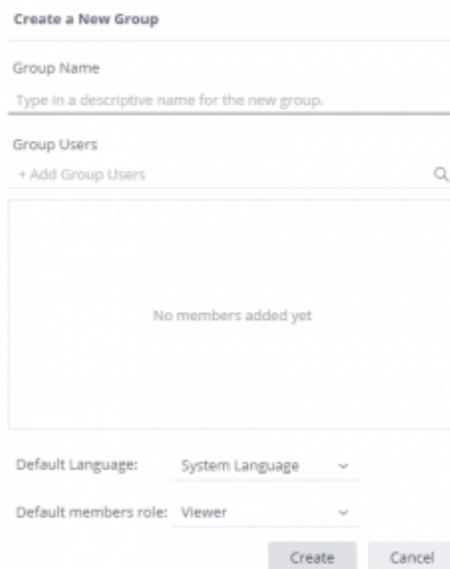
# Adding a New User Group

The following procedure describes how to add a new user group by selecting users from existing users in Sisense.

See [Adding a User Group from Active Directory](#) for a description of how to add a user group from Active Directory.

## To add a new user group:

1. Click **Admin** and select the **Groups** tab on the left.
2. Click **Add Group**.



3. In a **Group Name** field, enter the name for this user group.
4. In the **Group Users** field select each of the users that belong to this group. To add each user, start typing in the user's name. An auto-complete list is displayed showing the users that already exist in Sisense. Select a username to add to the group.  
**Note:** If you want, you can leave the group empty and add members to it later.
5. (Optional) The **Default Language** for a group is the language set by an Administrator for that server. From the Default Language list, you can select another default language for a group. For example, if your server supports groups located in different countries, you can set the language for each group.

6. (Optional) Select the default user role. If you have implemented SSO over SAML, each member of the group is assigned this role when they log in for the first time.
7. (Optional) The **Session Inactivity Timeout** options are displayed only if you have selected Session Inactivity for your session management. You can use the system default for a group or manually set how many minutes must pass before the group's users are logged out.  
**Note:** If the user belongs to multiple groups, the highest timeout settings are applied to that user.  
For more information about these settings, see [Managing User Sessions](#).
8. Click **Create**.

# Adding Users to an Existing User Group

The following procedure describes how to add one or more users to an existing user group.

## **To add multiple users to a user group:**

1. Click **Admin** and select the **Users** tab on the left.
2. Select the relevant users.
3. Click **+ Add to Group**.
4. In the **Group Users** field, you can type additional usernames to add more users. An auto-complete list is displayed showing the users that already exist in Sisense. Select a username to add to the group.
5. Click **Add**.

# Editing an Existing User Group

## To edit an existing user group:

1. Click **Admin** and select the **Groups** tab on the left.
2. Click  on the right of the user group's entry in the list.

**Note:** To remove a user from the Group, hover over the user name in the list and click  .

# Deleting a User Group

When a user group that was entered in Sisense is deleted, its users are not deleted. They remain active users in Sisense.

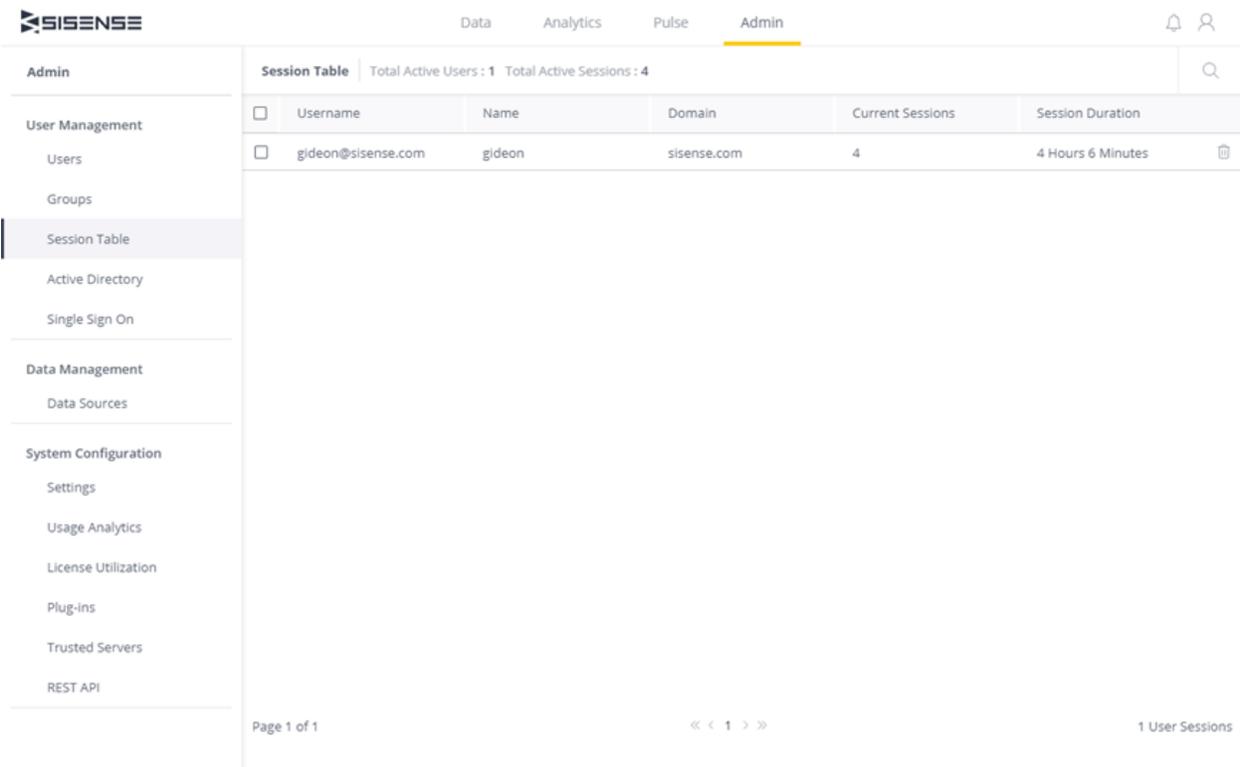
When a user group that was imported from Active Directory is deleted, a prompt is displayed asking whether you want to also delete all the users in this user group from Sisense.

## To delete an existing user group:

1. Click **Admin** and select the **Groups** tab on the left.
2. Click  at the right of the user group's entry in the list.

## User Sessions

You can view a list of all the users currently logged into your Sisense account in the Session Table.



The screenshot shows the Sisense Admin interface. The top navigation bar includes the Sisense logo, tabs for Data, Analytics, Pulse, and Admin (which is highlighted), and a user profile icon. The left sidebar contains a menu with categories: Admin, User Management (Users, Groups, Session Table, Active Directory, Single Sign On), Data Management (Data Sources), and System Configuration (Settings, Usage Analytics, License Utilization, Plug-ins, Trusted Servers, REST API). The main content area displays the 'Session Table' with a search icon and summary statistics: 'Total Active Users : 1' and 'Total Active Sessions : 4'. The table has columns for Username, Name, Domain, Current Sessions, and Session Duration. One user is listed: 'gideon@sisense.com' with name 'gideon', domain 'sisense.com', 4 current sessions, and a session duration of 4 Hours 6 Minutes. A trash icon is visible at the end of the row. The footer shows 'Page 1 of 1', navigation arrows, and '1 User Sessions'.

Username	Name	Domain	Current Sessions	Session Duration
gideon@sisense.com	gideon	sisense.com	4	4 Hours 6 Minutes

The Sessions Tables provides the following information about the users currently logged into your Sisense account:

- **Username:** The Sisense username
- **Name:** The user's first and last name
- **Domain:** The part of the user's email address after the @ sign
- **Current Sessions:** The number of sessions the user has open
- **Session Duration:** The amount of time from the first session until now

The Session Table is useful for monitoring who is currently logged in and who might be abusing their license. For example, if you need to declare some downtime, you can view the currently logged in users and end their sessions manually by revoking their session.

If you believe that a user is abusing their license, you can check the **Current Sessions** column to see how many sessions they currently have open. Users can open multiple tabs in their browser to open multiple sessions, or by sharing their account information with other users. To view which users have the most sessions currently open, click the **Current Sessions** table heading.

Current Sessions 

**Note:** User session in Sisense Mobile are not included in the Sessions table.

# Revoking a Session

From the Session Table, you can revoke a user's session. Revoking the session ends the user's current session. It does not revoke the user's API tokens.

When a user's session ends, the next time the user tries to perform an action in Sisense, a message is displayed that the session has ended and your users are prompted to log in again.

## To revoke a session:

1. In the **Admin** page, select **Sessions Table**.
2. Locate the relevant user in the Session table. You can click  and enter the user's information in the **Search** field to locate the user quickly. As you begin to type, the results are filtered.
3. For the relevant user, click  and confirm that you want to revoke the session by clicking **Revoke**. The next time the user performs an action in Sisense, they are redirected back to the Login page.

# Setting Session Inactivity Timeouts

When a user signs into Sisense, a session cookie is stored in their browser. The session cookie allows the user to remain logged in and authenticated even after ending a session for 7 days.

For security reasons, you may want to end your user's session sooner. Sisense provides two methods for ending a user session, according to the user's cookie or by the amount of session inactivity as recorded by Sisense.

When a user's session ends, the next time the user tries to perform an action in Sisense, a message is displayed that the session has ended and your users are prompted to log in again.

After making any changes to your users' session timeout settings, all of your users' sessions are terminated and the user is asked to log in again.

If you have implemented SSO, your users will be logged out and redirected to the IdP to reauthenticate.

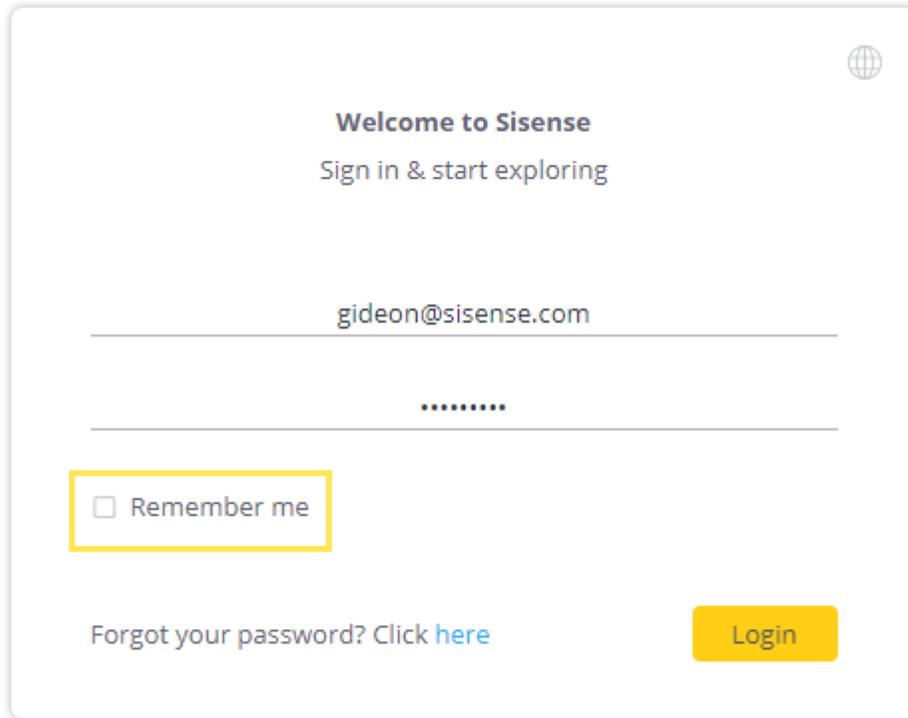
## To set the session inactivity timeouts:

1. In Sisense, open the **Admin** page and select **Settings**.
2. In the **Session Management** area, select the method you want to use for defining when a user's session ends:

**Cookie:** Allows you to define how much time must pass before a user's session is ended according to the user's cookie. This is the default behavior for Sisense.

- a. **Expiration(days):** The amount of days that must pass for a user's session to end. The default is 7 days. You can enter 0 so your users' sessions do not expire so long as they have the cookie.
- b. **Show "Remember Me" checkbox on Login screen:** Toggle to remove the **Remember Me** checkbox from the Login screen. Disabling this option means that if your user ends a session, they must log in

again the next time they try to access Sisense.



3. **Session Inactivity:** Allows you to define how many minutes of inactivity must pass before a user or an Administrator is logged out.  
**Note:** By selecting this option, you can set a custom session timeout for your Sisense user groups. See [Managing User Groups](#) for more information.
  - a. **Default Inactivity Timeout (min.):** Determines how much time must pass before a user is logged out. The default is 30 minutes.
  - b. **Admin Inactivity Timeout (min.):** Determines how much time must pass before an Administrator is logged out. The default is 30 minutes.
4. Click **Save** to save your changes.

## Customizing User Roles

This topic describes how to customize the permissions given to the Sisense user roles, using Sisense's [REST API](#). At the bottom of the page, you can find a video tutorial demonstrating the following procedure.

[Click here](#) to learn more about each role type.

**Note:** The API uses a previous terminology for Designers and Viewers. When entering the role name into the operation parameters, use Contributor (for Designer), and Consumer (for Viewer).

# Customizing Roles

You can customize the default roles with Sisense's REST API, by defining which permissions to enable or disable for each role. For example, you can prevent a user from drilling into widgets, or exporting data to a CSV file. On the other hand, you may want to let viewers change the widget type and format the style.

The following table lists all the actions that you can customize using the API.

True and False indicate values that you can change. Never indicates values that you cannot change.

## Dashboards

Dashboards	Admin	Data Admin	Data Designer	Designer	Viewer
create Create a new dashboard.	True	True	True	True	Never
duplicate Duplicate an existing dashboard.	True	True	True	True	False
change_owner Change the dashboard' owner.	True	True	True	True	False
toggle_edit_mode Select between Layout and View modes in the dashboard. Layout mode is the default mode in which you can add new widgets and	True	True	True	True	False

<b>Dashboards</b>	<b>Admin</b>	<b>Data Admin</b>	<b>Data Designer</b>	<b>Designer</b>	<b>Viewer</b>
rearrange them on the dashboard. In View mode, the dashboard cannot be rearranged or edited.					
edit_script Edit a dashboard using the Sisense JavaScript API.	True	True	True	True	Never
export_dash Export a dashboard as a .dash file that can be imported back into Sisense.	True	True	True	True	Never
export_jpeg Download a widget as an image in PNG format. The downloaded image represents the current view.	True	True	True	True	True
export_pdf Export the dashboard to PDF format. The exported file includes the current view.	True	True	True	True	True
restore Revert changes made in the dashboard to the latest dashboard that was shared.	True	True	True	True	True

<b>Dashboards</b>	<b>Admin</b>	<b>Data Admin</b>	<b>Data Designer</b>	<b>Designer</b>	<b>Viewer</b>
copy_to_server Copy the dashboard to another server.	True	True	True	True	False
import Import a .dash file into the Sisense Web Application.	True	True	True	True	Never
select_palette Change the selected color palette.	True	True	True	True	Never
replace_datasource Select a new data source for a dashboard.	True	True	True	True	False
undo_import_dash	True	True	True	True	False
Filters					
create Create a new filter.	True	True	True	True	False
on_off Switch filters on or off.	True	True	True	True	True
toggle_expansion Enable the expansion of filter settings to see more filter parameters.	True	True	True	True	True
toggle_auto_update Include the option to update	True	True	True	True	False

<b>Dashboards</b>	<b>Admin</b>	<b>Data Admin</b>	<b>Data Designer</b>	<b>Designer</b>	<b>Viewer</b>
the dashboard following every change					
modify_type Include in the filter options 'Ranking' and 'Starred', in addition to 'List' and 'Text'.	True	True	True	True	False
set_defaults Include in the dashboard menu the option to set the dashboard filters as the default filters for the dashboard.	True	True	True	True	False
advanced Include the Advanced menu option for creating custom filters.	True	True	True	True	False
use_starred Include the option to use filters that have been starred	True	True	True	True	False

### Widgets

<b>Widgets</b>	<b>Admin</b>	<b>Data Admin</b>	<b>Data Designer</b>	<b>Designer</b>	<b>Viewer</b>
create Create a new widget.	True	True	True	True	Never

<b>Widgets</b>	<b>Admin</b>	<b>Data Admin</b>	<b>Data Designer</b>	<b>Designer</b>	<b>Viewer</b>
delete Delete a widget.	True	True	True	True	False
rename Rename a widget.	True	True	True	True	False
duplicate Duplicate a widget.	True	True	True	True	Never
copy_to_dashboard Copy a widget to a dashboard. This functionality allows you to drag and drop widgets to additional dashboards in your Navigation Pane.	True	True	True	True	False
edit_script Edit a widget using the JavaScript API.	True	True	True	True	Never
change_type Change the chart type.	True	True	True	True	False
export_csv Export a widget's data to a CSV file.	True	True	True	True	True
export_png Download a widget as an image in the PNG format. The downloaded image represents	True	True	True	True	True

Widgets	Admin	Data Admin	Data Designer	Designer	Viewer
the current view.					
<p>modify_selection_mode</p> <p>Enable the option to change the filter behavior between 'Slice' (only filtered items are included in the chart) and 'Highlight' (filters items are highlighted).</p>	True	True	True	True	False
<p>drill_to_anywhere</p> <p>Enable the option to drill down into a field to get an in-depth view of a selected value. If False, the user can only drill down with predefined drill hierarchies (if defined).</p>	True	True	True	True	True
<p>add_to_pulse</p> <p>Include the option to add a widget to Pulse if the widget type is supported.</p>	True	True	True	True	True
<b>Items</b>					
<p>reorder</p> <p>Reorder the fields and values in the data panel and their representation in the chart.</p>	True	True	True	True	False

<b>Widgets</b>	<b>Admin</b>	<b>Data Admin</b>	<b>Data Designer</b>	<b>Designer</b>	<b>Viewer</b>
<p>modify_type</p> <p>Enables the option to edit values and categories in the data panel.</p>	True	True	True	True	Never
<p>on_off</p> <p>Enable the option to turn fields on or off in the widget.</p>	True	True	True	True	True
<p>select_hierarchies</p> <p>Include the option to select pre-defined hierarchies.</p>	True	True	True	True	False
<b>Filters</b>					
<p>on_off</p> <p>Switch filters on or off.</p>	True	True	True	True	False
<p>toggle_expansion</p> <p>Enable the expansion of filter settings to display more filter parameters.</p>	True	True	True	True	True
<p>modify</p>	True	True	True	True	False
<p>modify_type</p> <p>Include in the filter options 'Ranking' and 'Starred', in addition to 'List' and 'Text'</p>	True	True	True	True	False
<p>use_starred</p> <p>Enables the option to star</p>	True	True	True	True	False

Widgets	Admin	Data Admin	Data Designer	Designer	Viewer
(bookmark as favorite) a filter selection.					

## ElastiCubes

This object is not displayed in the Sisense REST API, however, you can still toggle these features through the **POST** /roles/{idOrName}/manifest/{path} endpoint. The value of path should be in the format of manage/elasticubes/[**value**] where [**value**] is the feature listed in the table below, add or schedule\_build.

Parameter	Value	Description	Parameter Type	Data Type
idOrName	<input type="text" value="dataAdmin"/>	The name or ID of the role.	path	string
path	<input type="text" value="manage/elasticubes/add"/>	The path to the manifest. Enter the path to the setting that you want to change as it appears in the hierarchy. Do not include the setting in the path. Example: To change the 'Create' setting for dashboard filters, enter dashboards/filters/. For creating dashboards, enter /dashboards. See the reference table above for more settings.	path	string

ElastiCubes	Admin	Data Admin	Data Designer	Designer	Viewer
add Allows the user to add new ElastiCubes.	True	True	True	False	False
schedule_build Allows the user to schedule builds for an ElastiCube.	True	True	True	False	False
modify_data_security_rights	True	True	True	False	False

ElastiCubes	Admin	Data Admin	Data Designer	Designer	Viewer
Allows editing the row-based data security permissions.					

**To change permissions for a user role:**

1. In the Sisense Web Application, click **Admin** at the top right of the screen, and then **REST API** in the left menu.
2. Click **REST API Reference** to view the API documentation.
3. In the API Documentation, select version **0.9**.
4. Click on **/roles** to access the different API operations.
5. Find the operation that you require (see examples below), and type in the required parameters in the operation. You can click on **Model Schema** to see and add example code for the parameters.

When changing a permission for a role, you must use the **Path** parameter, which defines the permission you want to update and its relative path in the hierarchy that appears in the table above. For example, to change the 'Create' setting for dashboard filters, enter dashboards/filters/. For creating dashboards, enter /dashboards. See the reference table above for more

settings.

<b>Dashboards</b>	<b>Admin</b>	<b>Designer</b>
create	true	true
duplicate	true	true
toggle_edit_mode	true	true
edit_script	true	true
export_dash	true	true
export_jpeg	true	true
export_pdf	true	true
restore	true	true
import	true	true
select_palette	true	true
<b>Filters</b>		
create	true	true
on_off	true	true
toggle_expansion	true	true
path	dashboards/filters/	
visibility_type	true	true

6. Click **RUN** to apply the changes.

Here are some examples of what you can do.

### See all permissions for all roles

Method: **Get /roles**

#### Parameters

- **includeManifest** – Use the default setting (true)
- **compiledRoles** – Use the default setting (true)

#### Example API Call

`/api/roles?includeManifest=true&compiledRoles=true`

### See all settings for a specific role

Method: **GET** `/roles/{idOrName}`

#### Parameters

- **idOrName** – The role name can be one of the following: admin, contributor, consumer
- **compiledRole** – Use the default setting (true)

### Example API Call

`/api/roles/consumer?compiledRole=true`

### Change a specific setting for a specific role

Method: **PUT** `/roles/{idOrName}/manifest/{path}`

#### Parameters

- **idOrName** – The role name can be one of the following: admin, contributor, consumer.
- **Path** – Enter the path to the setting that you want to change as it appears in the hierarchy. Do not include the setting in the path. Example: To change the 'Create' setting for dashboard filters, enter `dashboards/filters/`. For creating dashboards, enter `/dashboards`. See the reference table above for more settings.

- **Manifest** – Defines whether to enable or disable the setting. For example, to enable 'create', type in: {"create": true}

Parameters

Parameter	Value	Description	Parameter Type	Data Type
idOrName	consumer	The name or ID of the role.	path	string
path	dashboard/filters	The path to the manifest. Enter the path to the setting that you want to change as it appears in the hierarchy. Do not include the setting in the path. Example: To change the 'Create' setting for dashboard filters, enter dashboards/filters/. For creating dashboards, enter /dashboards. See the reference table above for more settings.	path	string

body

```
{
  "dashboards": {
    "create": true,
    "duplicate": true,
    "toggle_edit_mode": true,
    "edit_script": true,
    "export_dash": true
  }
}
```

The updated manifest object. Defines whether to enable or disable the setting. For example, to enable 'create', type in: create: true

Parameter content type: application/json

```
{
  "dashboards": {
    "create": true,
    "duplicate": true,
    "toggle_edit_mode": true,
    "edit_script": true,
    "export_dash": true,
    "export_jpeg": true,
    "export_pdf": true,
    "restore": true,
    "import": true
  }
}
```

Click to set as parameter value

### Example API Call

/api/roles/viewer/manifest/dashboards/filters

**Note:** You can use POST to do the same action, but for POST all other options will be reset to their default settings.

### Restore a specific setting for a specific role

Method: **DELETE** /roles/{idOrName}/manifest/{path}

### Parameters

- **idOrName** – The role name can be one of the following: super, dataDesigner, dataAdmin, admin, contributor, consumer.
- **Path** – Enter the full path of the setting you want to change, as it appears in the hierarchy. Example: dashboards/filters/modify\_type. See the reference table above for more settings.

### Example API Call

/api/roles/viewer/manifest/dashboards/filters/modify\_type

### Restore all settings for a specific role

Method: **DELETE** /roles/{idOrName}/manifest/{path}

### Parameters

- **idOrName** – The role name can be one of the following: super, dataDesigner, dataAdmin, admin, contributor, consumer.
- **Path** – / Enter just a slash to restore all settings.

### Example API Call

```
/api/roles/viewer/manifest/
```

## Integrating Active Directory

Sisense supports integrating Active Directory users and groups with your current Sisense users, so that you can share dashboards and email reports with any of your users. This works in parallel with Sisense's existing User Management, so you can continue to add users and user groups that are not part of the Active Directory domain.

To enable Sisense to identify your Active Directory users, you import your Active Directory users and groups into Sisense. When you import an Active Directory user, Sisense automatically creates a Sisense user. When you import an Active Directory group, Sisense does not automatically create a Sisense user for each member. The first time a member of an Active Directory group logs into Sisense, then Sisense automatically creates a Sisense user. Active Directory users that do not log in, are not assigned a Sisense user until their first login.

For secure connections to Active Directory, Sisense supports LDAP-S (LDAP over SSL), which encrypts communication between Sisense and your Active Directory. If your Active Directory supports SSL, you can upload your PFX certificate and PFX password for each Active Directory domain while defining your Active Directory settings in Sisense. PFX files contain the public key file (SSL certificate file) and the associated private key file in a single file. These files are generated and provided by you.

This topic describes how to configure Active Directory in Sisense, and how to import users and user groups from Active Directory.

# Adding an Active Directory Domain to Sisense

From the Active Directory page, you can add Active Directory domains to your Sisense configuration. Each Active Directory domain you add to your Sisense configuration is added to the Active Directory list displayed in the Active Directory page. The first time you open the Active Directory page, the Add Domain link is displayed, which you can click to begin adding Active Directory domains.

After you have added your first Active Directory domain, it is displayed in the Active Directory list.

From the Active Directory list, you can add additional Active Directory domains. Each Active Directory domain you add to Sisense enables you to share your dashboards and reports with users from any of the domains as well as Sisense users not part an Active Directory.

You can filter the list of the Active Directory domains by entering the name of the relevant domain in the Search field.

## To add an Active Directory:

1. Click **Admin** and select the **Active Directory** tab on the left.
2. In Active Directory, click **Add Domain**.
3. In the **Add Domain** window, fill in the Active Directory configuration fields to point to your Active Directory instance:
  - **Name:** The name of your Active Directory domain. This name is used to locate your Active Directory domain in Sisense when you have multiple domains.
  - **Connection String:** The full path to your Active Directory domain. Example: **ldap://dc.domain.com** (Or **ldaps://dc.domain.com** for secure connections)
  - **Base DN:** The Active Directory location from where a server searches for users. Enter the domain by its distinguished name (DN) with its

domain components (DC) connected by commas.

For example: For the domain domain.sisense.test, use

**DC=domain,DC=sisense,DC=test.**

**Note:** If you have a hierarchy of user groups in your organization, you can define the hierarchy in your string as follows:

- If your domain is domain.sisense.com/OU1/OU2, then use **OU=OU2, OU=OU1,DC=domain,DC=sisense,DC=com**. Only users and user groups from OU2 will be available. Users/groups under OU1 will not be available.
  - **Username:** Enter the username with domain. Example:  
**domain\username**
  - **Password:** Enter the password for your Active Directory instance.
  - **Sync data every:** Enter how often Sisense automatically synchronizes with the users and groups in this Active Directory domain in milliseconds.
  - **SSL Enabled:** Select to enable a secure connection to your Active Directory. When you select **SSL Enabled**, the **PFX Upload** area and **PFX Password** field are displayed. In the **PFX Upload** area, drag the PFX file into the browser or click **Browse** and navigate to the file. In **PFX Password**, enter the password you received after your PFX was generated.
4. Click **Test** to verify that your configuration successfully connects to your Active Directory domain.
  5. Click **Save**.

## Editing Active Directory Domain Settings

### To edit an Active Directory domain:

1. From the Active Directory list, select  for the Active Directory settings you want to edit. The Edit Active Directory window is displayed.
2. In the Edit Active Directory window, edit the relevant fields. See **Adding an Active Directory in Sisense** for more information about each field.
3. Click **Save**.

## Disabling an Active Directory Domain

Supports allows you to disable Active Directory domains in Sisense.

An Active Directory domain that has been disabled is still displayed in the Active Directory list, however, users from this domain will not be able to log in to Sisense, and you will not be able to share dashboards or reports with them.

Disabled Active Directory domains have the status Disabled in the Status column in the Active Directory list.

### **To disable an Active Directory domain:**

- From the Active Directory list, select ☰ > **Disable** for the Active Directory domain you want to disable. The Active Directory domain is disabled. You can enable the Active Directory domain by selecting ☰ > **Enable**.

## Deleting an Active Directory

You can delete an Active Directory domain from your configuration. When you delete an Active Directory domain, the users included in that domain are no longer displayed when sharing dashboards or reports. Deleted Active Directory domains cannot be restored, however you can add the Active Directory domain again from the Active Directory page.

### **To delete an Active Directory domain:**

1. From the Active Directory list, select  for the Active Directory settings you want to delete. The Delete Active Directory window is displayed.
2. Click **Delete**. The Active Directory domain settings are removed from Sisense.

# Adding a User from Active Directory

Once a connection with Active Directory has been established, an additional **Import from External Server** option is added to the **Add Users** button in the User tab.

The user's username and email will be added from Active Directory.

**Note:** You can add a user from Active Directory, even if you have not imported that user's user group from Active Directory.

See [Managing Dashboard Users](#) for general information about working with users/user groups in Sisense.

# Adding a User Group from Active Directory

Once a connection with Active Directory has been established, an additional **Import from Active Directory** option is added to the **Add Group** button in the **Groups** tab.

The following procedure describes how to import a list of users from a group in Active Directory to Sisense.

Users that are added in this way only appear in the Sisense users list after each user logs in to Sisense. In this way, your Sisense license slots are only consumed by users who actually log in to Sisense.

## To add a user group from Active Directory:

1. Click **Admin** and select the **Groups** tab on the left.
2. Click the **Add Group** button and then select the **From Active Directory** option from the menu.
3. In the **Search Active Directory Group** field start typing in the name of the desired Active Directory Group. An auto-complete dropdown menu is displayed from which you can select a user group.



If an Active Directory user group has already been imported into Sisense, the words **Group already selected** appear next to it, as shown above. You can add multiple Active Directory groups, by selecting one group after another. The groups appear in this window as follows:

4. Select the default Role for the members of this user group: Viewer or Designer. It will be assigned to new users from this group, when the users are created. Subsequently, user's role can be changed in the Sisense Users list.

Changes to a group's default role will affect users created after that time, but will not affect existing users.

5. Click **Add**.

## Introduction to SSO

Single Sign-On (SSO) is a mechanism that allows a system to authenticate users and subsequently tell Sisense that the user has been authenticated. The user is then allowed to access Sisense without being prompted to enter separate login credentials.

The SSO security mechanism allows Sisense to trust the login requests it gets from your corporate authentication system, and grant access to the users that have been authenticated by it. An SSO session begins when the authenticated user requests a secured resource from Sisense while logged into your site or application. The user's browser sends an HTTP request to Sisense that includes a cookie which contains session and authentication information. This information is then used for session validation.

Users who already have Sisense accounts can continue to access Sisense through the Sisense Login page with their current accounts. To prevent users from directly logging in to Sisense instead of your login page, your Sisense administrator can change the passwords of your current users forcing them to log in with your company's credentials in your company's login page.

Sisense recommends that administrators always keep a Sisense password, so that the administrator can access Sisense in case the SSO server is not available.

Sisense SSO supports two SSO protocols for securing the exchange of user authentication data, JSON Web Token (JWT) and SSO via SAML 2.0 (Security Assertion Markup Language). For more information, see [SSO via JWT](#) or [SSO via SAML 2.0](#).

## SSO via JWT

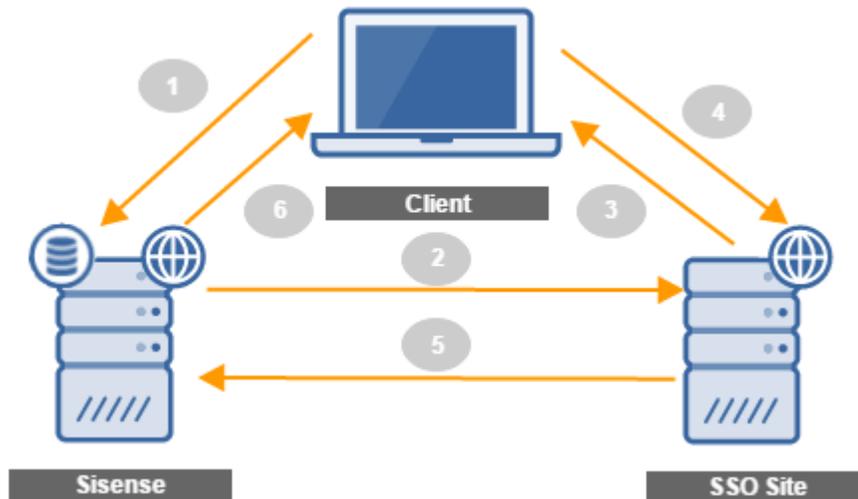
JWT is a technique that can be used for single sign-on (SSO) between your site and Sisense. JWT is a token that represents your users credentials wrapped in a single query string. In addition, Sisense uses the jti parameter (see below), which adds a unique ID to the token that prevents the token from being used more than once, thus preventing attacks on the system (also known as replay attacks).

The Sisense SSO via JWT authentication flow is explained in the following diagram.

**Note:** See the [Developers Community](#) for a tutorial that shows how to implement SSO via JWT.

# SSO Authentication Flow

The following is a diagram of the SSO authentication flow from your site or application to Sisense.



1. Your user requests a resource from Sisense, typically a dashboard.
2. Sisense recognizes that no authenticated cookie is present. If you have enabled SSO in Sisense, the SSO handler redirects the user to your Remote Login URL defined in the Sisense Web Application.

**Single Sign On Configuration**

Method  JWT  SAML 2.0 (i)

Remote Login URL  (i)

Remote Logout URL  (i)

Shared Secret  (i)

3. Your user is challenged to authenticate their account.
4. Your Remote Login application authenticates your user and generates a JWT (JSON Web Token).
5. You redirect the user back to Sisense with the encoded JWT in a query string. Sisense sets a cookie that authenticates the user's session until they

end it or you log them out via the Sisense REST API. For more information see [Logging Users Out](#).

6. Sisense provides the authenticated user with the request resource.

A common scenario that illustrates SSO is when an unauthenticated user navigates to your site in which Sisense is embedded via an iFrame. Sisense redirects this user to your SSO script. Your script authenticates the user through your login process and builds a JWT request with all the relevant credentials wrapped together. You then redirect the customer back to Sisense with the JWT payload. Sisense then decodes the user details from the JWT payload and then grants the user a session.

# Configuring SSO in Sisense

While SSO is highly customizable, there are generally four steps you should complete when configuring SSO:

**Note:** Configuring SSO requires technical expertise and should be conducted by an administrator or developer with SSO experience.

1. **Enabling SSO in Sisense:** Through the Sisense Web Application, an administrator can enable SSO in Sisense and define the relevant Login and Logout URLs.
2. **Creating a JWT:** After you authenticate a user, you generate a JWT with the user's credentials to Sisense, so Sisense knows this user is allowed to access resources from Sisense through your site.
3. **Configure Sisense as a sub-domain:** When authenticating users, you should configure SSO as a sub-domain.
4. **Logging Users Out:** A user can access Sisense so long as a session is maintained. To end a session, the user's cookie that Sisense provides must be deleted. To delete this cookie, you can use the Sisense REST API.

# Enabling SSO in Sisense

For Sisense to recognize that your users should be authenticated through SSO, you must enable SSO in the Sisense Web Application. In the SSO menu of the **Admin** page of the Sisense Web Application, you define the URL where Sisense redirects users to authenticate on your side and where Sisense redirects users after they log out from Sisense.

When you access the SSO menu of the **Admin** page, Sisense displays the Shared Secret key. The Shared Key is a JWT encryption public key used to encrypt the JWT payload. It is generated once when the SSO configuration is saved. You include this key in the JWT payload when redirecting the user back to Sisense after authenticating them on your side.

## To access and set up SSO:

1. Log into Sisense.
2. Select Admin screen and click **Single Sign On** in the left menu.
3. Fill in the following SSO configuration fields:
  - **Remote Login URL:** This is the URL that Sisense will invoke to attempt remote authentication. In that endpoint the participating application user authentication script is triggered and the JWT payload is generated.
  - **Remote Logout URL:** This is the URL that users will be redirected to after they log out from Sisense (i.e. the participating application's home page).
4. Click **Save**.

# Creating a JWT

Your script builds a JWT request that contains the user data.

The table below provides a list and descriptions of the attributes your JWT should contain.

In addition, several samples are provided below in various languages.

Attribute	Mandatory	Description
typ	Yes	Define the type of token with the attribute typ. You should also specify HS256 as the JWT algorithm in the header of your JWT payload. <pre>{   "typ": "JWT",   "alg": "HS256" }</pre>
iat	Yes	Issued at the time the token was generated. This is used to help ensure that a given token gets used shortly after it is generated. The value must be the number of seconds since UNIX epoch. Sisense allows up to five minutes clock skew. <b>Note:</b> The date must be an integer and not a float.
sub	Yes	Email of the user being signed in, used to uniquely identify the user in Sisense. If the user does not exist in Sisense, it will be created with default viewer privileges.
jti	Yes*	A unique string added to the token that is used to prevent replay attacks, by making sure the token

Attribute	Mandatory	Description
		is used only once.

\* You can set this attribute as optional in the Sisense REST API v1.0 through the POST settings/SSO endpoint.

SSO Code Samples

[C#](#)

[Java](#)

[Javascript](#)

[PHP](#)

[Ruby](#)

[Python](#)

## return\_to URL

When Sisense redirects a user to your login script, Sisense passes a `return_to` parameter in the URL. This parameter contains the page that Sisense will return the user to after the authentication succeeds. For example:

A customer visits your site opens a dashboard embedded through an iFrame.

Sisense recognizes that the user is not authenticated.

Sisense redirects the user to:

```
https://yourcompany.com/sisense/sso?return_to=https://yourcompany.sisense.com/dashboards/
```

All your script needs to do, is take the `return_to` value from the invoked URL and pass it back to Sisense when submitting the JWT token. In other words, upon authentication on your side, your script redirects the user to:

```
https://yourcompany.com/jwt?jwt=payload&return_to=https://yourcompany.sisense.com/dashboards/
```

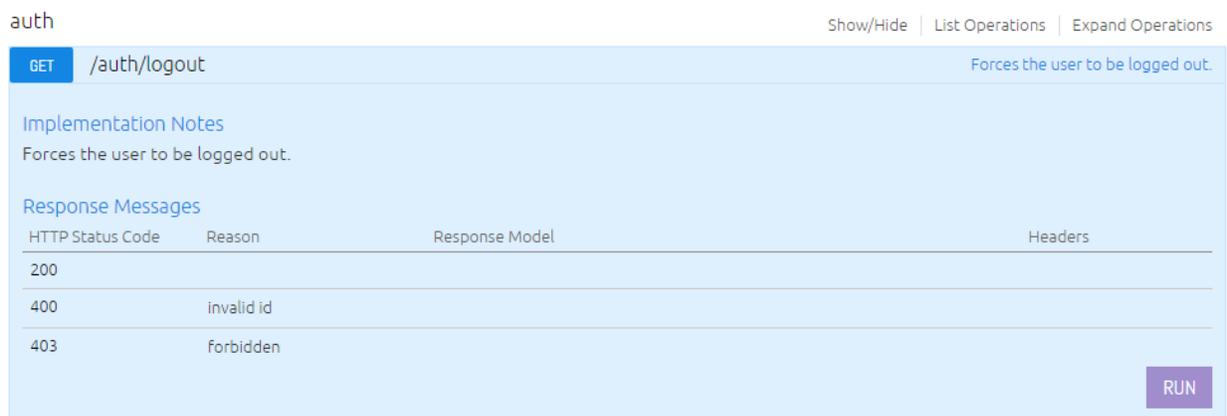
# Logging Users Out

When a user is logged in, anyone using that browser can access the session, or users may encounter an issue where they remain logged in until the Sisense cookie is cleared.

Users are logged out when the session ends. A session ends when the user closes their browser or according to the value of the attribute exp you send in the JWT payload.

You can log the user out through the Sisense REST API.

To manually log a user out, access version .9 of the REST API. Through the Auth method, you can issue a get request to log out specific users.



auth Show/Hide List Operations Expand Operations

**BET** /auth/logout Forces the user to be logged out.

**Implementation Notes**  
Forces the user to be logged out.

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
200			
400	invalid id		
403	forbidden		

**RUN**

While the logout REST API can delete the SSO authentication cookie, it can only delete it when the call is made from within the Sisense domain. Scripts on different pages can access each other only if the pages that executed them are at locations with the same protocol.

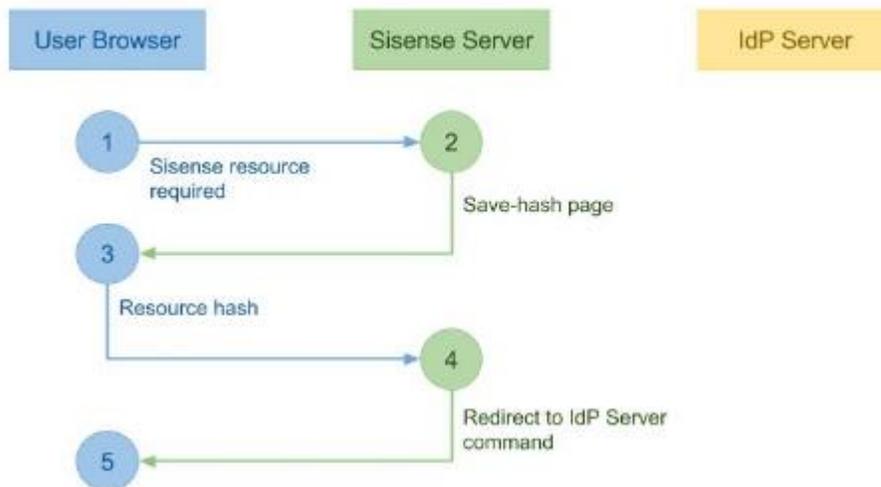
If you have embedded Sisense in an iFrame and you want to log out the user from your application and Sisense, you can use the window.postMessage method to call the logout when the users asks to logout from your application. This method overcomes any cross-origin communication limitations. Sisense has created a

plugin that implements a listener, which calls the Logout API when the `postMessage` method() is called. For more information, click [here](#).

## SSO via SAML 2.0

The Sisense SAML authentication process is based on the SAML 2.0 protocol, and is explained in the following diagrams:

### Sisense SAML auth workflow (1st phase)



# Phase 1

The first phase of this process begins when a user requests a resource from Sisense via their browser (1). The browser generates a resource request to the Sisense server. The server processes this request (2), and for unauthenticated users, returns a special save-hash page to save the requested hash data.

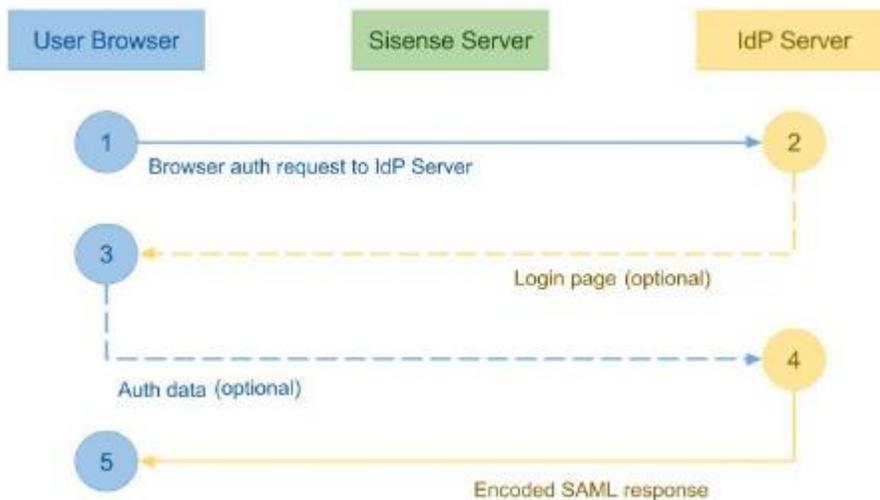
After the browser loads the save-hash page from the Sisense server, it runs a script, which loads the requested resource hash from the URL and sends it to the Sisense Server (3).

At the end of this phase, the Sisense Server converts the requested resource URL and sends it to the browser as a part of the redirect to IdP Server command (4).

## Phase 2

The second phase of the authentication process starts after the browser retrieves the redirect command from the Sisense Server, and sends the authentication request to the IdP Server (1) as described in the diagram below.

### Sisense SAML auth workflow (2nd phase)



The base URL for this request is taken from the Remote Login URL field in the Single Sign On section of the **Admin** page in the Sisense Web Application.

The requested resource address is passed to this request as a RelayState parameter. All other data is provided as a SAMLRequest parameter.

The next step of this phase depends on the user authentication state and the IdP implementation.

If the current user isn't logged in as an IdP user, the IdP server redirects the browser to your Login page (2), where the user enters their IdP credentials. After logging in, the IdP Server sends the browser the Auto-Sign-In page with encoded data about the currently logged-in user (4).

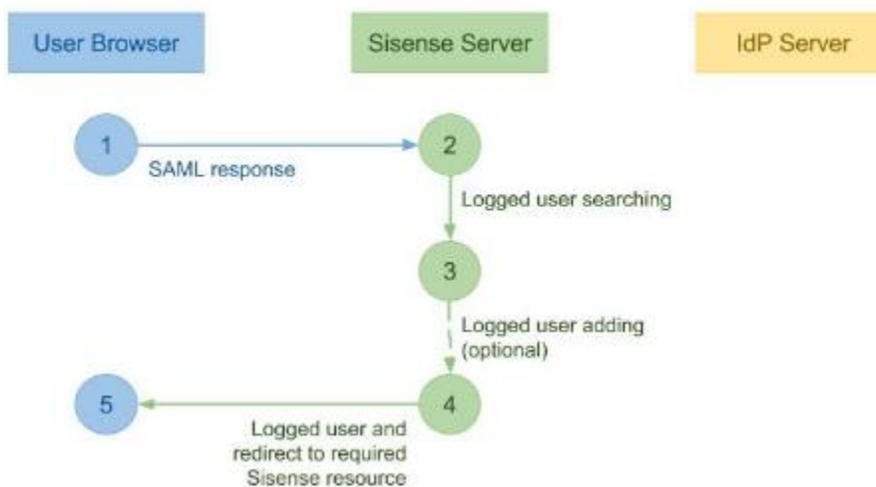
**Note:** Two-factor authentication for Sisense is supported for SSO providers that support two-factor authentication.

After this document is loaded in the user's browser, it runs a script which creates the POST-query to the Sisense Server API, and passes the SAML response to this query (1). At this time, the third phase of the authentication process begins.

## Phase 3

The Sisense Server handles the POST-query (2), decrypts it with the certificate specified in the Public X.509 Certificate field defined in the Single Sign On section of the **Admin** page. The Sisense Server uses the decrypted User ID as a key to locate the Sisense user in the internal database.

### Sisense SAML auth workflow (3rd phase)



If a user is not found in the system, Sisense creates a new user (3).

## Associating SSO Users with Sisense Groups

When Sisense creates a new user, Sisense analyzes the “memberOf” field to locate one or more groups related to the logged-in user.

If the “memberOf” field is empty, the user is assigned the role: “Viewer”. If the “memberOf” field contains one or more groups, and the groups were previously defined in Sisense, the newly created users’ Role is taken from the groups’ default role. When multiple Sisense Groups are found, the user is assigned the role with the maximum privileges.

After a user is created in the system, an administrator can modify the user role, if needed.

Below, is an example of an SAML XML where the “Test” group is specified:

```
<saml:Attribute
  NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic
  Name="memberOf">
  <saml:AttributeValue
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
    xsi:type="xs:string">
    Test
  </saml:AttributeValue>
</saml:Attribute>
```

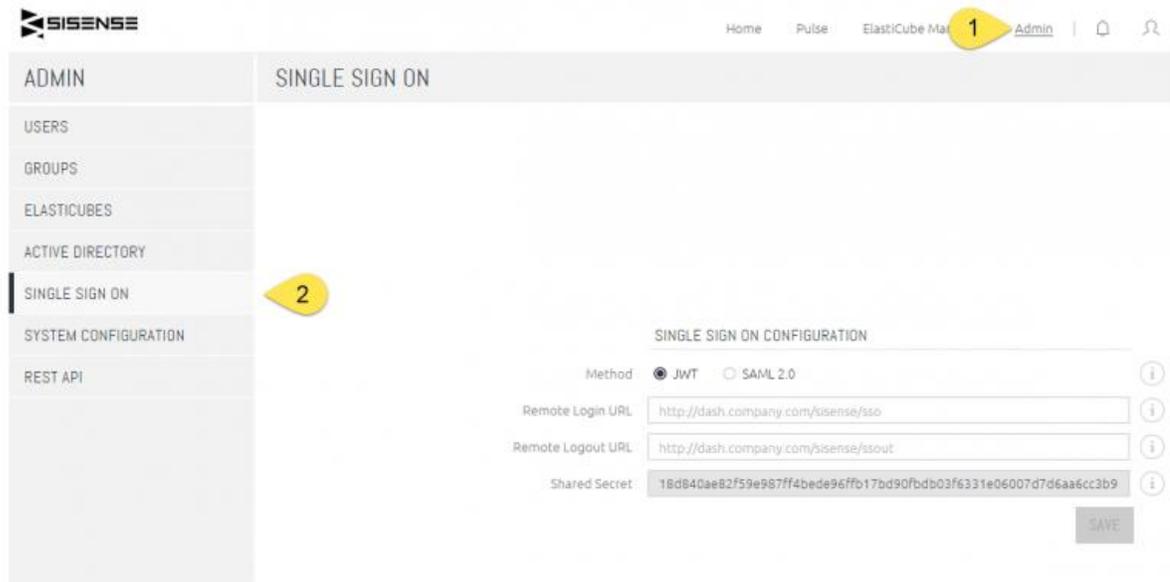
After the user is located or created, a new session is initialized, and the Sisense Server redirects the user browser to the requested resource (4).

# Enabling SSO via SAML in Sisense

After you have configured your SAML server, sign in to Sisense as an Administrator and follow the instructions below.

## To enable SAML in Sisense:

1. In the Sisense Web Application, click **Admin** and select **Single Sign On**.



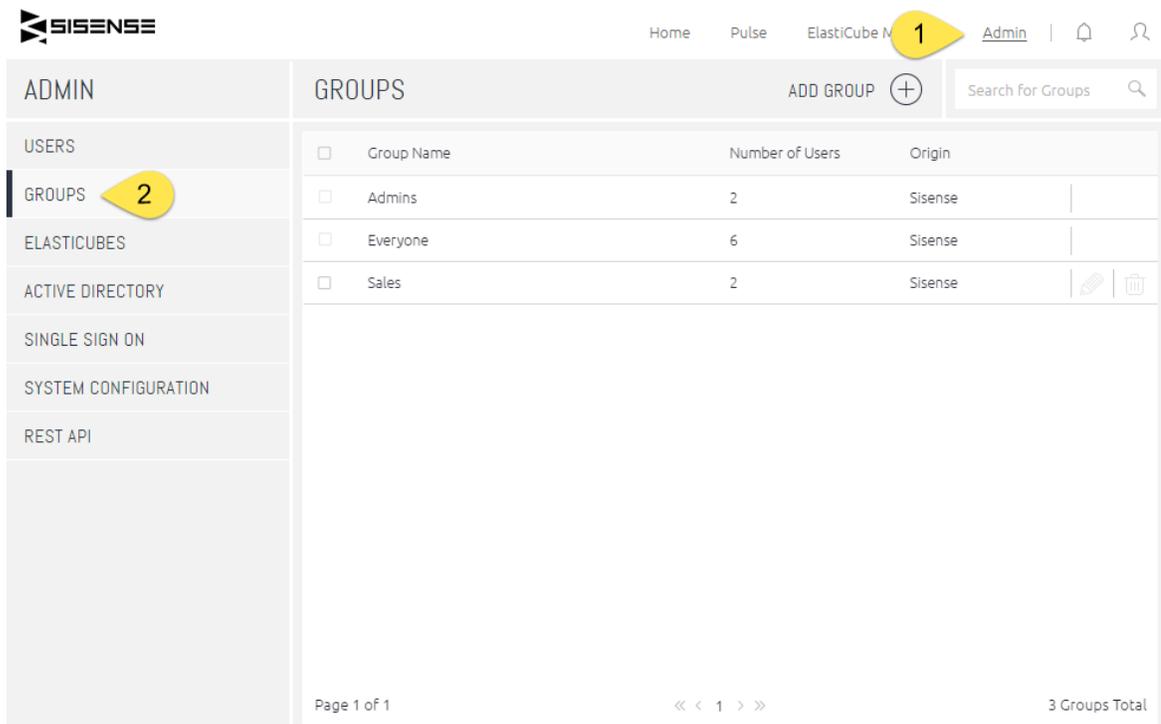
2. In the Single Sign On page, select **SAML 2.0**.
3. In the **Remote Login URL** field, enter the SAML Login endpoint. Sisense redirects the user to this field when they sign in. This value should be provided by the IdP Service.
4. In the **Remote Logout URL** field, enter the SAML Logout endpoint. Sisense redirects the user to this field when they sign out. This value should be provided by the IdP Service.
5. In the **Public X.509 Certificate** field, enter your public key for your SAML configuration. This value should be provided by the IdP Service.
6. Click **Save**. SSO via SAML 2.0 is configured.

# Sisense Default Role Set-Up

When an authenticated user is not found in the Sisense database, a new account is created. The user role is specified based on the user group/groups default role.

## To define a group's role:

1. In the Sisense Web Application, click **Admin** and select **Groups**.



The screenshot shows the Sisense Admin interface. The top navigation bar includes 'Home', 'Pulse', 'ElastiCube M', and 'Admin' (highlighted with a yellow circle labeled '1'). The left sidebar contains 'ADMIN', 'USERS', 'GROUPS' (highlighted with a yellow circle labeled '2'), 'ELASTICUBES', 'ACTIVE DIRECTORY', 'SINGLE SIGN ON', 'SYSTEM CONFIGURATION', and 'REST API'. The main content area is titled 'GROUPS' and features an 'ADD GROUP' button with a plus sign and a search bar labeled 'Search for Groups'. Below this is a table with the following data:

<input type="checkbox"/>	Group Name	Number of Users	Origin	
<input type="checkbox"/>	Admins	2	Sisense	
<input type="checkbox"/>	Everyone	6	Sisense	
<input type="checkbox"/>	Sales	2	Sisense	 

At the bottom of the page, it shows 'Page 1 of 1', navigation arrows, and '3 Groups Total'.

2. Click **Add Group**. The Create a New Group window is displayed.

3. In the **Create a New Group** window, select the default role of the group.

### Create a New Group

---

Group Name

Group Users

No members added yet

Choose default members role:  Viewer  Designer

4. Click **Save**.

**Note:** Changes to the group's default role are applied when new users are created, and do not affect existing users. After a user is created in the system, an administrator can modify the user role, if needed.

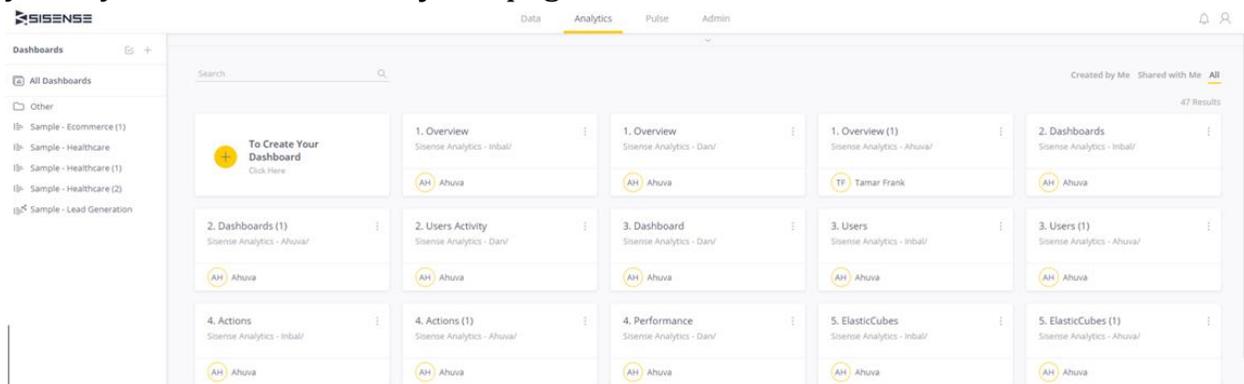
## Related Topics

For instructions on setting up Sisense with some 3rd-party providers, see:

- [Setting Up SSO SAML 2.0 with Okta](#)
- [Setting Up SSO SAML 2.0 with OneLogin](#)
- [Setting Up SSO SAML 2.0 with Auth0](#)
- [Setting Up SSO SAML 2.0 With Salesforce](#)
- [Setting Up SSO SAML 2.0 With ADFS](#)
- [Setting Up SSO SAML 2.0 With G Suite](#)

## Managing Your Users Dashboards

As a Sisense Administrator, you can manage and interact with all dashboards in the system, including your dashboards and dashboards of other users across your system from the **Analytics** page.



For Administrators, the **Analytics** page is a centralized location for managing all the dashboards in your system and performing administrative actions on them.

As an Administrator, you can perform the following operations:

- Sharing the dashboard with other users in the system, or change the dashboard's owner
- Accessing a user's dashboard as a Viewer
- Copying the dashboard to another server
- Downloading a PDF of the dashboard
- Deleting the dashboard

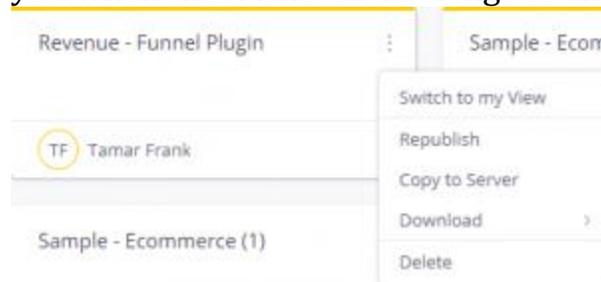
This is in addition to the functionality available to you for your own dashboards.

When you click the **All Dashboards** button on the **Analytics** page, you can see all your dashboards and the dashboards of your users. A yellow border displayed on top of the dashboard tile indicates that you are looking at the dashboard version of one of your users, and not your copy of the dashboard. This can be useful to view a dashboard that the owner modified, but has not yet republished the changes.



You can click the dashboard tile to open the dashboard. When opening a dashboard not shared with you, you will see the dashboard as a Viewer without editing rights. If you need to modify the dashboard, you must first make yourself the owner of the dashboard. You will then have full edit permissions for the dashboard.

The menu on the right of the dashboard tile is from where you can manage your or your users' dashboards. The example below includes the options available to you when the dashboard belongs to another user.



Keep in mind that the **All Dashboards** button is displayed only if the grid-view area of the **Analytics** page is available to you. If you have rebranded the whole **Analytics** landing page, the **All Dashboards** button will not be displayed. If you rebrand just the top part of the **Analytics** landing page, the **All Dashboards** button will be available. For Administrators, if you have enabled the [Dashboard](#)

[Admin](#) feature, the default Sisense **Analytics** page is displayed while all non-admin users continue to see the rebranded page.

You can rebrand the top part of the **Analytics** and **Data** pages by defining the URL of the marketingPart key, which is part of the [POST /branding endpoint](#). The marketingPart key is an iFrame that displays the Welcome banner and Sample dashboards and ElastiCubes.

**Note:** This feature can be disabled over REST API, if you are not interested in exposing the system dashboards to Administrators. See the isAdminAccessAllDashboards key of the [POST /settings/system](#) endpoint for more information.

For information on improving dashboard performance for your users, see [Maximize Dashboard Performance](#).

## Data Management

The topics below describe how you can share data models with other users and you can create drill hierarchies that you want to make available for Designers:

- [Adding and Removing ElastiCube Servers](#)
- [Sharing ElastiCubes](#)
- [Managing Drill Hierarchies](#)

## Adding and Removing ElastiCube Servers

This topic describes how to manage your ElastiCubes.

By default, the Sisense Web Application is connected to the ElastiCube server that is installed with it on the same machine.

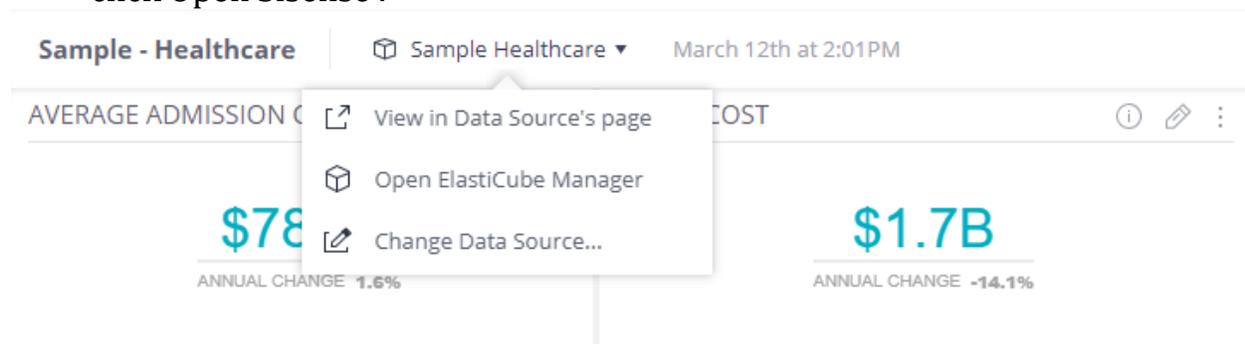
By default, all ElastiCubes on that machine are accessible to all dashboard users, and are visible under LOCALHOST.

# Opening an ElastiCube in Sisense

You have a number of ways to open your ElastiCube in Sisense:

From the Sisense Web Application:

- If you have an open dashboard, click on the ElastiCube name next to the dashboard name. Select View in ElastiCube's Page to open the ElastiCube management page in the web app. In the ElastiCubes page, select the ElastiCube and click Open. To open the ElastiCube of the open dashboard, click Open Sisense .

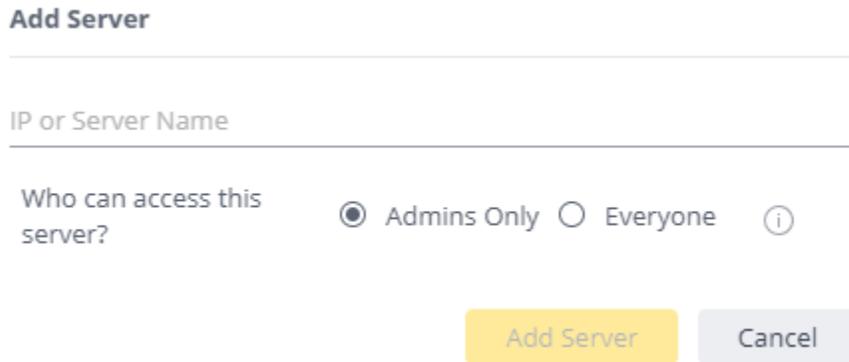


- From Windows: From the Start menu, open the ElastiCube Manager.

# Adding Another ElastiCube Server

## To add another ElastiCube Server to Sisense:

1. Click **Admin** and select the **Data Sources** tab on the left.
2. Click **Add Server**.



**Add Server**

---

IP or Server Name

---

Who can access this server?  Admins Only  Everyone ⓘ

**Add Server** Cancel

3. Select whether this ElastiCube Server is intended for Admins only. This selection also applies to all new ElastiCubes that will be added to this ElastiCube Server. A user that has access to an ElastiCube Server can access all the ElastiCubes on it.
4. Add the IP of the ElastiCube Server to which you want to connect and click Add Server. The new ElastiCube Server is added to the list with all its ElastiCubes.
5. Click Save. Once a new ElastiCube Server has been added, its ElastiCubes immediately become available to all designers (by default) when they create a new Dashboard.
6. To define which users are allowed to access the data of an ElastiCube Server, see [Assigning Rights to an ElastiCube Server](#).
7. To define which users are allowed to access each specific ElastiCube, see [Assigning Rights to an ElastiCube](#).

# Removing an ElastiCube Server

**To remove an ElastiCube server from the list:**

- In the Data Sources page, for the relevant server, click the Trash  button.

**Note:** The LOCAL MACHINE server cannot be removed.

## Sharing ElastiCube Models

You can share ElastiCube models with other Sisense users through the **Admin** page.

There are two types of access rights that can be assigned to a Sisense user, edit rights and query rights. Edit rights allow the user to modify the ElastiCube model, while query rights allow the user to query or create a dashboard based on the shared model, but not modify it.

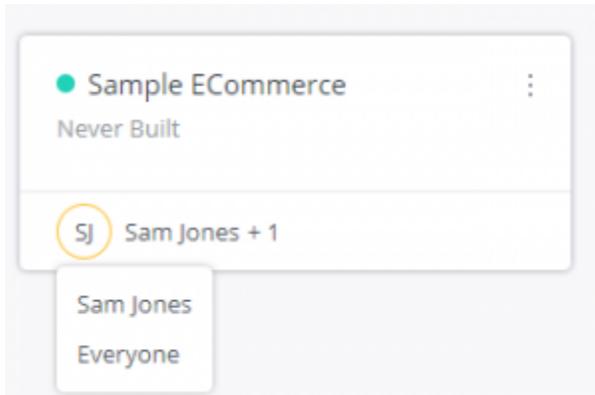
The user's role determines what access they have and what access can be assigned to them when the ElastiCube model is shared.

Administrators and Data Administrators have edit rights to the ElastiCube model, even without sharing the ElastiCube model with them. Data Designers can have edit or query rights, while Designers can only have query rights.

When an ElastiCube model is created on the **Data** page, it is not shared with anyone. This allows the model's creator to work on the model and publish it only once it's ready.

When you assign specific users/user groups access rights to this ElastiCube, then only the users/user groups that you assign can access the ElastiCube.

When an ElastiCube is shared with other users, the ElastiCube is displayed in the **Data** page with a list of the other shared users.



## To share an ElastiCube:

1. In the Sisense, select **Admin > Data Sources** and then the menu icon for the relevant ElastiCube.

SISENSE

Data Analytics 1 Admin

Admin

Elasticubes & Data Sets

Refresh List Add Build Alert Add Server Add Live Connection

Local Machine

ElastiCube	Editors	Last Model C...	Last Built	Size On Di...	Status
Sample ECommerce	Everyone (group)	10/12/17 1:28 PM		54.92MB	Running 3
Sample Healthcare	gideon@sisense.com+1	7/2/13 1:07 AM	10/19/17 5:31 ...	3.17MB	Running
Sample Lead Generation	Everyone (group)	3/3/14 7:46 PM	10/19/17 5:31 ...	2.87MB	Running

Live Data Sets

There are currently no Data Sets.

2

3

2. Click **Share**.

**Share ElastiCube: Sample ECommerce**

Search for users and groups 🔍

<b>S</b>	<b>Sam Jones</b> sam@sisense.com	Can Query ▾	🗑️
<b>AB</b>	<b>Everyone</b>	Can Query	
<b>CD</b>	2 users currently	Can Edit	🗑️

**Save** **Cancel**

3. Enter the names or groups of users with whom to share the ElastiCube. Selecting Everyone specifies that this ElastiCube is shared with all Sisense users and user groups.
4. The contact information of the people you add is listed in the center of the window and is displayed the next time you open this window.
5. In the dropdown menu next to each user/user group select what rights the user has. (You can only select the rights for a Data Designer as all other roles have predefined permissions.)
6. Click **Save**.

## Managing Drill Hierarchies

Dashboard viewers can drill down in most widgets to get an in-depth view of a selected value. To make it easier for viewers to select drill-down paths, dashboard designers can add predefined drill hierarchies to their widgets.

As an Administrator, you can create a pool of drill hierarchies, from which Dashboard Designers can select the drill hierarchies they want to include in the widgets they build.



This section describes how to create and manage the drill hierarchies that you want to make available for designers.

### To create a drill-down hierarchy:

1. Click **Admin** in the Sisense Web Application. In the left menu, select **Data Sources**.
2. For the relevant ElastiCube  $\ddagger$ , click and select **Manage Hierarchies**.
3. If this is the first hierarchy you are adding, select **Add Hierarchy**. If you already have hierarchies, click **Add Hierarchy** in the top-right corner.
4. Select the field. This will be the first field in the drill hierarchy, and the same field used in the widget.
5. Click **Add Field** to add the next level in the drill hierarchy. Repeat this step for as many levels as you need.

Source



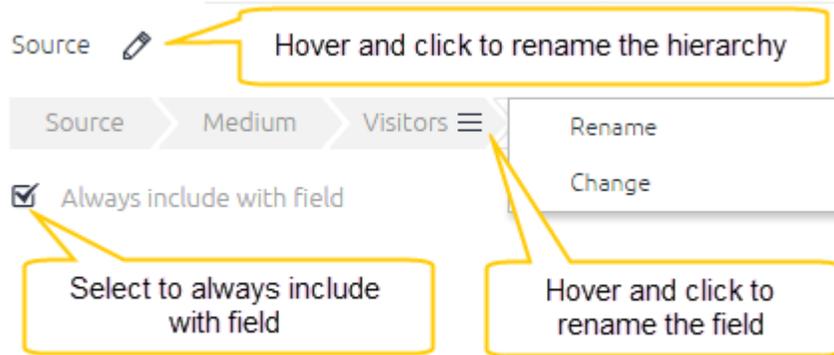
Additional options include:

**Always include with field:** Select this checkbox to always include the drill hierarchy in the widget for this field.

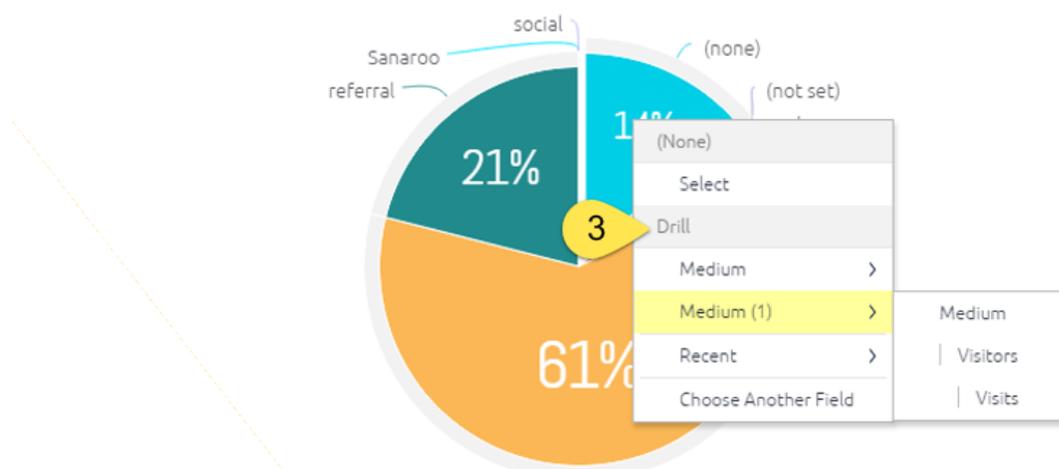
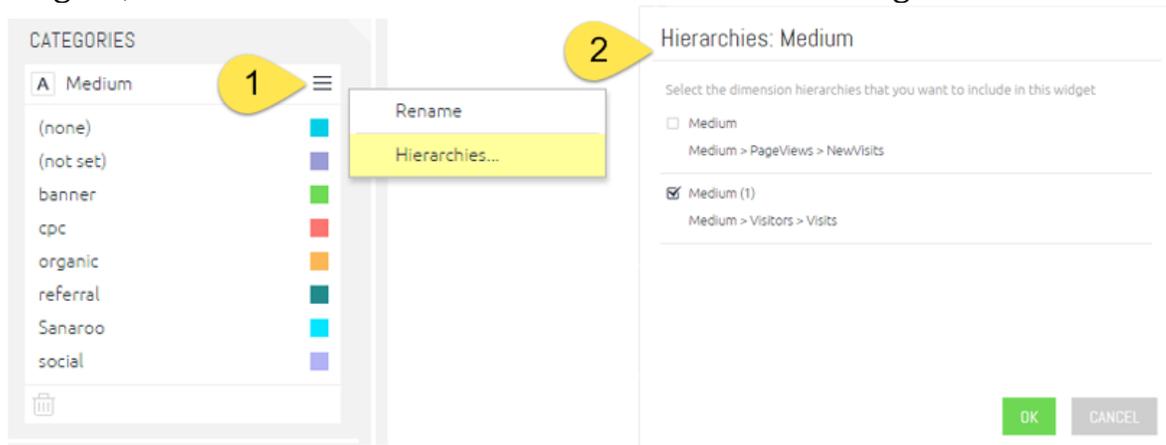
**Editing the Hierarchy name:** To edit the name of the drill hierarchy, click on the pencil icon, and type in the name. Click on the tick icon to confirm the change.

**Editing the field name:** To edit the name of a field in the hierarchy, hover over the field name and click on the menu > Rename. Type in the name, and

click on the tick icon to confirm the change.

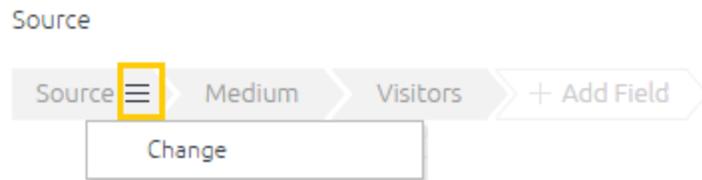


6. Add as many hierarchies as you need for the ElastiCube. When done, dashboard designers will have access to the hierarchies in the widget designer, and will be able to select and add them to the widget.



### To change a field in a hierarchy:

- Hover over the field in the hierarchy, and click on the menu that appears. Click **Change** and select a different field.



### **To delete a hierarchy:**

- Click on the bin to the right of the hierarchy.

## System Settings

Sisense Administrators can perform a variety of tasks related to implementing and maintaining an implementation of Sisense.

The following topics describe system-related tasks that Sisense Administrators are responsible for handling:

- [System Configuration](#)
- [Internationalization and Localization](#)
- [Managing Plug-ins](#)
- [Setting Up a Custom Email Server](#)
- [Sisense Narratives](#)

### Changing the Sisense Web Application's Language

Sisense supports multiple languages in the Sisense Web Application making it easier to extract meaningful insights from your data in your users' local language.

The following images display two interfaces, one in English and the other in Spanish.

**SISENSE**

Home

DASHBOARDS

- Sample - Ecommerce
- Sample Lead Generation

**Welcome gideon**

Explore Learn about our product and

Documentation

Our Video Library

Examples Need inspiration? Check o

E-Commerce

English

**SISENSE**

Principal

PANELES

- Sample - Ecommerce
- Sample Lead Generation

**Bienvenido gideon**

Explora Descubre más información sobre nuestro producto y ap

Documentación

Comunidad

Nuestros videos

Ejemplos ¿Busca inspiración? No se pierda estos ejemplos de pa

Comercio electrónico

Sanidad

Spanish

# What has been Translated?

Sisense has translated text that your users might see including menu names, buttons, messages, and other elements in the Sisense Web Application.

Translating the Sisense Web Application does not affect your data or how it is displayed.

The Sisense Web Application has been localized to the following languages:

- Chinese (Simple)
- French (France)
- German
- Italian
- Japanese
- Portuguese (BR)
- Russian
- Spanish (ES)
- Spanish (LA)
- Dutch

Note, the following content is not being localized:

- Built-in Javascript editor
- ElastiCube Manager and Server Console
- Sisense documentation and online help
- REST API and documentation
- Sisense automated emails

**Note:** You can use your own translation of Sisense automated emails by defining your own. [Click here for more information.](#)

# Changing Sisense's Default Language

Administrators can select the default language for the Sisense Web Application. Setting the default language affects all users and groups who access the Sisense dashboard. Additionally, Administrators can then set the language for specific groups and users when adding new users or by editing a group or user account. After selecting a new language, each string with a translation is displayed in that language across all your Sisense users' interfaces.

If you have defined a custom translation, the name of the language as defined in the settings.json is displayed in this list. See [Embedding Custom Languages](#) below for more information.

Sisense's default language is English. After changing the system language, you can revert back to the English or any other language by selecting it from the list.

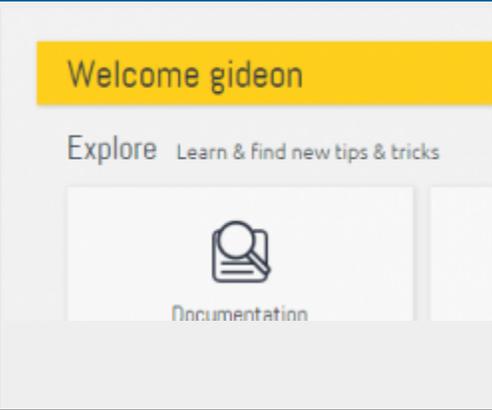
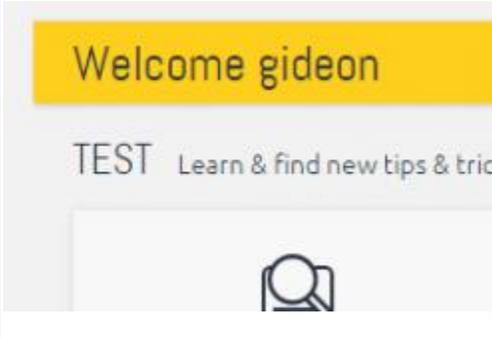
## **To modify your default language:**

1. In the Sisense Web Application, click **Admin** in the top menu, and then **Settings** in the left menu.
2. Select the relevant language from the Language list. After selecting the language, the UI is immediately translated.
3. Click **Save**. The Sisense Web Application is translated in the selected language.

# Embedding Customized Languages

Sisense enables you to display your own translations in the Sisense Web Application by replacing the values of strings in any language you choose. You can embed customized languages by modifying configuration and resource files replacing the string values according to your requirements.

For example, in the homepage of the Sisense Web Application, the word “Explore” is displayed. In the resource file home.js, you can modify the value of this string to display your own text as shown below.

	<pre>"welcome_user": "Welcome", "explore": { "explore": "Explore", "learn": "Learn &amp; find new tips &amp; tricks", "links": {</pre>
	<pre>"welcome_user": "Welcome", "explore": { "explore": "TEST", "learn": "Learn &amp; find new tips &amp; tricks", "links": {</pre>

In addition to modifying the resource files, you must modify the settings.json file and specify your language’s name and update the version. In this example, the language has been changed to “test” and the version increased from 1.0.0 to 1.0.1.

```
{ "displayName": "test", "version": "1.0.1" }
```

The value of “displayName” is displayed as your language’s name in Default Language list on the System Configuration page. In the example below, the displayed was set to Test.

Upgrading Sisense may replace the bundled translation files. If you wish to modify a bundled language, copy of all of the translation files to a new directory, and change the copied versions to prevent your translation from being overwritten.

### **To customize the Sisense Web Application’s language:**

1. Navigate to the directory `... \Program Files\Sisense\app\translations\`.
2. In the `\translations` directory, copy the **en-us(default)** directory, paste it in the `\translation` directory and rename it.
3. In the new directory, open each resource file and translate the strings into the desired language saving each file after your changes are made.
4. Open the settings.json file and replace the following key values:  
**displayName:** Enter the name of the language.  
**version:** Enter the name of the new version. Currently, the version is 1.0.0. For each version, increase this value, for example, 1.0.1.
5. Save the changes in a file called settings.json in your new language folder.
6. In the Sisense Web Application, click Admin in the top menu, and then Settings in the left menu.
7. Select your customized translation from the Language list. The Sisense Web Application is displayed in your customized translation.
8. Click **Save**.

## Internationalization and Localization

Sisense supports localization and internationalization by displaying your users’ native language in the Sisense Web Application. You can select the language displayed in the Sisense Web Application from a list of languages provided by Sisense or display your own translation. Each time you select a language, the Sisense Web Application interface is displayed in that language.

When you change the default language displayed in the Sisense Web Application, words that are part of the Sisense framework such as menu items, buttons, and messages are displayed in the selected language. Words specific to your dashboard, such as table names, field names, and filters, are not automatically translated when you select a new language to display. These words are called metadata and you can provide your own translation of these terms and display those translations to your users through a Dashboard script.

For instructions on how to change the language displayed in the Sisense Web Application, see [Changing the Sisense Web Application's Language](#).

For instructions on how to translate your metadata, see [Translating Sisense Metadata](#).

## Translating Sisense Metadata

Sisense enables you to translate metadata in your users' native language. Sisense metadata is displayed in your users' dashboards and includes the following:

- ElastiCube fields
- ElastiCube tables
- Widget Titles
- Dashboard Names
- Folder Names
- Saved formulas
- Default and custom hierarchies
- Customized renamed field titles

The following images display two interfaces with metadata, one in English and the other in Russian.

Brand	Бренд
Thrubanower Holdings Corp.	Thrubanower Holdings Corp.
Undimedat WorldWide	Undimedat WorldWide
Reseller	Reseller
APPLES	APPLES
WHITEBERRY	WHITEBERRY
MICRON	MICRON
Unerommazz Direct	Unerommazz Direct
LONY	LONY
POKIA	POKIA
NG	NG
Qwihupistor Holdings Group	Qwihupistor Holdings Group
« < 1 2 3 4 5 6 7 >	« < 1 2 3 4 5 6 7 >
English	Russian

As fields such as ElastiCube names and tables are defined by you, these fields are not translated when changing the default language of the Sisense Web Application.

You can translate your metadata by modifying the default values using a JavaScript plugin. Through the plugin, you define the new values to be displayed. You may also include additional logic to determine which users or user groups the translated values should be displayed to.

When you create a JavaScript plugin, you should create a folder for the plugin and place it inside the plugins folder in the following location:

**In Windows:**

... \Program Files\Sisense\app\plugins

**In Linux:**

/opt/sisense/storage/plugins

If the \plugins\ folder does not exist, create it and then create a new folder within the \plugins\ folder with the relevant files.

You can download an example of a Metadata Translation plugin [here](#).

Extract the enclosed folder into the plugins folder.

This file contains a folder called Metadata and two files:

**plugin.json:** The plugin.json file contains the name of the plugin and a link to the file that describes the plugin's functionality, in this example, run.js.

**run.js:** The run.js file contains the plugin's functionality and describes which metadata is to be translated and how it is to be translated.

This example, and any metadata translation JavaScript file you write should include three parts:

- **Provider Function:** These functions provide the mapping context contained in the Mapping object.
- **Event Subscription:** This event is triggered when a datasource changes, and calls the Provider function.
- **Mapping Object:** Contains key/value pairs with your default and translated metadata.

## Provider Functions

The code sample below is an example of two Provider functions, the first is a global provider function for folder and dashboard aliasing. The second is the datasource provider function for translating ElastiCube and formula names. You can replace the parameters in this example to provide the mapping context you define in the Mapping object.

### Global Provider

```
function (resolve, reject) {  
  //an ajax request or any other logic.  
  setTimeout(function () {  
    resolve(globalAliasing);  
  }, 0);  
};
```

The following table describes the Provider function and its arguments:

Parameter	Description
resolve	A callback function which should be called with the provided aliasing context object or without any parameter if aliases shouldn't be provided for the datasource.
reject	A callback function to be called in case of error, so the system can properly handle it.

### Datasource Provider

```
function (datasourceId, resolve, reject) {  
  //an ajax request or any other logic.  
  setTimeout(function () {  
    resolve(datasourceAliasing);  
  }, 0);  
};
```

```
};
```

The following table describes the Provider function and its arguments:

Parameter	Description
datasourceId	An object that represents the ID of the datasource with the following properties: <ul style="list-style-type: none"><li>• address</li><li>• database</li><li>• fullname</li><li>• ID</li><li>• title</li></ul>
resolve	A callback function which should be called with the provided aliasing context object or without any parameter if aliases shouldn't be provided for the datasource.
reject	A callback function to be called in case of error, so the system can properly handle it.

### Grouping Provider

The config object allows you to segment translated content per group. For example, Group 1 can see dashboards translated while group two can see the **Data** page translated. In the config object, you define the groupId and what alias is assigned to that group.

```
var config = [  
  {  
    "groupId": "5bbb4d8fd8528537d5604777",  
    "aliasName": datasourceAliasing  
  },  
  {  
    "groupId": "5bf3c5035f56071a1cf1b37c",  
    "aliasName": datasourceAliasing_1  
  }  
];
```

}

The following table describes the Provider function and its arguments:

<b>Parameter</b>	<b>Description</b>
groupID	Defines the group that an alias is to be assigned to. You can retrieve the groupID through the GET /groups endpoint in the Sisense REST API.
aliasname	Defines which alias object to use for the group defined in the groupID.

## Event Subscription

To trigger the Provider functions, your plugin should include a handler for a global event 'beforealiascontextinit'.

In the example below, `datasourceProviderFunction` and `globalAliasingProvider` are the Provider functions. The `timeout` parameter is optional and configures the amount of time in milliseconds which the system waits for the aliasing context to be resolved by the provider function. By default the system uses a timeout of 2000 ms

```
prism.on("beforealiascontextinit", function (ev, args) {  
  var timeout = 2000;  
  args.register(datasourceProviderFunction,  
    globalAliasingProvider, timeout);  
});
```

## Mapping Object

The translation context is a simple JavaScript object which contains key value pairs for tables, columns, hierarchies, formulas and custom item titles, all case-insensitive.

The sample below is a skeleton of a Mapping object that defines what metadata is to be translated and its new value.

```
/**
 * tables - aliases for table names
 * "tables": {
 * "table_name_1": "alias_1",
 * "table_name_2": "alias_2",
 * }
 * columns - aliases for column names in scope of table
 * "columns": {
 * "table_name_1": {
 * "column_name_1": "alias_for_column_name_1"
 * "column_name_2": "alias_for_column_name_2"
 * "column_name_3": "alias_for_column_name_3"
 * },
 * "table_name_2": {
 * "column_name_1": "alias_for_column_name_1"
 * "column_name_2": "alias_for_column_name_2"
 * "column_name_3": "alias_for_column_name_3"
 * }
 * }
 * titles - aliases for custom dimension titles, etc
 * "titles": {
 * "Custom title": "Alias",
 * "Sample Revenue": "Sample Profit"
 * },
 * formulas - aliases for custom formula titles
 * "formulas": {
 * "Custom title": "Alias"
 * },
 * hierarchies - aliases for hierarchy titles
```

```
* "hierarchies": {
* "Custom title": "Alias"
* }
*/
var datasourceAliasing = {
  "tables": {
    "brand": "бренд",
    "category": "категория",
    "country": "страна",
    "commerce": "коммерция"
  },
  "formulas": {
    "Revenue": "Чистый доход"
  },
  "hierarchies": {
    "Category by Brand and Age Range": "Категория через
Бренд и Год"
  },
  "titles": {
    "CUSTOM REVENUE": "ДОХОДЫ"
  },
  "columns": {
    "Brand": {
      "Brand": "Бренд",
      "Brand ID": "Бренд Id"
    },
    "Category": {
      "Category": "Категория",
      "Category ID": "Id Категории"
    },
    "Widgets": {
      "Revenue": "Чистый доход",
    },
    "Commerce": {
      "Age Range": "Возрастные диапазоны",
      "Brand ID": "Id Бренда",
      "Category ID": "Id Категории",
      "Condition": "Состояние",
      "Cost": "Затраты",
      "Country ID": "Id Страны",
```

```

    "Date": "Дата",
    "Gender": "Пол",
    "Quantity": "Количество",
    "Revenue": "Доходы",
    "Visit ID": "Id Посетителя"
  },
  "Country": {
    "Country ID": "Id Страны",
    "Country": "Страна"
  }
};
/**
 * Global object example
 * folder - aliases for folder names
 * "folder": {
 *   "folder_name_1": "alias_1",
 *   "folder_name_2": "alias_2",
 * }
 * dashboards- aliases for dashboard names
 * "dashboards- ": {
 *   "dashboard_name_1": "alias_1",
 *   "dashboard_name_2": "alias_2",
 * }*
 */
var globalAliasing = {
  "folders": {
    "New Folder": "Folder renamed",
    "my folder": "my folder renamed",
    "Any name": "любое название"
  },
  "dashboards": {
    "Sample - Ecommerce (1)": "Коммерческий Dashboard",
    "Sample - Healthcare": "Sample Healthcare renamed",
    "Sample - Healthcare (1)": "Healthcare renamed",
    "my dashboard": "my dashboard renamed"
  },
};

```

# System Configuration

The System Configuration settings let you define the web server's alias, the first day of the week, email settings, and more.

## To update one or more System Configuration settings:

1. Click **Admin** in the top menu, and then **Settings** on the left.
2. Update one or more of the following settings:
  - **Alias:** Define the web server's alias or IP address. This is especially important when you have customized your domain name (alias), for example, to prevent example, sharing dashboards from the incorrect IP address. Reports will be sent from the alias entered in this field.
  - **First Day of the Week:** Select the first day of the week so all of your Sisense users' queries will be calculated according to the day you define. By default, the first day of the week is Monday when querying an ElastiCube. Sisense defines Monday as the first day according to ISO 8601. This means that when selecting filters such as Week 1, Week 2, etc., the first day of the week is not that defined by first day of that year, but rather the first Monday of the year. Each week starts on a Monday and ends on a Sunday for 52-53 full weeks in a year.  
**Note:** Changing the first day of week after using a dashboard or widget filter/drill with Weeks filter in use might lead to bad results. To resolve the issue, the Weeks filter should be removed and added again.
  - **First Month of Fiscal Calendar:** Select the first month of your company's fiscal calendar. By default, the first month of the fiscal calendar is January.  
**Note:** Changing the first month of the fiscal calendar after using a dashboard or widget filter/drill with the related date fields is in use might lead to bad results. To resolve the issue the date filter should be removed and added again.
  - **Default Language:** Select the language to be displayed in Sisense. If you have defined a custom translation, the name of the language as defined in the settings.json is displayed in this list. Select it to display your custom translation. For more information about custom translations, [click here](#).
  - **Send Emails:** You can disable sending emails. This will disable emails sent when sharing dashboards as well as any scheduled email reports.

- **Pulse:** Toggle to remove the **Pulse** page from the top menu bar and the Pulse Notification Feed. This option hides the **Pulse** page from all users in your system and stops sending Pulse notifications.
- **Dashboard Administration:** Toggle to enable/disable the dashboard administration feature. Turn this feature off if you do not want Administrators to have access to all of the dashboards of every user in the system in the **Analytics** page. For more information, see [Managing Your Users Dashboards](#).
- **Widget Narrative:** Toggle to allow your Designers to add narratives to widgets. For more information about this feature, see [Sisense Narratives](#).
- **Field Suggestions:** You can toggle on field suggestions. Field suggestions are displayed to Dashboard Designers to help them add the relevant fields to widgets quickly and easily. For more information, see [Field Suggestions](#).
- **Session Management:** You can choose which method to use for handling the expiration of your users' sessions: Cookie or Session Inactivity. For more information, see [Managing User Sessions](#).
- **Embedded Domain White List:** You can define which domains can embed your dashboards into iFrames on their site. This is useful for controlling where your dashboards can be embedded. In the **Add Domain** field, enter each domain where your dashboards can be embedded and click **Add**. If you do not add any domains, then your dashboards can be embedded into any site. After adding a domain, your dashboards can be embedded only in those domains.

3. Click **Save** to update your system settings.

## License Utilization

Sisense provides your licensing information in the **Admin** page of the Sisense Web Application.

In the License Utilization section, you can see when your license expires, how many user licenses are in use for your account, and additional features that are supported in your account.

The License Utilization section includes the following information:

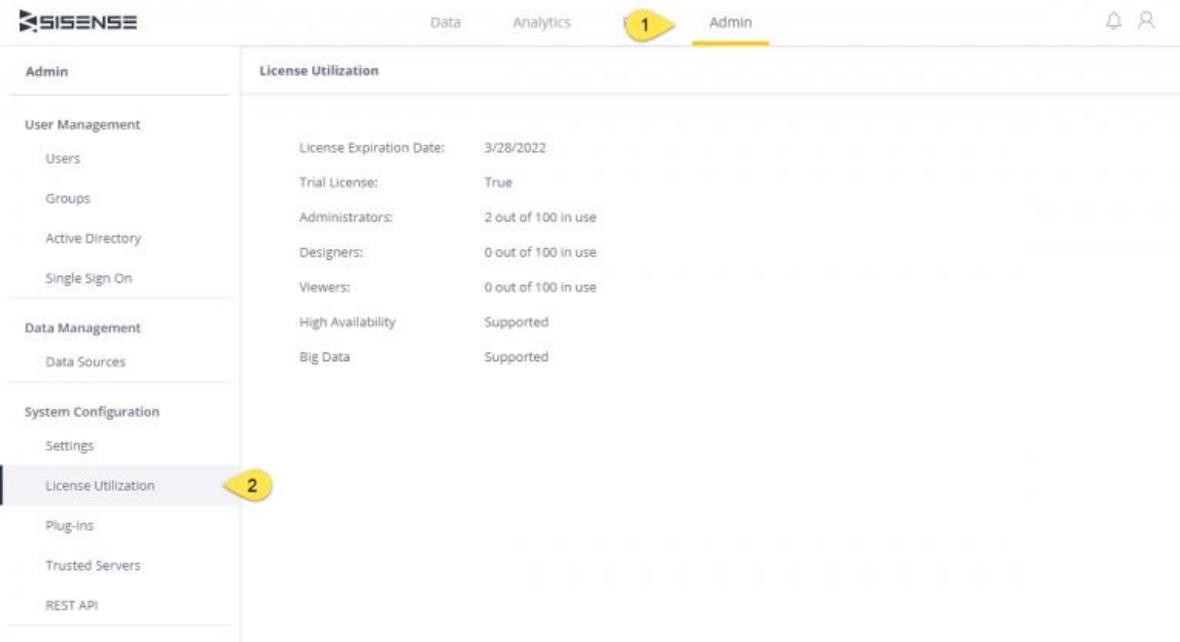
Details	Description
---------	-------------

Details	Description
License Expiration Date	The date when your Sisense license expires in the format of Month/Day/Year.
Trial License	Indicates if your license is for a trial or full version of Sisense. True indicates that you are using a trial license.
Administrators	The number of Administrator licenses in use out of the number of available licenses in your account. Administrator licenses includes Admin, Sys.Admin, and Data Admin.
Designers	The number of Designer licenses in use out of the number of available licenses in your account. Designer licenses include Designers and Data Designers.
Viewers	The number of Viewer licenses in use out of the number of available licenses in your account.
High Availability	Indicates if your account supports high availability.
Big Data	Indicates if your account supports big data. Big data as defined by Sisense is typically considered to be data sets above ~1 billion rows.

**To access the License Utilization section:**

**Note:** This section is accessible to System Administrators only.

- In the Sisense Web Application, click **Admin** and select License Utilization in the left menu.



## Configuration Manager

This topic provides an overview of the Sisense Configuration Manager, which you can use to configure and administer a variety of functionality and services in Sisense.

For example, you can use the Configuration Manager to perform the following tasks:

- Configure SSL in your Sisense environment
- Configure your application database for high availability
- Define where log files are stored and how detailed your logs are

The Configuration Manager can be accessed from your browser at <http://localhost:3030>.

Below you can find descriptions of the fields that appear in the Configuration Manager and how they relate to tasks you can perform in Sisense. When relevant, links are provided to topics that provide a more in depth explanation of the field and the tasks you typically use the field for.

# Web Server

The following fields are used to configure SSL for your Sisense environment. You should enable SSL if you have configured your web environment with SSL security (HTTPS) and certificates. This will enable HTTPS in the URLs of shared dashboards. For a complete explanation of these fields and their purpose, see [Setting up SSL](#).

- **Enable SSL:** Toggle to enable/disable SSL.
- **Upload Certificate:** Select the type of SSL certificate you are using.
- **SSL Certificate:** Enter your SSL certificate.
- **SSL Key:** Enter your SSL key.

**Port:** The port to be used when connecting to Sisense. The default value is 8081. For traffic over a secure connection, typically, this port is 443.

After setting a new port, restart the following services in Windows Services:

- Sisense.Identity
- Sisense.Galaxy
- Sisense.ECMServer
- Sisense.Gateway

**Proxy URL:** This field is used for implementing Base URLs. See [Configuring Base URLs](#) for more information.

**Domain Binding:** This field is used to allow external domains to access Sisense, for example, if you have implemented iFrames. You can add multiple domains separated by a comma and click **Save** to save your changes.

**Internal IP/Host:** The internal IP of Sisense. This is used when [setting Up SSL](#) and high availability.

**Build Node Server:** When implementing a multi-node configuration, for example for high availability, you must define which node is your build node in the Configuration Manager. The address of the build node should be the server IP address and not the server name.

# Application Database

The following fields are used to communicate with your Sisense application database. Typically, this is only necessary when implementing a high availability environment. For more information, see [High Availability](#).

**Connection String:** The connection string contains all the necessary attributes for accessing the Sisense application database.

**Encryption Key:** The value of this field is the encryption key used by Sisense to encrypt your application database users. This value is used when you need to create users for your application database. See [Hosting the Application Database](#) for more information

# Logs

Sisense stores logs locally on your Sisense server. These logs are typically used by Sisense Support when troubleshooting any issues with your Sisense environment. You can define the location of these logs and the level of logging.

**Path:** The current path when Sisense stores logs. You can enter a new path for your logs on your Sisense server.

**Level:** Select the level of logging.

# General

The following fields are used to communicate with your Sisense Configuration database and the Sisense Message Broker. Typically, this is only necessary when implementing a high availability environment. For more information, see [High Availability](#).

**ZooKeeper Connection String:** The connection string to the configuration database. Click **Test** to verify the connection. See [Replicating Zookeeper](#) for more information.

**RabbitMQ Connection String:** The connection string to the Sisense message broker. Click **Test** to verify the connection. See [Replicating the Message Broker](#) for more information.

**Manual Discovery Connections:** This is used internally by Sisense Support and does not need to be configured.

## Sisense Migration

Many times, you may want to set up Sisense or upgrade Sisense in a development environment before migrating it to your production environment. In a development environment, you can test and verify your dashboard behavior and data accuracy before publishing dashboards to a production environment.

The migration process for Sisense involves migrating your [ElastiCubes](#), [dashboards](#), and [Sisense system configuration](#).

Additionally, if you are using plugins, you will need to replace the Plugins folder with your current plugins located at `C:\Program Files\Sisense\app\plugins`.

[Backing up your installation](#) is recommended before a migration process, but also on a regular basis for securing your data and Sisense configurations.

**Note:** The original server and the target servers that Sisense is being migrated must be running the same version of Sisense.

## Next Steps

- [Migrating ElastiCubes](#)
- [Migrating Dashboards](#)
- [Migrating Sisense System Configurations](#)

## Migrating ElastiCubes

There are two ways to migrate your data to a new server, by exporting and importing .ecube files or exporting and importing .ecdata files.

An .ecube file includes the model metadata: model schema and data connection settings.

An .ecdata file includes both the model metadata and the data itself.

Both .ecdata and .ecube files can be used to make backups and transfer data between ElastiCube Servers, depending on whether you want to move the metadata only, or the metadata and data.. After building your ElastiCube from an .ecube file, your ElastiCube with the metadata and the raw data combined is stored in an .ecdata file.

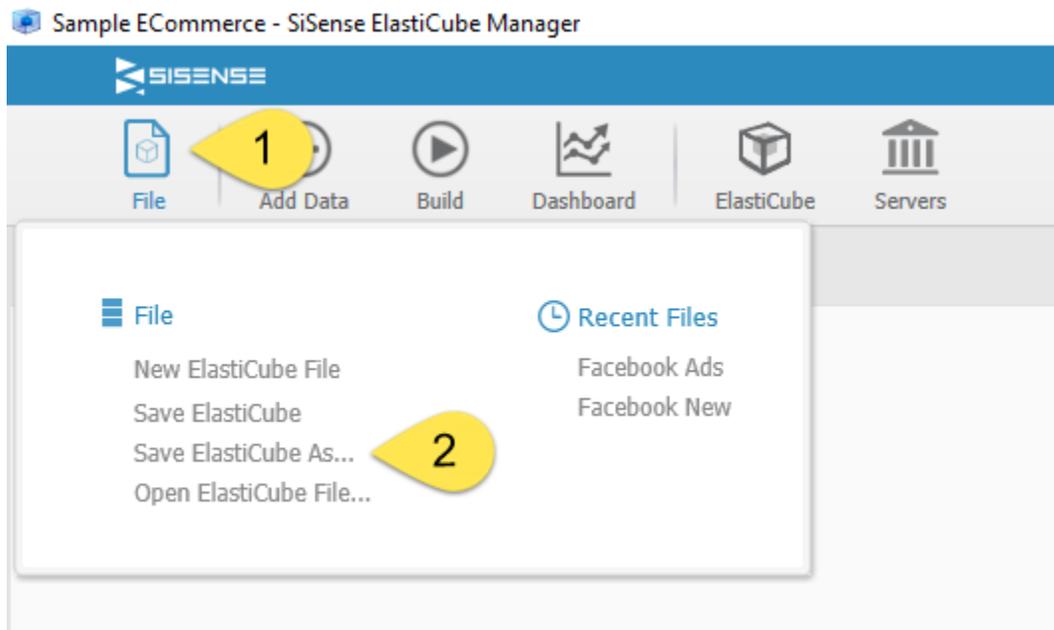
You can migrate ElastiCubes with .ecube files or .ecdata files. As .ecdata files can be quite large depending on how much data you imported, it may be easier to migrate your .ecube files. The steps below describe how to migrate your Sisense ElastiCubes via .ecube files. If you prefer to migrate the .ecdata files, see [Migrating .ecdata Files](#) for more information.

# Exporting .ecube Files

You can export .ecube files from the Sisense Desktop Manager. Once these files have been exported, you can back them up with your [source control](#) and import them into your target environment.

## To export .ecube files:

1. In the Sisense Desktop Manager in the source environment, open the ElastiCube you want to migrate.
2. Select **File > Save ElastiCube As...** and download the .ecube file to your machine.



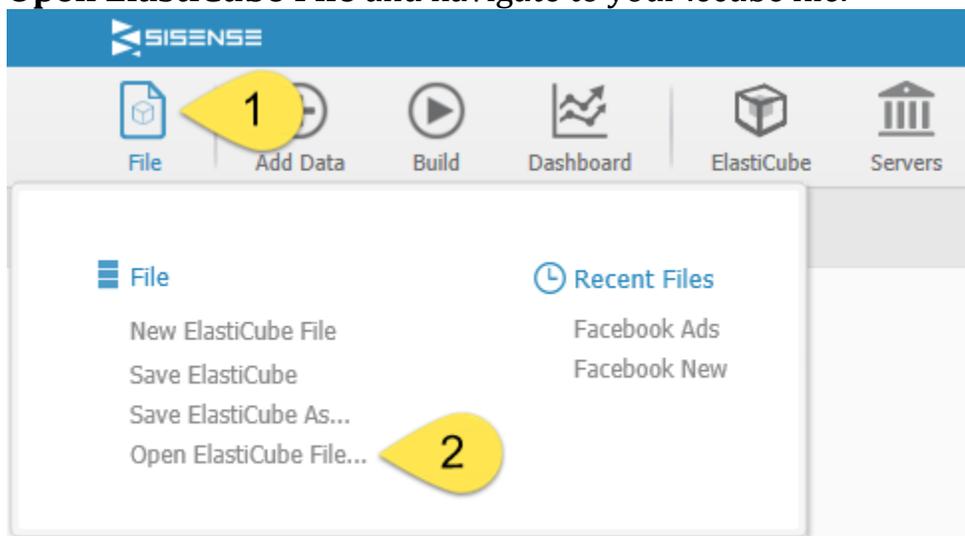
3. Save the .ecube file to your target environment.

# Importing .ecube Files

After exporting .ecube files, you can import them into Sisense through the Sisense Desktop Manager. Once the .ecube file have been imported, you must build the ElastiCube to use it with your dashboards.

## To import .ecube files:

- In the Sisense Desktop Manager in the target environment, select **File > Open ElastiCube File** and navigate to your .ecube file.



After selecting the relevant .ecube file, the schema is displayed, you can build the ElastiCube or if you are importing it in a new environment, see [Deploying ElastiCubes](#).

# Deploying ElastiCubes

As .ecube files only contain the metadata for your ElastiCube in your source environment, you need to [build](#) the ElastiCube in the target environment to bring in the data. If your target environment has different data sources from your source environment, for example, the source environment uses development databases whereas the production environment uses production databases, then the .ecube files need to be updated with the new data sources.

There are two ways to edit the ElastiCube with the new data sources:

- Sisense PSM Shell to automate the process
- Manually open the .ecube file and edit it

Both methods are outlined below.

# Automating ElastiCube Deployment

The following instructions describe a common scenario for deploying ElastiCubes in a new environment where .ecube files have been imported into a production environment and the data source connections need to be updated.

## Editing .ecube Files

Using the [Sisense Shell](#), or Sisense PSM, it is easy to edit the database connection attributes. This set of APIs allows you to edit Elasticube parameters via a command line interface. The most commonly used command in a dev-to-prod scenario redirects .ecube build queries to a production database. Below is an example where “myecube.ecube” is redirected from a development database called “tracking” to a production database called “trackingX”.

```
psm ecube edit connection database
ecube="c:\users\myuser\documents\mycube.ecube"
server="10.0.0.1" database="tracking"
newserver="localhost" newdatabase="trackingX"
newusername="mynewuser" newpassword="newpassword"
```

The server and database parameters contain the details of the previous database and the newserver and newdatabase parameters should contain the location and name of your updated data source on the production server.

## Building ElastiCubes with the PSM

The build can also be initiated on the production server through the Sisense interface or through the PSM. The following example code builds the Elasticube on the production server with the PSM:

- **Full Build, Local Server:** psm ecube build name="<elasticube name>" mode=restart serverAddress=localhost
- **Accumulative Build, Remote Server:** psm ecube build name="<elasticube name>" mode=full serverAddress=192.168.1.134

## Manually Deploying Elasticubes

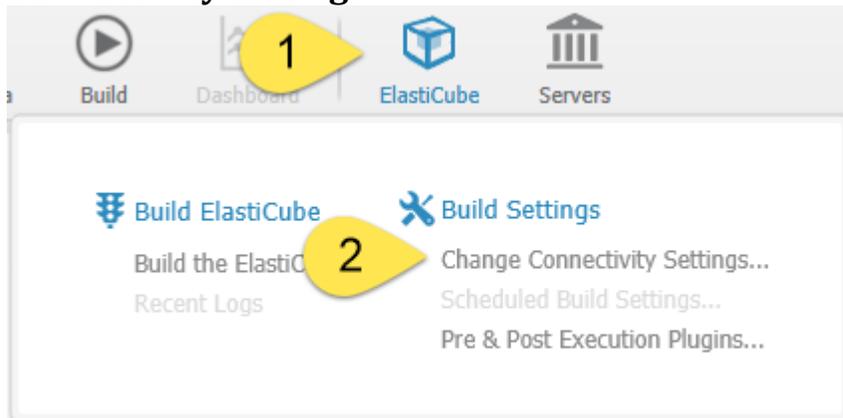
If you want to manually update your ElastiCube's connection settings to reflect your new data sources, on your target server, navigate to the imported .ecube file and double-click to open it. This will open the Elasticube Manager on the server. From here, you can manually edit the connection settings for each data source. This is only necessary if you need to update the connection settings. If your data sources have not changed between environments, you can build the ElastiCube without having to edit it.

### Editing the .ecube File

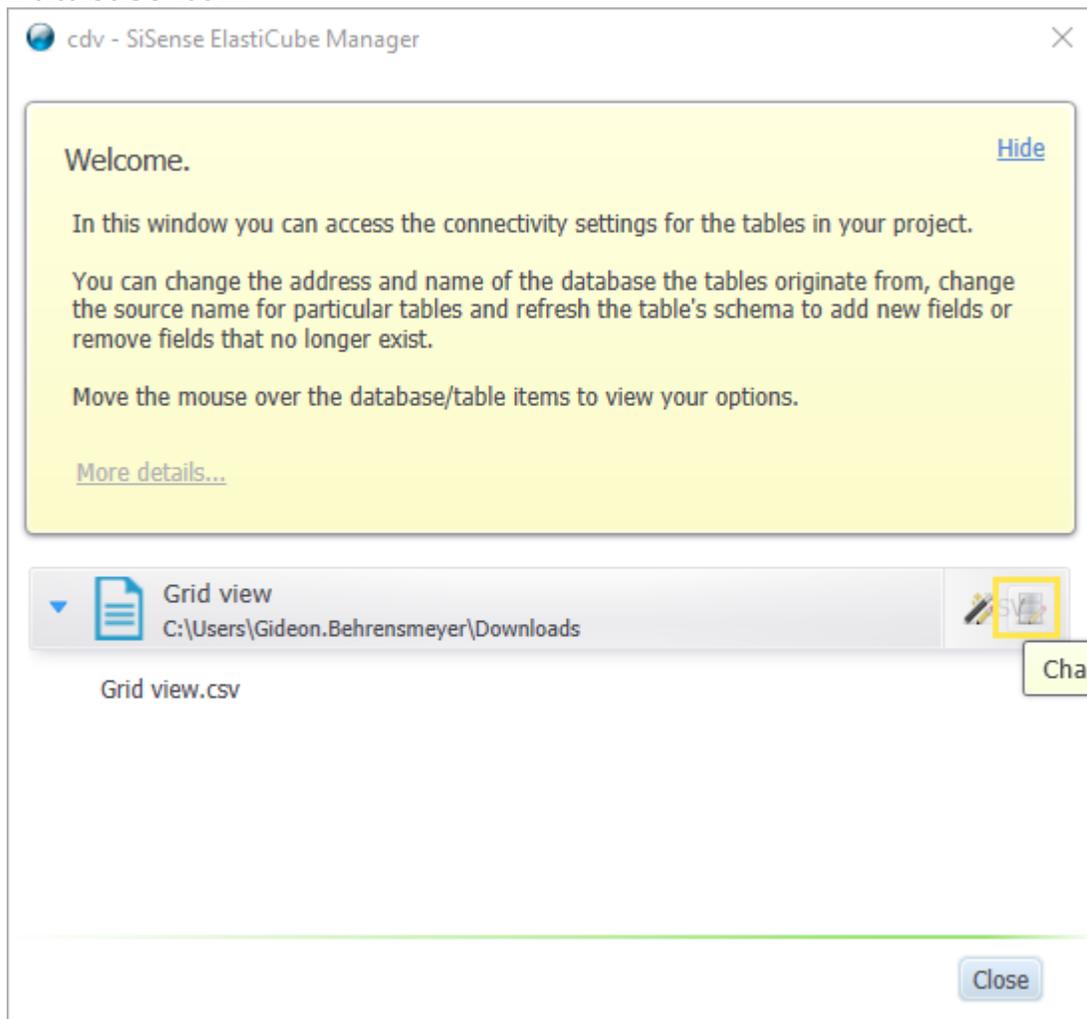
If you are pointing to different data sources in your production environment (e.g., a production vs development database) you can edit the connection settings in the Elasticube Manager.

### To edit .ecube file:

1. In the Desktop ElastiCube Manager, select **Elasticube** and then **Change Connectivity Settings**.



2. For each data source that needs to be changed, select the **Change Source Database** icon.



3. Choose your Database server location, then click Connect to Server. Make sure the correct database is selected under **Select Database** and then

select **OK** and **Close**.

Connect to SQL Server

Database server location:  
LAP-NY-RACHELB\SQLSERVER

Windows Authentication  
 Use The Following User Name & Password:

User Name: Sisense  
Password: ●●●●●●

Encrypt Connection  
 Trust Server Certificate

Connect to Server

---

Select Database:  
AdventureWorks2016CTP3

OK Cancel

## Next Steps

- [Migrating Dashboards](#)

## Migrating Dashboards in Sisense

There are several ways to migrate dashboards to other Sisense environments:

- Migrating Dashboards in Sisense (Recommended - described below)
- [Migrating Dashboards via REST API](#)
- [Exporting and importing .dash files](#)

The easiest way is to migrate the dashboards through Sisense as described below, however, if you want to implement version control for your dashboards, you may prefer to export .dash files that represent your dashboards in Sisense. Administrators and Designers with access to a remote Sisense server can copy dashboards from one Sisense server to another from the Sisense Web Application. When you copy a dashboard or a folder to a remote server, all the dashboards, including those in the selected folders are duplicated into the main Navigation Pane in the target server. Folders are not duplicated on the target server.

Before copying a dashboard to a target server, the Administrator of the target server must provide access to target server. The procedure below explains how to provide access to another server and then how to copy a dashboard from the source server to the target server through the Sisense Web Application.

**Note:** When migrating a dashboard to another server, you must be logged in to your local server with the IP address as the URL. You cannot copy dashboards from localhost.

### **To copy your dashboard to another Sisense server:**

1. On the target server, the Administrator needs to provide access for the source server to the target server. On the target Sisense server, select **Admin > Trusted Server List**.

The screenshot shows the Sisense Admin interface. The top navigation bar includes 'Data', 'Analytics', and 'Admin' (highlighted with a yellow callout '1'). The left sidebar menu has 'Admin' at the top, followed by 'User Management' (Users, Groups, Active Directory, Single Sign On), 'Data Management' (Data Sources), 'System Configuration' (Settings, License Utilization, Plug-ins), 'Trusted Servers' (highlighted with a yellow callout '2'), and 'REST API'. The main content area is titled 'Trusted Servers List' and contains a table with two columns: 'Server IP / Host' and 'Name'. The table lists two servers: 'http://192.168.228.1' (Sample Server 1) and 'http://192.168.228.2' (Sample Server 2). A '+ Add Server' button is located in the top right corner of the table area.

Server IP / Host	Name
http://192.168.228.1	Sample Server 1
http://192.168.228.2	Sample Server 2

The Trusted Server List page is displayed.

2. In the Trusted Server List page, click **Add Server**. The Create a New Server Access dialog box is displayed.

### Add Server

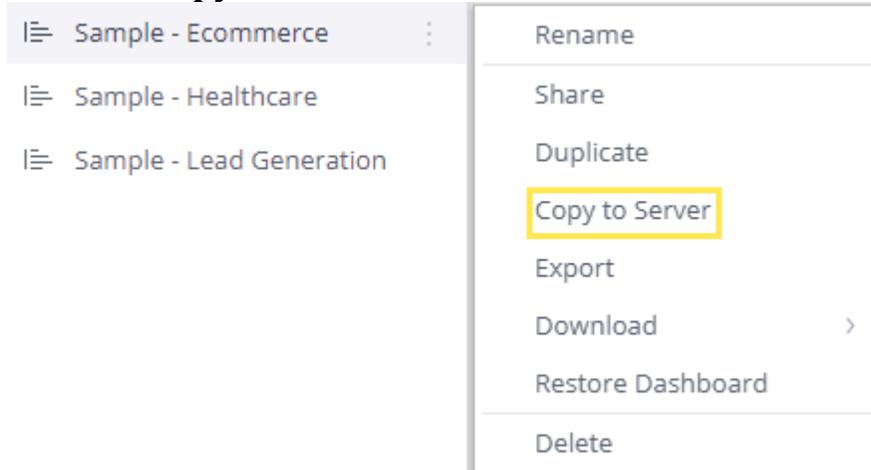
Server	<input type="text" value="Type URL or Hostname (Including Port)"/>
Name	<input type="text" value="Optional"/>

Cancel

**Note:** Some of the dialog boxes may be considered as pop-ups by your ad blockers. Make sure that Sisense is whitelisted or that your ad blocker is disabled when performing this procedure.

3. In the **Server** field, enter the IP address or hostname of the source server that contains the original dashboard.
4. In the **Name** field, enter a name for the source server. This is an optional identifier to help you keep track of your servers.

5. Click **OK**. The server is added to the list and now has copy access to the target server.
6. On the source server, from the Navigation Pane, click the dashboard's menu and click **Copy to Server**.



OR

To copy multiple dashboards, in the Dashboards list, click the Select Dashboards  button to display checkboxes next to the dashboard and folders, which allows you to select multiple dashboard to be exported. Then, select the relevant dashboards and folders to be exported and click . The Copy Dashboards to Server dialog box is displayed.

### Copy Dashboards to Server

Server

Republish dashboards after copying

Cancel

7. In the Copy Dashboards to Server dialog box, enter the following information:

**Server:** The URL or Hostname of the remote server. Sisense supports SSL connections so dashboards can be transferred securely when SSL is activated.

**Republish dashboards after copying:** Select to automatically republish the dashboard on the target server after the selected dashboards are copied. Republishing a dashboard refreshes the dashboard definitions (widget types, titles, scripts etc.) for each of the users who share the dashboard.

8. Click **OK**.
9. If the dashboards already exist on the recipient server, Sisense prompts you to select how you want to handle each dashboard that already exists. You can select one of the following options for each dashboard:
  - **Skip:** No changes are made to the dashboard on the recipient server.
  - **Overwrite:** The new dashboard writes the existing dashboard on the recipient server.
  - **Duplicate:** A new dashboard is created on the recipient server with a number added to the end of the dashboard's title, for example, "Sample Dashboard (1)".

After you have defined how the dashboards are to be handled, the selected dashboards are copied to the remote server and displayed in the Dashboard's list.

Once the copy is completed, you may need to update the data source for your dashboard or set data security rules for your ElastiCube (see the links below for detailed documentation):

- [Change the Data Source to the Production ElastiCube](#)
- [Set Security on the Production ElastiCube](#)
- [Share or Republish the dashboard to users](#)

## Next Steps

- [Migrating the Sisense Configuration](#)

# Migrating Sisense System Configurations

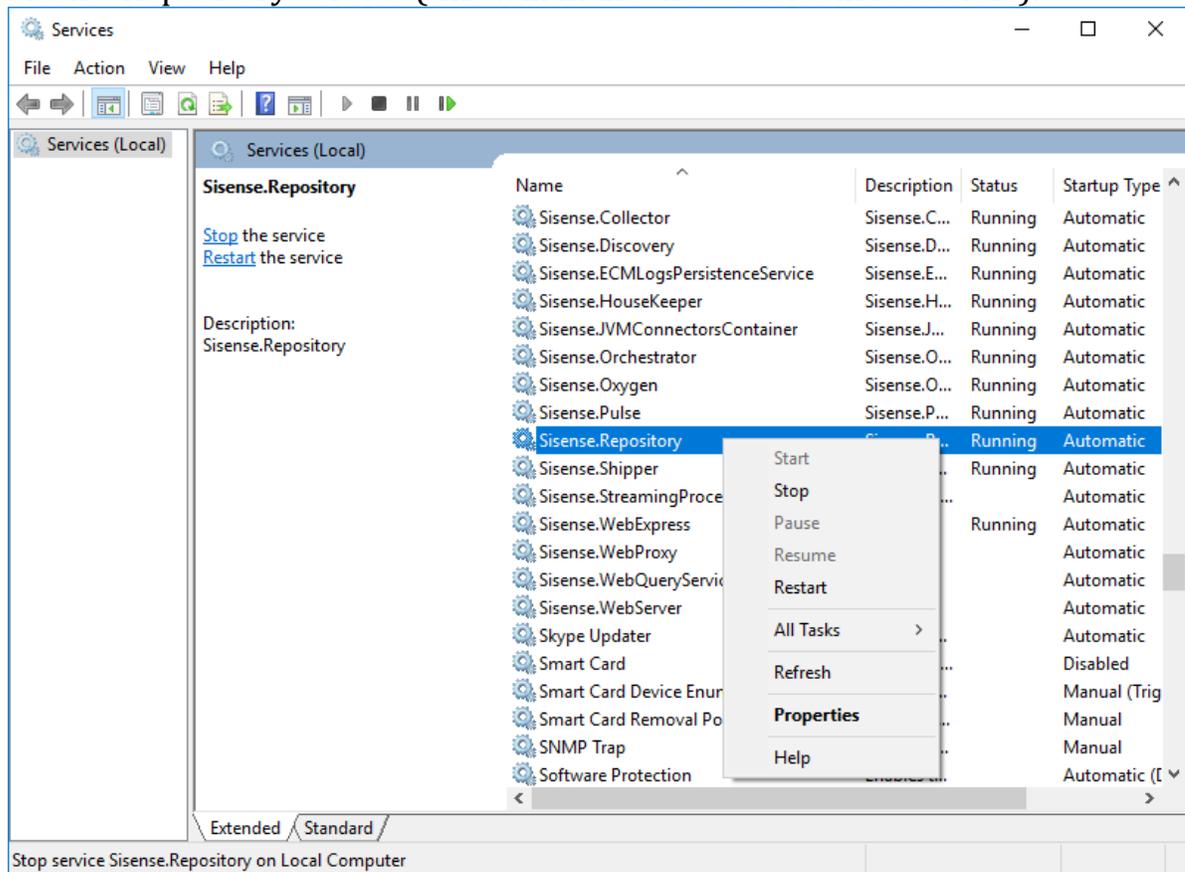
To migrate your Sisense Web Application configuration, you must copy several directories from the current environment and migrate them to the new environment.

In addition, your Sisense application database has an identifier associated with it when you install Sisense. This identifier must be deleted when migrating Sisense as each database should have its own unique identifier. A new identifier will be generated in the environment the next time Sisense is opened. You can access the database through RoboMongo and remove the identifier as described below.

## **To migrate the Sisense Web Application:**

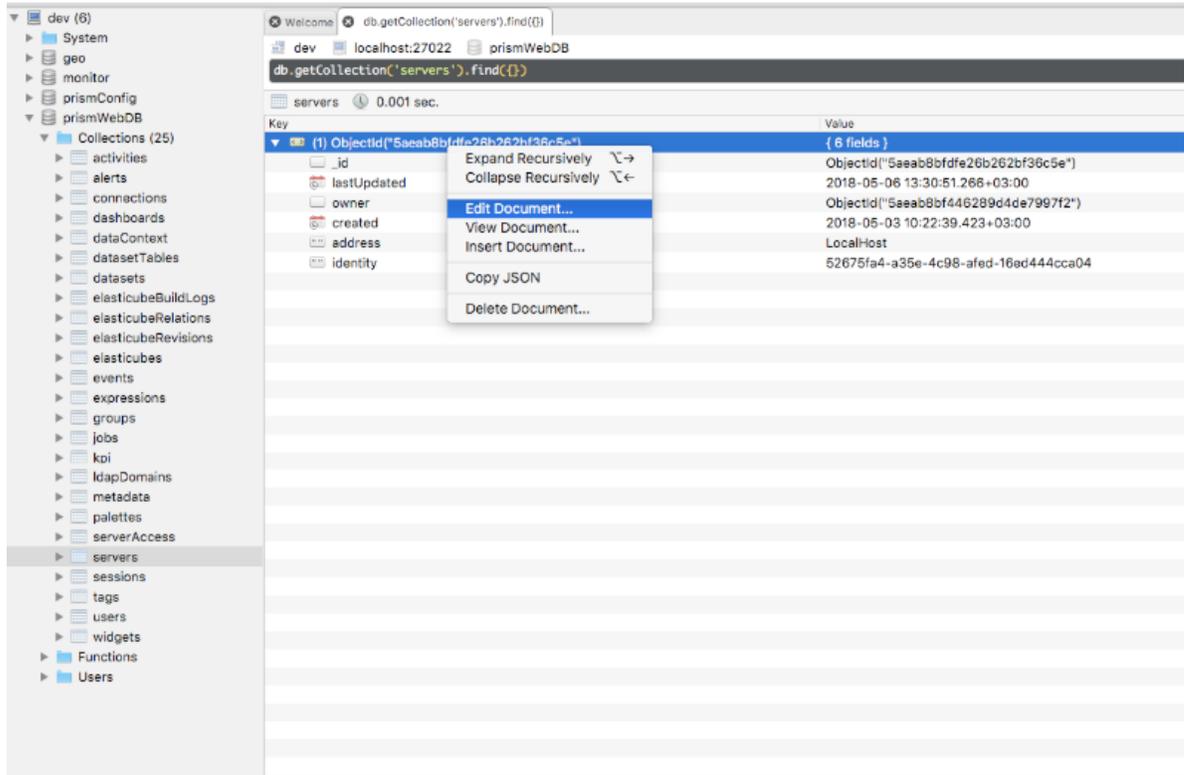
1. Back up the old environment, open your machine's Control Panel, and go to **System and Security > Administrative Tools > Services**, and stop the

SisenseRepository service (this will make the website inaccessible).

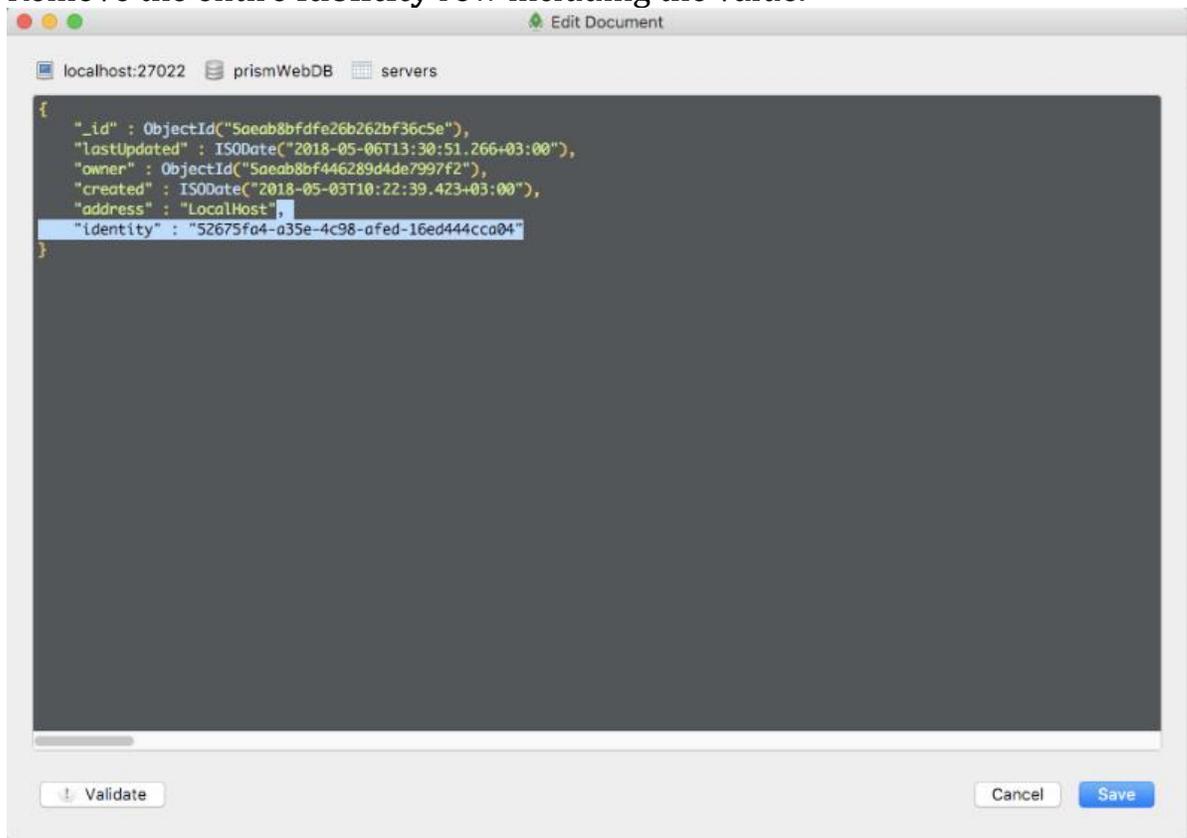


2. Create a copy of the entire Repository directory located at:  
%ProgramData%\Sisense\PrismWeb\Repository
3. Replace the existing **Repository** directory in the new environment with the one you backed up.
4. Create a copy of the entire **Discovery** directory located at:  
%ProgramData%\Sisense\Infra
5. Replace the existing **Discovery** directory in the new environment with the one you backed up.
6. In Windows Services, start the **SisenseRepository** service and **Sisense.Discovery** service.
7. Open Robomongo and connect to your application database with a WriteUser. For more information, see [Accessing the Application Database](#).

8. Under **PrismWebDB**, go to the **servers** collection and edit the file with address **LocalHost**.



9. Remove the entire **identity** row including the value.



10. Open Windows Services and restart the following services:

- Sisense.Configuration
- Sisense.Galaxy
- Sisense.Gateway
- Sisense.Identity
- Sisense.Plugins
- Sisense.ECMServer
- Sisense.ECMLogs
- Sisense.Jobs
- Sisense.StorageManager

11. Open the IIS Manager and restart the website SisenseWeb.

## Embedded Analytics

Sisense's embedded analytics solution is an end-to-end, single stack BI solution embedded within your application that lets your customers easily prepare, analyze and visualize complex data. Sisense's embedded analytic solution offers

built-in features and exposes a variety of functionality through several interfaces, such as the Sisense JavaScript and REST APIs for customizing your integration.

You can accomplish the following through the Sisense embedded analytics solution:

- Seamlessly integrate Sisense with your current interface and branding
- Manage data, reporting and visualization
- Integrate SSO and Active Directory
- Govern user and data security
- Support high availability

This documentation is intended for two audiences, product managers and developers. For product managers, those who want to enhance their products and applications with Sisense embedded analytics, this section describes the features and functionality Sisense offers as part of an embedded analytics solution. For developers, those who embed Sisense in the application, this section describes how to embed Sisense analytics through the relevant APIs.

# Overview

When you embed Sisense analytics, you can provide your users with Sisense's analytical capabilities such as data management, reporting and visualization, built into your business applications and solutions.

Each OEM has their own requirements when embedding Sisense's analytic solution. This section covers the multiple ways in which Sisense's OEM features can be embedded to provide BI capabilities in your applications. Sisense exposes functionality through several interfaces. For example, Sisense's REST API allows you to white label Sisense and Sisense's command shell allows you to couple Sisense with your on-premise installation to modify connection settings. You determine which features you want to embed according to the type of OEM services you provide to your customers.

The following table lists the various OEM-related tasks you can perform with Sisense and the description with links to the relevant documentation for completing each tasks:

Task	Description
Rebranding Sisense	Sisense enables you to white label the Sisense Web Application to display your company's logos and colors and mask the URL to display your company's domain.  For documentation regarding how you can customize and rebrand your Sisense solution, see the following topics: <a href="#">White Labeling Sisense</a> <a href="#">Rebranding Sisense Automated Emails</a> <a href="#">Displaying Custom URLs</a>
Embedding Sisense	Sisense also allows you to embed widgets and dashboards directly into your site or application through the use of iFrames

<b>Task</b>	<b>Description</b>
	<p>or SisenseJS, Sisense's custom JavaScript library for embedding widgets.</p> <p>For documentation regarding how you can embed Sisense, see the following topics:</p> <p><a href="#">Embedding Dashboards and Widgets via iFrames</a></p> <p><a href="#">Embedding Dashboards and Widgets via JavaScript</a></p>

# Considerations

Before embedding Sisense analytics, you should keep in mind several considerations that may affect your design and deployment:

- **Users:** Who will be using your application? This determines what interfaces need to be white labeled and which API calls you should leverage.
- **Performance:** As with any deployment, you must provide sufficient CPU and memory on the client and the server for your anticipated user load.
- **Availability:** What is your desired up-time and how many resources can you dedicate to maintaining system availability.

## OEM Architecture

Sisense has many customers who use Sisense as an OEM Business Intelligence platform. This means that a customer is using Sisense to provide analytics to their own external customers. Customers' customers, can be referred to as "Tenants".

# Rebranding and Embedding Sisense

OEM deployments comprise of two main features: rebranding and embedding. In the white-labeled use-case, a Sisense server is completely rebranded. Customers replace the Sisense logo and name, change the color palette used by the application, customize the emails sent by the system, and more. The tenants access the Sisense server directly, and experience Sisense analytics as customized by the customer. They are not necessarily aware that the analytics server they are accessing is actually a Sisense server.

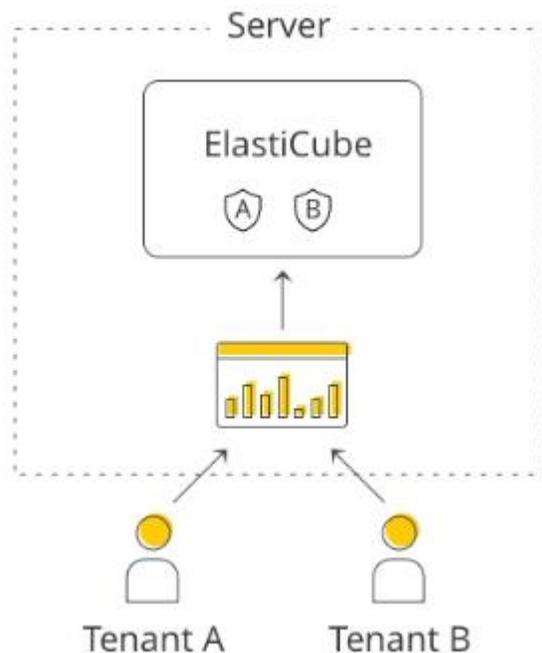
In the embedded use-case, Sisense customers have their own web application. They embed Sisense within it. Sisense supports embedding the full Sisense application, including the data modeling, analytics and administration areas, or embedding specific dashboard and widgets. using iFrames. Widgets can also be embedded within web pages using the SisenseJS infrastructure.

Both of these use-cases, the white-labeled server and the embedded offering, rely on strict data governance to ensure tenants get access only to their own assets and data.

This type of governance can be achieved using a few different OEM architectures. The selected architecture depends on the customer requirements and needs, and their own preferences. Sisense provides the flexibility to support OEM deployments in the way that customers want to work.

In this topic, a few different ways to support OEM use-cases are outlined, and describe the benefits and disadvantages of each.

## Option 1: Shared ElastiCubes with Row-Based Data Security



**Advantages:** Low hardware costs, high resource utilization, simple asset change management

**Disadvantages:** Tenant resource usage may affect other tenants

**Typically Best for:** Tenants with identical data models and dashboard requirements

The first type of architecture for OEM deployments utilizes shared Sisense servers for multiple tenants, and shared ElastiCubes and dashboards.

Segregation between tenants is achieved by using row-based data security within ElastiCubes. All of the customers' data resides in a shared ElastiCube, but each of the tenants gets access only to their own data.

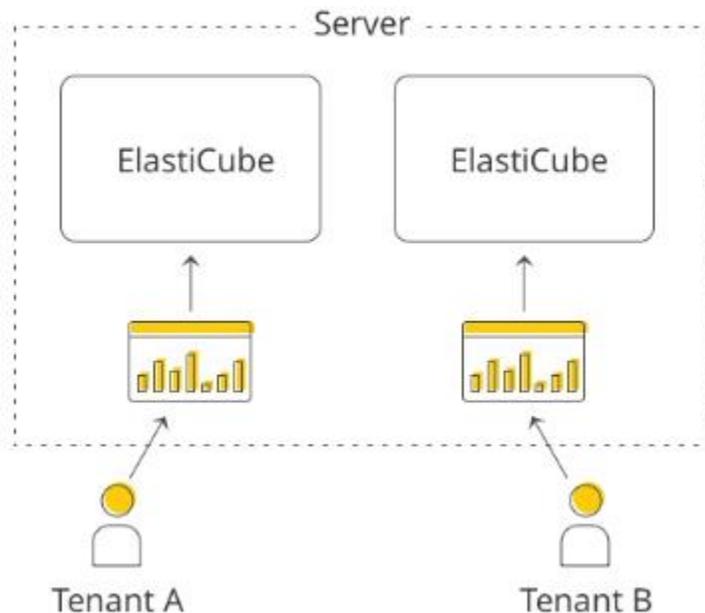
This methodology makes it easy to maintain system assets, and handle changes that occur during the lifecycle of the asset. Modifying the ElastiCube schema is done only once in the shared ElastiCube for all users. Changing the shared dashboards is also done once for all users.

This methodology is well suited for customers providing a service based on identical data sources and reports. For example, a customer analyzing shopping statistics for tenants on a shared shopping portal will take data for all of the customers from the shopping portal analytics data using each tenant's credentials. But all of the data has the same exact structure, so the generated dashboards are common analytics of this data.

This methodology provides good utilization of the server resources, and ensures that asset maintenance remains easy.

Note that as the server is shared by multiple tenants, they are also sharing resources. High resource usage by one of the tenants, for example, generating multiple reports or heavy builds that require a lot of CPU, may affect other tenants.

## Option 2: Dedicated Cube per Tenant



**Advantages:** Low hardware costs for small number of tenants, high resource utilization

**Disadvantages:** High hardware costs for large number of tenants, more complicated asset change management, tenant resource usage may affect other tenants

**Typically Best for:** Tenants who require customized data models

The second architecture for OEM deployments utilizes shared Sisense servers for multiple tenants, together with providing a **dedicated ElastiCube and dashboards for each tenant**. In this deployment, multiple tenants use the same server. Typically, the OEM has default ElastiCubes and dashboards, and creates a dedicated copy of them for each of the tenants. The ElastiCubes and dashboards can be identical copies for each of the tenants, or customized per tenant. The OEM uses access control for ElastiCubes and dashboards to ensure each of the tenants only has access to their own data. Typically a user group is created for each of the tenants. All of the tenants users are assigned to the same group. The relevant ElastiCubes and Dashboards are shared with the tenants group. In this

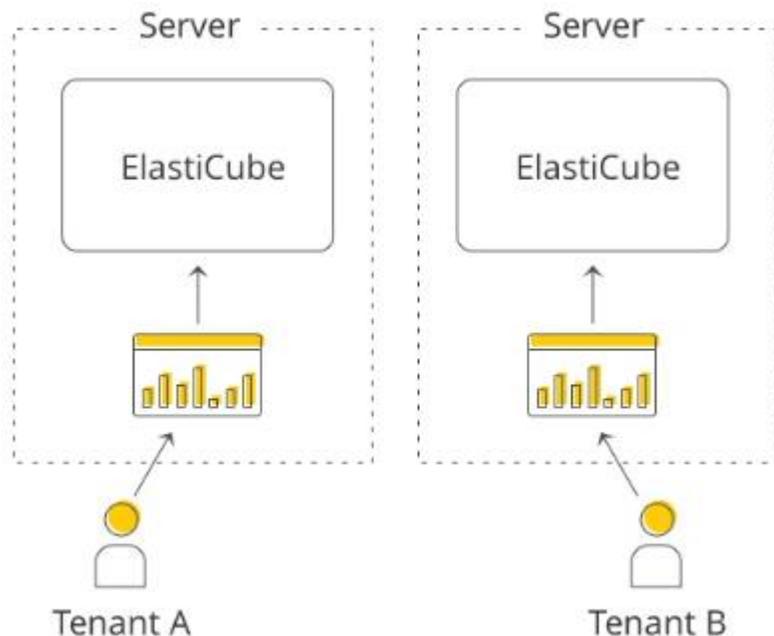
way, the asset access control layer ensures that users of each tenant only gets access to their own data.

This methodology is well suited for customers whose data sources are not identical. It allows for customizing the data import process so that the data preparation and ETL processes can handle the specific customers data structure, and transform it to the desired target structure. This methodology is also well suited for tenants who have different reporting needs. While the tenant is initially provisioned with default ElastiCubes and default dashboards, customizations can be made per tenant, without affecting other tenants. It is easier to accommodate the specific needs and requirements of each tenant. In this use case, the OEM needs to manage multiple copies of ElastiCubes and dashboards. Making a change to an ElastiCube schema or to dashboards needs to be replicated across all of the tenants. When there are many tenants, the cost of making changes to assets is high.

This methodology provides better utilization of the server, and can reduce the cost of ownership as multiple tenants are sharing the same server. But as with the previous option, shared servers means that multiple tenants are also sharing the server resources. The behavior of one tenant may affect other tenants using the server.

With this solution, you need to consider how the system scales to support your future needs, to support additional tenants. There are limitations to the number of ElastiCubes that can be deployed on a single machine. While initially, you enjoy shared server resources, as you add more tenants, you may have to provision additional servers, increasing the hardware costs of this solution.

## Option 3: Dedicated Sisense Server per Tenant



**Advantages:** Highest level of security, dedicated resources per tenant

**Disadvantages:** Low resource utilization, high hardware costs, complicated asset change management

**Typically Best for:** Tenants with strict security regulations, such as financial or healthcare institutes and tenants that need a high level of schema and dashboard customizations

The third architecture for OEM deployment is to provide a dedicated Sisense server for each tenant. Typically an OEM customer will have a server image including default ElastiCubes and dashboards. Each of the customers receives their own instance of the server. The ElastiCubes and dashboards can be identical between the servers, or customized for each of the tenants. The data for each tenant is completely separate as each server has its own assets, including configuration, users, ElastiCubes and dashboards.

This option offers a high level of customization per tenant. Each of the tenants has their own ElastiCubes and dashboards, and they can be modified to accommodate the tenants specific needs, without affecting any other customers. This option also provides a very high level of governance and security for each of the tenants, as the OEM can be assured that no tenant can access assets of another tenant. Additionally, the OEM is assured that the system behavior of one tenant will not affect other tenants. If there is a heavy system load caused by multiple queries or reports generated by one tenant, it will not have any effect on the other tenants.

This deployment architecture is suitable for cases where there may be legal restrictions that mandate that data of different tenants cannot be located on the same physical machine.

However, for multiple reasons, this option can be expensive and not very cost effective to maintain. A tenants server may be underutilized if it's not used throughout the day. Or if most of the users reside in the same timezone, the server may be idle for long periods of the time. Additionally, a Sisense server is a high performance server and capable of supporting a large number of concurrent users. If a tenant does not have many users, the server may be underutilized even during work hours of the tenant, and could have possibly been used by additional users.

Another issue is the configuration maintenance of the ElastiCubes and dashboards. If an OEM needs to make a change to the ElastiCubes and dashboards that are provided to the tenants, then all the changes will have to be made on each one of the servers. For example if the OEM wants to create a new dashboard, and share it with all of their tenants, they will have to copy the dashboard to each one of the tenants servers. This can require quite a lot of work.

In this deployment option there's a trade-off between complete segregation and dedicated resources, and the cost of ownership and maintenance.

## Summary

Sisense supports a range of flexible architectures for OEM deployment. The type of architecture suitable for a specific customer depends on the use-case, the needs of the customer, the resources that can be dedicated to the deployment (both allocated people, and allocated hardware), and preferences.

### Embedding Dashboards and Widgets

This article is about getting started with embedded Sisense dashboards. If you're new to Sisense, check out our [embedded analytics solutions](#).

You can embed Sisense dashboards and widgets in non-Sisense environments, such as in your own websites or applications by concatenating parameters to the URL of your dashboard or widget.

In addition, you can customize what aspects of the dashboard and widget are displayed in your environment.

For example, if you want to remove the Sisense header from the dashboard, you can concatenate the “?embed=true&h=false” parameter in the URL of the dashboard to hide the header.

`mysite.com/app/main#/dashboards/5541dc7a80a4e2181e00011a?embed=true&h=false`

In this example, the “embed=true” embeds the dashboard into your environment. When you embed a dashboard or widget, viewers can only view the dashboard or widget and do not have any editing privileges. The “h=false” parameter hides the Sisense header allowing you to whitelabel the dashboard.

This article provides a full list of the parameters you can concatenate, their functionality, and examples.

# Embedding Dashboards

To embed a dashboard, add “?embed=true” to the URL of the dashboard, and refresh the page. If your dashboard is within a folder, then first delete “?folder=” and everything that follows, before adding “?embed=true”.

This will open the dashboard without the surrounding Sisense environment.

For example:

Dashboard URL:

`mysite.com/app/main#/dashboards/5541dc7a80a4e2181e00011a`

Embedded URL:

`mysite.com/app/main#/dashboards/5541dc7a80a4e2181e00011a?embed=true`

Dashboard URL for dashboard within folder:

`mysite.com/app/main#/dashboards/550952417404b2981a000029?folder=55095a27404b2981a00003b`

Embedded URL for dashboard within folder:

`http://localhost:8081/app/main#/dashboards/550952417404b2981a000029?folder=55095a27404b2981a00003b`

You can use this URL to embed the dashboard in an iframe, for example:

```
<iframe id="ifm" name="ifm" width="100%" height="100%"
frameborder="0"
src="http://mysite.com/app/main#/dashboards/536f2b70d093e26c280000d5?embed=true" scrolling="auto"></iframe>
```

## Dashboard Embedding Configuration Options

You can control which aspects of the Sisense environment are available when embedding by concatenating the following parameters to the embedded URL:

### **Show/Hide the left Navigation Panel**

Parameter: "l" (left) or "i" to hide it completely

Default: false

Example: ?embed=true&l=true

### **Show/Hide the right filter panel**

Parameter: "r" (right)

Default: true

Example: ?embed=true&r=false

### **Show/Hide the toolbar**

Parameter: "t" (toolbar)

Default: false

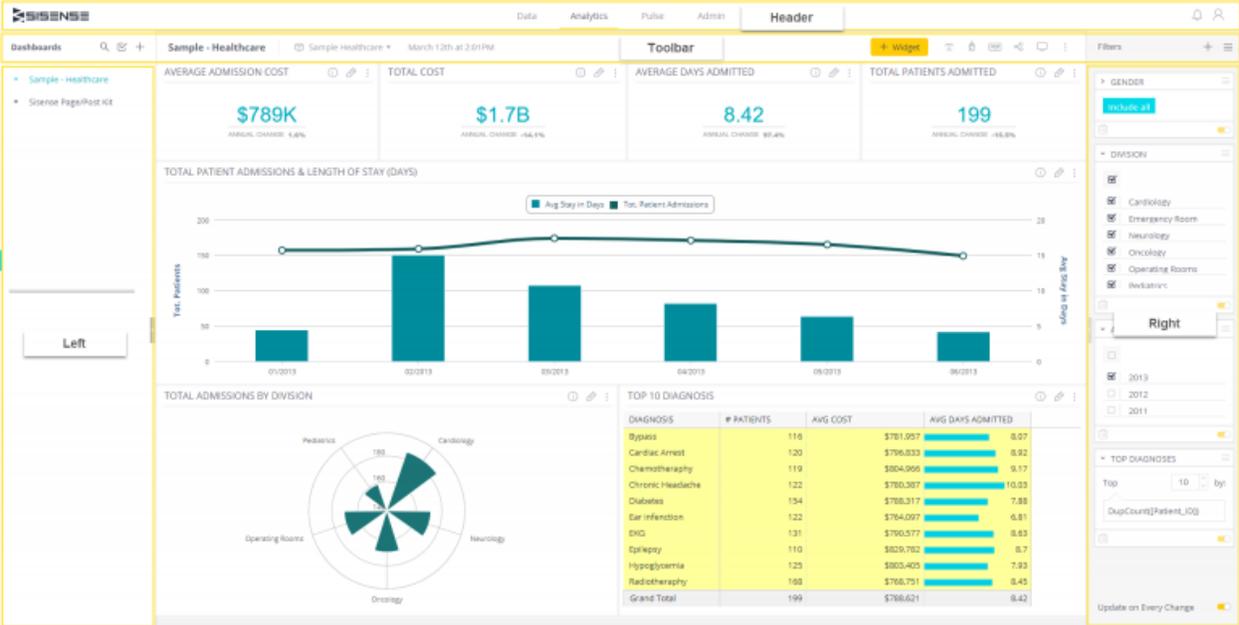
Example: ?embed=true&t=true

### **Show/Hide the environment header**

Parameter: "h" (header)

Default: false

Example: ?embed=true&h=true



# Embedding Widgets

To embed a single widget, add “?embed=true” to the URL of the widget when it edit mode.

This will open the widget without the surrounding Sisense environment.

For example:

## **Dashboard URL:**

<http://mysite.com/app/main#/dashboards/536f3a54d093e26c28000114/widgets/536f3a54d093e26c2800011b>

## **Embedded URL:**

<http://mysite.com/app/main#/dashboards/536f3a54d093e26c28000114/widgets/536f3a54d093e26c2800011b?embed=true>

You can use this URL to embed the widget in and iframe, for example:

```
<iframe id="ifm" name="ifm" width="100%" height="100%" frameborder="0"
src="http://mysite.com/app/main#/dashboards/536f3a54d093e26c28000114
/widgets/536f3a54d093e26c2800011b?embed=true"
scrolling="auto"></iframe>
```

## Widget Embedding Configuration Options

You can control which aspects of the Sisense environment are available when embedding by concatenating the following parameters to the embedded URL:

### **Show/Hide the left data panel**

Parameter: "l" (left)

Default: false

Example: ?embed=true&l=true

### **Show/Hide the right design panel**

Parameter: "r" (right)

Default: false

Example: ?embed=true&r=false

### **Show/Hide the toolbar**

Parameter: "t" (toolbar)

Default: false

Example: ?embed=true&t=true

### **Show/Hide the environment header**

Parameter: "h" (header)

Default: false

Example: ?embed=true&h=true

## Authentication in Embedded Dashboards and Widgets

All Sisense dashboards and widgets require authentication. When embedding dashboards and widgets outside of the Sisense environment, usually the use case is to not enforce the Sisense environment authentication, but rather use SSO (Single Sign On) with existing corporate authentication. To read more on configuring Single Sign On in Sisense, [click here](#).

### Displaying Custom URLs

You can display your company's URL in the address of your Sisense dashboards by defining an alias URL from the Sisense **Admin** page.

This is useful when you are providing OEM services and you want to mask Sisense's URLs and branding.

#### **To implement an alias URL in Sisense:**

1. Click **Admin** in the top menu, and then **Settings** in the left menu.
2. In the **Alias** field, enter the web server's alias or IP address. This is especially important when you have configured your IIS with a domain name (alias), but when you share a dashboard, the IP address and/or the incorrect port is sent instead. Reports will be sent from the alias entered in this field.
3. Click **Save**.

### Rebranding Sisense Automated Emails

Sisense automated emails are sent to users according to predefined scenarios.

When a predefined scenario such as a build alert is triggered, an automated email is sent to the relevant Sisense users.

You can customize these emails to match your company's branding.

This page describes the Sisense automated emails and how you can replace them with your own so each time a predefined scenario is triggered, Sisense sends your customized emails to your Sisense users.

# Sisense Automated Emails

You can rebrand the following automatic emails:

- **Build Alert:** Email sent each time a build alert is triggered.
- **Dashboard Errors Reports:** Email sent when a dashboard report fails.
- **Dashboard Errors Reports Bulk Fail:** Email sent when a large amount of email reports fail.
- **Dashboard Reports:** Email sent to the recipient of a shared dashboard when you share a dashboard.
- **Empty Dashboard Report:** Email sent to an administrator when an empty dashboard report is sent.
- **KPI Alert:** Email sent each time a KPI alert is triggered.
- **New User Invitation:** Email sent to a new user invited to Sisense.
- **Password Recovery:** Email sent to a user who requested a new password.
- **Share ElastiCube:** Email sent to the recipient of a shared ElastiCube.
- **Share With:** Email sent to the recipient of a shared dashboard.
- **Transfer Ownership:** Email sent to a user who received ownership of a dashboard from another owner.
- **User Created:** Email sent to a user when a user is created.
- **User Created AD:** Email sent when a user is created in Sisense after being added from Active Directory.

For each automated email, Sisense provides an email template in the format of EJS and LESS files. These files are located within template folders inside the Template Parent folder located at:

## In Windows:

```
...\Program Files\Sisense\app\galaxy-service\src\features\emails\templates
```

## In Linux:

```
/opt/sisense/storage/emails
```

## LESS Files

All templates contain the style.less file that refers to a stylesheet that defines the branding used in Sisense's automated emails. The content of every style.less is as follows:

```
@import '../styles.less';
```

The stylesheet referred to by all the email templates is called styles.less located at:

### **In Windows:**

```
...\Program Files\Sisense\app\galaxy-  
service\src\features\emails\templates
```

### **In Linux:**

```
/opt/sisense/storage/emails
```

By replacing the content of the styles.less file with your own CSS, you can redefine the branding used in all of the email templates to your custom branding.

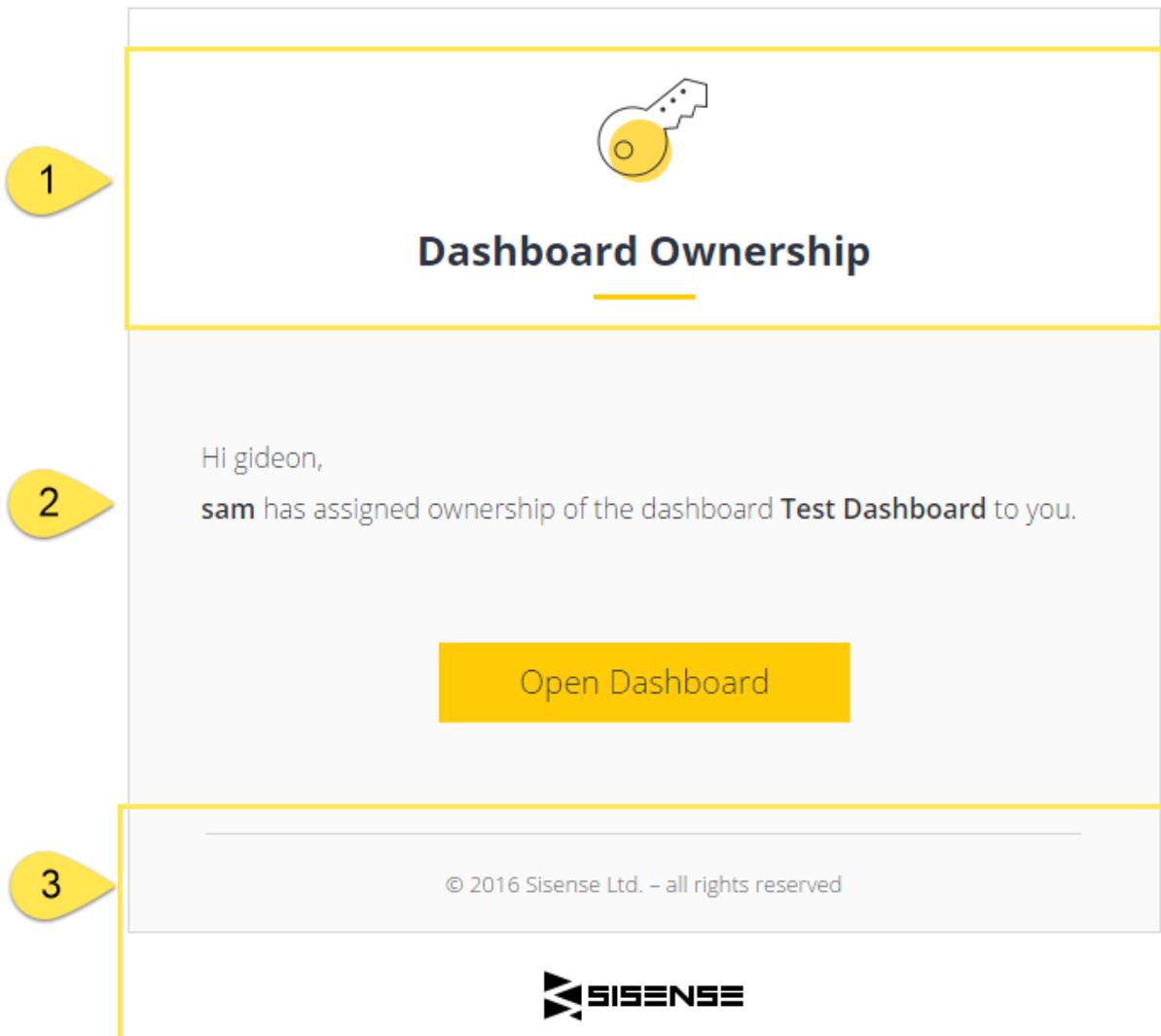
## EJS Files

Each template folder contains an EJS file, which contains HTML and JavaScript that describe the content of the email.

You can use any text/HTML editor to modify the HTML.ejs files according to your needs.

The following is an example of an HTML.ejs file.

```
<% include ../header %>
  <tr>
    <td>
      <div class="contentText">
        <p class="userText" style="margin: 0;"><%=
i18nContent.hi %> <%= newOwnerUserName %>,</p>
        <p class="generalInfo" style="margin:
0;"><%- i18nContent.generalInfo %></p>
      </div>
    </td>
  </tr>
<% include ../footer %>
```



Each EJS file has three sections, a header, body, and footer; each of which describes various areas in the email your users receive.

Section 1 is the Header section defined in the EJS file as `<% include ../header %>`.

This is a reference to the header.ejs file located at:

```
..\Program Files\Sisense\app\galaxy-service\src\features\emails\templates
```

The header.ejs file defines the title and image used in the header section of each email template. You can modify the header by replacing the header.ejs file with or replace the `<% include ../header %>` tag from each template with your own HTML and JavaScript.

Section 2 is the Body section, which contains the content displayed in the email your users receive. There are two types of content displayed in automated emails, tokens and strings. Tokens refer to Sisense variables such as the Sisense username or dashboard owner's name. For example, the `<%= newOwnerUserName %>` token displays the new owner of a dashboard's Sisense username in the automated email.

```
<tr>
  <td>
    <div class="contentText">
      <p class="userText" style="margin: 0;"><%=
i18nContent.hi %> <%= newOwnerUserName %>,</p>
      <p class="generalInfo" style="margin: 0;"><%=
i18nContent.generalInfo %></p>
    </div>
  </td>
</tr>
```

When you replace the HTML.ejs file with your own files, you can still use the Sisense tokens. For more information about tokens you can add to your automated emails, see [Applying Sisense Tokens](#).

In addition, each template contains an object such as `i18nContent.hi` that displays strings predefined by Sisense. These strings cannot be modified, however, you can remove and replace the object with your own content.

The final section, Section 3 contains the footer. Like the header, the footer, `<% include ../footer %>`, is a reference to a footer.ejs file located at:

**In Windows:**

```
...\Program Files\Sisense\app\galaxy-
service\src\features\emails\templates
```

**In Linux:**

```
/opt/sisense/storage/emails
```

The footer.ejs file defines the copyright text and image used in the footer section of each email template. You can modify the footer by replacing the footer.ejs file with or replace the `<% include ../footer %>` tag from each template with your own HTML and JavaScript.

Images that are displayed in the Header and Footer are stored in the following location:

**In Windows:**

```
...\Program Files\Sisense\app\galaxy-  
service\src\features\emails\templates\images
```

**In Linux:**

```
/opt/sisense/storage/emails/images
```

You can replace these with your own images by maintaining the same file names used, however, it is recommended to replace the content of the Header and Footer sections altogether with your own content.

# Replacing Sisense Automated Emails

If you prefer to implement your own email templates for Sisense's predefined scenarios, you can replace Sisense's emails with your own through the Sisense REST API.

**Note:** Rebranding Sisense emails must be enabled for your license. Contact your Account Manager for more information.

For each automated email, Sisense provides an email template in the format of EJS and LESS files. These files are located within template folders inside the Template Parent folder located at:

## **In Windows:**

```
...\Program Files\Sisense\app\galaxy-service\src\features\emails\templates
```

## **In Linux:**

```
/opt/sisense/storage/emails
```

Some template folders may contain an html-org.ejs file. These files are legacy templates and are maintained for backwards compatibility. Newer developments should only modify or replace the html.ejs files.

**Note:** While it is possible to modify the content of your automated emails through the language.js file located in the templates folder, the recommended method is modify the email templates as described in the procedure below.

After you have prepared your new automated emails, through the /branding endpoint of the REST API, you can configure Sisense to send the customized automated emails when the relevant event is triggered.

## **To define the location of your branded emails:**

1. Copy the contents of the directory:

### **In Windows:**

```
...\Program Files\Sisense\app\galaxy-service\src\features\emails\templates
```

**In Linux:**

/opt/sisense/storage/emails

2. Paste all the folders in the following directory:

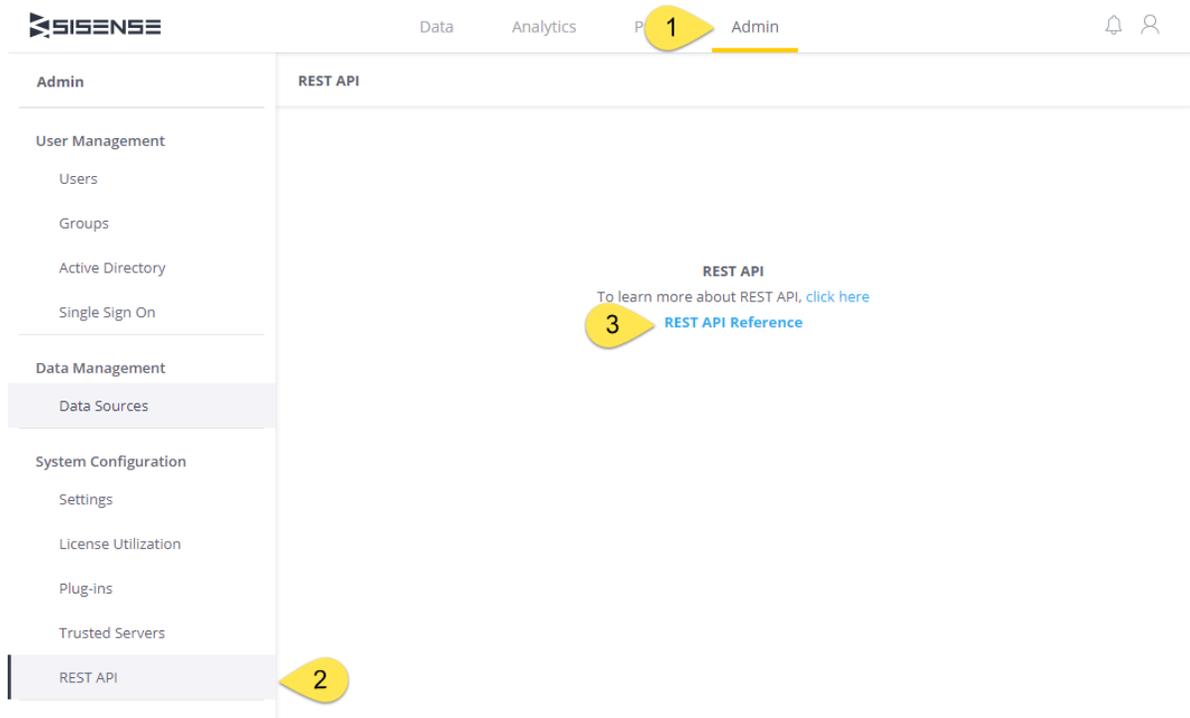
... \Program Files\Sisense\app\galaxy-service\src\features\emails\

**In Linux:**

/opt/sisense/storage/emails

**Note:** You will modify the templates in this folder and configure Sisense to send them instead of the original **Sisense automated emails**.

3. For each template you want to modify, change the contents of html.ejs files and styles.less as described in **Sisense Automated Emails**.
4. Access the Sisense REST API.



5. In version .9 of the REST API, select the POST /branding.

**POST** /branding Adds new branding to your Sisense dashboards

**Implementation Notes**  
Branding includes customized texts, logos and emails.

**Response Class (Status 200)**

Model | Model Schema

```

"emails": {
  "senderEmail": "string",
  "senderName": "string",
  "templates_directory": "string",
  "passwordRecoverySubject": "string",
  "newUserInviteSubject": "string",
  "createdUserSubject": "string",
  "shareWithNewUserSubject": "string",
  "shareWithExistingUserSubject": "string",
  "transferOwnership": "string"
},
  
```

6. In the email object define the following information:

**Note:** If you have already rebranded your Sisense Web Application, copy the current settings through the GET /branding endpoint and paste them into the POST /branding endpoint modifying or adding the emails object. If you only modify the emails object, the remaining objects and keys are set to their default Sisense and will overwrite any existing settings.

Key	Description
senderEmail	The email address from which the automated email is to be sent. When you define the senderEmail value, you must verify your domain in Sisense's Mandrill email service. Click <a href="#">here</a> for more information.
senderName	The name to be displayed as the Sender in the automated email.
templates_directory	The directory location of your template folders. Sisense accepts /resources/branding/emails as a custom location for automated email templates. This directory is located on your Sisense

Key	Description
	server at ...Program Files\Sisense\app\galaxy- service\src\features\emails
passwordRecoverySubject	The subject line of the password recovery email.
newUserInviteSubject	The subject line of the new user email.
createdUserSubject	The subject line of the new user invite email.
shareWithNewUserSubject	The subject line of the shared dashboard with a new user email.
shareWithExistingUserSubject	The subject line of the shared dashboard email.
transferOwnership	The subject line of the dashboard transfer ownership email.

For example:

```
"emails": {
  "senderEmail": "sys-admin@company.com",
  "senderName": "sys-admin",
  "templates_directory": "/resources/branding/emails",
  "passwordRecoverySubject": "Recover Password",
  "newUserInviteSubject": "new User Account Activation",
  "shareWithNewUserSubject": "A Dashboard has been
shared with you:",
```

```
"shareWithExistingUserSubject": "A Dashboard has been  
shared with you:"
```

```
},
```

7. Click **Run**. The next time an automated email is sent, your templates are sent instead of Sisense's predefined emails.

# Applying Sisense Tokens

Most Sisense automated emails include tokens that you can apply to your automated emails. When the email is sent, Sisense populates the token with the relevant value. For example, the `<%= user %>` token represents the Sisense's username.

Each template has tokens defined by Sisense that you can apply listed below. Tokens from one automated email cannot be applied to another. Applying tokens to other automated emails, for example adding `<%= user %>` to the KPI alert email results

**build\_alert:** Sent when a build alert is triggered.

No Template Tokens Available

**Dashboard\_errors\_report:** Sent when an Dashboard email report fails.

Template Tokens Available

```
<%= errors[i].error %>
```

```
<%= errors[i].widgetFailed %>
```

```
<%= errors[i].widgetCount %>
```

**Dashboard\_errors\_report\_bulk\_fail:** Sent when a Dashboard email report sent to a large amount of users fails.

Template Tokens Available

```
<%= errors.groups[i] %>
```

**Dashboard\_report:** An automated email that contains a dashboard report sent to users.

Template Tokens Available

```
<%= url %>
```

```
<%= images[i] %>
```

**Empty\_dashboard\_report:** An automated email that contains an empty dashboard report sent to users.

No Template Tokens Available

**Kpi\_alert:** Sent to shared users when a KPI alert is triggered.

Template Tokens Available

<%= measure %>

<%= value %>

<%= message %>

**New\_user\_invitation:** Sent to a new user that you have invited to join.

Template Tokens Available

<%= owner %>

\* The folder for this template contains a file called HTML-ORG.ejs. This file is for older versions of Sisense and is maintained for backward compatibility.

**Password\_recovery:** Sent when a user requests their password from the Forgot Password page.

No Template Tokens Available

\* The folder for this template contains a file called HTML-ORG.ejs. This file is for older versions of Sisense and is maintained for backward compatibility.

**Share\_with:** Sent to a new recipient of a dashboard when the dashboard is shared with them.

No Template Tokens Available

\* The folder for this template contains a file called HTML-ORG.ejs. This file is for older versions of Sisense and is maintained for backward compatibility.

**transfer\_ownership:** Sent to the new owner of a dashboard when ownership is transferred.

Template Tokens Available

<%= newOwnerUserName %>

<%= oldOwnerUserName %>

<%= sharingDashboardName %>

**User\_created:** Sent to the recipients of new Sisense accounts.

No Template Tokens Available

**User\_created\_ad:** Sent when a user is created in Sisense after being added from Active Directory.

No Template Tokens Available

## White Labeling Sisense

Sisense enables you to rebrand (white label) the Sisense Web Application into your native application or site by using the Sisense REST API to replace existing icons and images with your own.

In addition, you can rebrand the automated emails Sisense sends to match your company's branding.

White labeling must be enabled in your license. To enable this feature, please contact your Sisense representative or open a request through our Help Center.

In addition, Sisense now offers a premium white labeled version of the Sisense Mobile app that can be customized according to your branding, for more information see [Rebranding Sisense Mobile](#).

# Rebranding Sisense

You can rebrand the Sisense Web Application and the automated emails sent by Sisense through Sisense's REST API.

There are two steps to rebranding the Sisense Web Application. The first step is to place your branded content into the Sisense branding directory where Sisense is installed. For a list of the files you can rebrand and their specific dimensions, see [White Labeling Configuration Options](#).

The second step is to access the Sisense REST API, where you can embed your content directly into Sisense and modify text displayed to customers.

## To rebrand the Sisense Web Application:

1. Create a subfolder with your brand name in this directory (on the machine where Sisense is installed):

### In Windows:

"...Program Files\Sisense\app\resources\".

### In Linux:

/opt/sisense/storage/rebranding

Put your branded files in this directory. The following files are supported:

- Desktop Logo (for example: Logo-Desktop.png) – This is the main icon that is visible at the top left of the Sisense Web Application.
- Tablet Logo (for example: Logo-Tablet.png)
- Mobile Logo (for example: Logo-Mobile.png)
- Favicon (for example: favicon.ico)

**Note:** Optimal dimensions for logo images: 92×26

- In the Sisense Web Application, click **Admin**, and then **Rest API**, followed by **REST API Reference**.
2. In the REST API Reference section, select version 0.9 in the top right corner.
  3. Click /branding and then click the POST operation /branding.
  4. To get the required JSON format for the request, click the sample model schema to place the sample text in the body parameter.

POST /branding Adds new branding to your Sisense dashboards

**Implementation Notes**  
Branding includes customized texts, logos and emails.

**Response Class (Status 200)**  
Model | Model Schema

```

{
  "logo": {
    "desktop": {
      "small": "string",
      "large": "string"
    },
    "tablet": {
      "small": "string",
      "large": "string"
    },
    "iphone": {

```

Response Content Type: application/json

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
body	<pre> {   "logo": {     "desktop": {       "small": "string",       "large": "string"     },     "tablet": { </pre>	branding object that needs to be added to the collection	body	Model   Model Schema

Parameter content type: application/json

```

{
  "logo": {
    "desktop": {
      "small": "string",
      "large": "string"
    },
    "tablet": {
      "small": "string",
      "large": "string"
    },
    "iphone": {

```

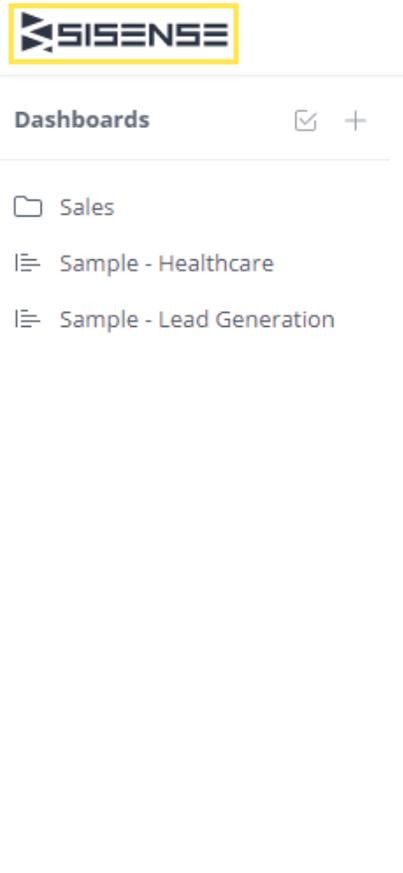
Click to set as parameter value

Replace the sample values with your required values. For an explanation of each parameter and how it modifies the Sisense Web Application, see [White Labeling Configuration Options](#).

5. Click **RUN**. Refresh the Sisense Web Application page to view your changes. **Note:** You can change just some of the parameters by leaving out those you do not want to update.

# White Labeling Configuration Options

The following table provides a list of the features you can rebrand within the Sisense Web Application and the JSON code that must be modified to rebrand it. Click the thumbnail image of the feature to see the default branding provided by Sisense and where your branding will appear.

Name	Description
<p data-bbox="186 688 386 730">Sisense Logo</p>  <p data-bbox="217 842 358 869">Dashboards</p> <p data-bbox="217 932 321 959">Sales</p> <p data-bbox="217 989 480 1016">Sample - Healthcare</p> <p data-bbox="217 1045 542 1073">Sample - Lead Generation</p>	<p data-bbox="781 688 1437 961">The Sisense logo can be replaced with your own. You need to supply two sizes of the logo, for each device (Desktop / Tablet / Mobile). These logos affect the following screens:</p> <ul data-bbox="820 976 1299 1113" style="list-style-type: none"><li>• Login Screen</li><li>• Account Activation Screen</li><li>• 'Forgot Password' Screen</li></ul> <p data-bbox="781 1136 1437 1234">Below is a list with the ideal logo dimensions (pixels):</p> <p data-bbox="781 1312 878 1354"><b>Small</b></p> <ul data-bbox="820 1360 1144 1497" style="list-style-type: none"><li>• Desktop: 120x48</li><li>• Tablet: 92x24</li><li>• Phone: 92x24</li></ul> <p data-bbox="781 1577 878 1619"><b>Large</b></p> <ul data-bbox="820 1625 1144 1761" style="list-style-type: none"><li>• Desktop: 204x60</li><li>• Tablet: 163x74</li><li>• Phone: 198x168</li></ul> <p data-bbox="781 1841 1076 1883"><b>Important Notes:</b></p>

1. Images must be in URI format (web address)
2. No spaces are allowed in the address of the images
3. Transparent .png format is preferred

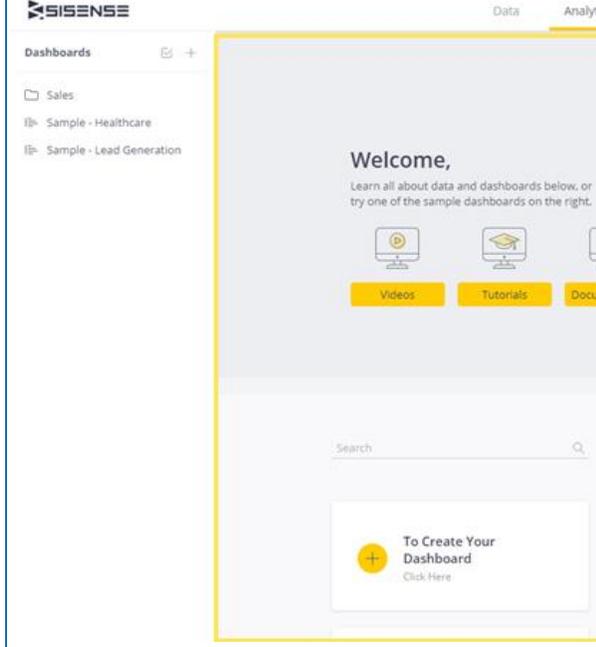
```
"logo":{  
  "desktop":{  
    "small":"/resources/<YourBrand>/logoDesktopSmall.png",  
    "large":"/resources/<YourBrand>/logoDesktopLarge.png"  
  },  
  "phone":{  
    "small":"/resources/<YourBrand>/logoPhoneSmall.png",  
    "large":"/resources/<YourBrand>/logoPhoneLarge.png"  
  },  
  "tablet":{  
    "small":"/resources/<YourBrand>/logoTabletSmall.png",  
    "large":"/resources/<YourBrand>/logoTabletLarge.png"  
  }  
},
```

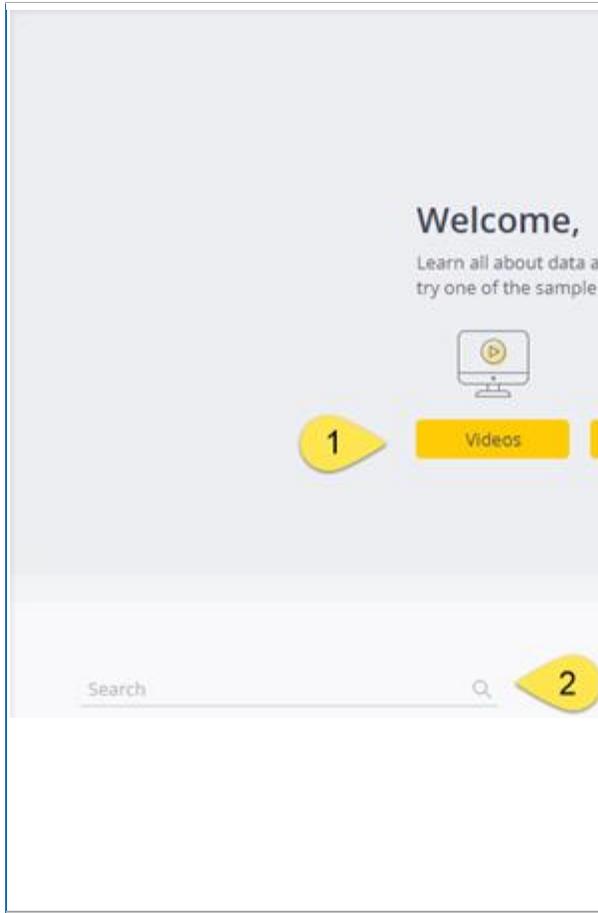
#### Page Title and Favicon



The page title that appears in the browser tab/page and the Sisense favicon.

**Note:**

	<ol style="list-style-type: none"> <li>1. Image must be in URI format (web address)</li> <li>2. .ico format is preferred</li> </ol> <p>You can modify these images by updating the values of the following JAQL parameters in the REST API:</p> <p>"favicon" :  "/resources/&lt;YourBrand&gt;/fav.ico"  "pageTitle": "Sisense",</p>
<p>Homepage</p>  <p>The screenshot shows the Sisense homepage. On the left, there is a 'Dashboards' sidebar with a 'Sales' folder containing 'Sample - Healthcare' and 'Sample - Lead Generation'. The main content area features a 'Welcome,' banner with the text 'Learn all about data and dashboards below, or try one of the sample dashboards on the right.' Below the banner are three buttons: 'Videos', 'Tutorials', and 'Documentation'. At the bottom, there is a search bar and a 'To Create Your Dashboard Click Here' button.</p>	<p>The home screen when your users log into Sisense. Rebranding the homepage does not affect the view of the Administrator. Administrators see the default homepage, while all other users see the rebranded page.</p> <p><b>Note:</b> The prefix http:// or https:// must be included.</p> <p>“homePage” : “http://www.mysite.com”,</p>
<p>Homepage Config</p>	<p>The Welcome banner of the homepage displayed when users log into Sisense. You define the following properties in the homePageConfig object:</p> <p><b>hideSearchInNavver:</b> Indicate if you want to display the Search bar below or hide it. By default, the Search bar is</p>

 <p>The screenshot shows a user interface with a 'Welcome,' message and a 'Videos' button. A yellow callout bubble with the number '1' points to the 'Videos' button. Below the 'Videos' button is a search bar with the word 'Search' and a magnifying glass icon. A second yellow callout bubble with the number '2' points to the search bar.</p>	<p>displayed.</p> <p><b>marketingPart:</b> The URL of the Marketing Banner iFrame. This iFrame is displayed on the top part of the <b>Analytics</b> and <b>Data</b> pages where the Welcome message and Sample Dashboards and ElastiCubes are displayed. You can replace this iFrame with your own content by defining the URL as the value of this key.</p> <p><b>tutorialsLink:</b> The Tutorials button. Enter a URL that opens your site.</p> <p><b>documentationLink:</b> The Documentation button. Enter a URL that opens your documentation site.</p>
<p>Login Page</p>	<p>The Login page title and subtitles.</p> <p>“loginTitle” : “WELCOME TO SISENSE”, “loginSubtitle” : “Sign in &amp; start exploring”</p>



Welcome to Sisense

Sign in & start exploring

gideon@sisense.com

.....

Remember me

Forgot your password? Click [here](#)

Copyright Text

The copyright text that appears in Sisense.  
"copyrightText": "Copyright © 2018  
Sisense inc. All rights reserved",



**Welcome to Sisense**  
Sign in & start exploring

gideon@sisense.co

.....

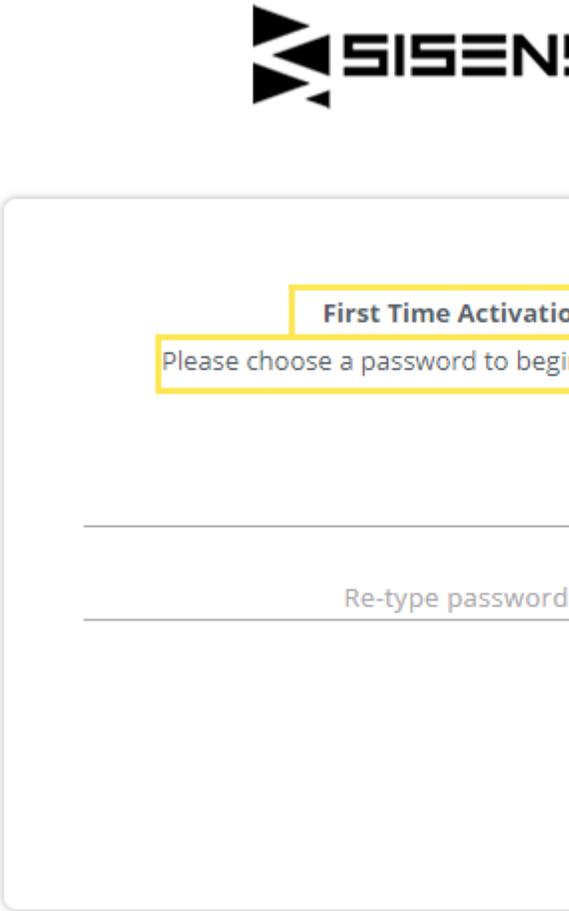
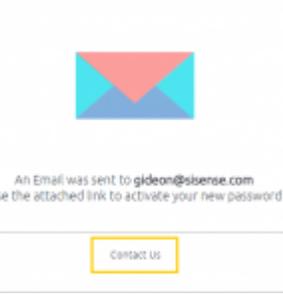
Remember me

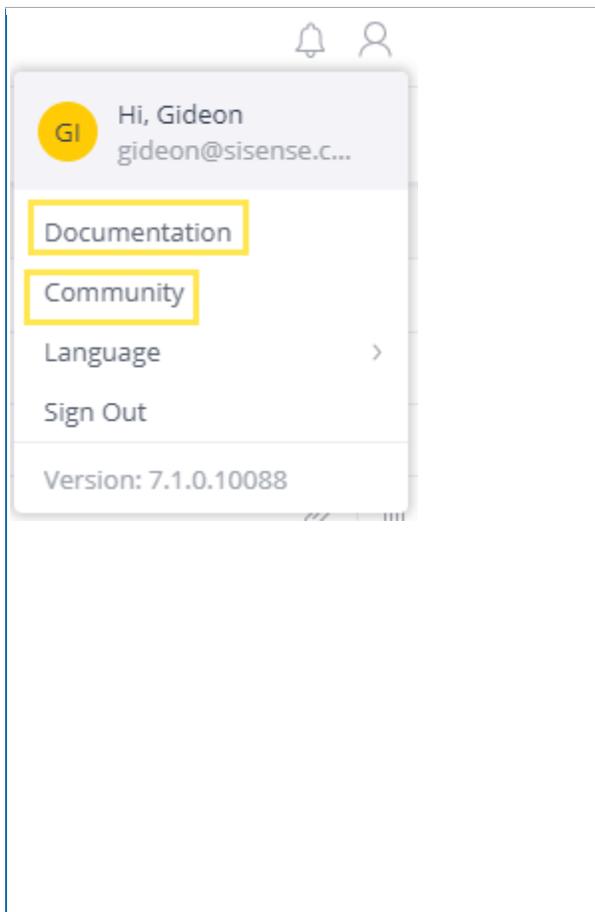
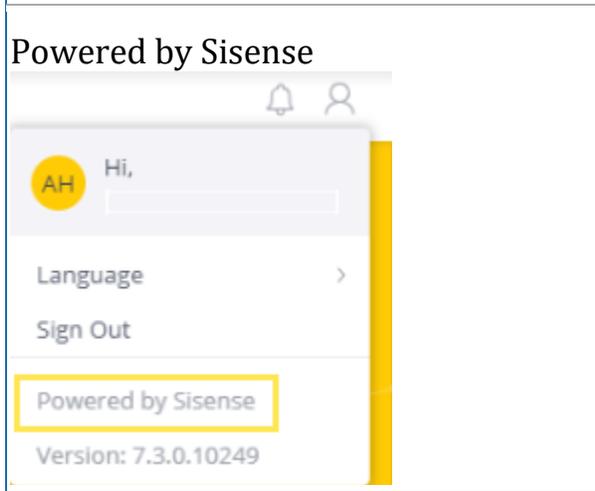
Forgot your password? Click [here](#)

Copyright © 2018 Sisense inc. All

Activation Titles

The titles on the activation page for new users signing in to Sisense.  
"activationTitle": "First Time Activation", "activationSubtitle": "Please choose a password to begin using Sisense",

	
<p><b>Contact Us</b></p> <hr/> 	<p>The Contact Us text is displayed when you request a password and enter your email on the Recover Password page. This text displays a message and provides a link to your Support services.</p> <p>“contactUsText” : “For further assistance, please contact our success management team!: &lt;a href='mailto:support@sisense.com'&gt;support@sisense.com&lt;/a&gt;”,</p>
<p><b>User Support</b></p>	<p>The User Support text is displayed in the</p>

	<p>top right corner of the Sisense Web Application.</p> <p>By default, the links redirect customers to the Sisense user forums and documentation site. You can modify these URLs to redirect users to your user forums or documentation. This is useful for example when providing OEM services or if you have special forums or instructions to display to your users.</p> <p>“documentationUrl”: “string”,  “forumUrl”: “string”</p> <p>Replace the string with the URL to the relevant locations to redirect your users to your support sites.</p>
<p>Powered by Sisense</p> 	<p>This string is displayed by default in the User Profile for rebranded servers. You can remove this string by setting the boolean <b>PoweredBySisense</b> value to false.</p>
<p>Emails</p>	<p>The sender information and subject line text displayed in automated emails.</p> <p>For further customization, see <a href="#">Rebranding Sisense Automated Emails</a>.</p> <p>Note: Branded emails (for example, using</p>

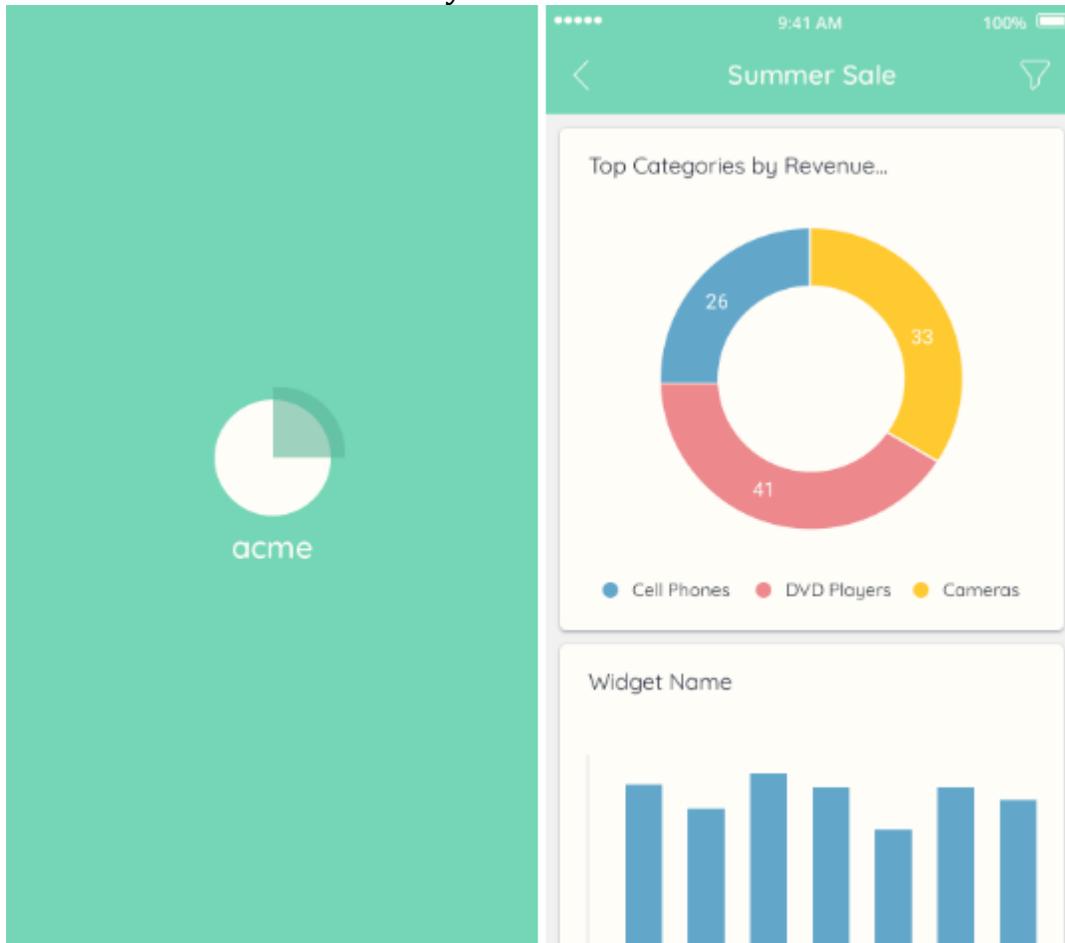
the senderEmail and senderName paramaters) will be used for new user/group invitation emails and password recovery emails. When sharing a dashboard report, or for subscription emails, the email will be sent without the original name and email of the dashboard's creator.

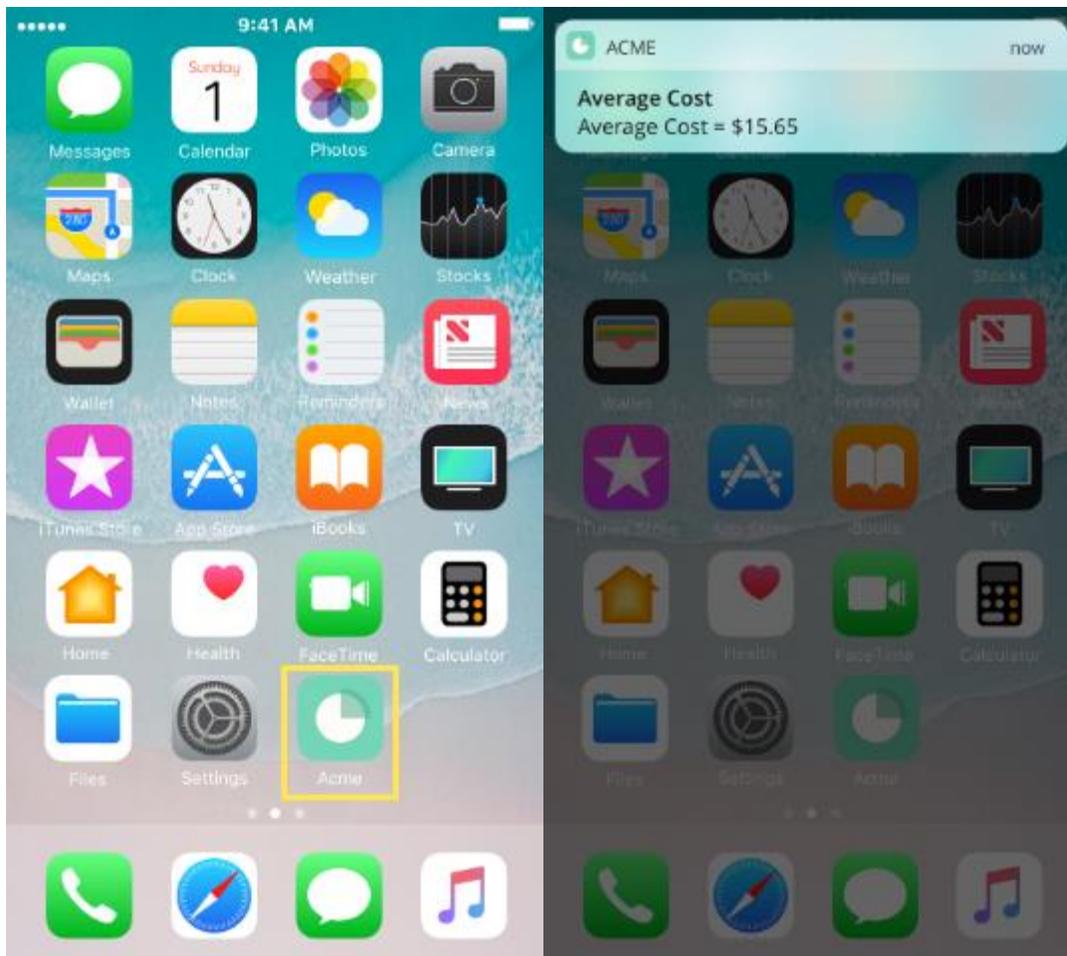
```
"emails" :
{ "senderEmail": "sys-admin@company.com", "senderName": "sys-admin", "templates_directory":
"/resources/<YourBrand>/emails", "passwordRecoverySubject":
"Sisense password recovery",
"newUserInviteSubject":
"Sisense account activation",
"shareWithNewUserSubject":
"A Dashboard has been shared with you:",
"shareWithExistingUserSubject":
"A Dashboard has been shared with you:"
"transferOwnership":
```

```
"string" }
```

## Rebranding Sisense Mobile

Sisense enables you to provide your users with a fully white labeled version of Sisense Mobile that matches your brand's look and feel.





The following objects can be rebranded:

- App icon and title
- Splash screen
- Mobile notification logo image and colors
- Mobile color palette
- App description and information that appears in the app store

Rebranding Sisense Mobile is a premium Sisense offering. Contact your CSM if you are interested in your own white labeled version of Sisense Mobile.

# Sisense Mobile Checklist

Once you have purchased a rebranded version of Sisense Mobile, you need to [download](#) and complete this [checklist](#), which details all the information and resources you must provide Sisense.

The resources include things like icons in different sizes (iPhone, Android), specific colors, mobile app name, and other configurable options within Sisense Mobile. These resource files should be zipped together and the zipped file and your completed checklist should be sent to [rebranded\\_mobile@sisense.com](mailto:rebranded_mobile@sisense.com) with your CSM CCed in the email.

After receiving all the assets and your completed [checklist](#), Sisense will create a rebranded mobile app and provide you with a version to try out. The white labeled mobile app will be fully tested by Sisense QA, but Sisense recommends that you test it out for yourself before uploading it to the app stores.

Sisense will provide instructions for signing the application and loading it up to the app store.

Rebranding Sisense Mobile typically takes up to one month.

# Resources Required for Rebranding Sisense Mobile

Information	Description	Default Value
Date	The date you completed the form.	None. Mandatory
Company Name	The name of your company.	None. Mandatory
Contact Person	The full name of the person responsible for managing rebranding Sisense Mobile.	None. Mandatory
Contact Email	Your email address.	None. Mandatory
CSM Name	The name of your Sisense CSM.	None. Mandatory
Display Name	The application name that is displayed below the icon on the phone home screen.	None. Mandatory
Android package ID	Google Play identifier. The ID must match the criteria described <a href="#">here</a> . This ID must be unique. You can verify whether or not the ID is taken by opening this URL	None. Mandatory

Information	Description	Default Value
	<p><a href="https://play.google.com/store/apps/details?id=com.mycompany.myapp">https://play.google.com/store/apps/details?id=com.mycompany.myapp</a> replacing com.mycompany.myapp with your preferred app ID.</p>	
iOS Bundle ID	<p>App Store identifier. This ID must be unique. For iOS, you need to sign in to your account at <a href="https://developer.apple.com">https://developer.apple.com</a> and click Certificates, IDs &amp; Profiles &gt; App IDs &gt; Registering an App ID (+ sign) &gt; select Explicit App ID.</p>	None. Mandatory
Android Signing Certificate	<p>Google certificate in the form of a JKS file. See <a href="#">Publishing Your App</a> for more information.</p>	None. Mandatory
Apple Signing Certificate	<p>Apple certificate in the form of a p12 file. See <a href="#">Publishing Your App</a> for more information.</p>	None. Mandatory
Server IP Address/hostname	<p>If this option is provided, the rebranded mobile app will be bound to this specific IP address. This means that users of your app will be able to skip the screen for providing the server IP address/hostname.</p> <p>Keep in mind that if you do provide this address, your app can be used only for this IP address. If you intend to use the mobile app for different Sisense servers, or you are not sure your IP address/hostname will remain your permanent identifier, you should not provide it.</p>	None. Optional

Information	Description	Default Value
Primary background color	Background color of the top navigation bar, the splash screen and the buttons.	#FFCB05 (Sisense yellow)
Primary text (foreground) color	The color of the text that is positioned on top of the components with primaryBGColor background.	#000 (black)
Title highlight color	The color of highlighted text that is displayed on a white background (e.g. Recent Dashboards and My Dashboards).	#F2B900 (dark yellow)
Title border color	The color of separators and pipes that are displayed around titleHighlightColor components.	#FFCB05 (Sisense yellow)

## Icon Resource Files

The following icon resources are used by the mobile app for push notifications and for the store listing in iOS. All sizes are mandatory. Please use the exact file names specified below.



Platform	Size (Width x Height)	File Name	Note
iOS	40x40	ios_icon_40x40.png	
iOS	80x80	ios_icon_80x80.png	
iOS	60x60	ios_icon_60x60.png	
iOS	120x120	ios_icon_120x120.png	
iOS	180x180	ios_icon_180x180.png	
iOS	76x76	ios_icon_76x76.png	
iOS	152x152	ios_icon_152x152.png	
iOS	167x167	ios_icon_167x167.png	
iOS	29x29	ios_icon_29x29.png	
iOS	58x58	ios_icon_58x58.png	
iOS	87x87	ios_icon_87x87.png	
iOS	1024x1024	ios_icon_1024x1024.png	App Store Icon

<b>Platform</b>	<b>Size (Width x Height)</b>	<b>File Name</b>	<b>Note</b>
Android	36x36	android_icon_36x36.png	
Android	48x48	android_icon_48x48.png	
Android	72x72	android_icon_72x72.png	
Android	96x96	android_icon_96x96.png	
Android	144x144	android_icon_144x144.png	
Android	192x192	android_icon_192x192.png	
Android	24x24	android_notice_24x24.png	Notifications Icon
Android	36x36	android_notice_36x36.png	Notifications Icon
Android	48x48	android_notice_48x48.png	Notifications Icon
Android	72x72	android_notice_72x72.png	Notifications Icon
Android	96x96	android_notice_96x96.png	Notifications Icon

## Splash Screen

Splash screen images are displayed on the full screen until the app is loaded when the app is cold-launched. All sizes are mandatory.



Platform	Size (Width x Height)	File Name
iOS	320x480	ios_splash_320x480.png

Platform	Size (Width x Height)	File Name
iOS	640x960	ios_splash_640x960.png
iOS	768x1024	ios_splash_768x1024.png
iOS	1536x2048	ios_splash_1536x2048.png
iOS	1024x768	ios_splash_1024x768.png
iOS	2048x1536	ios_splash_2048x1536.png
iOS	2208x1242	ios_splash_2208x1242.png
iOS	1242x2208	ios_splash_1242x2208.png
iOS	750x1334	ios_splash_750x1334.png
iOS	640x1136	ios_splash_640x1136.png
iOS	1125x2436	ios_splash_1125x2436.png
iOS	2436x1125	ios_splash_2436x1125.png
Android	320x240	android_splash_320x240.png
Android	480x320	android_splash_480x320.png
Android	800x480	android_splash_800x480.png
Android	1280x720	android_splash_1280x720.png
Android	1600x960	android_splash_1600x960.png
Android	1920x1280	android_splash_1920x1280.png
Android	240x320	android_splash_240x320.png
Android	320x480	android_splash_320x480.png
Android	480x800	android_splash_480x800.png
Android	720x1280	android_splash_720x1280.png
Android	960x1600	android_splash_960x1600.png

<b>Platform</b>	<b>Size (Width x Height)</b>	<b>File Name</b>
Android	1280x1920	android_splash_1280x1960.png

## Additional File Resources

Platform	Purpose	Filename	Notes
iOS+Android	Logo image that is displayed after the Splash screen.	login_screen_logo.svg	Must be a vector file (svg) to fit all resolutions.
iOS+Android	Login screen background.	login-bg-image.svg	Must be a vector file (svg) to fit all resolutions.
iOS	Firebase project configuration file for pulse alerts (push notifications)	GoogleService-Info.plist	Unique for each app. Generate the file on <a href="https://firebase.google.com">https://firebase.google.com</a> If not provided, notifications from Pulse alerts will not work.
Android	Firebase project configuration file for pulse alerts (push notifications)	google-services.json	Unique for each app. Generate the file on <a href="https://firebase.google.com">https://firebase.google.com</a> If not provided, push notifications from Pulse alerts will not work.

# **Publishing Your App**

Before Sisense provides you with your rebranded mobile app, Sisense signs the app for you. To sign your app, you must provide Sisense with the relevant certificates as described below. Once the app has been signed, Sisense returns the signed app to you and you can publish the app in Google Play or the Apple App store. Apple and Android require that all apps be digitally signed with a certificate.

## Android

- For instructions on how to sign your rebranded Android-based mobile app, see [Sign your app](#).

## iOS

1. Go to <https://developer.apple.com> and create an identifier (App ID) for the new app.
2. Under **App Services**, select **Push Notifications**.
3. If you do not have an iOS production certificate, create one on <https://developer.apple.com>.
4. Download the certificate to the Mac that has your private key, and add it to the key chain (double click).
5. Open Keychain Access on your Mac, and under **My Certificates** locate the iPhone Distribution Certificate you have created for the app.
6. Select **File > Export Item**. Make sure the file format is p12.
7. In [developer.apple.com](https://developer.apple.com), create a Distribution Provisioning Profile for your app, using the same certificate you created above.
8. Download the certificate to your Mac.
9. Double click the certificate. This will start the process to install it.

For more information, see [Code Signing](#).

# Sisense Mobile Upgrades

Sisense regularly releases updates for Sisense Mobile. Sisense will periodically notify (about once a quarter) when an upgraded rebranded version is available and you can choose whether to upload the app to your store or not.

## Security at Sisense

Sisense provides a variety of ways in which you can control which data is exposed to users and secure your connections to Sisense.

The following topics describe how you can configure and maintain the security of your data in Sisense:

- [Security Overview](#)
- [Security Bulletin](#)
- [Data Security](#)
- [SSL](#)
- [Optional Security Features](#)

## Security Bulletin

### **Security vulnerability in NodeJS library used by Sisense 7.2.1 and 7.2.1 Service Pack 1**

We have recently discovered a low severity security vulnerability in the NodeJS libraries used in version 7.2.1. The issue is resolved in 7.2.1 service pack 3 (7.2.1.13003), released Dec. 6th, 2018. This issue is not relevant for any other versions.

We have found that a third party package used by Sisense, NodeJS, included malicious code that could be used to steal Bitcoin from Bitpay and Copay wallets. The malicious code targeted developers at the Copay company that had a very specific development environment setup: **running the payload in any other**

**environment has no effect.** This specific type of targeting means that, ultimately, most applications are not affected even if the malicious module is mistakenly deployed. **Specifically to Sisense, this vulnerability could not be used to access any Sisense data, and was not used by Sisense code, but antivirus programs are identifying it as malware.**

For more information about this vulnerability, see [here](#).

#### **Remediation:**

If you are using one of the builds mentioned above and are concerned about the AntiVirus alerts we recommend you upgrade to the latest 7.2.1 version, available for download [here](#).

#### **Update regarding a security vulnerability resolved in Sisense V6.7.1 on May 23, 2018**

Starting from Sisense V6.7, there was an issue with data security rules. This rare issue occurred only when a single dashboard contained at least two widgets from entirely disconnected tables - meaning that there was no relationship path leading from one table to the other.

If a filter was applied on one of the tables, data security rules of the second table were disabled. The widget showing data from the second table would display all data, without data security rule restrictions.

Note that the issue did not occur when a dashboard only used widgets related to a single data security field. The issue also did not occur if there was any relationship path between the tables of the data security fields. The relationship path did not have to be a direct relationship between the tables, it could be a relationship that went through other tables.

The issue is resolved in Sisense V6.7.1 (build 6.7.1.17004), and in Sisense V7.1.2.

## **Update regarding Meltdown (CVE-2017-5754) and Spectre (CVE-2017-5753 and CVE-2017-5715)**

The Meltdown/Spectre vulnerabilities are the recent vulnerabilities found in Intel's processors. These are vulnerabilities in all of the Windows operating systems. Sisense provides an application and does not provide the server hardware or the Windows OS. These are provided by customers. As such, it's the customers' responsibility to secure their OS.

We recommend customers follow all security recommendations of the vendors who provided the customers OS. The current Windows recommendation is to install the latest available security updates for the OS. Sisense has run extensive performance tests on the OS patch (Windows patch) and found a minimal impact on performance.

For more information about these vulnerabilities and how to negate them, see:

[Meltdown and Spectre](#): A high-level description of the vulnerabilities.

[Project Zero](#): Describes the theory behind the vulnerabilities by those who discovered them.

[Intel Official Announcement](#): Describes Intel's response to the vulnerabilities.

[AMD Processor Security](#): Describes AMD's response to the vulnerabilities.

## Sisense Security Overview

# Sisense Security Architecture

Sisense is built around a robust and flexible security architecture that is both comprehensive and intuitive. This architecture has been designed to ensure security processes are enforced while scaling to enterprise deployments of Sisense. This includes the ability to secure dashboards and data as well as implement custom security requirements that suit your organization. This section provides a general overview of the main security features.

Security is based around three levels associated with sets of security features. The diagram below maps this security architecture on a system, data and object level.



# System Level Security

System-level security encompasses security features for role-based settings and integration options. This includes user and server management, connection to an active directory, Single Sign-On (SSO) implementation, and use of the security REST API.

## User Management

You can assign five primary roles to Sisense users:

- Administrators
- Data Admin
- Data Designer
- Designer
- Viewer

These roles can be defined on either a user or group level to determine sharing, access and security. To learn more about user roles, [click here](#).

## ElastiCube Server Access Rights

You can assign access rights to different ElastiCube servers for individual users, groups or to everyone. These settings allow the management of different environments such as a testing and production server, or servers for specific projects or departments. See also ElastiCube Server and Data Model Security.

**Note:** Following improvements to ElastiCube security in Sisense V7.0 and later, ElastiCubes created prior to Sisense V7.0 are accessible to everyone by default, unless you have defined the ElastiCube's access rights. For ElastiCubes created after Sisense V7.0, the default access is only for the ElastiCube owner. When the ElastiCube is ready, it should be shared with the relevant people, or with "Everyone" if that is the desired access policy.

## Active Directory

Connect existing users and groups from your organization's Active Directory to define security and sharing properties and reduce deployment time. This

removes 'password fatigue' as users can rely on existing credentials while organizational policies around security credentials such as updates can be enforced. See also [Integrating Active Directory](#).

### **Single Sign-On (SSO)**

SSO facilitates seamless integration between Sisense and other systems in your organization while offering standardization of authentication policies across your organization. This can improve user productivity by avoiding password fatigue and reduce support overhead. See also [Configuring SSO](#).

### **REST API**

The Security REST API provide access to parameters to integrate and automate restrictions and access control based on existing settings and standards. Specify access rights and security to dashboards, data models and data. Manage users via the API to create, edit and assign new users or groups. [Click here](#) to visit the API documentation site.

# Object Level Security

Object security defines access rights for different users and groups to various components within Sisense. The two main objects are dashboards and data models.

## **Dashboards**

You can share dashboards on either a user or group level. The sharing options include the configuration of access rights for all users as well as whether users defined as designers may edit the dashboard. The sharing options also include subscription settings that define which users and groups will receive email reports. See also [Sharing Dashboards](#).

## **Data Models**

You can define access rights to different Data Models on a user or group level. This enables flexibility to create models for specific user or group needs while offering strict access control. See also [Sharing ElastiCubes](#).

# Data Level Security

Data access must provide data to people only to the extent that they need to complete their jobs. Data Level Security provides the necessary control to enforce varying degrees of data visibility and access to support the separation of duties. A single dashboard can be shared with many users, but each viewer sees only data relevant to their needs. This reduces both development time and provides for security.

## **Security on the Row Level**

You can grant user and group permissions to specific rows in the data. For each data model, you can apply multiple rules to enforce granular access control.

## **Row Level Defaults**

Control which data is accessible for users or groups that do not have explicit security rules. For example, enable new employees to access a restricted data set until they are added to relevant groups. You can set defaults to include everything, nothing or view based on a security rule.

See more in Data Security below.

# Security Levels

Sisense provides two levels of security:

- [Data Model Security](#)
- [Data Security](#)

# Data Model Security

## What is Data Model Security?

Sisense enables you to define access rights to control which users can access which models, whether they are creating new dashboards or trying to access shared dashboards.

## Data Model Security – Use Case Example

You may have an data model named Marketing and only want the CEO and Marketing team to have access to it. You can grant rights only to them using Data Model Security, thus denying anyone else access.

## How Does Data Model Security Work?

- You can define which users/user groups have access to a data model.
- By default, only the data model's creator, Administrator and Data Administrator can access a data model. Once you start assigning users/user groups access rights to a data model, then those users/user groups will have access to the model. The type of access is determined their role and what access you assign to the user.
- When a user attempts to access a dashboard using a direct link and that dashboard is based on a data model to which that user does not have access rights, a security message is displayed.

## Data Security

Data security in Sisense can be divided into two types, data encryption and data access. Data communication is related to how data is secured by Sisense while be imported into Sisense and written on your server's disk.

When data is imported into Sisense or when you connect directly to a data source, the protocol used depends on the protocols supported by the data source. Sisense supports importing data over SSL, if the source supports it. Sisense supports SSL for data retrieval, for example, when viewing data in dashboards.

Configuration data, such as account credentials and authorization profiles, are encrypted prior to being written to the disk. The encryption technology used by Sisense includes:

1. SHA-256
2. TripleDES
3. AES-256

For data at rest, Sisense supports OS based disk encryption, Windows file system encryption. For more information, click [here](#).

The second type of data security is data access. This type of data security refers to who can access your data after its imported into Sisense and displayed in a dashboard.

### **What is Data Access Security?**

Sisense enables you to define Data Security Rules that control which users can access which portions of the raw data in a data model, down to row granularity.

For example:

- Each widget only shows the data permitted by the Data Security Rules that apply, including totals, averages and so on.
- The data browsers used while building dashboards and widgets only show the data permitted by the Data Security Rules that apply.
- You can define a single dashboard that automatically displays different results for each user (or user group), based on the rows that user is permitted to see.

### **Use Case Example**

- A Sales Order table has a column representing the salesperson that closed a deal.
- You created a quarterly performance dashboard for your salespeople, but want each of them to see only their own data.
- You do not want any of them to be exposed to data that represents the performance of others.

### **How Does Data Access Security Work?**

The Sisense security model is designed to work in a 'Grant access' model, and not in a 'Deny access' model.

By default, when applying a data security rule, access is blocked for everyone, and the best practice is to leave 'Everyone else' set to 'Nothing', while granting groups and users with access to specific data values. The model accumulates grants, meaning that the most permissive combination wins. So if both a group and one of its members have conflicting rules, the permissive combination wins. Each data model contains tables and each table contains fields.

A Data Security Rule defines that a specific user can only see any data of an entire row of a table, if a specific field in that row has a specific value(s).

For example, in a Sales widget a salesperson (for example, Dan) will only see the sales amounts from the rows of a Sales model whose Salesperson field contains the value Dan (rows 1 and 4).

**Sales Table**

#	Salesperson	Product	Amount
1	Dan	HD-TV	\$100
2	Matthew	TV	\$300
3	Amber	Media Center	\$700
4	Dan	Player	\$200
5	Matthew	Air Conditioner	\$600

Dan will not see any part of a row in the data model that does not contain the value 'Dan' in the Salesperson field, nor will any amounts from this row be included in totals.

**Note:** The entire row of data is not seen by the relevant user even when the field to which the rule applies does not appear in the widget.

If a widget that shows the amount spent per product is shared with Dan, then he will only see HD-TV and Player and the sales total will be 300\$.

## Defining Data Access Security for a Data Model

Each Data Security Rule applies to a specific field in a data model and to specific user(s)/user group(s). It enables you to define the values that must be contained in a specific field to enable that entire row of data to be available to a user.

If you want to exclude or hide data from certain users, you can define these rules in the Sisense REST API. For more information, see [Restricting Data Access for Data Models](#).

### To access Data Security:

1. Click **Admin** and select the **Data Sources** tab in the menu.
2. For the relevant data model, select and click **Data Security**.

If no data security rules have yet been defined for this data model, then the following message is displayed:



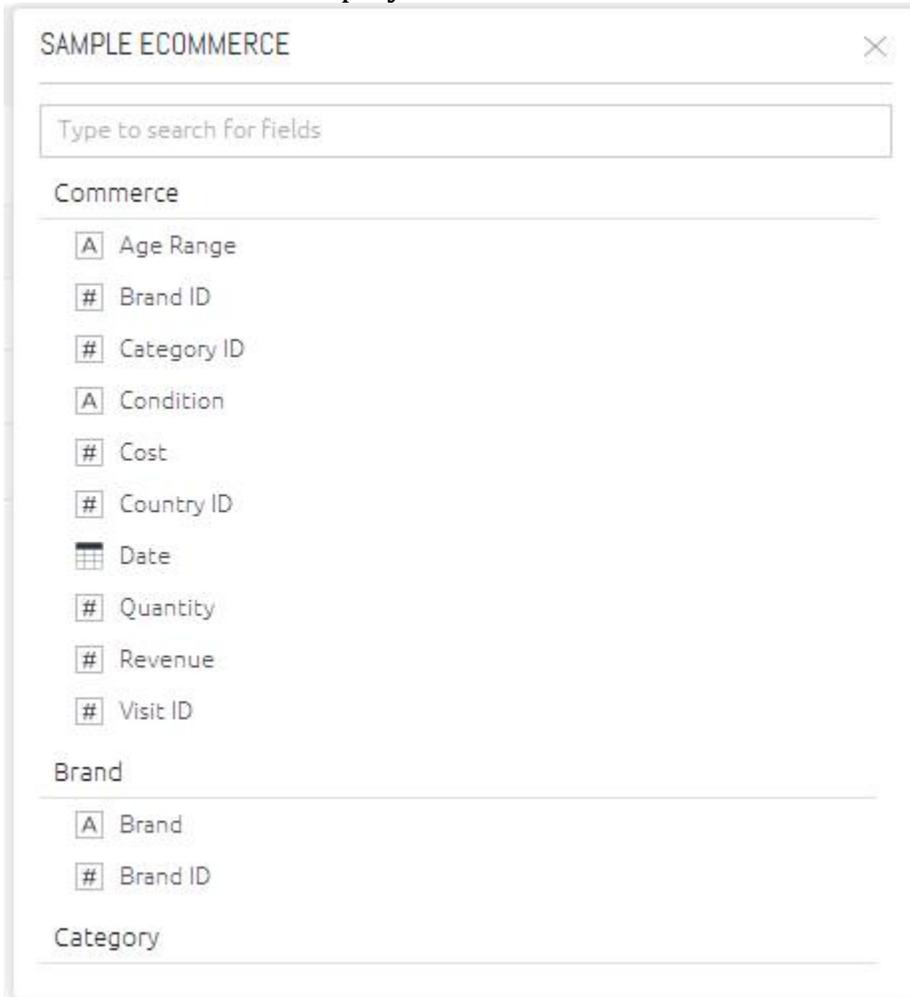
The screenshot shows a warning message with the following text: "Data Security Has Not Been Set", "All users who can access Facebook Ads Demo on (localhost) can view all records.", "To restrict users from accessing specific records, click 'Add Field'", and a blue "Add Field" button.

Data Security Has Not Been Set

All users who can access Facebook Ads Demo on (localhost) can view all records.  
To restrict users from accessing specific records, click "Add Field"

Add Field

3. Click **Add Field** to display a list of the fields in this data model.



**Note:** Row-based data security rules may cause reduced performance when applied to floats.

4. Select a field. For example, **Brand**. The following window is then displayed in which you can define rules.

**Note:** You cannot select date type fields.



The left side of this table enables you to define which users/user groups can access this data. Click **+ Add Restriction** and start typing into the **Restricted User/Groups** field to get a drop-down list.

Add as many users/user groups, as necessary.

The right side of this window enables you to define which values the

specified users/user groups are permitted to see.

Start typing into the **Values** field to get a list.

Multiple values can be selected.

The value of numeric type fields must be typed into this field, as no auto-complete option appears for numeric type fields.

Alternatively, you can start typing in one of these values (in both text and numeric type fields):

- **Everything:** To specify that the selected users/user groups can see this data no matter what the value is in this field.
- **Nothing:** To specify that the selected users/user groups cannot see this data no matter what the value is in this field.



For example, you can define that the following Users/User Groups must have the following values in the Product Category column to enable them to see their data row in a widget.

#	User/User Group	Product Category
1	Management	Everything
2	Bob	Apple Mac Desktops
3	Don	Calculators, Camera Flashes
4	Everyone else	Nothing

This means that management can see the data of all Product Categories, Don can only see the data of Calculators and Camera Flashes, Bob can only see the data of Apple Mac Desktops, and Everyone else won't see anything.

### How Does Data Level Security Work for Tables with Relationships?

Tables in a data model may have a relationship between them.

As described above, each widget only shows any data of an entire row of a table, if a specific field in that row has a specific value.

In addition, a widget may further restrict the data shown to a specific user when a rule is defined for a table that has a relationship to a table that has a field in the widget.

This means that a widget only shows the data permitted by the **combined** Data Security Rules assigned to all the tables that have **any field in the widget**.

As described above, the entire data row is restricted even when the field to which the rule applies does not appear in the widget. The entire row of data is also restricted even when the field of the relationship between the two tables does not appear in the widget.

#### **Use Case Example – Expanding Upon the Example Above**

- The Sales table has a column that has a relationship with a Deal Contacts table that holds the contacts that were involved in each deal.
- You created a Deal Contacts widget for your salespersons.
- As described in the example above, the Sales table has a Data Security Rule that maps each user to their matching field value, so that each sales person only sees their own data.
- Even though the Deal Contacts table does not have any Data Security Rules defined for it, the Deal Contacts widget only enables each sales person to see the contacts associated with their own sales, because of the Data Security Rule assigned to the Sales table.

## Restricting Data Access for Data Models

In Sisense, all users who have access to your data models can see all of the data. If you define any data security rules, the default behavior is inclusionary, which means you define what values of a field a user is allowed to see. For example, you can allow each Sales rep to see transactions for their own customers, and prevent Sales reps from seeing the transactions of other customers. In this case, you define a row-based data security rule for each Sales rep, based on the customer IDs of each customer.

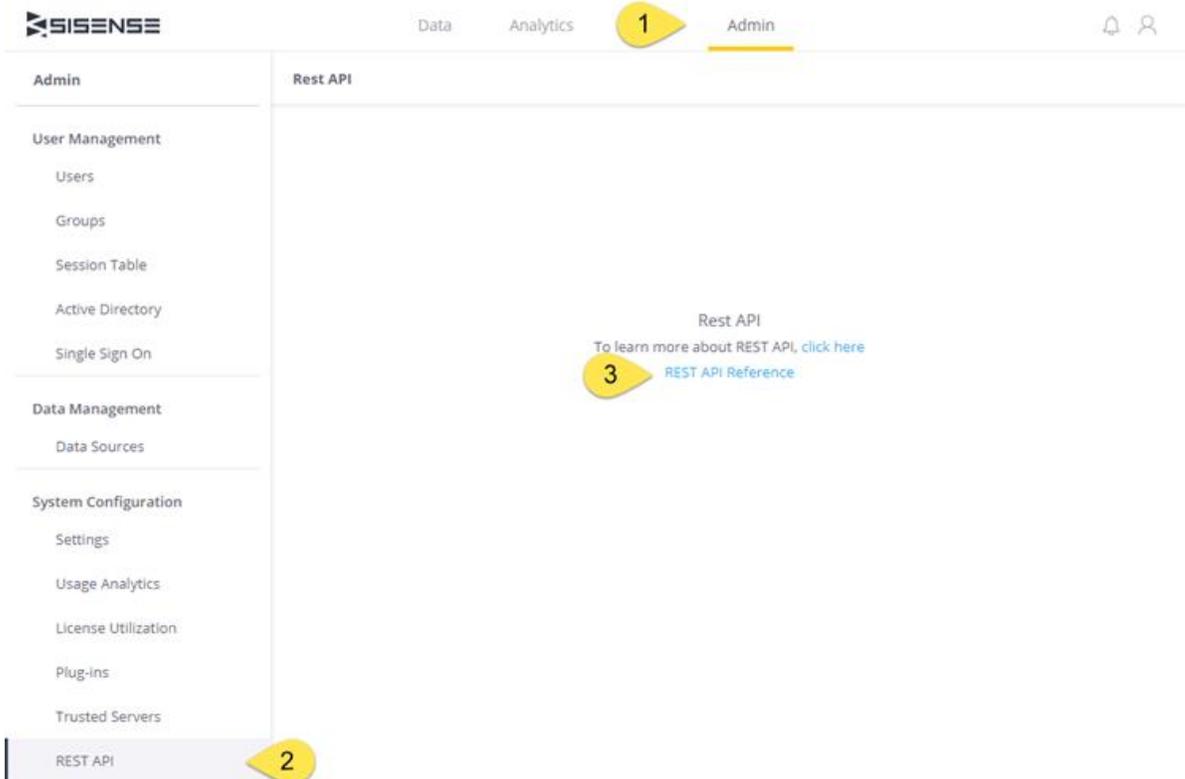
In some cases, you may want to allow all of your users to see your data except for a specific user or group of users. In this case, exclusionary rules are preferred. For example, let's assume that your company has thousands of customers, and your policy is that all Sales reps can see information for most of your customers, not only for their own customers. You may have certain customers whose data is sensitive and should only be accessed by certain authorized Sales reps. In this case, it's easier to manage a definition that allows access to everything, except the few restricted customers, than to manage a list of the thousands of customers whose data is freely available to all Sales reps.

To restrict certain rows of data to a specific user or group of users, through the Sisense REST API, you can change the default data security behavior to exclusionary, which allows you to hide or restrict access to data to certain users. You can combine inclusionary rules with exclusionary rules. In case the rules conflict, the exclusionary rules take precedence.

**Note:** When multiple data security rules exist for a specific field-user or field-group combination, the "inclusionary" rules will be combined with "OR" logic between them. "Exclusionary" rules will be combined with "AND" logic between them.

**To change the data security behavior for a rule:**

1. In Sisense, click Admin, and then Rest API, followed by REST API Reference.



2. In Version 0.9 of the REST API, select **elasticubes**, then POST `/elasticubes/datasecurity`.

```
[
  {
    "allMembers": true,
    "column": "string",
    "datatype": "string",
    "elasticube": "string",
    "exclusionary": true,
    "members": [
      "string"
    ],
    "server": "string",
    "shares": [
      {
        "party": "string",
        "type": "string"
      }
    ],
    "table": "string"
  }
]
```

3. In the body of your call, update the value of “**exclusionary**” to **true**.
4. Click **Try it out** and then **Execute**.

## Setting Up SSL

SSL (Secure Sockets Layer) is a security protocol that encrypts the server-client data channel. In Sisense, SSL secures the link between the Sisense server and the clients' browsers (where dashboards are created and viewed). With SSL in place, you can access Sisense via an HTTPS secure connection, which is password protected. This is recommended when you want users to be able to access dashboards from outside the organization.

Keep in mind, configuring SSL on your Sisense server and in your IT environment should be performed by an IT Specialist or Web Administrator.

**Note:** If you are using a reverse proxy or load balancer, you should ensure that the communication between the proxy and Sisense is HTTPS.

The procedure below describes how to set up SSL in Sisense from Sisense V7.2 and later. If you are upgrading from an earlier version to Sisense V7.2 or later, see [Configuring SSL after Upgrading](#) at the end of this topic.

### To configure your Sisense SSL settings:

1. In your browser, open the Configuration Manager located at <http://localhost:3030>.
2. Toggle the **Enable SSL** switch to **Enabled**.
3. Upload or enter your SSL certificate. Sisense supports two types of certificates, PFX and CERT files. Typically, these files are provided by a third party provider.

**When using SSL, the certificate must include the root certificate and any intermediate certificates.**

**PFX:** PFX files contain the public key file (SSL certificate file) and the associated private key file in a single file. If you select **PFX**, drag the PFX file into the browser or click **Browse** and navigate to the file. In **PFX Password**, enter the password you received after your PFX was generated.

**CERT-Key:** A .cert file is the public key, which is used to verify client authentication requests. It is what is received by an HTTP client from a server in the SSL handshake. If you select **CERT-KEY**, two boxes are displayed **CERT File** and **Key File**. In **CERT File**, drag the .cert file into the browser or click **Browse** and navigate to the file. In **Key File**, drag the key file into the browser or click **Browse** and navigate to the file. The .key file is

the private key to the certificate.

Alternatively, if the files are not provided, and you have received a coded certificate and key, you can enter these values in the **SSL Certificate** and **SSL Key** fields.

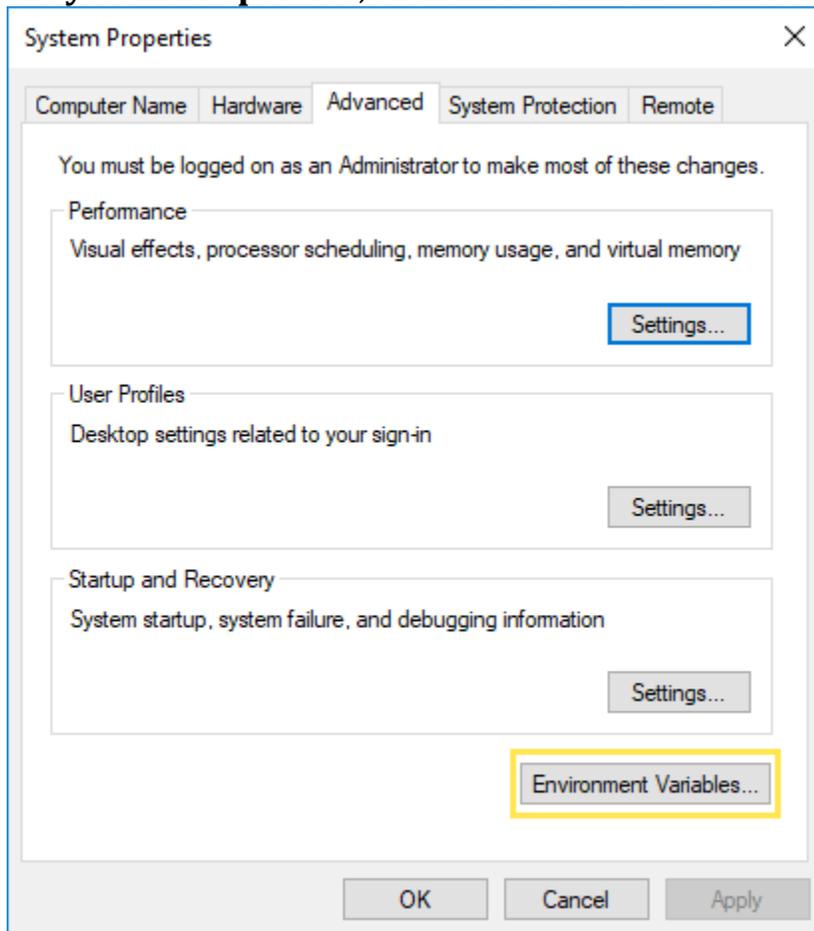
4. In **Port**, enter the port to be used when accessing Sisense. By default, this is 8081, however, if you are implementing SSL, typically the port is set to 443.
5. In **Internal IP**, enter your system DNS name, for example, test.sisense.com.
6. After you have finished defining these settings, in the Configuration Manager, click **Save**.

# Self-Signed Certificates

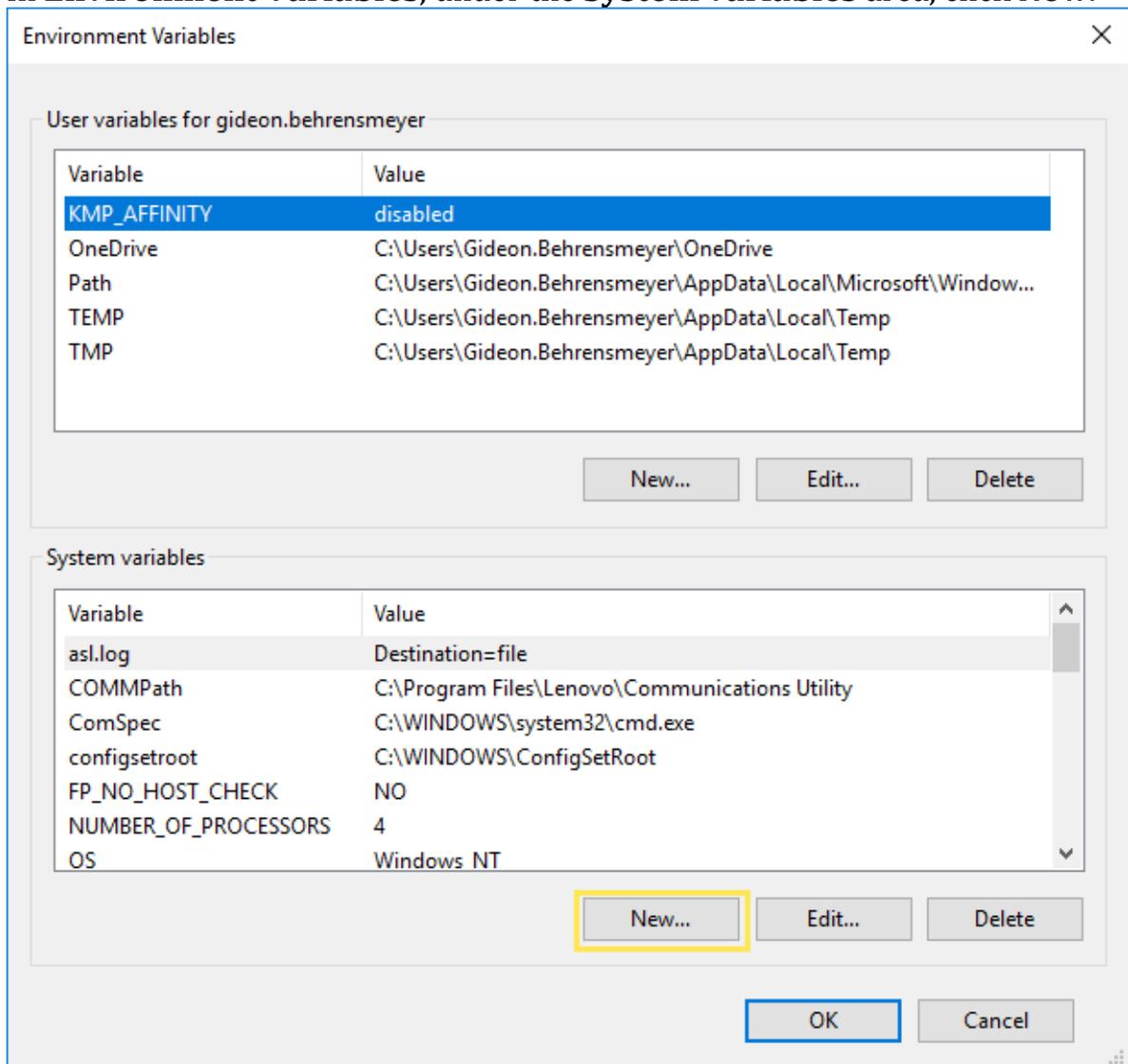
If you are using a self-signed certificate, you may experience problems connecting to data sources. This is caused by the self-signed certificate being rejected. To fix this, you need to give the `NODE_EXTRA_CA_CERTS` environment variable a file path to a file containing your certificate.

## To prevent self-signed certificates from being automatically rejected:

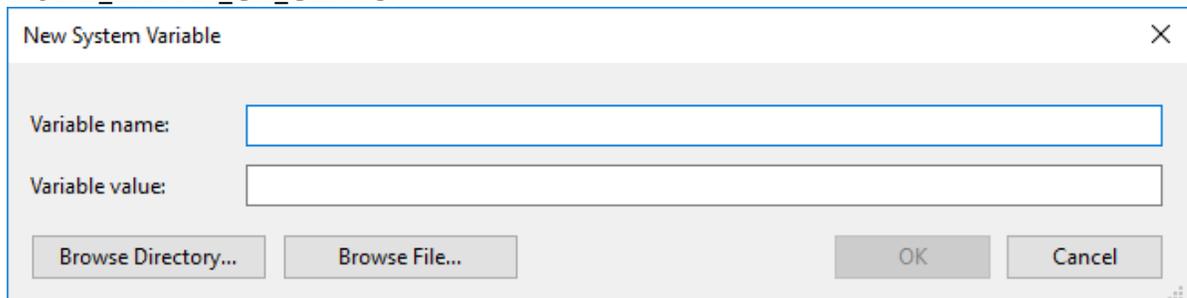
1. On your Sisense server, right-click on **This PC** (Or My Computer in older versions of Windows) and select **Properties**.
2. On the left side, click **Advanced system settings**.
3. In **System Properties**, click **Environment Variables**.



4. In **Environment Variables**, under the **System variables** area, click **New**.



5. In the **New System Variable** dialog box, in **Variable name**, enter **NODE\_EXTRA\_CA\_CERTS**.



6. In **Variable value**, enter the address of your **.ca** file.  
OR  
Select **Browse file** and navigate to the **.ca** file.
7. Click **OK**.

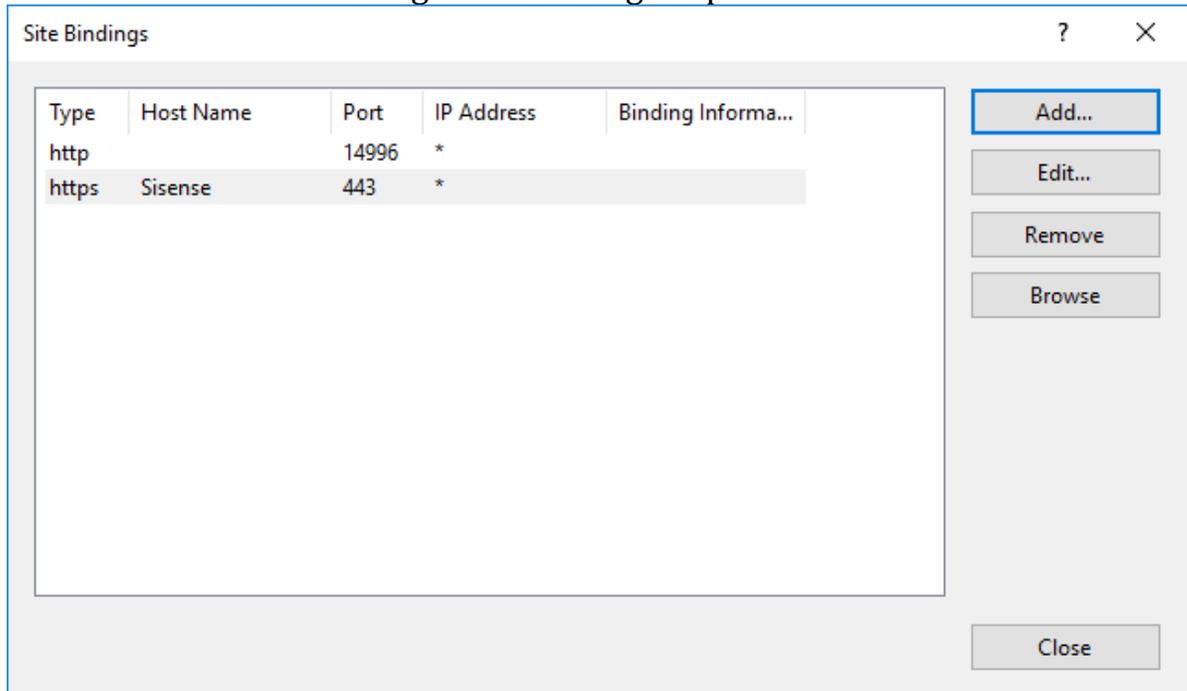
8. Restart your computer. This should resolve connection problems caused by sign certificates.

# Configuring SSL After Upgrading

Earlier versions of Sisense used IIS, which meant that part of your SSL configuration was stored in the IIS Manager. As NodeJS is now used as the application server, your Sisense SSL bindings must be removed when you are upgrading to Sisense V7.2 and later from an earlier version for SSL to continue to work.

## To configure SSL after upgrading:

1. On the Sisense Server, open the IIS Manager.
2. Remove from the IIS Manager the bindings to port 443.



3. Reset IIS.
4. In your browser, open the System Configuration at <http://localhost:3030/>.
5. Verify that **Enable SSL** is enabled.
6. Upload or enter your SSL certificate. Sisense supports two types of certificates, PFX and CERT files. Typically, these files are provided by a third party provider.

**When using SSL, the certificate must include the root certificate and any intermediate certificates.**

**PFX:** PFX files contain the public key file (SSL certificate file) and the

associated private key file in a single file. If you select **PFX**, drag the PFX file into the browser or click **Browse** and navigate to the file. In **PFX Password**, enter the password you received after your PFX was generated. **CERT-Key**: A .cert file is the public key, which is used to verify client authentication requests. It is what is received by an HTTP client from a server in the SSL handshake. If you select **CERT-KEY**, two boxes are displayed **CERT File** and **Key File**. In **CERT File**, drag the .cert file into the browser or click **Browse** and navigate to the file. In **Key File**, drag the key file into the browser or click **Browse** and navigate to the file. The .key file is the private key to the certificate.

Alternatively, if the files are not provided, and you have received a coded certificate and key, you can enter these values in the **SSL Certificate** and **SSL Key** fields.

7. In **Port**, enter the port to be used when accessing Sisense. By default, this is 8081, however, if you are implementing SSL, typically the port is set to 443.
8. In **Internal IP**, enter the system URL (for example, test.sisense.com).
9. Click **Save**.
10. Click **Yes** to restart the Sisense services.

## Optional Security Hardening for Sisense Web Pages

This topic provides additional security options that can be applied for hardening the security of Sisense web pages for your needs.

# Cookie Security

By default, Sisense web pages have cookies that contain a secureFlag. This flag instructs the browser that the cookie should only be returned to the Sisense Web Application over encrypted connections (HTTPS).

# Strict Transport Security

HTTP Strict Transport Security (HSTS) is a method for preventing any communications from being sent over HTTP to the specified domain and allows only communication over HTTPS. This is useful for preventing man-in-the-middle attacks or users with invalid certificates from accessing your dashboards. This is automatically applied when you enable SSL for the Sisense Web Application.

# Allowed Domains for Embedded Dashboards

If you are embedding a dashboard on your website, you can control who can access the website by adding allowed domains to a whitelist.

Allowed Domains enable you to limit where your embedded dashboards can be viewed, even if someone takes the embed code from your page.

When you add a domain to the whitelist, Sisense includes the domain in the X-Frame Options header of the dashboard web page.

For example:

```
<add name="X-Frame-Options" value="ALLOW-FROM  
https://dashboardurl.com" />
```

## To add your domain to a white list:

1. In the **Admin** page, select **Settings**.
2. Under Security Settings, enter your domain.

Security Settings

Embedded Domain White List



+ Add Domain

Add

3. Click **Add**.
4. Click **Save** at the bottom of the page.

## Multi-Node Deployments

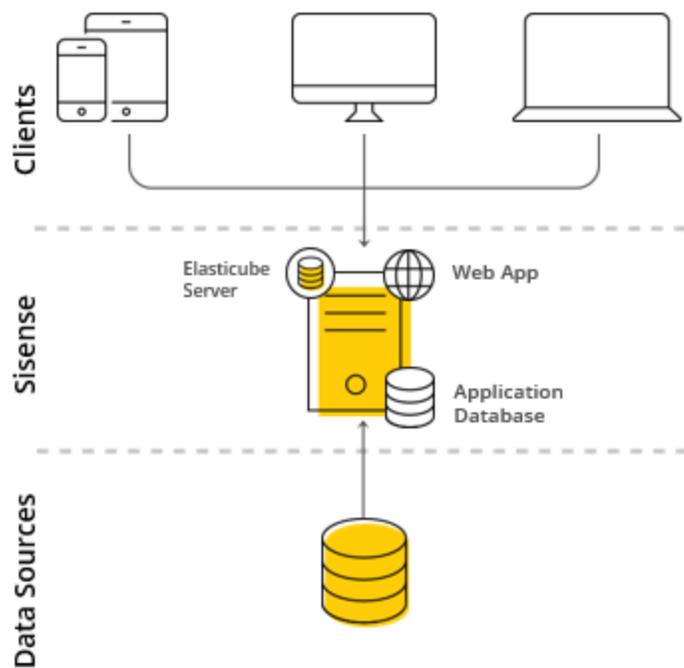
# Overview

Sisense provides flexible design choices for supporting high availability (HA) and scalability for your Sisense deployments. Typically, you want to implement multi-node deployments when you want to optimize performance or build in redundancy. Before implementing a multi-node environment, you can ask yourself the following questions:

- Do you want to improve dashboard load time?
- Are your ElastiCubes taking a long build time to build?
- Do you need to support a lot of concurrent users?
- Do you have a lot of ElastiCubes?
- Do you need high availability for your system?

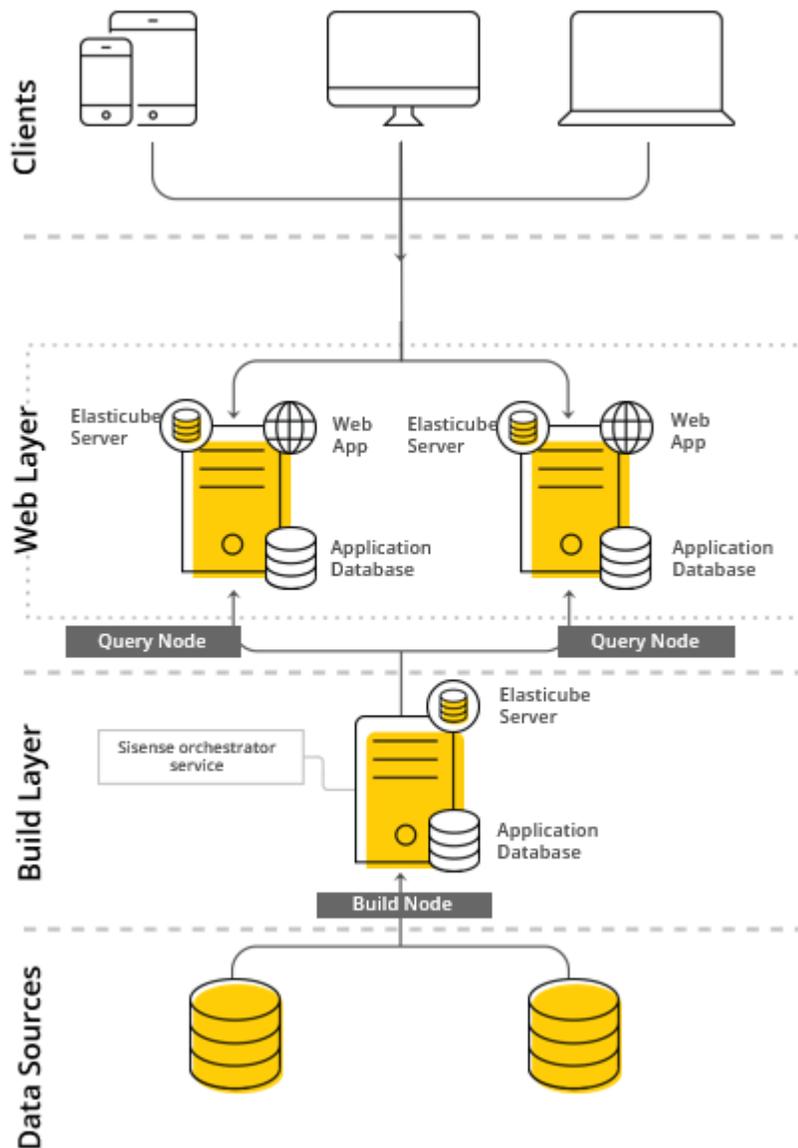
If you answered yes to any of these questions, you may want to consider scaling out Sisense.

The basic Sisense architecture looks like the diagram below where Sisense is installed on a single machine. Your users connect to your machine and you import or connect to your data sources from the same machine. This machine supports the Sisense web application, your data models, and all of your users.



One way to scale out Sisense is to improve your hardware with more storage, CPU, and memory; however, this can get quite expensive and doesn't provide any redundancy.

Another way to scale out Sisense is to add more machines, thus improving performance and building in redundancy into the system as shown in the diagram below.



While Sisense is fully-functional in a single node environment, a multi-node deployment is necessary for scalability to support large amounts of concurrent users and redundancy in case of a failure of one of the Sisense components. Replicating each of these components provides redundancy and fault tolerance against the failure of any single component. The replicated components are combined into nodes. There are three types of nodes, a build node, query nodes, and an application node. These nodes and their components are described below.

# Nodes

# Build Node

The build node is responsible for building ElastiCubes and distributing the build to query nodes via the Sisense Orchestrator Service. The Sisense Orchestrator Service is an automated service that you configure on the build node to synchronize and distribute built ElastiCubes to the query nodes. For more information, see [Distributing ElastiCube Builds to Query Nodes](#).

Build nodes include an ElastiCube Server, application database, Sisense plugins, and the Sisense Orchestrator Service.

The build node is not replicated as its failure only prevents building new ElastiCubes, not issuing queries from Sisense.

**Note:** Sisense plugins must be located on the build node.

# Query Nodes

Query nodes are responsible for supporting queries from Sisense dashboards on the application layer. ElastiCube models are distributed by the build node to the query nodes. The query nodes' ElastiCube models are then combined into ElastiCube Sets to increase redundancy by separating the web and ElastiCube servers across multiple query nodes. If a build node is distributing a build to one ElastiCube server, Sisense automatically directs any queries to the other ElastiCubes in the ElastiCube Set.

The query node can be configured as a single application stack where each node hosts Sisense, ElastiCubes, and an application database. In this configuration, if the machine hosting the components fails, the whole query node will fail. Queries will then be redirected to the next available query node. Another option is to host each component of the query node separately in a distributed application stack. In this configuration, if a component of the query node fails, the rest of the query node is not affected. For an example of a single application stack, see Scenario 1 and for an example of a distributed application stack, see Scenario 2.

# Application Nodes

The application node supports your Sisense application. This is the interface you see when you log into Sisense, including the Model Editor, dashboards, etc. In some models, this resides on the same node as the query node.

# Components

Sisense has many components that reside on each of the nodes. These components are highlighted in the diagram below. Some of these components are responsible for supporting Sisense, such as the application database, configuration database, and message broker. Other components, such as a load balancer, Multi-Node Deployment Wizard, and ElastiCube Sets are responsible for supporting high availability in Sisense. Each of these components is described in more detail below.

# Application Database

The application database is installed with Sisense and supports Sisense. The application database is a central repository for Sisense metadata including user information, permissions, data sources, dashboards, jobs, etc.

If the application database fails, the Sisense web application will fail.

To achieve redundancy and high availability of the application database, a minimum of three nodes is required.

# Message Broker

The message broker is a component of Sisense and is responsible for the communication of events between various Sisense services across your Sisense configuration. Sisense availability and functionality are heavily dependent on the broker service. It should be replicated with at least two nodes to ensure that the services can continue to communicate with each other in case the message broker fails.

# Configuration Database

The configuration database provides a single representation for the cluster regarding the topology, configurations, and state.

To achieve redundancy and high availability, the configuration service should be replicated with at least three nodes to ensure that your configuration is up-to-date across your entire deployment.

# Load Balancer

To support a multi-node configuration, you must handle load balancing on your side prior to directing traffic to one of your Sisense nodes. Load balancing spreads requests across multiple query nodes according to an algorithm you define and the current status of the query node.

When implementing ElastiCube Sets, Sisense's query nodes operate in active-active mode. This means that each of the query nodes is active and can handle requests when the node is not building and its components are available. For example, traffic could be spread 50-50 across two web servers and if a component fails, a load balancer should redirect traffic to the other available web server.

# ElastiCube Sets

SisenseElastiCube Sets are collections of identical ElastiCube models that allow you to query running ElastiCubes within the ElastiCube Set while other ElastiCubes are building.

# Components that Cannot be Replicated

Several Sisense components are deployed as single services that can not be replicated and do not have redundancy:

- ElastiCube web management
- Plug-ins
- Scheduled reporting jobs
- Sisense Orchestrator
- Build nodes

# Multi-Node Configuration

To support more concurrent users and queries, and build in redundancy into your deployment, you must provide additional machines and configure the orchestration between the various Sisense nodes and their services. Sisense makes it easy to implement a multi-node deployment with the Multi-Node Deployment Wizard. This wizard automates the configuration of your nodes. If you want to implement high availability, after you have run the wizard, you configure the Sisense Orchestrator service that manages the distribution of ElastiCubes across multiple machines.

The following pages describes the multi-node deployments Sisense supports, how to configure them in the Multi-Node Deployment wizard, and how to configure the distribution of ElastiCubes with the Sisense Orchestrator.

## Next Steps

- [Supported Deployments](#)
- [Setting Up Multi-Node Deployments](#)
- [Distributing ElastiCubes to Query Nodes](#)
- [Setting Up ElastiCube Sets](#)
- [Securing the Message Broker's Communication](#)

## Installing the Multi-Node Deployment Wizard

### Previous Steps

- [Supported Deployments](#)

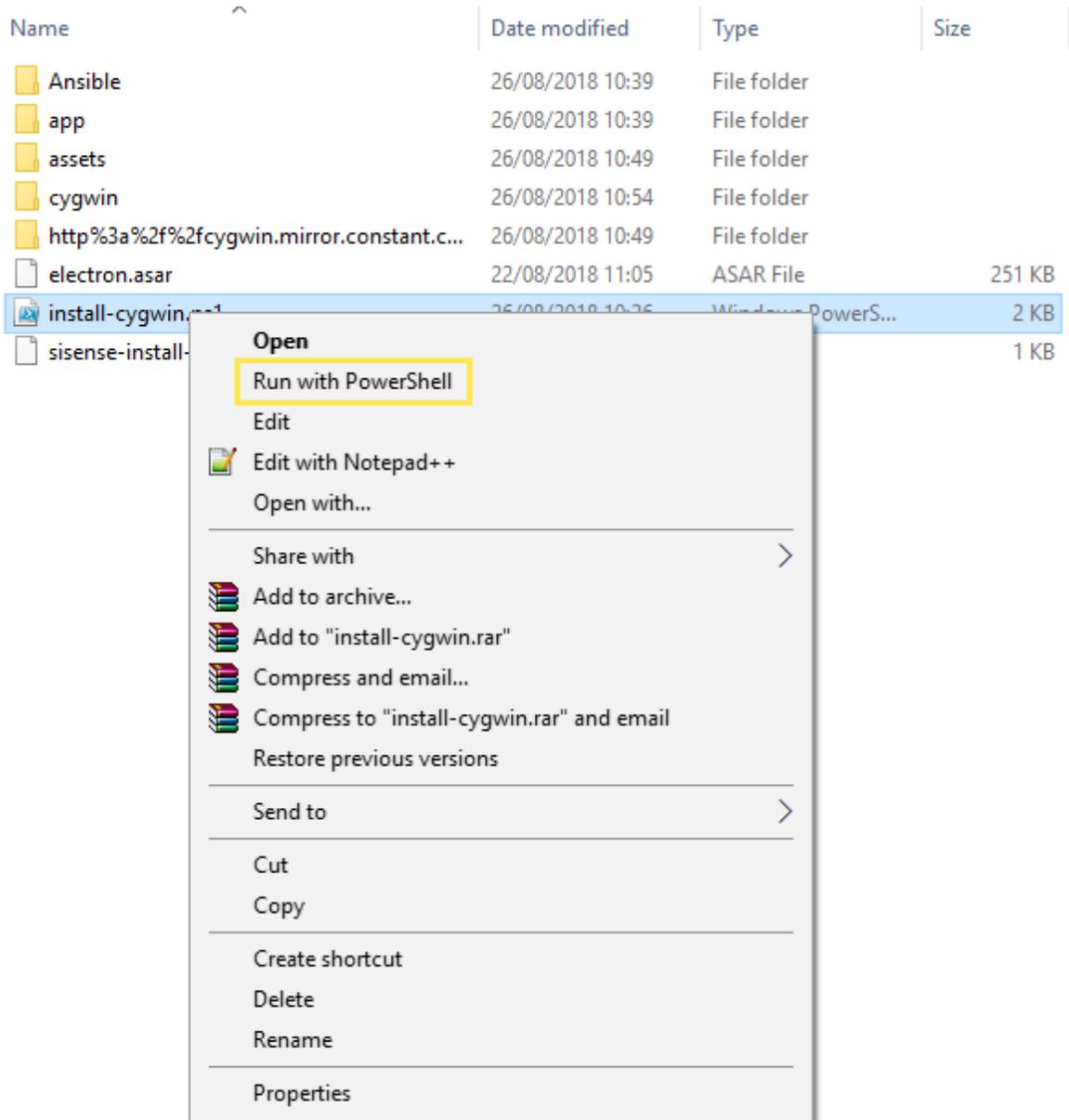
With a multi-node deployment, you can increase the reliability of your configuration further by adding additional query and build nodes, and an application node.

To simplify the process of setting up a Multi-Node environment, Sisense provide a Multi-Node Deployment Wizard that automates most of the setup process for you.

**Note:** Windows 8, 10, and Windows Server 2012, 2016 are supported.

### To install the Multi-Node Deployment Wizard:

1. Download the [Sisense High Availability Multi-Node Deployment Wizard](#) to your build node. When saving the file, make sure there are no spaces in the file name. This can happen if you download the file to the same directory multiple times, for example, MultiServerDeploymentWizard.zip(1).
2. Extract the contents of the zip file.
3. Download the [Cygwin script](#) and [Sisense-install-ansible.bsh](#) and save the files to the directory  
`...MultiServerDeploymentWizard/resources/` located in the directory where you extracted the contents of the Multi-Node Deployment Wizard zip file.
4. Run `install-cygwin.ps1` with Powershell. This process can take between 20-30 minutes.



During the process, Cygwin is downloaded and installed. Once the installation is complete, you are prompted to press any button to close the script.

```
Windows PowerShell

Directory: C:\Users\Gideon.Behrensmeier\Desktop\SisenseHAConfigurationWizard.win32

Mode                LastWriteTime         Length Name
----                -
d-----          10/10/2018         10:22     assets
=====
=== Starting Cygwin download ===
=====
Downloaded, installing..
note: Hand installation over to elevated child process.
Done!
=====
Starting Ansible + dependencies install via Cygwin/Bash
=====
Running bash script via Cygwin at C:/Users/Gideon.Behrensmeier/Desktop/SisenseHAConfigurationWizard.win32/sisense-install-ansible.bsh
/usr/bin/bash: C:/Users/Gideon.Behrensmeier/Desktop/SisenseHAConfigurationWizard.win32/sisense-install-ansible.bsh: No such file or directory
Done!
Press any key to continue..._
```

The Multi-Node Deployment Wizard is installed.

## Next Steps

- [Model 1: 1 Application Node, 1 Query Node/Build Node](#)
- [Model 2: 1 Application Node, 1 Query Node, 1 Build Node](#)
- [Model 3: 2 Application/Query Nodes and 1 Build Node](#)
- [Model 4: 2 Application Nodes, 2 Query Nodes, and 1 Build Node](#)

### Using ElastiCube Models in Multi-Node Environments

You can manage your ElastiCube models in multi-node deployments with the following configuration and behavior:

## Step 1

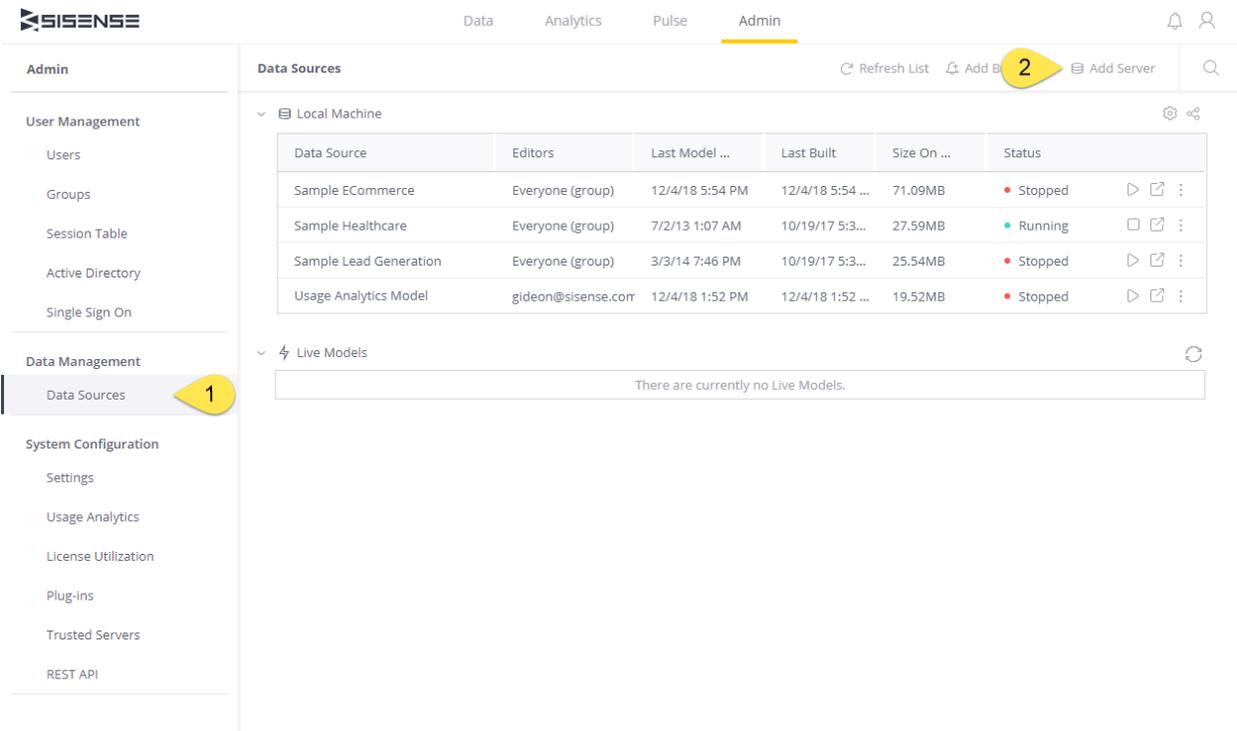
The application must know the IP address of the build server. You can specify the build node server IP address in the Configuration Manager:

1. In your browser, open the Configuration Manager at: [server address]:3030, for example, 127.61.98.2030:3030.
2. In **Build Node Server**, enter the IP address of the build node.

## Step 2

To edit and build Elasticubes in the **Data** page, you must add the build node as a remote server.

In the **Admin** page, select **Data Sources** and click **Add Server**.



The screenshot shows the Sisense Admin interface. The left sidebar has 'Data Sources' highlighted under 'Data Management' with a yellow callout '1'. The main content area shows 'Data Sources' with a table of local machines and a section for live models. The 'Add Server' button is highlighted with a yellow callout '2'.

Data Source	Editors	Last Model ...	Last Built	Size On ...	Status	
Sample ECommerce	Everyone (group)	12/4/18 5:54 PM	12/4/18 5:54 ...	71.09MB	Stopped	▶ ↗ ⋮
Sample Healthcare	Everyone (group)	7/2/13 1:07 AM	10/19/17 5:3...	27.59MB	Running	◻ ↗ ⋮
Sample Lead Generation	Everyone (group)	3/3/14 7:46 PM	10/19/17 5:3...	25.54MB	Stopped	▶ ↗ ⋮
Usage Analytics Model	gideon@sisense.cor	12/4/18 1:52 PM	12/4/18 1:52 ...	19.52MB	Stopped	▶ ↗ ⋮

Enter your IP address. Make sure to use the server IP address and not the server name when you add

### Add Server

IP or Server Name

Who can access this server?

Admins Only  Everyone ⓘ

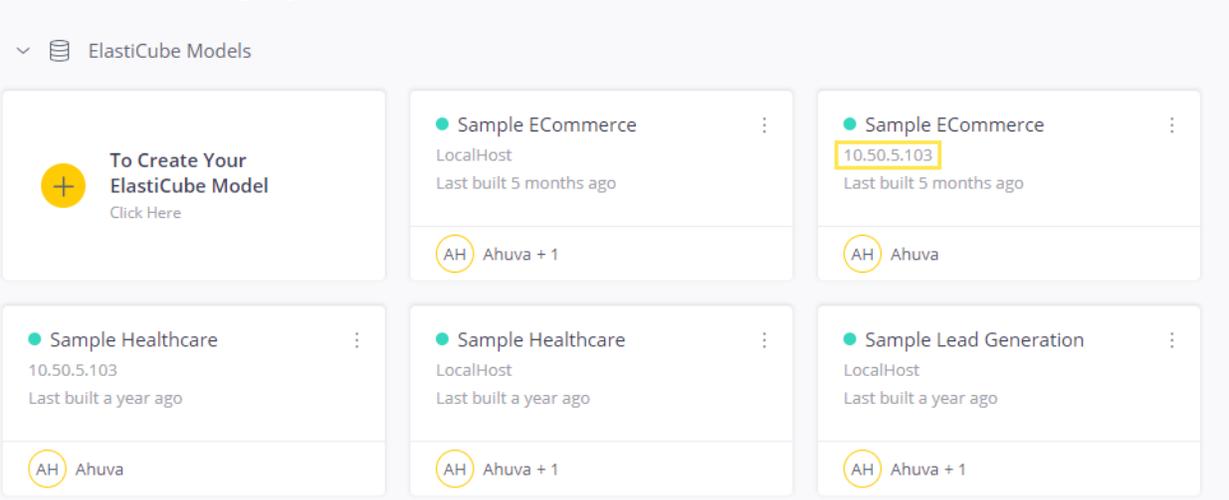
Add Server

Cancel

it.

**Note:** If you initiate a build from the **Data** page at the same time that a build is initiated by the Orchestrator service or when a build is already in progress, one of the builds may fail. If this happens, initiate another build.

You will now see the ElastiCubes from the build server on your **Data** page. This ElastiCube is displayed with the IP address of the build node.



You can now edit and build ElastiCubes that reside on your build server from the **Data** page.

## Step 3

If you want to use multiple query nodes, configure the [Orchestrator](#) to distribute your ElastiCube models from the build node to the query nodes.

For more information, see [Activating the Sisense Orchestrator Service](#).

## Step 4

Connect your dashboards to an ElastiCube or ElastiCube Set.

If your dashboard is new, you can select the data source when creating the dashboard. For existing dashboards, see [Changing a Dashboard's Data Source](#).

If you want to use ElastiCube Sets, perform the following steps:

1. Add the query nodes as remote servers in the **Admin** page. See Step 2 for information on how to set up a query node as a remote server.
2. Make sure the Orchestrator has completed distributing the ElastiCubes to the query nodes. You will be able to see the ElastiCubes from the query nodes on the **Data** page.
3. [Create the ElastiCube Sets](#) using the ElastiCubes on the query nodes.
4. Connect the dashboards to the ElastiCube sets. See [Changing a Dashboard's Data Source](#) for more information.

**Note:** You must edit only the ElastiCube used for the build, meaning the ElastiCube that is located on the build node. You should not edit the ElastiCubes that are on the query nodes, and are part of the ElastiCube sets, as their data is overwritten upon the next build.

## Supporting Usage Analytics in Multi-Node Environments

Usage Analytics is supported in multi-node deployments. The information is collected from all the nodes in the system, stored on the build node, and then displayed in your dashboard on the application node.

If you are using the Usage Analytics feature in a multi-node deployment, these are the steps you need to perform to support Usage Analytics. In some cases, you may have defined these settings already, if so, verify that each of these steps is complete.

## Step 1

Sisense must know the IP address of the build server. You can specify the build node server IP address in the Configuration Manager on your application node:

1. In your browser, open the Configuration Manager at: [server address]:3030, for example, 127.61.98.2030:3030.
2. In **Build Node Server**, enter the IP address of the build node.

Internal IP/Host

Build Node Server

### Application Database

Connection String

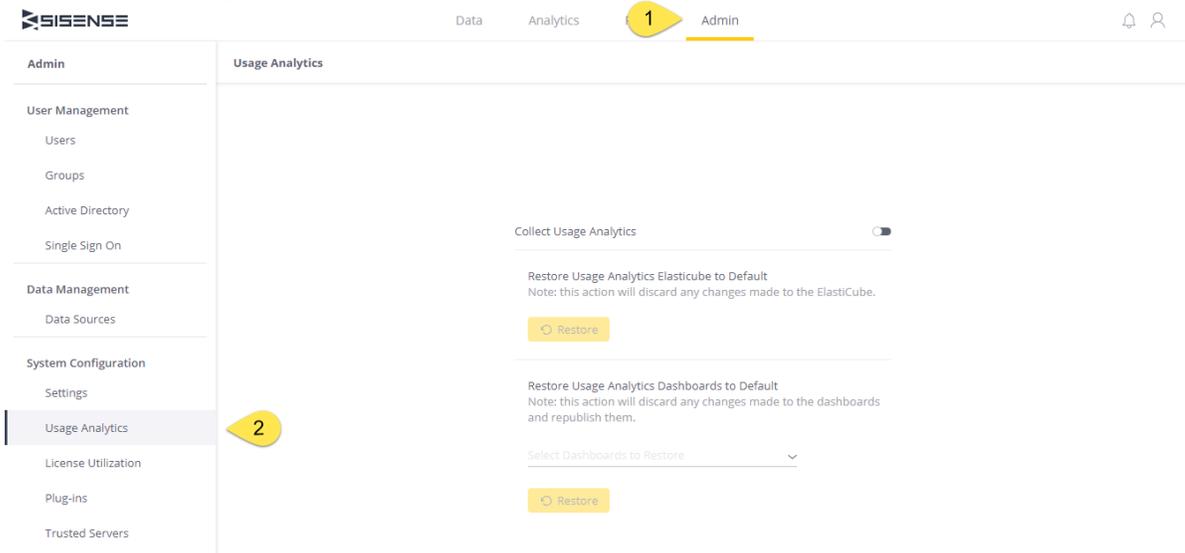
```
mongodb://AppUser:5s5bQGkDPuY6zjxRIUeAgg==@localhost:27018/
```

Test

## Step 2

On the application node, enable Usage Analytics.

1. In the **Admin** page, select Usage Analytics.



2. Toggle **Collect Usage Analytics** to enable.  
Enabling Usage Analytics creates the Usage Analytics Model Elasticube on the build node.

## Step 3

Add the build node as a remote server to the application node.

In the **Admin** page, select **Data Sources** and click **Add Server**.

The screenshot shows the Sisense Admin interface. The top navigation bar includes 'Data', 'Analytics', 'Pulse', and 'Admin'. The left sidebar has 'Admin' selected, with sub-items like 'User Management', 'Data Management', and 'System Configuration'. 'Data Sources' is highlighted in the 'Data Management' section with a yellow callout '1'. The main content area shows 'Data Sources' with a table of local machines and a 'Live Models' section. The 'Add Server' button is highlighted with a yellow callout '2'.

Data Source	Editors	Last Model ...	Last Built	Size On ...	Status	
Sample ECommerce	Everyone (group)	12/4/18 5:54 PM	12/4/18 5:54 ...	71.09MB	Stopped	▶ ↗ ⋮
Sample Healthcare	Everyone (group)	7/2/13 1:07 AM	10/19/17 5:3...	27.59MB	Running	◻ ↗ ⋮
Sample Lead Generation	Everyone (group)	3/3/14 7:46 PM	10/19/17 5:3...	25.54MB	Stopped	▶ ↗ ⋮
Usage Analytics Model	gideon@sisense.corr	12/4/18 1:52 PM	12/4/18 1:52 ...	19.52MB	Stopped	▶ ↗ ⋮

Enter your IP address. Make sure to use the server IP address and not the server name when you add

### Add Server

IP or Server Name

Who can access this server?  Admins Only  Everyone

it.

Then, on the build node, schedule builds for the Usage Analytics Model ElastiCube.

### Schedule Build

ElastiCube: Tutorial

Last Build: 03/04/2018 1:39 PM

Choose one of the scheduling options:



Disable



Daily



Interval

Select the days and time to build your ElastiCube automatically and click **Save**.

- |                                            |                                              |                                               |                                              |
|--------------------------------------------|----------------------------------------------|-----------------------------------------------|----------------------------------------------|
| <input checked="" type="checkbox"/> Monday | <input checked="" type="checkbox"/> Tuesday  | <input checked="" type="checkbox"/> Wednesday | <input checked="" type="checkbox"/> Thursday |
| <input checked="" type="checkbox"/> Friday | <input checked="" type="checkbox"/> Saturday | <input checked="" type="checkbox"/> Sunday    |                                              |

12:00 AM

Save

Cancel

See [Scheduling Builds](#) for more information.

## Step 4

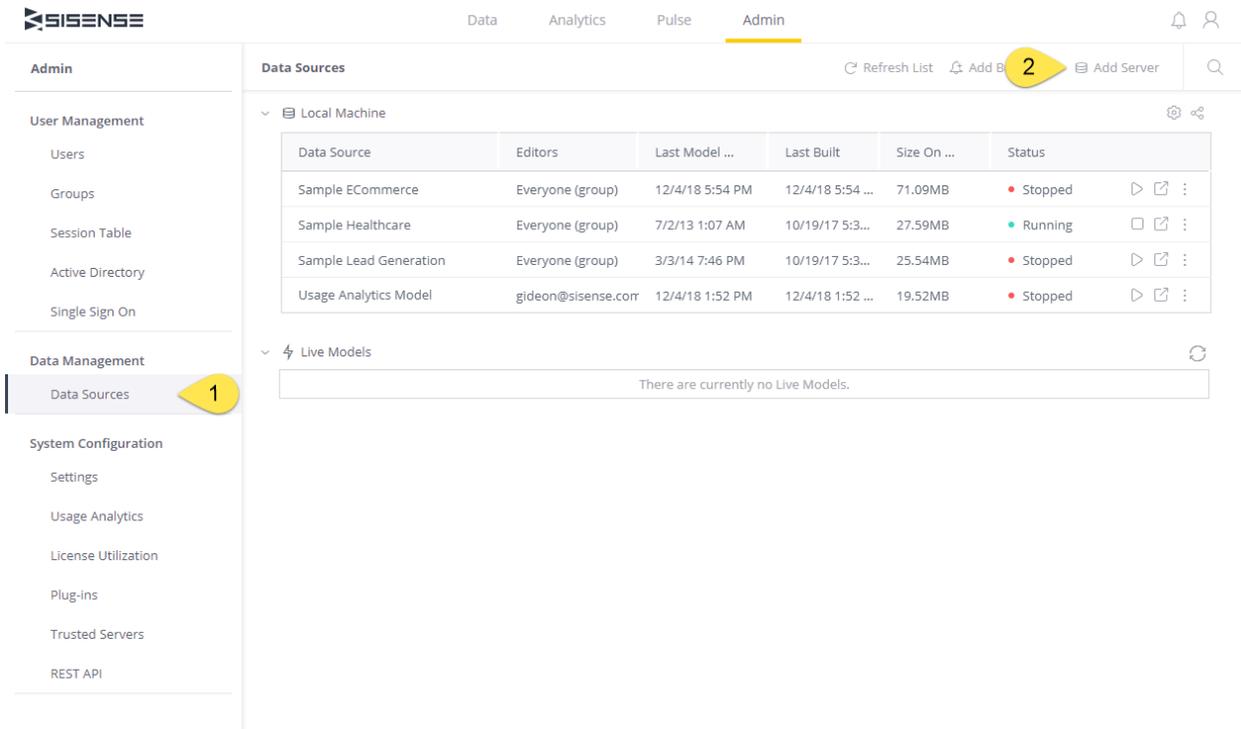
Configure the Orchestrator on your build node to distribute the Usage Analytics Model ElastiCube to the query nodes.

See [Distributing ElastiCube Builds to Query Nodes](#) for more information.

## Step 5

On your application node, add your query nodes as remote servers.

In the **Admin** page, select **Data Sources** and click **Add Server**.



The screenshot shows the Sisense Admin interface. The top navigation bar includes 'Data', 'Analytics', 'Pulse', and 'Admin' (which is selected). The left sidebar contains 'Admin', 'User Management', 'Data Management', and 'System Configuration'. Under 'Data Management', 'Data Sources' is selected and highlighted with a yellow callout '1'. The main content area shows the 'Data Sources' page with a table of local machines and a section for live models. The 'Add Server' button is highlighted with a yellow callout '2'.

Data Source	Editors	Last Model ...	Last Built	Size On ...	Status	
Sample ECommerce	Everyone (group)	12/4/18 5:54 PM	12/4/18 5:54 ...	71.09MB	Stopped	▶ ↗ ⋮
Sample Healthcare	Everyone (group)	7/2/13 1:07 AM	10/19/17 5:3...	27.59MB	Running	◻ ↗ ⋮
Sample Lead Generation	Everyone (group)	3/3/14 7:46 PM	10/19/17 5:3...	25.54MB	Stopped	▶ ↗ ⋮
Usage Analytics Model	gideon@sisense.corr	12/4/18 1:52 PM	12/4/18 1:52 ...	19.52MB	Stopped	▶ ↗ ⋮

Enter your IP address. Make sure to use the server IP address and not the server name when you add it.

### Add Server

IP or Server Name

Who can access this server?

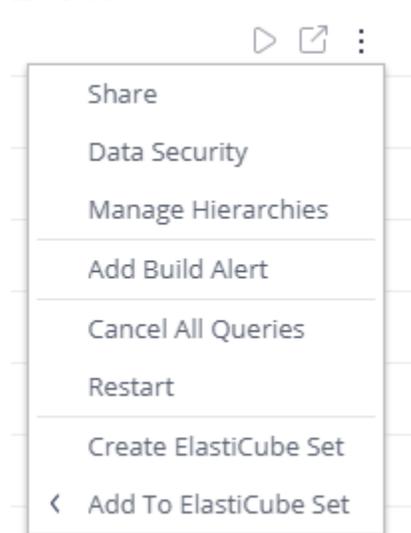
Admins Only  Everyone (i)

Add Server

Cancel

## Step 6

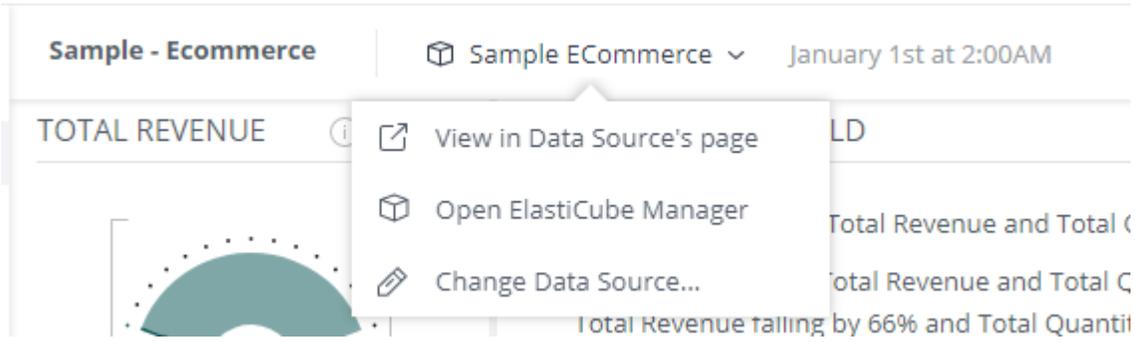
Create ElastiCube Sets for the Usage Analytics Model ElastiCubes for the query nodes.



See [Creating ElastiCube Sets](#) for more information.

## Step 7

Update your default Usage Analytics dashboards to use the Usage Analytics Model ElastiCube Set.



See [Changing a Dashboard's Data Source](#) for more information.

You are now ready to view your Usage Analytics data in a Multi-Node deployment topology.

## Distributing ElastiCube Builds to Query Nodes

### Previous

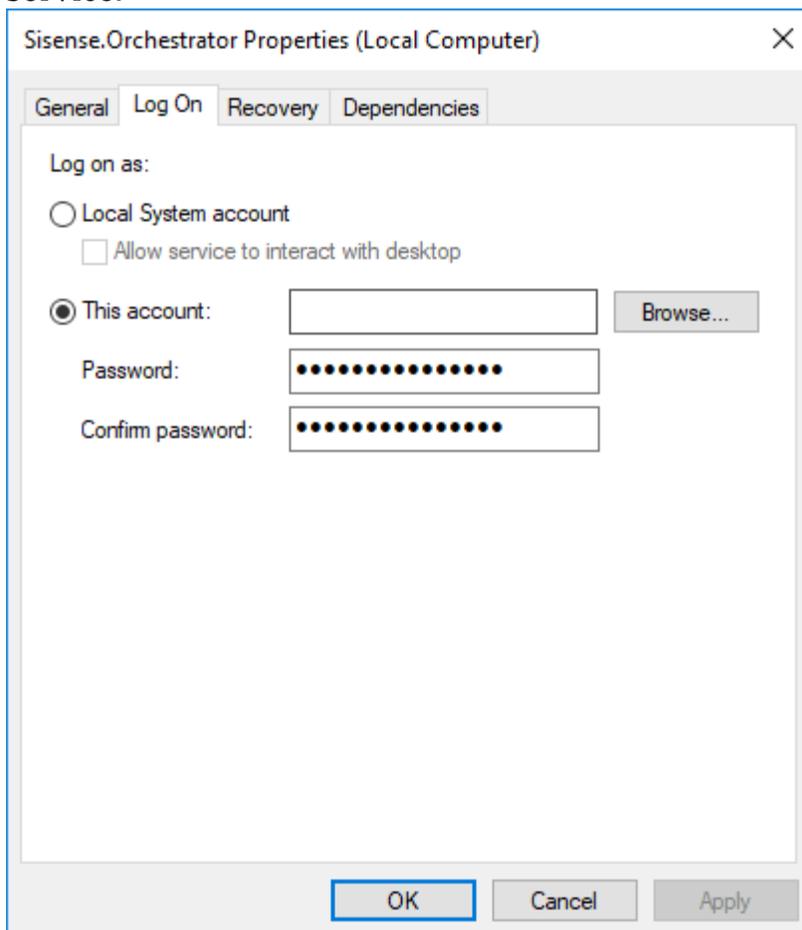
- [Model 4: 2 Application Nodes, 2 Query Nodes, and 1 Build Node](#)

The Sisense Orchestrator Service is an automated service that distributes builds and balances queries across an [ElastiCube Set](#). The Sisense Orchestrator Service is installed during the default installation of Sisense. To distribute ElastiCube builds across all query nodes, you must configure the Sisense Orchestrator Service. When an ElastiCube is building, or receiving a build from the build node, the Sisense Orchestrator Service redirects requests to another available ElastiCube in the set.

After you have set up all of your nodes, you can verify that each query node is working by stopping the Sisense.ECMS service. When this service is stopped, queries should be redirected to the other query nodes in your configuration.

**Note:** After configuring the Sisense Orchestrator Service, the directory `C:\ProgramData\Sisense\PrismServer\ElastiCubeData\` and the relevant ElastiCube folders within it must be shared with permissions for 'Everyone' so Sisense can build or update the ElastiCubes in those folders. Keep in mind that each of your nodes must have sufficient disk space to support the build node.

In addition, the user of the build node must have Administrator access to the service Sisense.Orchestrator on each of the query nodes, so the build node can write to them. You can set this in the properties of the Sisense.Orchstrator service.



The Sisense Orchestrator Service is defined in a JSON file called `config.json` in the `Sisense.Orchestration Config` folder

(C:\ProgramData\Sisense\Sisense.Orchestrator\config.json ) of your Sisense Installation folder of your build node.

The config.json file has three main objects that you must define. The first object is the ElastiCube object that defines which ElastiCube is to be built on the build node and which ElastiCubes are located on query nodes. The second object is the Tasks object that determines how and when the build ElastiCube is to be distributed to the query nodes. The final object that you must define is the Schedule object that defines when an ElastiCube is to be built and distributed. To distribute your ElastiCube builds through the Sisense Orchestrator Service, you define and save the config.json file. After saving the file, the Sisense Orchestrator Service automatically begins to build ElastiCubes in your ElastiCube Set. You can review the progress of the build in the

C:\ProgramData\Sisense\application-logs\orchestrator-service folder in a log file whose file name you define in the JSON file.

The objects that you must include in the config.json file and their descriptions are provided in the table below:

Key	Example	Value
general	<pre>{   "general": {     "logLevel": "INFO",     "emails": [       {         "email": "john.test@sisense.com",         "name": "Example"       }     ]   } }</pre>	The General object contains two objects, logLevel and emails that define what types of logs Sisense generates and where to send them to.
logLevel	<pre>"logLevel": "INFO",</pre>	The type of info returned in the log.

Key	Example	Value
		The possible values you can enter are Info, debug, and Error.
emails	<pre> <b>"emails":</b> [   {     "email": "john.test@sisense.com",     "name": "Example"   } ] </pre>	This object defines who receives an email and their email address when an email is triggered according to the events you define in the mail object.
email	<pre> <b>"emails":</b> [   {     <b>"email":</b> "john.test@sisense.com",     "name": "Example"   } ] </pre>	The email address where emails are to be sent depending on the value of the When key.
name	<pre> <b>"emails":</b> [   {     "email": "john.test@sisense.com",     <b>"name":</b> "Example"   } ] </pre>	The name of the recipient of the email.

Key	Example	Value
cubes	<pre> <b>"cubes":</b>{   "HA_QueryWeb1":{     "ecube": "Sample ECommerce",  "url": "\\127.128.129.91\\C:\\ProgramData\\ Sisense\\PrismServer\\ElasticCubeData\\Sample ECommerce",  "localPath": "C:\\ProgramData\\Sisense\\PrismServer\\ElasticCubeData\\Sample ECommerce"   },   "HA_QueryWeb2":{     "ecube": "Sample ECommerce",  "url": "\\127.128.129.92\\C:\\ProgramData\\ Sisense\\PrismServer\\ElasticCubeData\\Sample ECommerce",  "localPath": "C:\\ProgramData\\Sisense\\PrismServer\\ElasticCubeData\\Sample ECommerce"   } }, </pre>	<p>The cubes object contains all your ElastiCubes in your configuration and their location. As the Sisense Orchestrator Service is installed on the build node, the build cubes are local, so you must define the name of the ElastiCube. For ElastiCubes on a query node, you must define the ElastiCube name, URL of the remote server, and its directory.</p>
build ElastiCubes	<pre> <b>"HA_QueryWeb1":</b>{   "ecube": "Sample ECommerce",  "url": "\\127.128.129.91\\C:\\Prog </pre>	<p>The Build ElastiCube object defines the ElastiCubes that</p>

Key	Example	Value
	<pre>ramData\\ Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce", "localPath":"C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce" },</pre>	<p>will be used as Build ElastiCubes nodes.</p>
ecube	<pre>"HA_QueryWeb1":{     "ecube":"Sample ECommerce", "url":"\\\\127.128.129.91\\C:\\ProgramData\\ Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce", "localPath":"C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce" },</pre>	<p>The name of the ElastiCube.</p>
url	<pre>"HA_QueryWeb1":{     "ecube":"Sample ECommerce", "url":"\\\\127.128.129.91\\C:\\ProgramData\\ Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce", "localPath":"C:\\ProgramData\\Sisen</pre>	<p>The URL address of the ElastiCube server. If the URL is to a secure address, the value should include the username and password</p>

Key	Example	Value
	<pre>se\\PrismServer\\ElasticubeData\\Sample ECommerce"     },</pre>	<p>credentials. For example, "ssh://username:password@10.50.1.128:/C/ecubes". See Scenario 1 for an example.</p>
localPath	<pre>"HA_QueryWeb1":{     "ecube":"Sample ECommerce",      "url":"\\\\127.128.129.91\\C:\\ProgramData\\Sisense\\PrismServer\\ElasticubeData\\Sample ECommerce",      "localPath":"C:\\ProgramData\\Sisense\\PrismServer\\ElasticubeData\\Sample ECommerce"     },</pre>	<p>Directory of the Elasticubes.</p>
tasks	<pre>"tasks":{     "task1":[         {             "build":{                 "cube":[                     "HA_QueryWeb1",                     "HA_QueryWeb2"                 ],                 "queue":[                     "entire"                 ]             }         }     ] }</pre>	<p>The Tasks object contains a task array that defines which Elasticube should be built, the type of build, and to which Elasticubes the</p>

Key	Example	Value
	<pre>       }     }, </pre>	<p>build should be distributed to.</p> <p>In high availability scenarios, running multiple concurrent builds is not recommended and can result in stability issues.</p> <p>Sisense recommends that tasks are spaced out so each build is complete before the next one begins.</p>
task	<pre> "tasks":{   "task1": [     {       "build":{         "cube": [           "HA_QueryWeb1",           "HA_QueryWeb2"         ],         "queue": [           "entire"         ]       }     }   ] } </pre>	<p>The name of the task array.</p> <p>Currently, you can define only one task. Defining multiple tasks in the config.json file may cause builds to fail.</p> <p>In high availability</p>

Key	Example	Value
	<pre> }, </pre>	<p>scenarios, running multiple concurrent builds is not recommended and can result in stability issues. Sisense recommends that tasks are spaced out so each build is complete before the next one begins.</p>
build	<pre> "build": {     "cube": [         "HA_QueryWeb1",         "HA_QueryWeb2"     ], </pre>	<p>The build object defines the ElastiCube to be built and distributed.</p>
cube	<pre> "build": {     "cube": [         "HA_QueryWeb1",         "HA_QueryWeb2"     ], </pre>	<p>The name of the cube to be built.</p>
queue	<pre> "queue": [     "accumulate", </pre>	<p>The value of this key is the type of build</p>

Key	Example	Value
	<pre> "entire" ] </pre>	<p>Sisense should attempt and in what order. There are two possible values delimited by a comma:</p> <p><b>accumulate:</b> Attempts an accumulative build.</p> <p><b>entire:</b> Attempts an entire build.</p> <p><b>schemaChanges:</b> Attempts to update the build only if changes were made to the schema since the previous build. Sisense recommends the following value: "accumulate","entire"</p> <p>In this example, Sisense attempts to do an accumulative build first, and if it fails, Sisense attempts an entire</p>

Key	Example	Value
		<p>build. If the build fails for all builds, Sisense retries again according to the next scheduled build.</p>
<p>distribute</p>	<pre> {     "distribute": [         "HA_QueryWeb1",         "HA_QueryWeb2"     ] }, </pre>	<p>Determines which cubes the latest build should be distributed to. The value should be the ElastiCube name for your ElastiCubes on your query nodes. For example, ["cube1, cube2, cube3"]. You can add multiple ElastiCubes delimited by a comma separator.</p>
<p>reattach</p>	<pre> "reattach": {     "deleteOldDbfarm": true,     "cube": [         "HA_QueryWeb1",         "HA_QueryWeb2"     ] }, </pre>	<p>Reattach is an object that contains two objects, DeleteOldDbfarm and Cube.</p>

Key	Example	Value
delete OldDb farm	<pre> "reattach":{   "deleteOldDbfarm":true,     "cube":[       "HA_QueryWeb1",       "HA_QueryWeb2"     ]   } }, </pre>	<p>A boolean value that determines if an old ElastiCube is to be deleted after it has been updated.</p>
cube	<pre> "reattach":{   "deleteOldDbfarm":true,     "cube":[       "HA_QueryWeb1",       "HA_QueryWeb2"     ]   } }, </pre>	<p>The name of the ElastiCube to be reattached or not.</p>
mail	<pre> {   "mail":"build" } </pre>	<p>Indicates when an email alert is triggered. Email alerts can be triggered for the following scenarios:</p> <p><b>None:</b> No email is ever sent.</p> <p><b>Build:</b> An email is sent when a build is complete regardless</p>

Key	Example	Value
		<p>if the build was successful or not.</p> <p><b>Fail:</b> An email is sent when a build fails.</p>
scheduler	<pre>"scheduler": [   {     "task": "task1",     "schedule": "15 14 * * *",     "enabled": true   } ]</pre>	<p>The Scheduler object contains an array of tasks that define when a task is to be task is to be initiated.</p>
task	<pre>"scheduler": [   {     "task": "task1",     "schedule": "15 14 * * *",     "enabled": true   } ]</pre>	<p>Defines which task to perform and the order. Currently, Sisense only supports one task.</p>
schedule	<pre>"scheduler": [   {     "task": "task1",     "schedule": "15 14 * * *",     "enabled": true   } ]</pre>	<p>The time that a task is to be initiated in Cron format. Some examples: -To run a build each night at midnight, enter the value "0 0 * * * *" -To run a build</p>

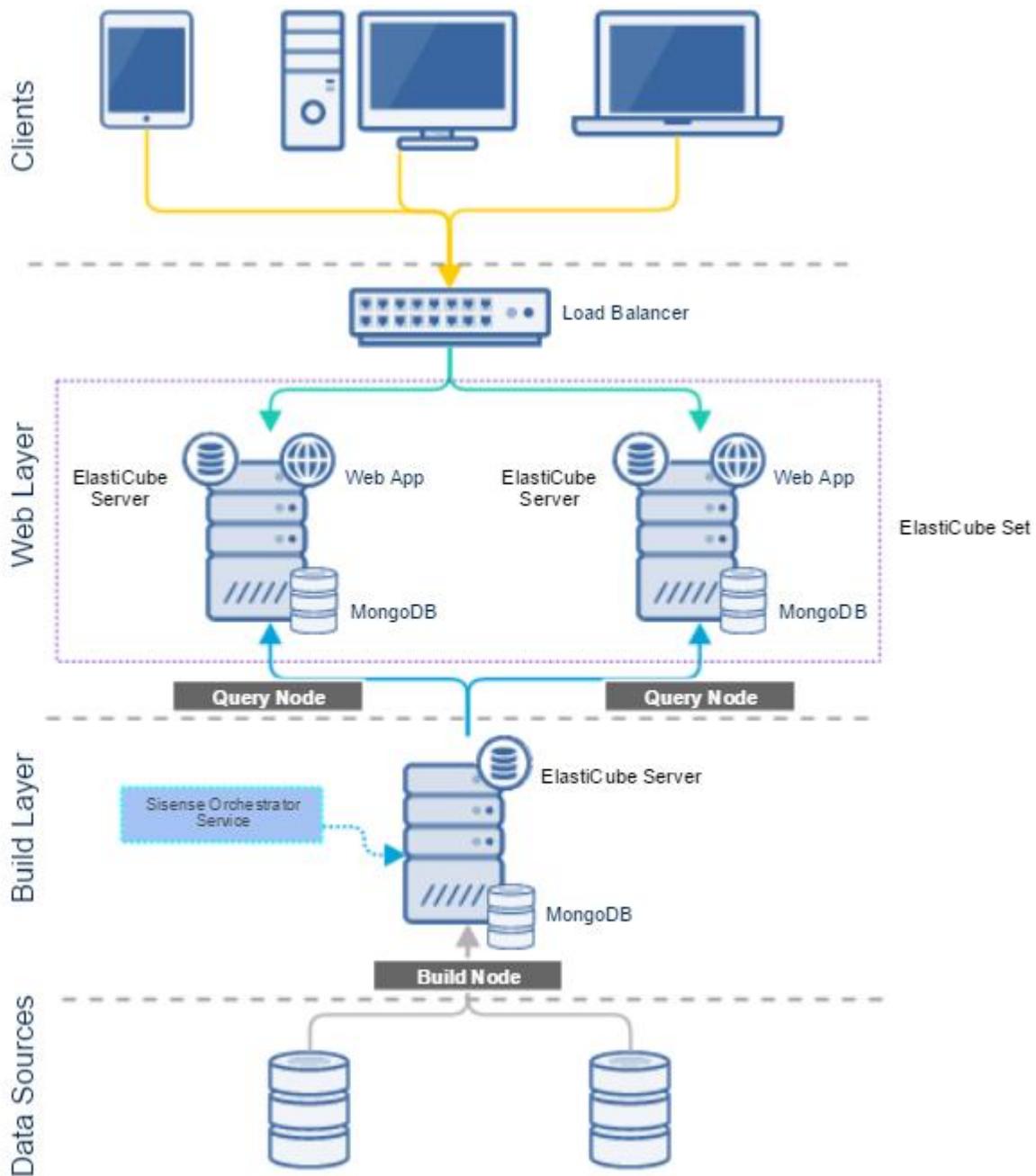
Key	Example	Value
		every hour, enter the value "0 * * * * *" See the full Cron format reference <a href="#">here</a> .
enabled	<pre>"scheduler": [   {     "task": "task1",     "schedule": "15 14 * * *",     "enabled": true   } ]</pre>	A boolean value that indicates if the task is to be executed or not.

# Use Cases

The following section presents two use cases and an example config.json file that demonstrates how to support these use cases. For assistance with additional HA configurations, contact your Customer Service Manager.

## Scenario 1 - Single Stack

In this example, there is one build node and two query nodes. Each query node is hosted on a separate machine while the components that make up the query node are hosted on the same machine as a single application stack.



In this example, the build cube is called “Sample ECommerce”. This ElastiCube is distributed to two ElastiCube servers defined in the cube1 and cube2 objects. The build cube is distributed first to cube1 and then to cube2 as defined in the tasks object. The first time the Sisense Orchestrator Service tries to build the BuildTest1 ElastiCube, it attempts an accumulative build and if that fails, an entire build. After each build is replicated, the previous version of the ElastiCube

is deleted as specified in the reattach object. This task takes place hourly. If the distribution fails, Sisense sends an email to johndoe@Sisense.com with a debug log attached.

```
{
  "general":{
    "logLevel":"INFO",      //Determines type of log
events recorded
    "emails":[
      {
        "email":"john.test@sisense.com",    //Who
to send reports
        "name":"Example1"
      }
    ]
  },
  "cubes":{
    "localCubeBuildTest1":{
      "ecube":"Sample ECommerce"      //Name of the
ecube to be distributed
    },
    "cube1":{
      "ecube":"Sample ECommerce",    //Name of the
ecube on the query node

"url":"\\\\127.128.129.91\\C:\\ProgramData\\Sisense\\Pr
ismServer\\ElasticubeData\\Sample ECommerce",

"localPath":"C:\\ProgramData\\Sisense\\PrismServer\\Ela
stiCubeData\\Sample ECommerce"
    },
    "cube2":{
      "ecube":"Sample ECommerce",    //Name of the
ecube on the query node

"url":"\\\\127.128.129.92\\C:\\ProgramData\\Sisense\\Pr
ismServer\\ElasticubeData\\Sample ECommerce",

"localPath":"C:\\ProgramData\\Sisense\\PrismServer\\Ela
```

```

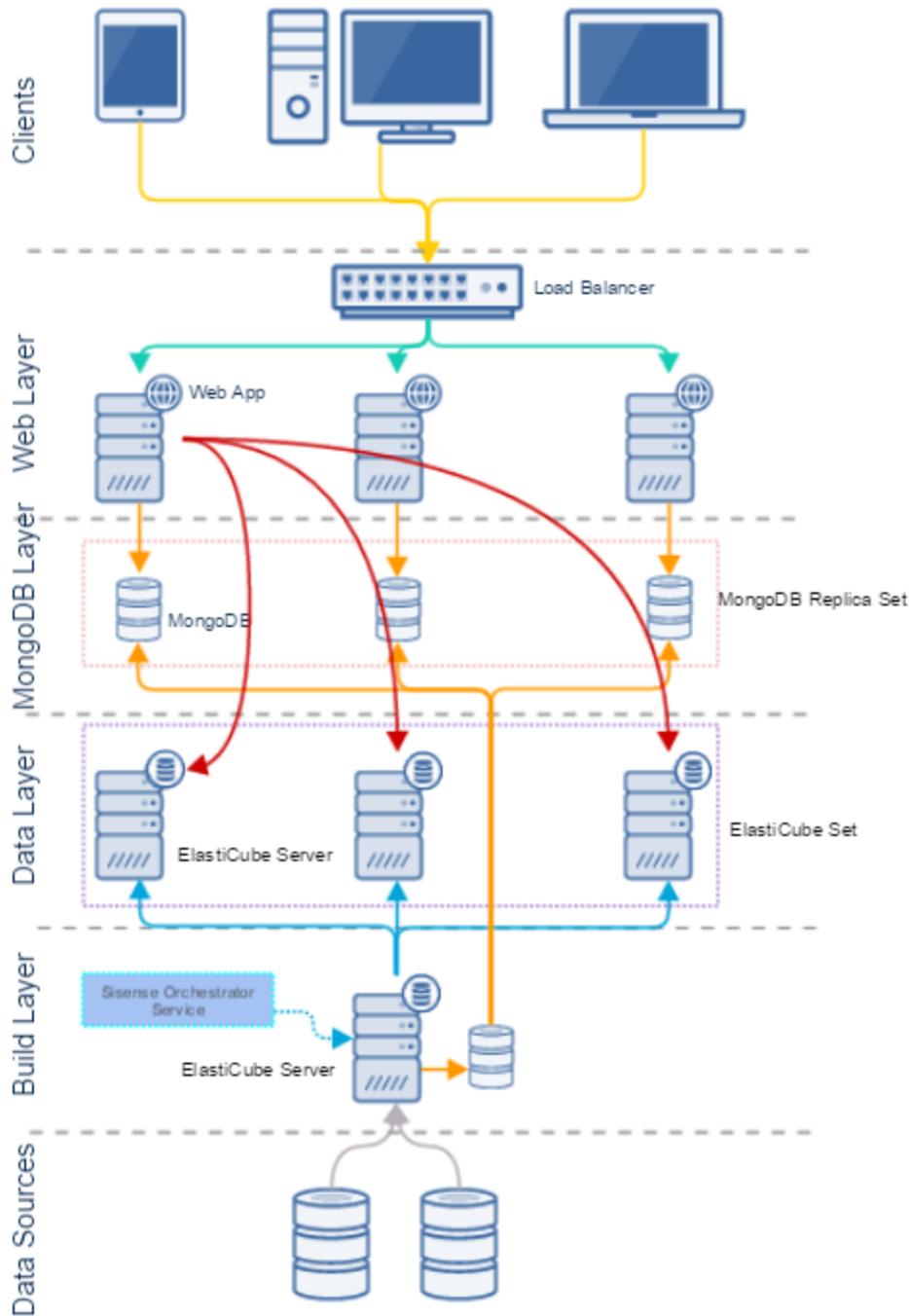
stiCubeData\\Sample ECommerce"
    }
  },
  "tasks":{
    "task1":[
      {
        "build":{
          "cube":[
            "cube1",
            "cube2"
          ],
          "queue":[
            "accumulate", //The first type of
build to be attempted
            "entire" //The second type of build to
perform if the first fails
          ]
        }
      },
      {
        "distribute":[ //The order of how ecubes
on the build node is distributed to query nodes
          "cube1",
          "cube2"
        ]
      },
      {
        "reattach":{
          "deleteOldDbfarm":true,
          "cube":[
            "cube1",
            "cube2"
          ]
        }
      },
      {
        "mail":"build" //What event triggers an
email
      }
    ]
  }
}

```

```
    },
    "scheduler": [
      {
        "task": "task1",    //The name of the task
        object that is to be executed
        "schedule": "0 * * * *",    //How often to
        perform a task in Cron format
        "enabled": true
      }
    ]
  }
}
```

#### **Scenario 2 - Distributed Stack**

In this example, there are three query nodes in which each component is located in a different remote location. This scenario, while costly to implement, demonstrates a distributed application stack configuration where each component is replicated and located on a separate server independent of the other components. If any component fails, the rest of the architecture remains intact. For example, the web server is on one server, the ElastiCubes are stored on another server, and the MongoDB is located on another server.



The build cube in this example is called "BuildTest1". This ElastiCube is distributed to three remote ElastiCube servers defined in the cube1, cube2, and cube3 objects. The build cube is distributed first to cube1, then cube2, and then cube3 as defined in the tasks object. The URLs are standard URLs for cube1 and

cube2 while the URL for cube3 is secured through SSH. When the URL is secured, you must provide the URL and the required credentials to access it.

The first time the Sisense Orchestrator Service tries to build the BuildTest1 ElastiCube, it attempts an accumulative build and if that fails, an entire build. If both builds fail, Sisense attempts to update the schema only. After each build is replicated, the previous version of the ElastiCube is deleted as specified in the reattach object. This task takes place At 14:15 on the 1st of every month. After a build is successful, Sisense sends an email to johndoe@Sisense.com.

```
{
  "general":{
    "logLevel":"DEBUG",
    "emails":[
      {
        "email":"john.test@sisense.com",
        "name":"Example1"
      }
    ]
  },
  "cubes":{
    "localCubeBuildTest1":{
      "ecube":"Sample ECommerce"
    },
    "cube1":{
      "ecube":"Sample ECommerce",

"url":"\\\\127.128.129.91\\C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce",

"localPath":"C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce"
  },
  "cube2":{
    "ecube":"Sample ECommerce",

"url":"\\\\127.128.129.92\\C:\\ProgramData\\Sisense\\PrismServer\\ElastiCubeData\\Sample ECommerce",
```

```

"localPath":"C:\\ProgramData\\Sisense\\PrismServer\\Ela
stiCubeData\\Sample ECommerce"
    },
    "cube3":{
        "ecube":"Sample ECommerce",

"url":"\\\\127.128.129.92\\C:\\ProgramData\\Sisense\\Pr
ismServer\\ElastiCubeData\\Sample ECommerce",

"localPath":"C:\\ProgramData\\Sisense\\PrismServer\\Ela
stiCubeData\\Sample ECommerce"
    }

},
"tasks":{
    "task1":[
        {
            "build":{
                "cube":[
                    "cube1",
                    "cube2",
                    "cube3"
                ],
                "queue":[
                    "accumulate",
                    "entire",
                    "schemachanges"
                ]
            }
        },
        {
            "distribute":[
                "cube1",
                "cube2",
                "cube3"
            ]
        },
        {
            "reattach":{

```

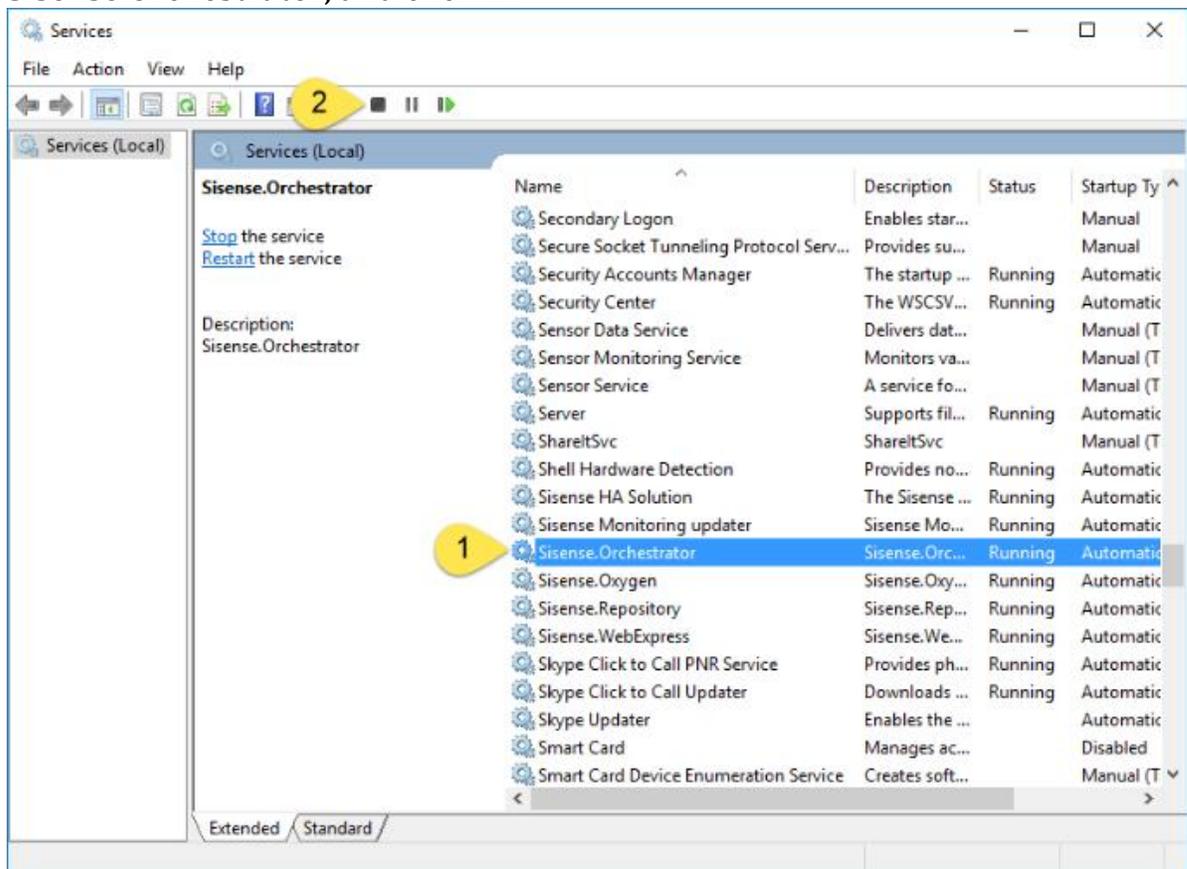
```
        "deleteOldDbfarm":true,
        "cube":[
            "cube1",
            "cube2",
            "cube3"
        ]
    },
    {
        "mail":"fail"
    }
]
},
"scheduler":[
    {
        "task":"task1",
        "schedule":"15 14 1 * *",
        "enabled": true
    }
]
}
```

## Add Query Nodes

In some configurations, replicating components can improve performance by scaling out and preventing any potential bottlenecks. Each component you replicate must be added to the ElastiCube Set and the config.json file. You can continue to add query nodes according to your requirement so long as the nodes are included in the config.json file.

### To add a query node:

1. Stop the Sisense Orchestrator Service. Open Windows Services, select Sisense.Orchestrator, and click .



2. In the Sisense Web Application, click **Admin** and select the **Data Sources** tab on the left.
3. Hover over the ElastiCube you want to add to the ElastiCube Set and click on the menu that appears. Click **Add to ElastiCube Set** and select the set that you want to add the ElastiCube to.
4. Edit the config.json located at Sisense/Sisense.Orchestration/Config/ and add the new ElastiCube to the Cubes object.

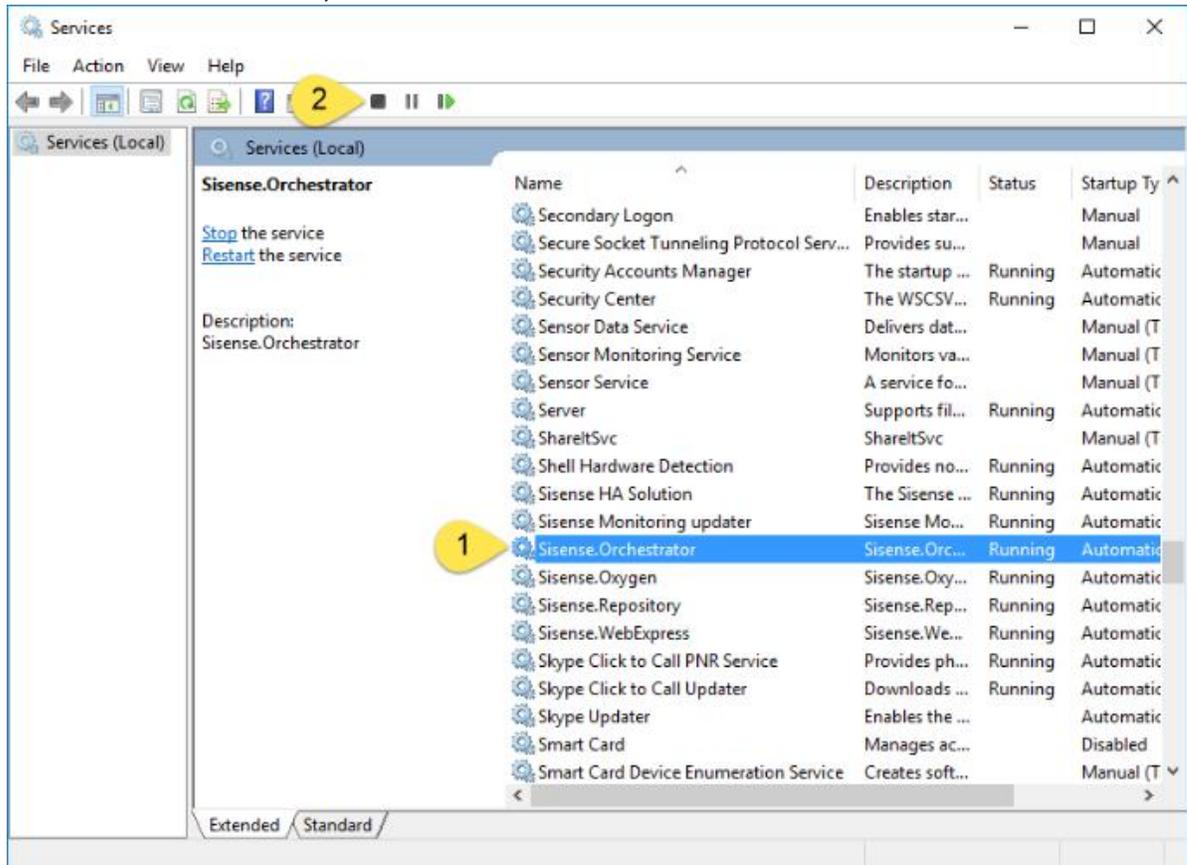
5. Save the config.json file.
6. Restart the Sisense Orchestrator Service.

## Remove Query Nodes

You can remove query nodes by removing them from the ElastiCube Set and the config.json file.

To remove a query node:

1. Stop the Sisense Orchestrator Service. Open Windows Services, select Sisense.Orchestrator, and click .



2. In the Sisense Web Application, click **Admin** and select the **Data Sources** tab on the left.
3. In the ElastiCube Set table, click and clear the checkboxes of the ElastiCubes to be removed. Click **Save** after you have selected the relevant ElastiCubes.
4. Edit the config.json located at Sisense/Sisense.Orchestrator/Config/ and delete the relevant ElastiCubes from the Cubes object.
5. Save the config.json file.
6. Restart the Sisense Orchestrator Service.

## Next Steps

- [Setting Up ElastiCube Sets](#)

## Setting Up ElastiCube Sets

### Previous Steps

- [Distributing ElastiCubes](#)

Sisense ElastiCube Sets are collections of ElastiCube models with identical schemas that allow you to query running ElastiCubes within the ElastiCube Set while other ElastiCubes are building.

**Note:** From Sisense V6.5 onwards, you can query ElastiCubes during accumulative builds without ElastiCube Sets. However, to ensure that your users can query ElastiCubes after a failed build, you can implement ElastiCube Sets to handle your queries while Sisense rebuilds the failed ElastiCube.

Grouping ElastiCubes into ElastiCube sets has several benefits:

- Allowing viewers to query the most up-to-date cubes within an ElastiCube Set.
- Reducing server load during builds by using multiple nodes in which only the non-building node is queried.
- Achieving data redundancy by running builds interchangeably across multiple nodes.

To maximize availability and reduce build and dashboard response times, administrators can distribute resource allocation across multiple Sisense nodes. By adding ElastiCubes across multiple servers to a single ElastiCube Set, administrators can configure ElastiCube Sets to run builds interchangeably via the Sisense Orchestrator Service so the most up-to-date and complete ElastiCubes are queried.

The Sisense Orchestrator Service is an automated service that can be configured to synchronize builds across the ElastiCube Set. If you change your schemas

within any of the ElastiCubes, you must manually change the schemas in all of the ElastiCubes and rebuild the cube entirely.

This page provides the following information:

1. Instructions on [Creating](#) and [Deleting ElastiCube Sets](#).
2. A walkthrough for how you can create a set and configure the Sisense Orchestrator Service. See [Working with ElastiCube Sets](#).
3. (Optional) Instructions on how to implement and configure the Sisense Orchestrator Service. See [Activating the Sisense Orchestrator Service](#).
4. How to update an ElastiCube within an ElastiCube Set.

# Creating ElastiCube Sets

From the **Admin** page, Administrators can create ElastiCube Sets.

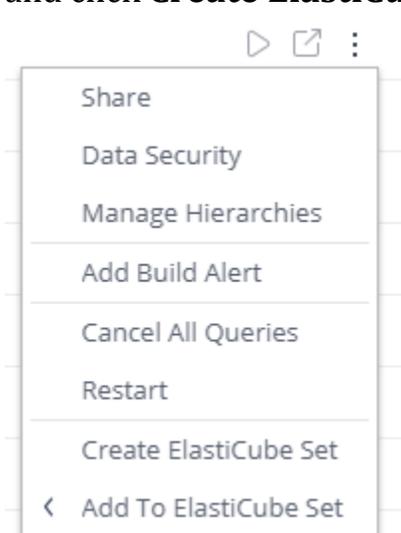
ElastiCube Sets require at least two ElastiCubes. The Administrator can add multiple ElastiCubes to a set, however, ElastiCubes can only be part of a single ElastiCube Set at a time. After an Administrator creates an ElastiCube Set or adds an ElastiCube to an existing set, the ElastiCube is disabled in the original server and no further actions can be performed outside of the set.

Sisense recommends that you develop ElastiCubes in a development environment or separate ElastiCube to ensure they work before adding them to an ElastiCube Set.

**Note:** Build nodes used in high availability configurations should not be included in your ElastiCube Set. If you have implemented high availability, only query nodes should be included in the ElastiCube Set.

## To create an ElastiCube Set:

1. Click **Admin** and select the **Data Sources** tab on the left.
2. For the relevant ElastiCube you want to add to the ElastiCube Set, select : and click **Create ElastiCube Set**.



3. In the Create ElastiCube Set window, enter a name for the set and select the ElastiCubes you want to include in the set.

4. From the Routing Strategy list, select a strategy for how you want to route traffic to your Elasticube Set. See [Routing Queries in Elasticube Sets](#) for more information.
5. From the Failover Elasticube list, select which Elasticube you prefer to use if the other Elasticubes in the set are not available. Failover Elasticubes only handle queries when an Elasticube that is part of the Elasticube Set fails.(Note: this feature is available in Sisense V6.7 onwards).

### Create Elasticube Set

Elasticube Set Name

Routing Strategy

Single Elasticube

Elasticubes

Search Elasticubes...	
<input type="checkbox"/> Facebook Ads Demo	4.01MB
<input type="checkbox"/> new	39.46MB
<input type="checkbox"/> Sample ECommerce	59.09MB
<input type="checkbox"/> Sample Healthcare	1.56MB
<input checked="" type="checkbox"/> Sample Lead Generation	2.25MB
<input type="checkbox"/> Training	814.28KB

1 Elasticube Selected

Failover Elasticube

None

Cancel

6. Click **Create**.
7. In the Existing Dashboards popup message, indicate if you want to associate your Elasticube's dashboards with the Elasticube Set or maintain the

association with the original ElastiCube.

Existing Dashboards in added ElastiCubes

Connect existing dashboards to new ElastiCube Set

Leave existing dashboards with their current ElastiCube

8. Click **OK**. The ElastiCube Set is created and displayed in the **Admin** page. After you have created an ElastiCube Set, you can configure the ElastiCube [Set's Hierarchies](#), [Access Rights](#), and [Data Security](#).

## Adding ElastiCubes to ElastiCube Sets

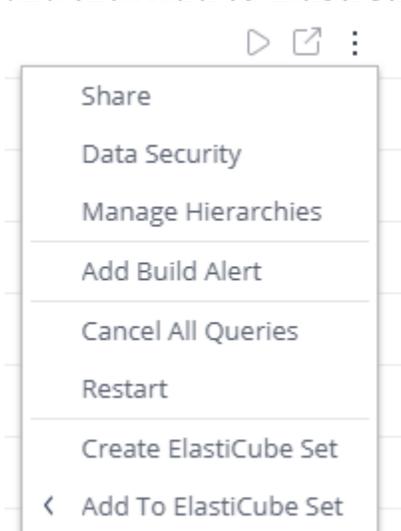
After you have created an ElastiCube Set, you can add more ElastiCubes to the set.

When you add an ElastiCube to an existing ElastiCube Set, any dashboards associated with that ElastiCube are passed on to the ElastiCube Set.

**Note:** You cannot add 'localhost' to an ElastiCube Set. Instead, you can add your localhost as a new server with its IP address as the server name and then add ElastiCubes from it to an ElastiCube Set.

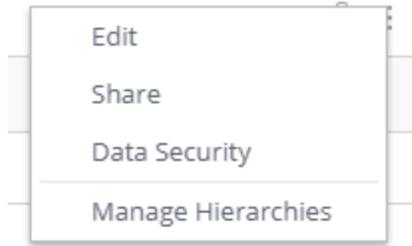
### To add an ElastiCube to the ElastiCube Set:

1. Click **Admin** in the upper right corner and select the **ElastiCubes** tab on the left.
2. For the relevant ElastiCube you want to add to the ElastiCube Set select **:** and click **Add to ElastiCube Set**.



OR

In the ElastiCube Set table, click **:** and select **Edit** and select the ElastiCubes to be added to the set. Click **Save** after you have selected the relevant ElastiCubes.



The ElastiCube is added to the ElastiCube Set and displayed in the ElastiCube Set table on the **Admin** page.

## Deleting ElastiCube Sets

Administrators can delete ElastiCube Sets. Deleting an ElastiCube Set removes the set from the **Admin** page and any dashboards associated with that set.

### **To delete an ElastiCube Set:**

1. Hover over its title and click the Trash  icon.
2. Click **Delete** to confirm that you want to delete the ElastiCube Set.

# Working with ElastiCube Sets

This section describes how to create your first ElastiCube Set and configure it for high availability.

In this example, let's assume that you have an ElastiCube, ElastiCube A, and you want to add it to a set to ensure that the data can always be queried, even during a build.

The first step is to create a copy of ElastiCube A. The ElastiCubes can have different names, but the schemas must be identical.

**Note:** If your ElastiCubes are located on a single node, the ElastiCubes cannot have the same name.

In Sisense, we will save a new version of ElastiCube A as ElastiCube B. In Sisense, you can see both ElastiCubes are identical (Select **Admin > Data Sources**)

Local Machine

ElastiCube	Status	Created On	Last Built	Size On Disk	
Elastic Cube A	Running	3/3/16 9:11 AM	3/3/16 9:11 AM	39.42MB	Open  
Elastic Cube B	Running	3/3/16 9:11 AM	3/3/16 9:14 AM	39.42MB	Open  

Now, let's create an ElastiCube Set that contains both ElastiCubes. Click : > **Create ElastiCube Set**. For more information, see [Creating ElastiCube Sets](#). Select the ElastiCubes to be added to the set. For this set, we will select ElastiCube A and B, which have different names, but identical schemas.

# Create ElastiCube Set

ElastiCube Set Name

ElastiCube Set Example

ElastiCubes

Search ElastiCubes... 

<input checked="" type="checkbox"/>	Elastic Cube A	39.42MB
<input checked="" type="checkbox"/>	Elastic Cube B	39.42MB
		⌵
		⌵
		⌵
		⌵
		⌵
		⌵

2 ElastiCubes Selected

CREATE

CANCEL

Click **Create** and confirm that you want to create a new set with these ElastiCubes. The set and its ElastiCubes are now displayed at the top of the **Admin** page.

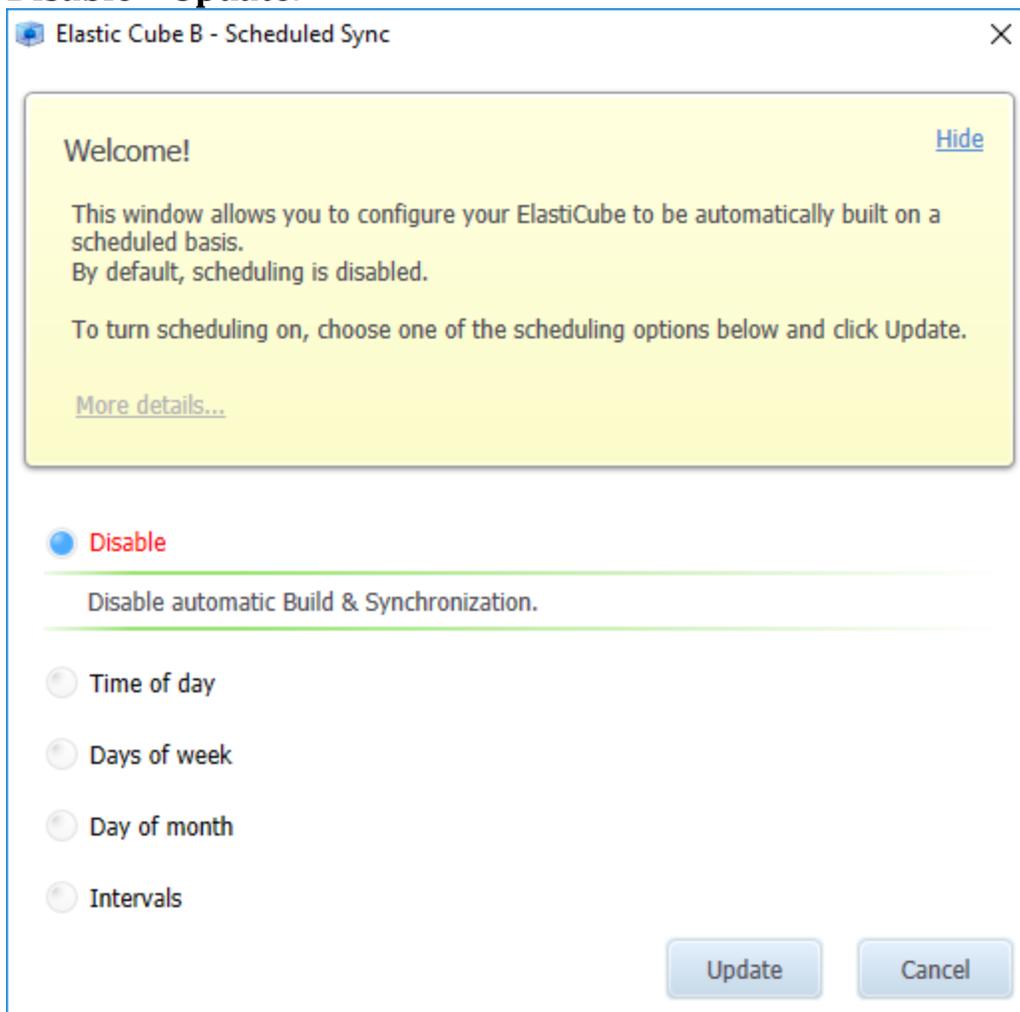
 ElastiCube Example [Manage Hierarchies](#) | [Access Rights](#) | [Data Security](#) |  | 

Server	ElastiCube	Status	Created On	Last Built	Size On Disk	
LocalHost	Elastic Cube A	<span style="color: green;">●</span> Running	3/3/16 9:11 AM	3/3/16 9:11 AM	39.42MB	<a href="#">Open</a> 
LocalHost	Elastic Cube B	<span style="color: green;">●</span> Running	3/3/16 9:11 AM	3/3/16 9:14 AM	39.42MB	<a href="#">Open</a> 

Now that the set is built, it's time to schedule automatic builds with the Sisense Orchestrator Service. This service automatically builds the ElastiCubes according to your preferences. For example, you can schedule the builds to run every 10 minutes or at midnight when queries are at their fewest.

To schedule automatic builds, first you should disable any scheduled builds you have for the ElastiCubes. In Sisense in the top menu, select **ElastiCube** >

**Schedule Build Settings** for each ElasticCube with a scheduled build, then select **Disable > Update**.



Now that all the ElasticCubes' scheduled builds are disabled, it's time to configure the Sisense Orchestrator Service to build the ElasticCubes within the set. To configure the Sisense Orchestrator Service, you must modify the Config.JSON file that was installed when you installed the latest version of Sisense. This file defines how the ElasticCube Set is being built, the order of the build, and how frequently a build occurs.

This file is located in the Sisense.Orchestration Config folder (/Sisense.Orchestration/Config/) of your Sisense Installation folder.

For this set, let's assume that you want to schedule an accumulative build for midnight. Within the **elastiCubes** object, we listed both ElastiCubes to be built by the Sisense Orchestrator Service. The order of the ElastiCube builds is in the order of oldest to newest. Next, in the **timer** object, we listed the hour when the build is to take place. The value is defined in Cron format, so midnight equals 0 minutes 0 hours "0 0 \* \* \* \*".

Finally, we define the flow of the build. We want an accumulative build, but if for some reason this fails, we want to run an entire build as a back-up. Remember, when an accumulative build fails, the ElastiCube needs to be completely rebuilt to ensure that the data is intact. In the build object, we define the type of builds to be initiated and their order. The order of the values determines what build type is initiated first. In this example, the value is "FullUpdateExisting","Full", which means run an accumulative build first and if that fails, run a full build. For a complete explanation on how to define the rest of the objects list in the Config.JSON file, see [Activating the Sisense Orchestrator Service](#).

Now, save the file and Sisense automatically begins to run the builds according to the schedule defined in the timer object.

```
{
  "servers": {
    "localhost": {
      "ip": "localhost",
      "port": "8081",

      "apiKey": "eyJ0eXAiHDHDHsXKSSZJdnUzI1NiJ9.eyJ1c2VyIjoiNT
      Y1N332RW2eygL0uVLufGYC-8vY"
    }
  },
  "tasks": [
    {
      "name": "taskName",
      "elasticsearch": [
        {
```

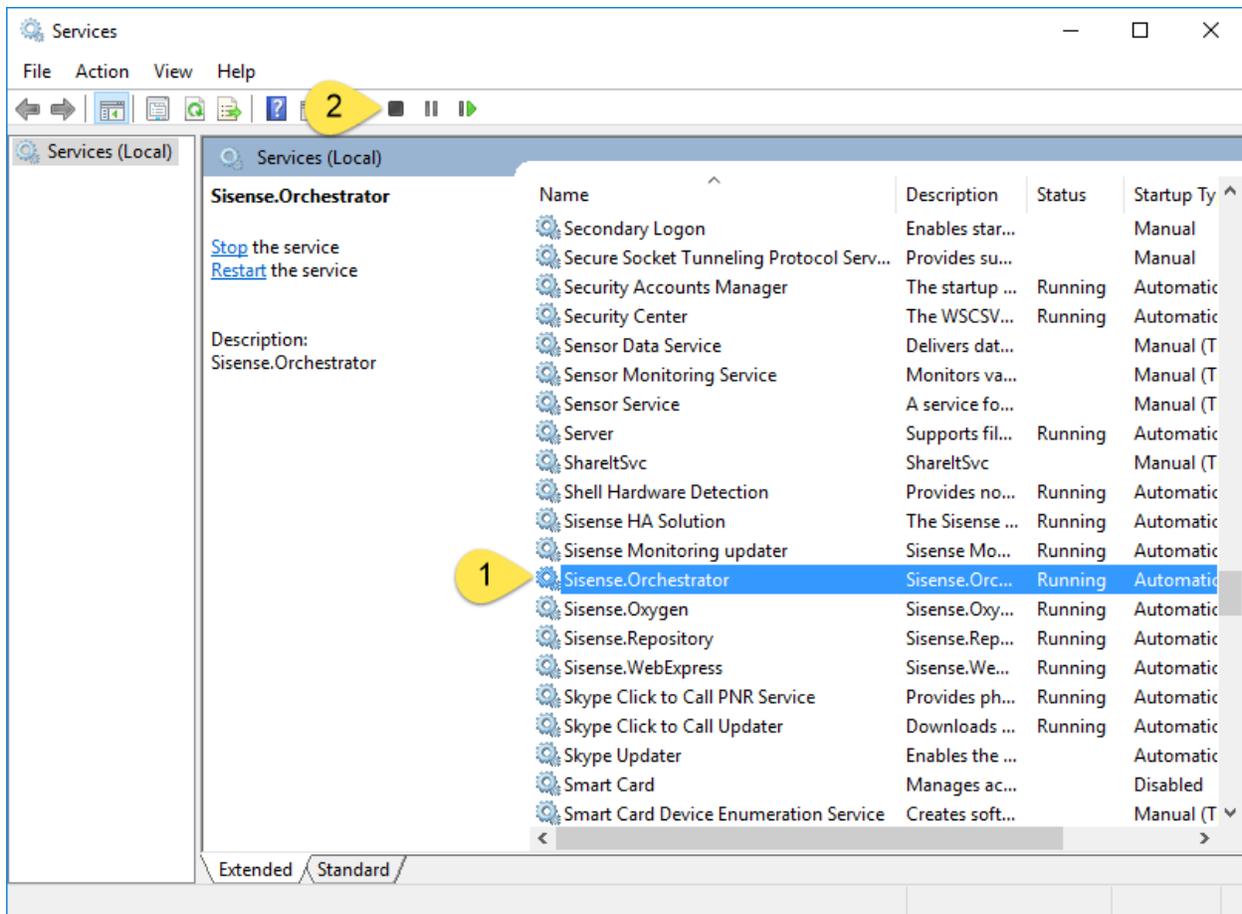
```

        "server":"localhost",
        "cube":"ElastiCube A"
    },
    {
        "server":"localhost",
        "cube":"ElastiCube B"
    }
],
"timer":"0 0 * * *",
"build":{
    "queue":[
        "FullUpdateExisting",
        "Full"
    ],
    "flow":"sync"
},
"email":{
    "when":"Fail"
}
}
],
"emails":[
    {
        "email":"JohnDoe@sisense.com",
        "name":"John Doe"
    }
],
"logLevel":"info"
}

```

The Sisense Orchestrator Service automatically builds the ElastiCubes within the set, but if the schema changes, you must manually rebuild each ElastiCube within the set so the schemas match across all the ElastiCubes. While working with the schemas of an ElastiCube that is part of an ElastiCube Set, you should stop the Sisense Orchestrator Service.

To stop the Sisense Orchestrator Service, open Windows Services, select Sisense.Orchestrator, and click .



After the ElastiCubes are rebuilt, restart the Sisense Orchestrator Service by selecting Sisense.Orchestrator and click ▶.



The ElastiCube Set is now configured and running ElastiCubes can be queried while the other ElastiCubes are being built.

# Activating the Sisense Orchestrator Service

The Sisense Orchestrator Service automates the build process for ElastiCube Sets and determines how frequently the build process occurs.

To activate the Sisense Orchestrator Service, you must create and save a JSON file called config.json in the Sisense.Orchestration Config folder

(/Sisense.Orchestration/Config/) of your Sisense Installation folder. This file defines how the ElastiCube Set is being built, the order of the build, and how frequently a build occurs.

After you save the file, the Sisense Orchestrator Service automatically begins to build ElastiCubes in your ElastiCube Set as defined by you in the config.json file.

You can review the progress of the build in the /Sisense.Orchestration/Logs folder in a log file whose file name you define in the JSON file.

Note: During this phase it is recommended to stop the Sisense Orchestrator Service while importing data and reactivate it after the importing process is completed.

The objects that you must include in the config.json file and their descriptions are provided in the table below:

Key	Value
Servers	The servers block contains a server object that defines each server that hosts an ElastiCube within your ElastiCube Set. Each server must include a server name, IP, port, and a unique API key. For example, if you have two different servers, each server must be represented as a separate object with its own name, IP, and port.
ServerName	Your server's name. The value of this key is used when associating the ElastiCube Set with a server in the value of

Key	Value
	the Tasks key.
IP	The IP address of your server.
Port	The port of your web server.
API	The access token used to identify your requests to the Sisense web server. For more information about the access token, see Retrieving a Token.
Tasks	The Tasks object contains an array of ElastiCube objects. This object can contain multiple ElastiCube objects, but a minimum of two are required for a set.
Name	The name of the plan. The value of this parameter is the filename of the log file for the Sisense Orchestrator Service.
ElastiCubes	The ElastiCube object that holds an array of ElastiCubes. Each ElastiCube must include the name of its server and the name of the cube as defined in the Elastic Manager.
Server	The name of the server that hosts the ElastiCube.
Cube	The name of the cube to be included in the set.
Timer	The frequency of builds in Cron format. Some examples: -To run a build each night at midnight, enter the value "0 0 * * *"-To run a build every hour, enter the value "0 * * * *" See the full CRON format reference <a href="#">here</a> .
Build	The Build object contains two keys, Queue and Flow, which determine the order in which the build is completed and the type of build.
Queue	The value of this key is the type of build Sisense should

Key	Value
	<p>attempt and in what order. There are two possible values delimited by a comma: FullUpdateExisting: Attempts an accumulative build. Full: Attempts an entire build. Sisense recommends the following value: "FullUpdateExisting", "Full" In this example, Sisense attempts to do an accumulative build first, and if it fails, Sisense attempts an entire build. If the build fails for all builds, Sisense retries again according to the next scheduled build.</p>
Flow	<p>Determines the order of the build. A build occurs according to the value of Timer. For example, if your builds are scheduled for midnight, then the Orchestator begins to build the ElastiCubes at midnight according to the order you define below. Single: Builds a single cube, then waits for the next time a build is scheduled according to the value of Timer. The cube to be built is a cube that previously failed or the cube with the oldest build. Sync: Builds the cubes with the oldest build first and then the remaining cubes in the order of oldest to newest until all ElastiCubes are updated. All: Builds the cube with the oldest build first and then the remaining cubes asynchronously.</p>
Email	<p>This object determines what triggers activate Sisense to send an email to users you define in the Emails block</p>
When	<p>Indicates when an email alert is triggered. Email alerts can be triggered for the following scenarios: None: No email is ever sent. Build: An email is sent when a build is</p>



```
GSSDFVfasddgdeygL0uVLufDFeY"
  }
},
"tasks":[
  {
    "name":"taskName",
    "elasticubes":[
      {
        "server":"Server1",
        "cube":"ElastiCube1"
      },
      {
        "server":"Server2",
        "cube":"ElastiCube2"
      }
    ],
    "timer":"*/10 * * * *",
    "build":{
      "queue":[
        "FullUpdateExisting",
        "Full"
      ],
      "flow":"sync"
    },
    "email":{
      "when":"Fail"
    }
  }
],
"emails":[
  {
    "email":"JohnDoe@sisense.com",
    "name":"John Doe"
  }
],
"logLevel":"info"
}
```

## Overcoming Build Failures in ElastiCube Sets

In Sisense, if an accumulative build fails, you must rebuild the ElastiCube that failed entirely to ensure the validity of the ElastiCube.

If you use the Sisense Orchestrator Service, Sisense recommends that you define the value of the **Queue** key as “FullUpdateExisting, Full”

```
"queue": ["FullUpdateExisting", "Full"]
```

In this configuration, the Sisense Orchestrator Service attempts to build the set using accumulative builds and in the event of a failure, a full build is initiated.

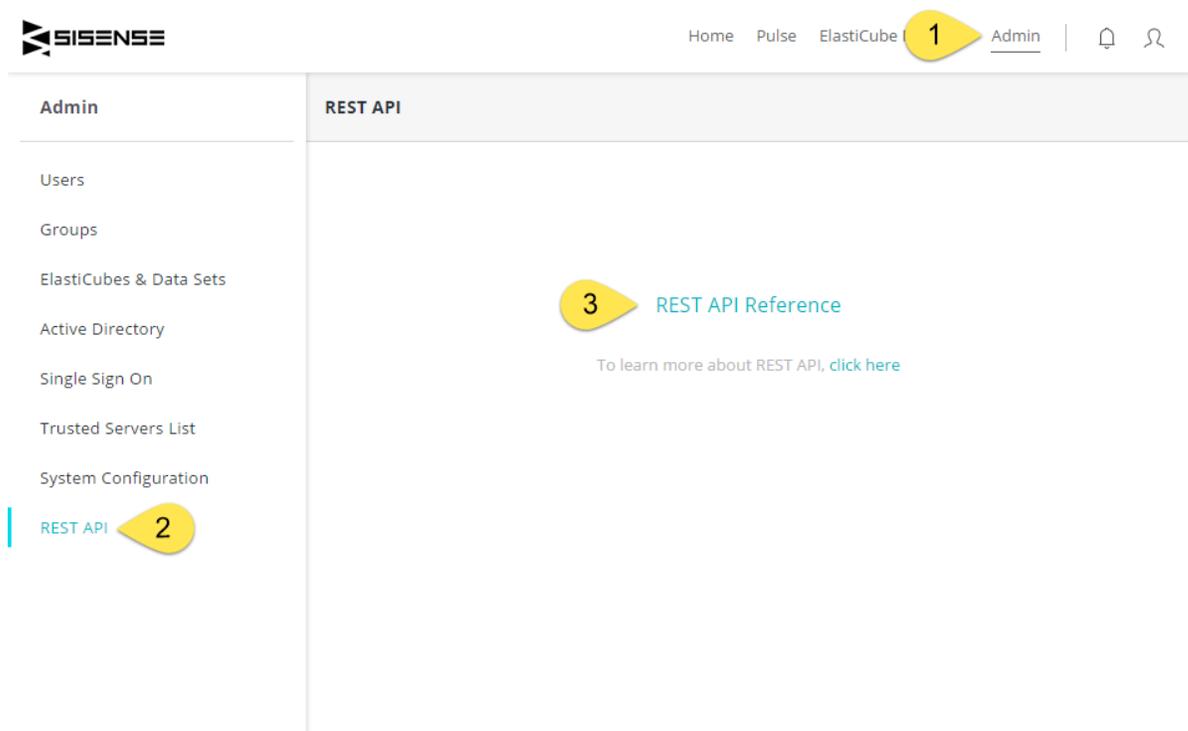
## Retrieving a Token

As part of the config.json file, you must provide a token that Sisense uses for authenticating your ElastiCube Sets.

You can retrieve this token through Sisense's API documentation. After you retrieve the token, save it as the value of apiKey in the config.json file.

### To retrieve a token:

1. In Sisense, access the API documentation, select **Admin** > **REST API** > **REST API Reference**.



2. In the REST API Reference site, select Version 1.0 in the top-right corner of the page.
3. Open the authentication method.

4. Enter your login credentials and click **Run**. The access\_token is displayed in the Response Body below.

authentication Show/Hide | List Operations | Expand Operations

**POST** /authentication/login Authenticate and receive token

**Implementation Notes**  
The login endpoint validates passed credentials and returns an API token for subsequent requests to the API.

**Response Class (Status 200)**  
Model | Model Schema

```
{
  "success": true,
  "message": "string",
  "access_token": "string",
  "profile": {}
}
```

Response Content Type:

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
x-device-id	<input type="text"/>	The device id.	header	string
username	<input type="text" value="(required)"/>	User's username or email	formData	string
password	<input type="text" value="(required)"/>	User's password in plain text	formData	string
localeId	<input type="text"/>	User's locale, as ISO language code (See <a href="#">table</a> )	formData	string

**Response Messages**

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized: Invalid credentials. Wrong username or password.		

**RUN**

5. Copy the token and paste it into the value of apiKey in the config.json file.

[Hide Response](#) RUN

**Curl**

```
curl -X POST --header "Content-Type: application/x-www-form-urlencoded" --header "Accept: application/json" -d "username=gideon%40"
```

**Request URL**

```
http://localhost:8081/api/v1/authentication/login
```

**Response Body**

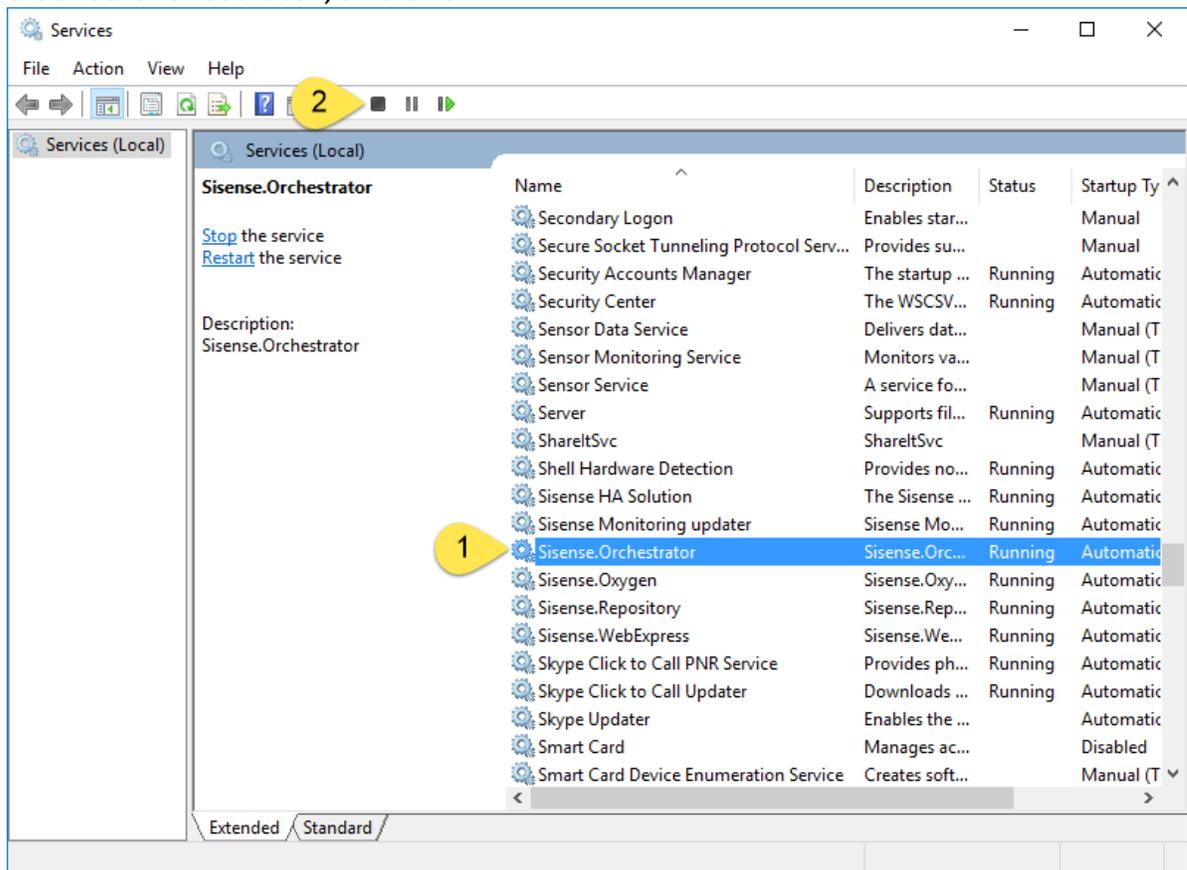
```
{
  "success": true,
  "message": "",
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiaWNTZi0wM3YTFiZGQ2YmViNDY1MDNjZWNIiwiYXBPu2VjcmV0IjoiaWNTkMDU3N",
  "profile": {}
}
```

## Updating ElastiCubes within an ElastiCube Set

The Sisense Orchestrator Service automatically updates your data across ElastiCubes within an ElastiCube Set. If you want to modify the schema, for example, by adding new tables, you must manually rebuild the ElastiCubes and then import that ElastiCube to the other machines hosting your ElastiCube Set. While rebuilding your ElastiCube, you should deactivate the Sisense Orchestrator Service to prevent it from updating the ElastiCubes with different schemas. After the schemas are identical for all the ElastiCubes across the ElastiCube Set, you can reactivate the Sisense Orchestrator Service to automatically update your ElastiCubes.

### To update an ElastiCube within ElastiCube Set:

1. Stop the Sisense Orchestrator Service. Open Windows Services, select Sisense.Orchestrator, and click ■.



2. Apply your schema changes to the ElastiCube and run a full build.
3. Export the ElastiCube and import it on your other machines. For more information, see [Importing and Exporting ElastiCube Data](#).
4. In Windows Services, restart the Sisense Orchestrator Service by selecting Sisense.Orchestrator and clicking ▶.

## Next Steps

- [Securing the Message Broker's Communication](#)

## Routing Queries in ElastiCube Sets

When a user queries an ElastiCube that is part of an ElastiCube Set, Sisense routes the query to a single ElastiCube with the latest ElastiCube build. By default, ElastiCube Sets operate in active-passive mode. In active-passive mode, one ElastiCube, by default the ElastiCube with the latest build, handles incoming queries. This ElastiCube is active while the remaining ElastiCubes are on standby in passive mode.

Sisense also supports active-active mode where queries are spread across available ElastiCubes within an ElastiCube Set. In active-active mode, each ElastiCube that is not currently building can handle queries.

You can modify how queries are routed to ElastiCubes within ElastiCube Sets by selecting one of the following routing strategies:

**Single:** (Default). ElastiCubes operate in active-passive mode where the ElastiCube with the freshest data receives all the queries while the remaining ElastiCubes are on standby.

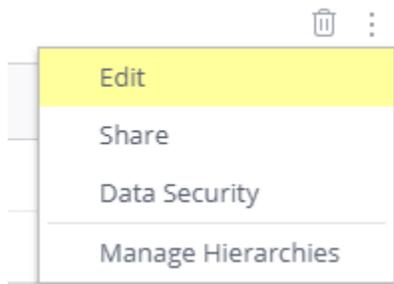
**Multiple ElastiCubes – Fresh Data:** ElastiCubes operate in active-active mode where queries are spread across all ElastiCubes with the latest data.

**Multiple ElastiCubes – Best Spread:** ElastiCubes operate in active-active mode where queries are spread across the largest group of ElastiCubes with identical data.

**To set a query routing strategy:**

1. Click **Admin** and select the **Data Sources** tab on the left.

2. For the relevant ElastiCube Set, select **Edit**.



3. From the Routing Strategy list, select the relevant strategy.
4. Click **Save**.

## Securing the Message Broker's Communication

**Note:** The procedure described below is an optional step for hardening the security of the Sisense Message Broker.

The message broker is responsible for communicating events across Sisense components. To ensure that your communication is secure, RabbitMQ, which Sisense uses as the message broker, supports Transport-Layer-Security (TLS) connections. TLS encrypts traffic between Sisense components and verifies that various Sisense components are authentic.

The communication is encrypted through the use of certificates. You are responsible for generating your certificate bundle, which includes the Certificate Authority (CA file), a certificate (public key) file and a private key file. To encrypt and decrypt traffic, you must provide Sisense with certificate/key pairs in the Sisense Configuration Manager. The following article explains how you can generate TLS certificates.

### **To enable TLS-connections for the Sisense message broker:**

1. Open the Configuration Manager in your browser located at <http://localhost:3030/>.

2. Under the section Message Broker, toggle **TLS Enabled** to enable TLS support.

#### Message Broker

Connection String

amqp://127.0.0.1:5672

Test

TLS Enabled



CA Certificate

Place CA Certificate Here

Private Certificate

Place Private Certificate Here

Private Key

Place Private Key Here

3. Enter the following details:

**Connection String:** The connection string to the Sisense message broker. The RabbitMQ uses the [AMQP URI Specification](#) for connections.

**CA Certificate:** Enter your CA certificate. This is provided when you generate your certificate bundle.

**Private Certificate:** Enter your private certificate. This is provided when you generate your certificate bundle.

**Private Key:** Enter your private key. This is provided when you generate your certificate bundle.

4. Click **Save**.

## Overview

Sisense Usage Analytics is a set of pre-defined dashboards and ElastiCube that enable you to monitor your Sisense user and dashboard activity. With Sisense Usage Analytics, you can better understand how users are interacting with Sisense dashboards and optimize your configuration accordingly. For example, you can understand which dashboards are making an impact and how fast the dashboards are loading.

Sisense collects usage data for the following actions:

- Opening a dashboard
- Changing a filter
- Selecting areas (filter)
- Drilling down into widgets
- Exporting a dashboard to PDF
- Exporting a dashboard to images
- Adding KPIs to Pulse

The data displayed in your Usage Analytics dashboards is collected once Usage Analytics is enabled. When you first enable the feature, there will only be a small amount of data. Data is accumulated while the feature is enabled, and stored on your Sisense Server for 30 days.

Sisense Usage Analytics includes the following dashboards:

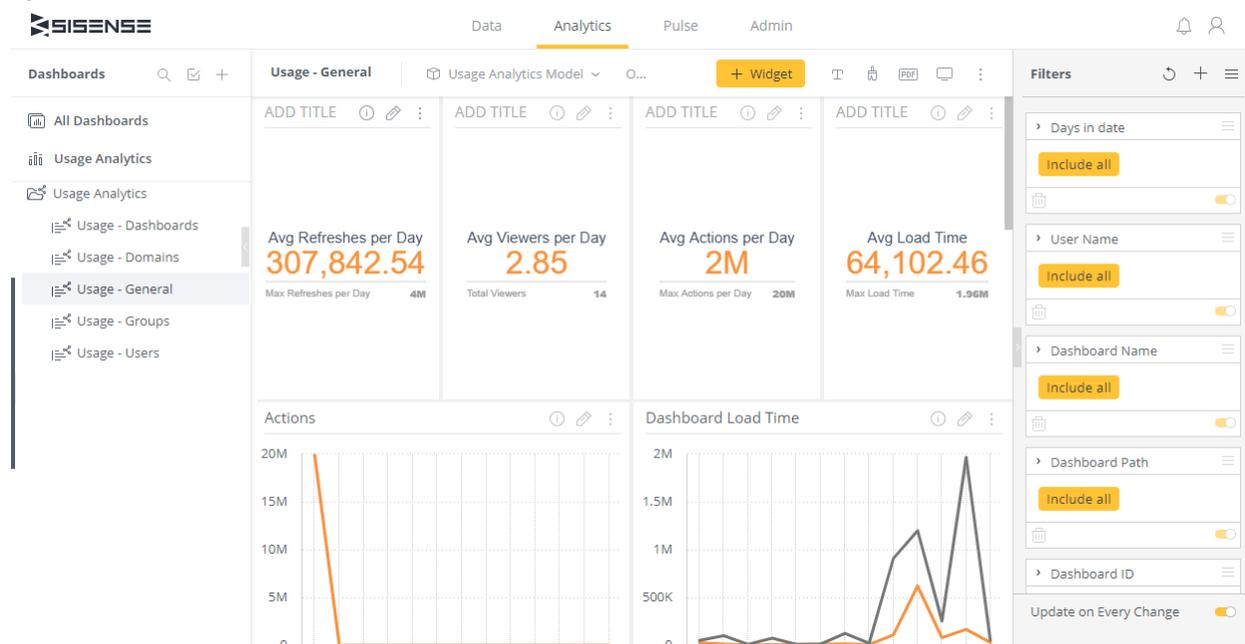
**Usage - Dashboards:** Provides insights regarding the number of dashboards in your system, how often they are used, and their performance.

**Usage - Domains:** Provides insights regarding the behavior of your users, from different email domains in your system. This is useful for OEMs who have tenants from different email domains. From this dashboard you can monitor system usage for each tenant.

**Usage - General:** Provides a general summary of dashboard performance, dashboard usage, and who is viewing your dashboards.

**Usage - Groups:** Provides insights regarding group activity in your system and their dashboard usage.

**Usage - Users:** Provides insights regarding the behavior of specific users in the system.



The Usage Analytics dashboards included with Sisense are visible after you [activate](#) the Usage Analytics feature in the **Admin** page. After activating Usage Analytics, Sisense displays the default Usage Analytics ElastiCube in the **Data** page and your default Usage Analytics dashboards in the **Analytics** page.

You can fully customize the default ElastiCube and dashboards as you like.

**Note:** The Usage Analytics ElastiCube is built on a CSV file where your data is stored. This CSV should not be modified in any way or the ElastiCube will not build. To update the Usage Analytics data, you should perform a full build. If you modify your ElastiCube and it no longer builds, or if you ever need to revert back to the original ElastiCube and dashboards, you can do so through the **Admin** page or through the REST API. See [Restoring Usage Analytics](#) for more information.

If you have implemented a multi-node environment and want to view usage data across all your nodes, there are additional steps you must take to set it up. See [Supporting Usage Analytics in Multi-Node Environments](#) for more information.

**Note:** Groups with ';' in their name will lead to incorrect data being displayed in your Usage Analytics as this causes the groups to be parsed as two separate groups.

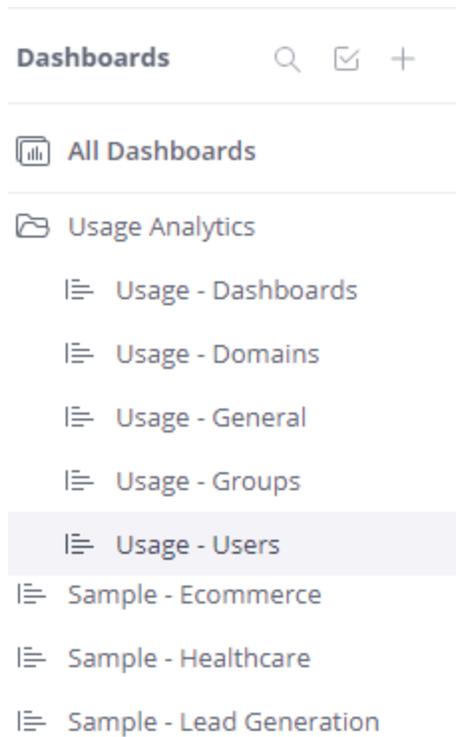
## Related Topics

- [Activating Usage Analytics](#)
- [Dashboard Analytics](#)
- [Domain Analytics](#)
- [General Analytics](#)
- [Groups Analytics](#)
- [Users Analytics](#)
- [Restoring Usage Analytics](#)
- [Supporting Usage Analytics in Multi-Node Environments](#)

## Activating Usage Analytics

There are two steps to activating [Usage Analytics](#). The first step is to enable Usage Analytics from the **Admin** page, and the second step is to build the model called **Usage Analytics Model** that is automatically created when Usage Analytics is enabled.

Once Usage Analytics is enabled, your server's activity is recorded on your Sisense Server. Enabling Usage Analytics adds a Usage Analytics ElastiCube to the **Data** page and a Usage Analytics folder to your Dashboards list where all the pre-defined usage dashboards are stored.



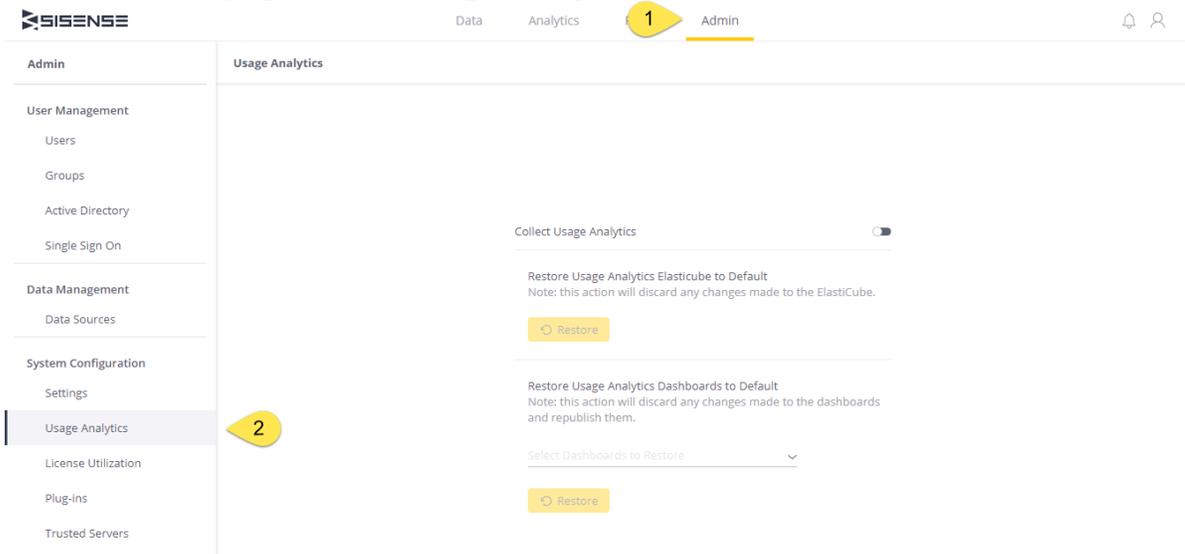
These dashboards are initially empty until you complete the second step, which is to build the pre-defined Usage Analytics model. After you build your Usage Analytics model, your Usage Analytics dashboards will display the updated usage data. The data displayed in your Usage Analytics dashboards is collected when Usage Analytics is enabled. All the data displayed in your Usage Analytics is stored on your Sisense Server for 30 days.

If you have implemented a multi-node environment and want to view usage data across all your nodes, there are additional steps you must take to set it up. See [Supporting Usage Analytics in Multi-Node Environments](#) for more information.

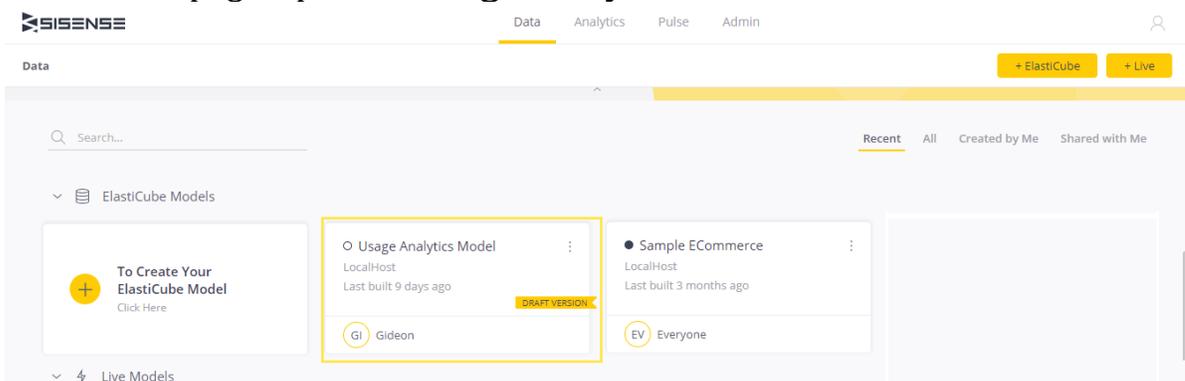
**Note:** To display the latest usage analytics data, you need to perform a full build on the Usage Analytics model. You can automate this step by [scheduling periodic builds](#).

**To activate Usage Analytics:**

1. In the **Admin** page, select Usage Analytics.



2. Toggle **Collect Usage Analytics** to enable. Sisense automatically adds the **Usage Analytics Model** to your **Data** page.
3. In the **Data** page, open the **Usage Analytics Model**.

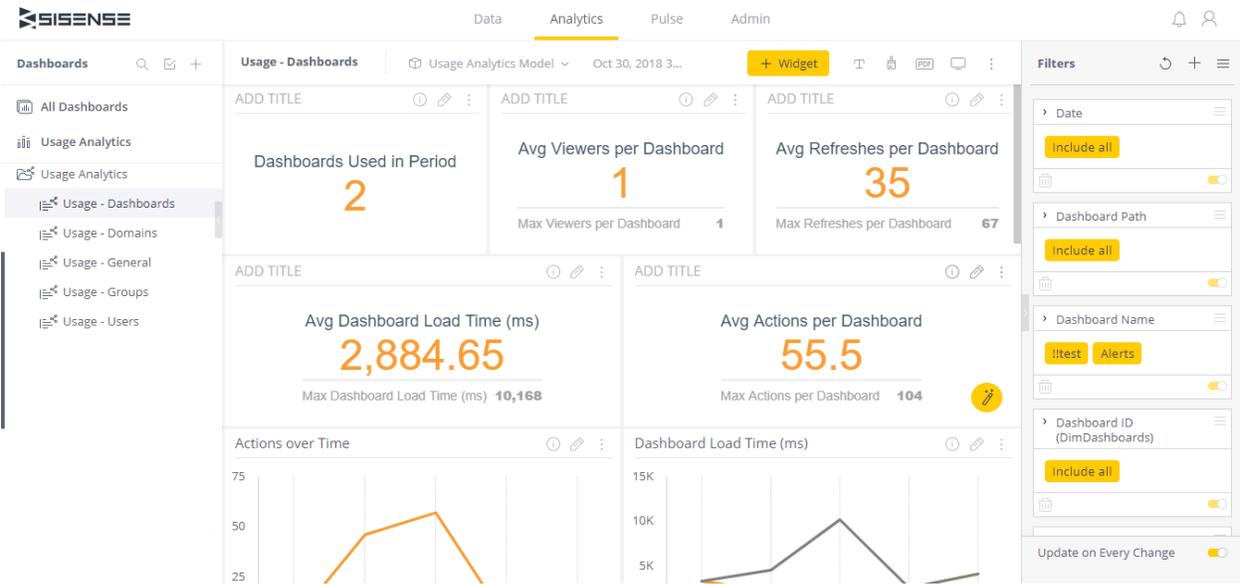


4. Build the Usage Analytics Model. Your Usage Analytics dashboards display all the information collected since you began collecting usage analytics data. To update your dashboards with the latest data, you need to run a full build each time.

## Dashboard Analytics

The Usage - Dashboard dashboard provides you with the ability to see how often your dashboards are being used, how often Viewers are interacting with them, and how quickly your dashboards are loading.

You can filter the information displayed in your widgets with a set of filters on the right. With these filters, you can filter the details about your dashboard by various parameters, such as a specific dashboard or user or group.



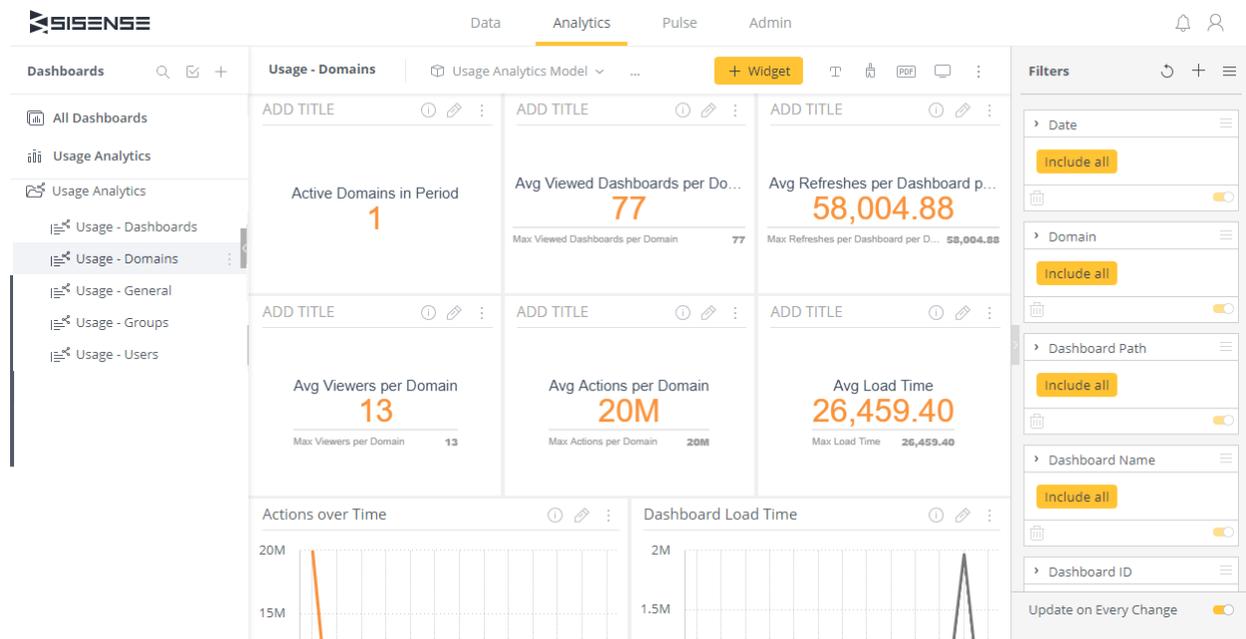
This dashboard is useful for answering the following questions:

Question	Widget
How many dashboards have been used?	Dashboards Use in Period
How many Viewers have been looking at my dashboards?	Avg Viewers per Dashboard
How many times have my dashboards been refreshed?	Avg Refreshes per Dashboard, Dashboard Refreshes and Viewers
How long does it take to load a dashboard?	Average Dashboard Load Time, Avg Action per Dashboard, Dashboard Load Time
How much are Viewers interacting with my dashboards?	Avg Actions per Dashboard, Dashboard Actions, Dashboard Usage

Question	Widget
Who is looking at my dashboards?	Dashboard Viewers

## Domain Analytics

Domain Analytics provide you with insights regarding users from different email domains active in your account. A domain represents each unique email domain of your users in your Sisense system. For example, if you have users with the email addresses john@acme.com and sally@sisense.com, these represent two different domains. This dashboard is useful for OEMs who want to monitor and track their tenants' activity in their Sisense system.



The Usage - Domain dashboard contains data regarding how many domains were active for the recorded period of time, the number of Viewers per domain, how often they are using your dashboards, and what type of dashboard load performance your customers are experiencing.

You can filter the information displayed in your widgets with a set of filters on the right. With these filters, you can filter the details displayed in your dashboard by each domain.

With the Usage - Domain dashboard, you can keep track of how many users you have per domain. For example, in the Users per Domain and Number of Dashboard Viewers per Domain tables, you can view how many users your customers have, and monitor their license utilization.

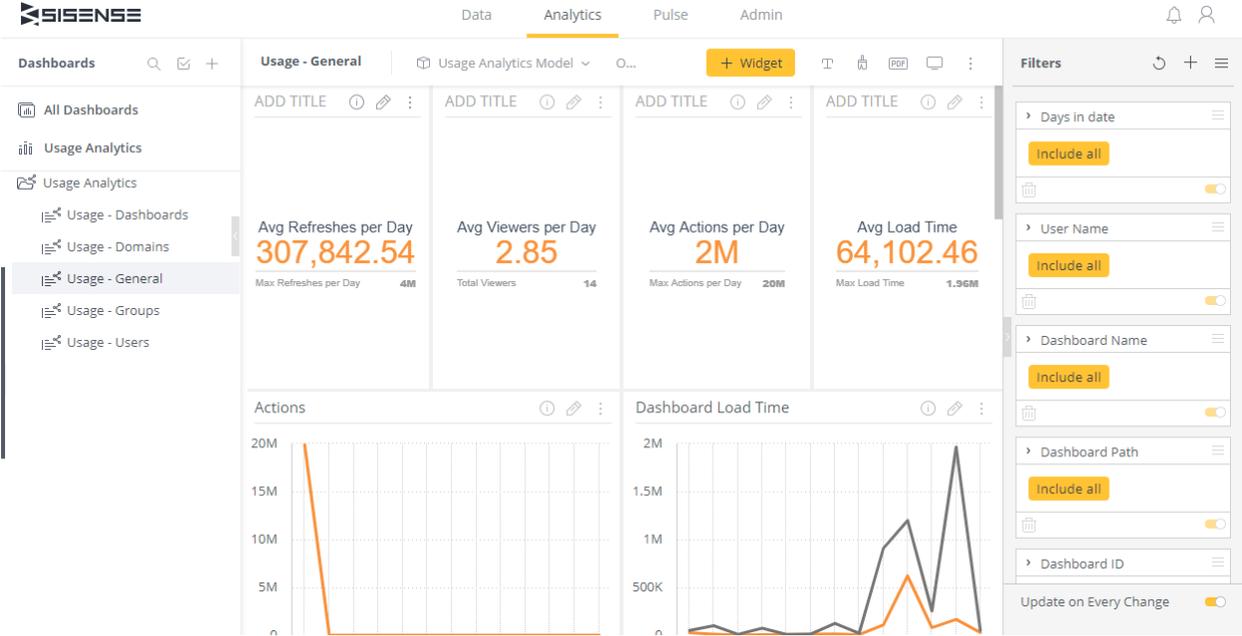
This dashboard is useful for answering the following questions:

Question	Widget
How many domains are using my dashboards?	Active Domains in Period
How many dashboards are users from each domain viewing?	Avg Viewed Dashboards per Domain, Avg Viewers per Domain, Users per Domain
How often are dashboards being refreshed by users from the domain?	Avg Refreshes per Dashboard per Domain, Dashboard Refreshes and Viewers
How many Viewers do I have per domain?	Avg Viewers per Domain, Number of Dashboard Viewers per Domain
How much are users from each domain interacting with my dashboards?	Avg Actions per Domain, Actions over Time
How well are my dashboards loading?	Avg Load Time, Dashboard Load Time
Who is viewing my dashboards?	Dashboard Viewers per Domain

## General Analytics

The Usage - General dashboard provides you with insights regarding your Sisense system, including how often Sisense is being used and its performance.

You can filter the information displayed in your widgets with a set of filters on the right. With these filters, you can filter the details about your system activity by an individual dashboard or per user or group.

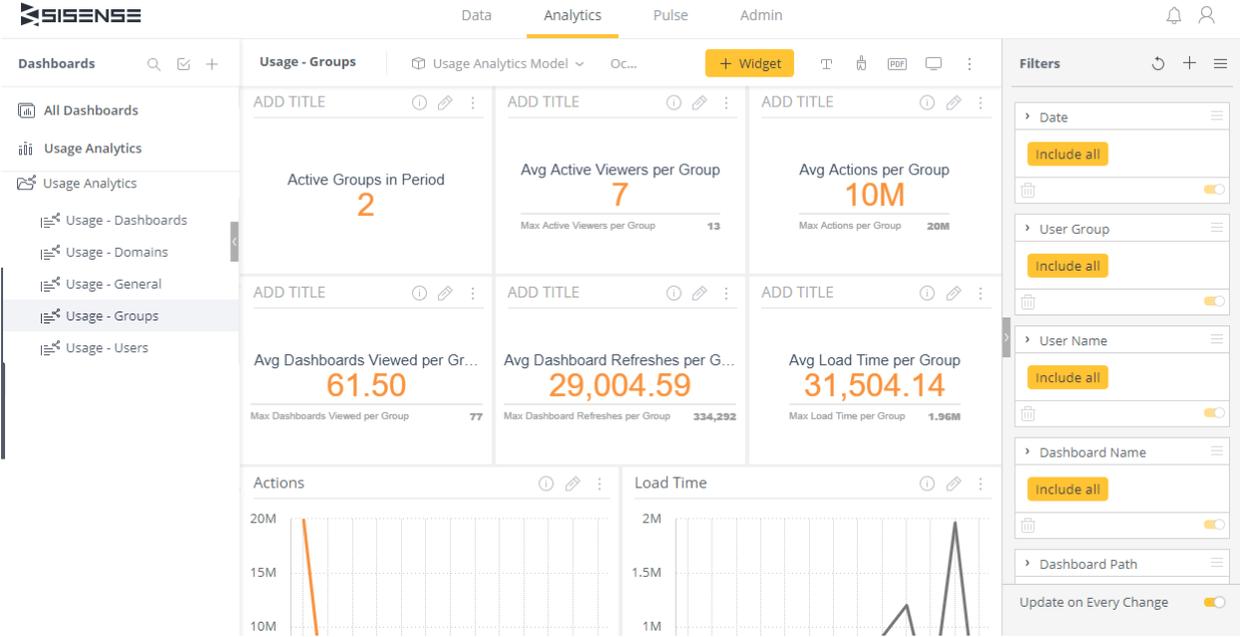


This dashboard is useful for answering the following questions:

Question	Widget
How often are dashboards being refreshed?	Avg Refreshes per Day, System Usage
How many Viewers are active each day?	Avg Viewers per Day, System Usage
Have many actions occur in my dashboards per day?	Avg Actions per Day, Actions, System Usage
How long does it take to load a dashboard?	Avg Load Time, Dashboard Load Time, System Usage
Which dashboards are being viewed?	Dashboard Viewers
Who is viewing my dashboards?	Dashboard Viewers

# Groups Analytics

The Usage - Group dashboard provides you with insights regarding your Sisense groups. This dashboard is useful to see which groups are active and check the system performance for dashboards for your groups.



The Usage Domain dashboard contains data regarding how many groups were active for the recorded period of time, how often they are using your dashboards, and what type of dashboard load performance your users are experiencing.

You can filter the information displayed in your widgets with a set of filters on the right. With these filters, you can filter the details displayed in your dashboard by each user, group, or dashboard.

With the Group Analytics dashboard, you can keep track of your groups and remove or edit any groups that aren't being used.

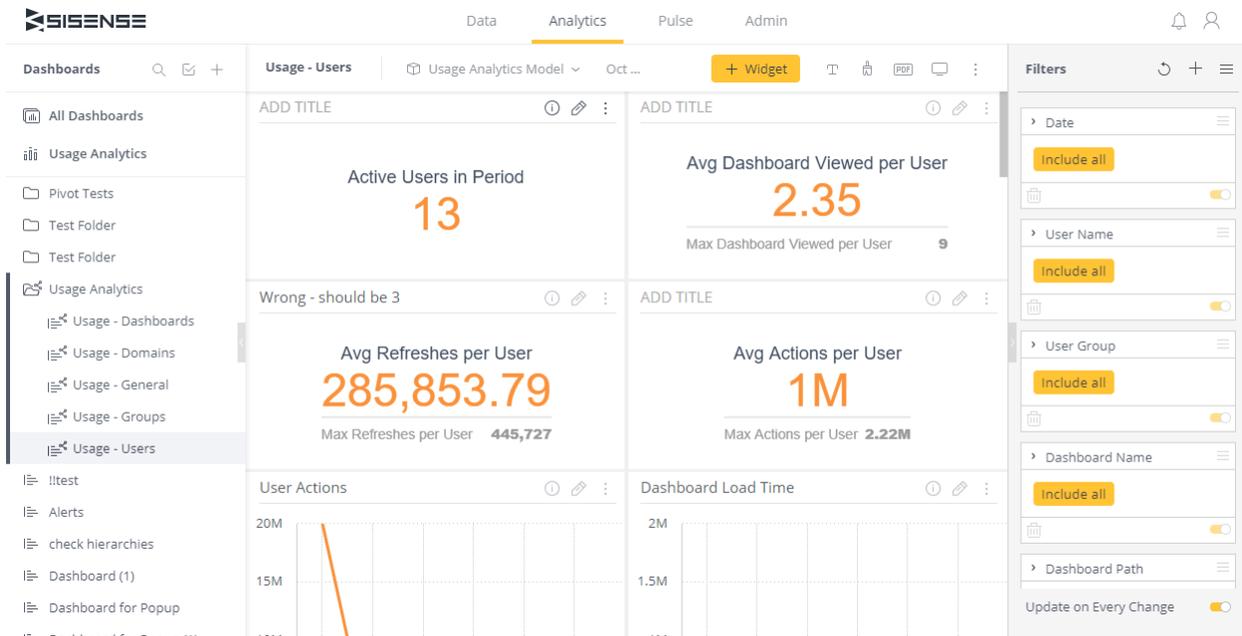
This dashboard is useful for answering the following questions:

Question	Widget
How many active groups do I have?	Active Groups in Period, Dashboard Refreshes and Viewers

Question	Widget
Can unnecessary groups be deleted?	Active Groups in Period, Avg Active Viewers per Group, Dashboard Viewers per Group
How many active users are in a group?	Avg Actions per Group, Actions, Dashboard Refreshes and Viewers
How many actions does a group perform?	Avg Actions per Group, Actions
How many dashboards were viewed by a group?	Avg Dashboards Viewed per Group
How many times was a dashboard refreshed by a group?	Avg Dashboards Refreshed per Group
Whats the dashboard load time for my groups?	Avg Load Time per Group, Load Time
Who is in my groups?	Dashboard Viewers per Group

## User Analytics

The Usage - User dashboard provides you with insights regarding your Sisense users. This dashboard is useful to see which users are active and which users are using a license unnecessarily.



The User Analytics dashboard contains data regarding how many users were active for the recorded period of time, how often they are using your dashboards, and what type of dashboard load performance your users are experiencing. You can filter the information displayed in your widgets with a set of filters on the right. With these filters, you can filter the details displayed in your dashboard by each user, group, or dashboard.

This dashboard is useful for answering the following questions:

Question	Widget
How many active users do I have?	Active Users in Period, Dashboard Refreshes and Viewers, User Activity
Can unnecessary users be deleted?	Active Users in Period, Viewed Dashboards per User, User Activity
How often are my users viewing dashboards?	Avg Refreshes per User
How many times is a dashboard being refreshed?	Dashboard Refreshes and Viewers

Question	Widget
How many times are users interacting with my dashboards?	Avg Actions per User, User Actions
How fast are my dashboards loading?	Dashboard Load Time
Who is viewing my dashboards?	Dashboard Viewer, Viewed Dashboards per Users

## Restoring Usage Analytics

You can work with your Usage Analytics ElastiCube and dashboards like you would with any other ElastiCube or dashboard you own.

If you have modified the Usage Analytics ElastiCube and cannot build it anymore, you can restore the default ElastiCube from the **Admin** page or the REST API.

Restoring the ElastiCube or a dashboard returns it to the original state. Any changes you made will be deleted.

### To restore your Usage Analytics ElastiCube and dashboards in Sisense:

1. In the **Admin** page, click **Usage Analytics**.
2. If **Collect Usage Analytics** is not enabled, toggle the switch to enable it.
3. Click **Restore** to for the relevant assets or click both **Restore** buttons to restore the ElastiCube and dashboards.

### To restore your Usage Analytics ElastiCube and dashboards through the

#### REST API:

1. In the **Admin** page, click **REST API** and then select **API Reference**.
2. For the ElastiCube, run the following endpoints to restore your Usage Analytics assets:  
`admin, POST /usageanalytics/restore/cube`  
 For your dashboards, run the following endpoint:

admin, POST /usageanalytics/restore/dashboards

**POST** /usageanalytics/restore/cube Restore usage analytics cube

Endpoint run restore process for default Usage Analytics cube.

**Parameters** Try it out

Name	Description
authorization string (header)	The user's API token preceded by the keyword Bearer (with space between it and the token). For more information, see <a href="#">API tutorial</a> .

**Responses** Response content type: application/json

Code	Description
200	<p><b>Execution result</b></p> <p><a href="#">Example Value</a>   <a href="#">Model</a></p> <p>true</p>

**POST** /usageanalytics/restore/dashboards Restore usage analytics dashboards

## Managing Plug-ins

Sisense plugins enable you to extend the functionality and capabilities of your Sisense dashboard through standard web development languages such as JavaScript and HTML.

For more information about developing Sisense Plug-ins, see [JavaScript API & Plugins](#).

In the Sisense **Admin** page, you can view a list of your current plugins, their version as defined in your plug-in's script, and their status, enabled or disabled.

Admin	Plug-ins					Refresh List	Search
	Plug-in Name	Version	Last Update	Status			
User Management	<input type="checkbox"/>	accordionPlugin	n/a	06/12/2017 1:56 pm	Disabled	<input type="checkbox"/>	
Users	<input type="checkbox"/>	beforeshareduserssuggested	n/a	06/12/2017 1:56 pm	Disabled	<input type="checkbox"/>	
Groups	<input type="checkbox"/>	DynamicBuckets	n/a	04/21/2016 6:07 pm	Enabled	<input checked="" type="checkbox"/>	
Active Directory	<input type="checkbox"/>	funnel	n/a	06/05/2017 3:55 pm	Disabled	<input type="checkbox"/>	
Single Sign On	<input type="checkbox"/>	iframeWidget	n/a	06/05/2017 3:55 pm	Disabled	<input type="checkbox"/>	
Data Management	<input type="checkbox"/>	jaqline	n/a	06/05/2017 3:55 pm	Disabled	<input type="checkbox"/>	
Data Sources	<input type="checkbox"/>	jumpToDashboard	n/a	06/05/2017 3:55 pm	Disabled	<input type="checkbox"/>	
System Configuration	<input type="checkbox"/>	kmeans	n/a	09/18/2016 12:32 pm	Enabled	<input checked="" type="checkbox"/>	
Settings	<input type="checkbox"/>	no name	n/a	07/06/2015 11:28 am	Enabled	<input checked="" type="checkbox"/>	
License Utilization	<input type="checkbox"/>	parameters	n/a	08/15/2016 1:56 pm	Enabled	<input checked="" type="checkbox"/>	
Plug-ins	<input type="checkbox"/>	plotWidget	n/a	06/15/2017 9:22 am	Enabled	<input checked="" type="checkbox"/>	
Trusted Servers	<input type="checkbox"/>	RegressionLine	n/a	06/05/2017 3:55 pm	Enabled	<input checked="" type="checkbox"/>	
REST API	<input type="checkbox"/>	smartLabel	n/a	03/02/2017 5:40 pm	Enabled	<input checked="" type="checkbox"/>	
	<input type="checkbox"/>	SwitchDimension	n/a	02/02/2016 2:36 pm	Enabled	<input checked="" type="checkbox"/>	
	<input type="checkbox"/>	tableWidget	n/a	09/01/2015 3:20 pm	Enabled	<input checked="" type="checkbox"/>	

Page 1 of 1      << < 1 > >>      16 Plug-ins Total

From the Plug-ins page, you can enable or disable your plugins. This is useful for testing new plugins or troubleshooting issues with your dashboard. When a plug-in is disabled, its functionality is no longer available in the Sisense Web Application.

### To enable or disable a plug-in:

1. In the Sisense Web Application, select **Admin > Plugin-Ins**.
  2. Toggle the Enable/Disable switch to activate or deactivate a plug-in.
- OR

To activate/deactivate multiple plugins, select the checkbox of the relevant plugins. When you select any of the checkboxes, an Enable/Disable switch is displayed in the menu bar. Toggle the switch to enable or disable all the selected plugins.



After enabling or disabling a plug-in, Sisense rebuilds your Plug-ins' list. The next time you refresh your dashboard, the plugins will be enabled or disabled as defined in the Plug-ins page.

# Setting Up a Custom Email Server

Sisense includes a default email server configuration, which uses Mandrill by MailChimp to send automated emails such as dashboard reports or password reset emails. This configuration can be used for POC and testing environments. For production environments, you should provide your own email server. Custom email servers allow all your email transactions to be completely managed on your side.

Through the Sisense REST API, you can send an HTTP POST request to issue Sisense emails through your own custom email server instead of the default Sisense email server. The REST API exposes properties that you can configure for the custom email servers such as security, timeout, and logging properties.

When authenticating an HTTP request, Sisense only supports the basic authentication through the Authorization method using plain text.

## To set up a custom email server:

1. In the Sisense Web Application, click Admin, and then Rest API, followed by REST API Reference.
2. In version 1.0 of the REST API, select Settings, then POST `/settings/email_server`.
3. In the body of your call, define the values of the following keys:

Key	Value
enable	If True, allows access to the custom email server instead of the default Sisense email server. <b>Note:</b> This property is different from the enable property found in version .9 of the REST API POST <code>/settings/system</code> request. In .9, enable activates Sisense emails. In version 1.0, enable sends Sisense emails through your custom server.
host	The hostname or IP address to connect to. By default, this is 'localhost'.

Key	Value
port	The port to connect to (defaults to 25 or 465).
secure	<p>When you send an email, some email servers check if secure transport (TLS) is available for that address or domain.</p> <p>This value determines if the connection should use TLS (if True) or not (if False). The default value is false. The connection may be upgraded to TLS if the email server requires it.</p>
ignoreTLS	<p>If this is True and secure is False, TLS will not be used even if the email server attempts to upgrade the connection to TLS-secure.</p> <p>This field should be set to True when connecting to a non-secure SMTP server.</p>
requireTLS	If this is True and secure is False, it forces the connection to be upgraded to TLS-secure even if the email server does not advertise support for TLS.
connectionTimeout	How many milliseconds to wait for the connection to establish.
greetingTimeout	How many milliseconds to wait for the greeting after connection is established.
socketTimeout	How many milliseconds of inactivity to allow.
user	<p>The user name of the user.</p> <p>As non-secure or restricted SMTP servers do not require authentication, do not include this field when connecting to a non-secure SMTP server.</p>

Key	Value
pass	The user's password for the email server. As non-secure or restricted SMTP servers do not require authentication, do not include this field when connecting to a non-secure SMTP server.
logger	If set to True, Sisense logs events such as which server you connected to.If the value is not defined or is False, then nothing is logged.
debug	If set to True, then Sisense logs SMTP traffic. When False, Sisense logs only transaction events.Debug logs record everything and may increase in size quickly and impact performance. You should only set activate debug logging when troubleshooting a specific issue and then reset the logging level back to info.

4. Click **Run**.
5. After you have set up the email server, you need to change the sender email address to the domain of your custom email server so they are consistent. You can update the sender email through the POST 0.9 /branding endpoint in the SisenseREST API.

```
{
  "emails": {
    "senderEmail": "username@newdomain.com"
  }
}
```

6. Click **Run**.

# Activating a Sisense AMI from the AWS Marketplace

Sisense provides a BYOL (Bring Your Own License) AMI (Amazon Machine Image) on the Amazon AWS Marketplace. You can utilize your image to run your Sisense instances on AWS. Before you create an instance, make sure you have your Sisense account user and password available to initialize the instance.

**Note:** The payment to AWS covers the price of the machine on AWS, using an existing Sisense license. Your Sisense license payment is part of the existing Sisense account billing, and is not charged to your AWS account.

## To initialize your Sisense instance:

1. Instantiate the Sisense 7.x AMI from the AWS Marketplace to create your Sisense server.



## Sisense Activation

Welcome to your Sisense instance!  
In order to start, enter your credentials.

Sign In

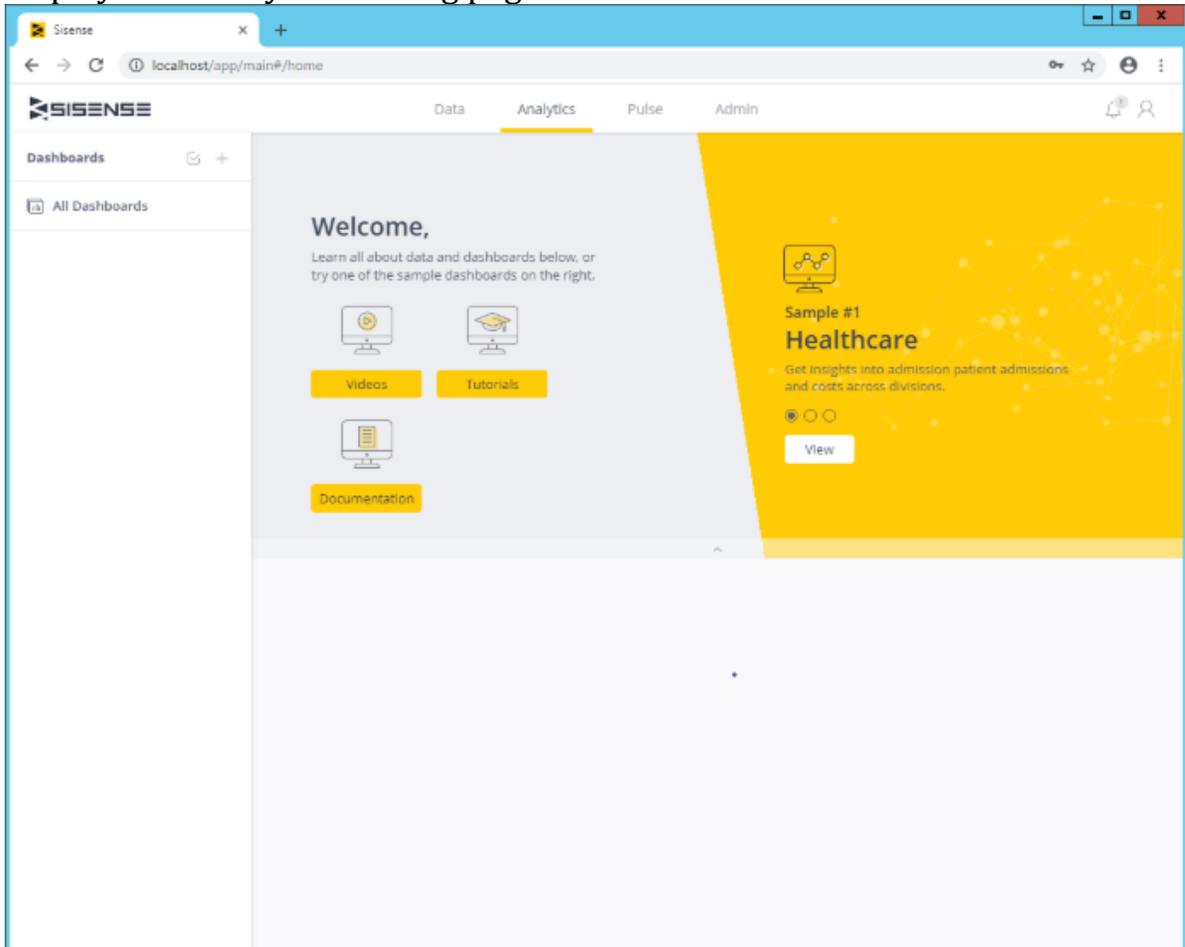


[Contact Support](#)

[Sign In >](#)

2. Connect to your Sisense server using Remote Desktop. The Sisense Activation screen is displayed.
3. Sign in using your Sisense credentials: the email address and password you provided when acquiring your Sisense license.

4. Click **Sign In** to continue. Your account information will be verified, and your server automatically initialized. After completing the initialization, your browser will automatically be launched, displaying the Sisense login window (<http://localhost/app/account#/login>).
5. Enter the same user name and password you provided to activate the server, and click **Login** to login to Sisense server. You Sisense server is now active and available for your use. The Sisense Web UI is available - and displays the Analytics landing page.



### Troubleshooting:

- In case you activate Sisense server, and login to it, and the Sisense WebUI isn't displayed, restart your Sisense server. As you have already activated the Sisense instance, your Sisense application is ready. Click **Start** to launch the application, and display the login screen.

# Welcome!

Your Sisense application is ready

Start >

[Contact Support](#)

Provide your Sisense user and password to begin working with the application.

# Viewing Dashboards on Mobile Devices

Sisense Mobile connects you to your dashboard anytime, anywhere on your Android or iPhone mobile devices. Sisense Mobile decreases time to insight with an intuitive UI, search bar for locating dashboards, and responsive visualizations; You can explore your dashboards with simple gestures optimized to let you dive deep into your data or apply filters to view the data you want. Check out a demo video below to see how you can tap, zoom, and swipe to interact with your data.

Sisense now offers a premium white labeled version of the Sisense Mobile app that can be customized according to your branding, for more information see [Rebranding Sisense Mobile](#).

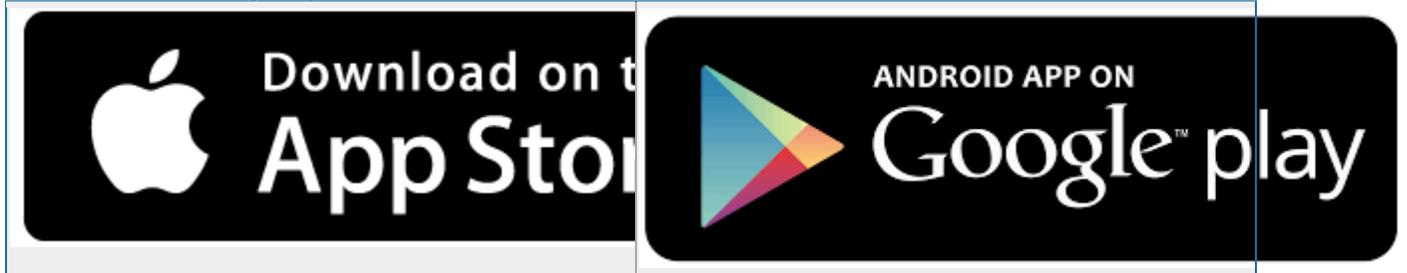
# Getting Started

## Supported Phones and Operating Systems

- iPhone – Requires iOS 9 or later, Supported iPhone 5 device or higher.
- Android – Requires Android versions 4.4 or later.

## Get Sisense Mobile

To download the Sisense Mobile App for your device, click the relevant link below according to your version of Sisense.



**Note:** Sisense Mobile is supported by Sisense V6.5 and later.

## Logging in to Sisense Mobile

After you have downloaded Sisense Mobile, open the app to launch the login screen and enter the server address and your Sisense user credentials.

The server address should include the server IP or domain name (alias). If you are not using the default port (80) you will need to add the port as well, for example: `http://xx.xxx.xx.xx:8081`. When working with SSL security protocol, use the 'https' prefix.

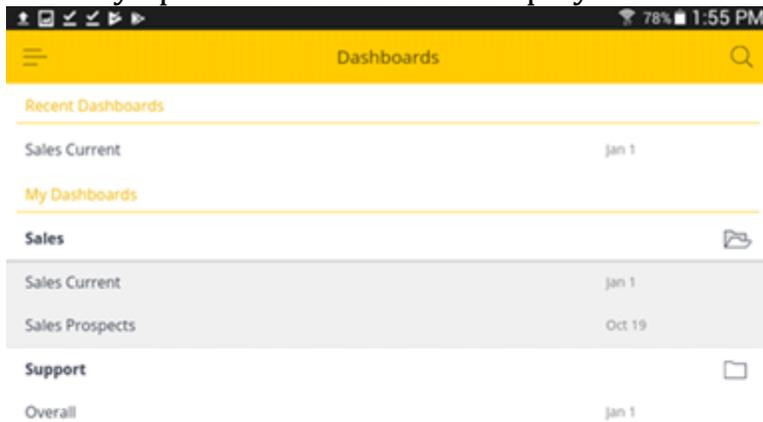
You can sign out by opening the navigation menu and tapping Sign Out.

If you have problems connecting, check the following:

- If your server is not on port 8081, check that the port has been added correctly, for example, 192.168.1.1:8080
- Try to access the server from your phone's web browser
- Try to access the server from other computers connected to the same WIFI connection as your phone

## ***Navigating Sisense Mobile***

When you first log in to Sisense Mobile, a list of your folders, dashboards, and recently opened dashboards is displayed.

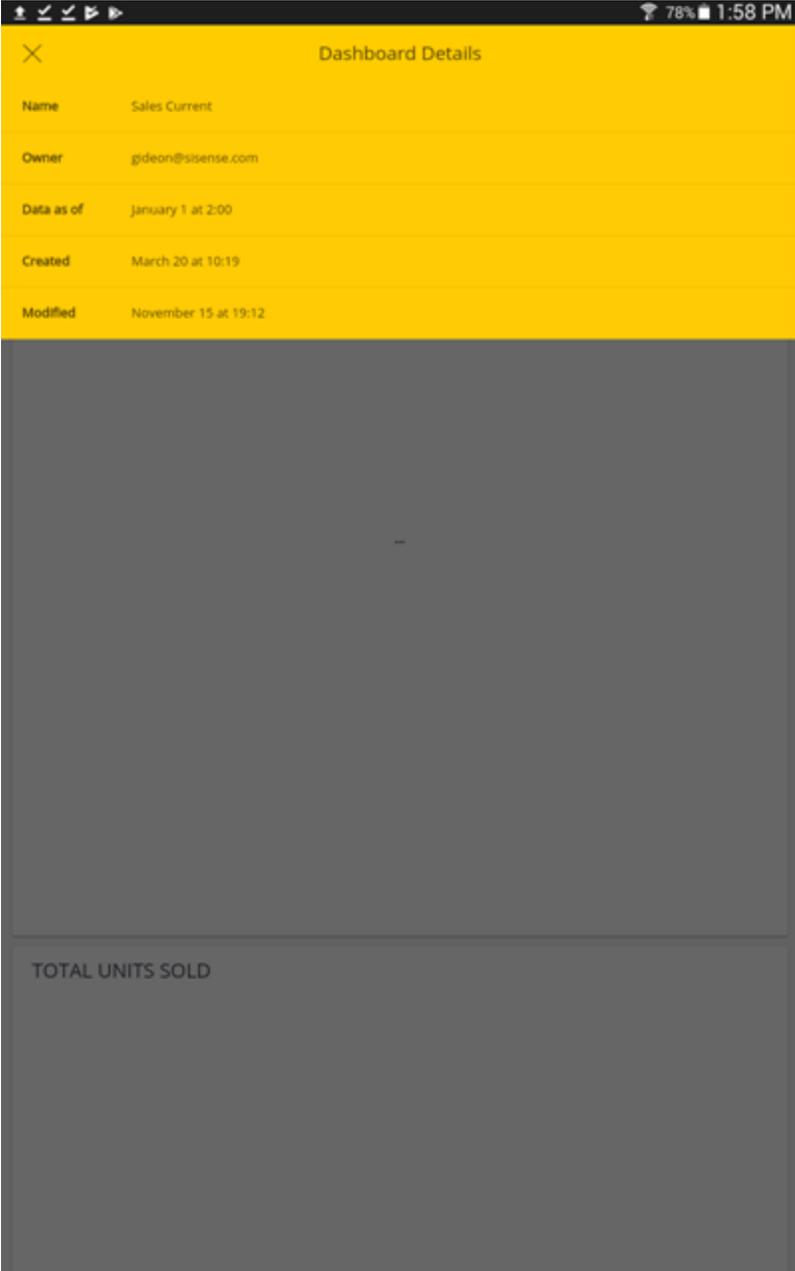


By tapping a dashboard, you can open the dashboard or if your dashboard is located in a folder, tap the folder then tap the relevant dashboard to open it. You can also locate a dashboard quickly by tapping the search icon and typing the name of the dashboard. As you begin to type, any dashboards that match the

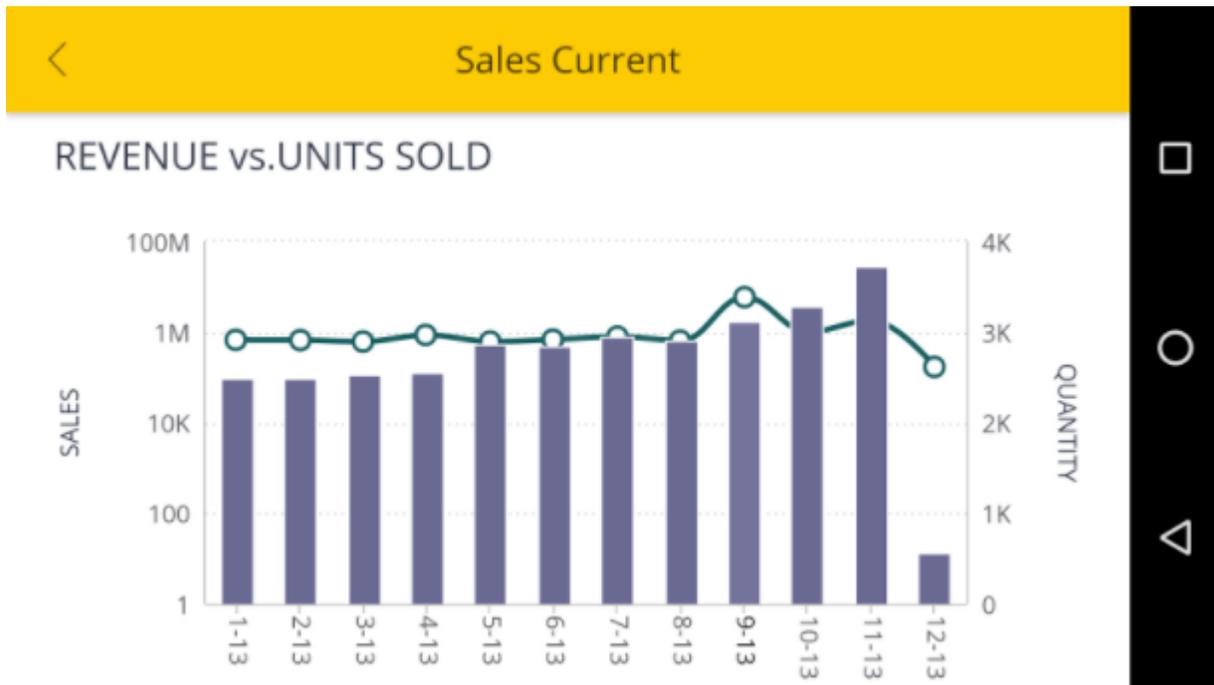
characters you entered are displayed.



Once the dashboard is displayed, you can tap the dashboard's title to display more details.



By rotating your device horizontally, you can view your dashboard in landscape view.

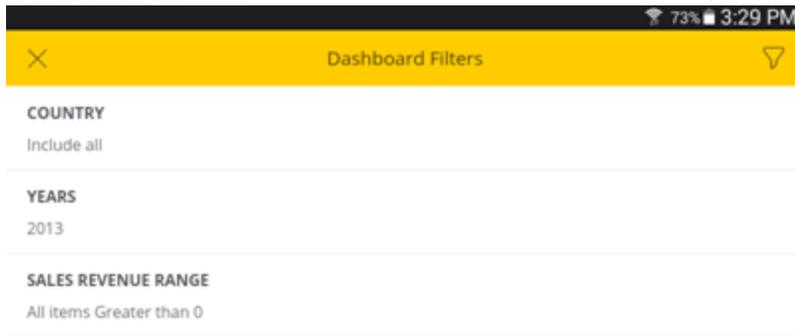


### *Exploring Data*

In Sisense Mobile, you can filter and drill down into your data much like you do from your PC.

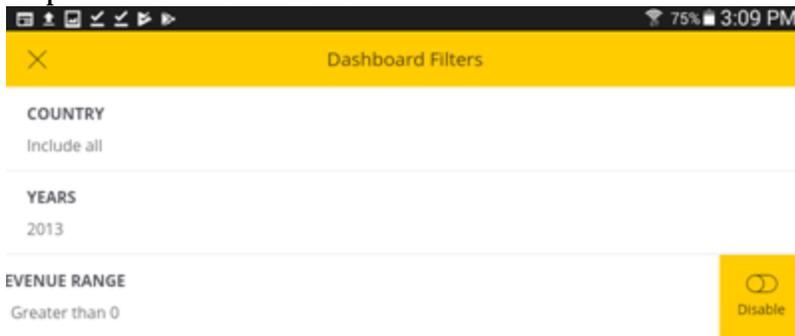
## Filtering Widgets

To filter a dashboard, you can tap the filter icon . This displays the list of available filters.

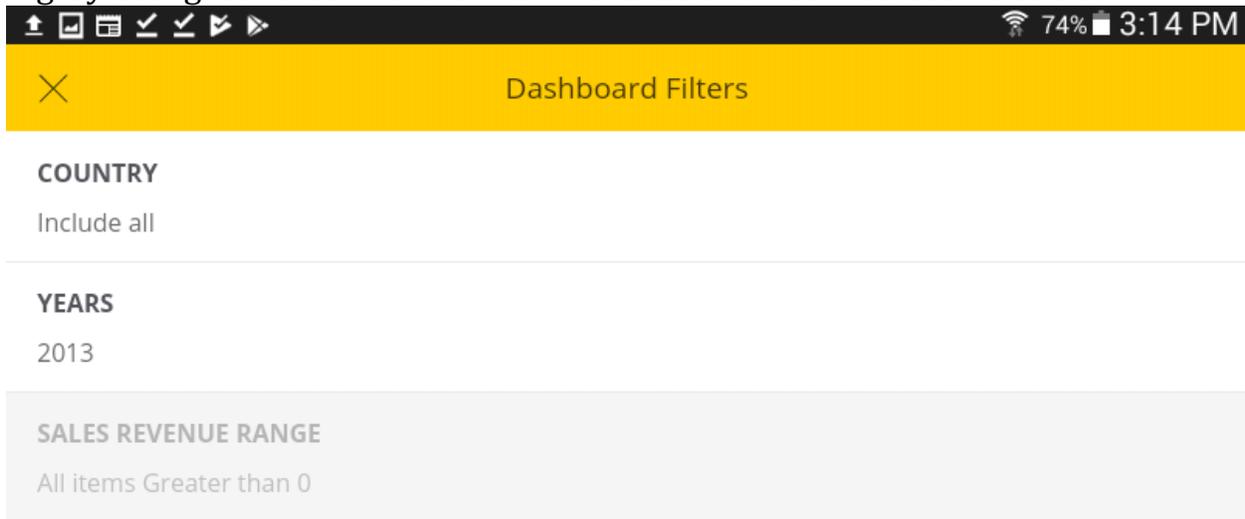


You can disable your filters by swiping left on the filter to display a toggle button.

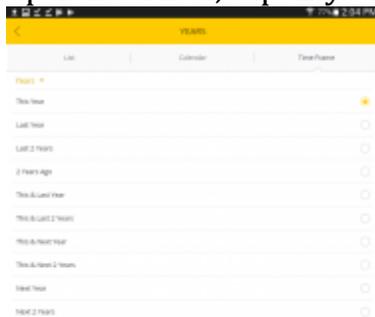
Tap it to disable a filter.



A gray background behind the filter indicates the filter has been disabled.



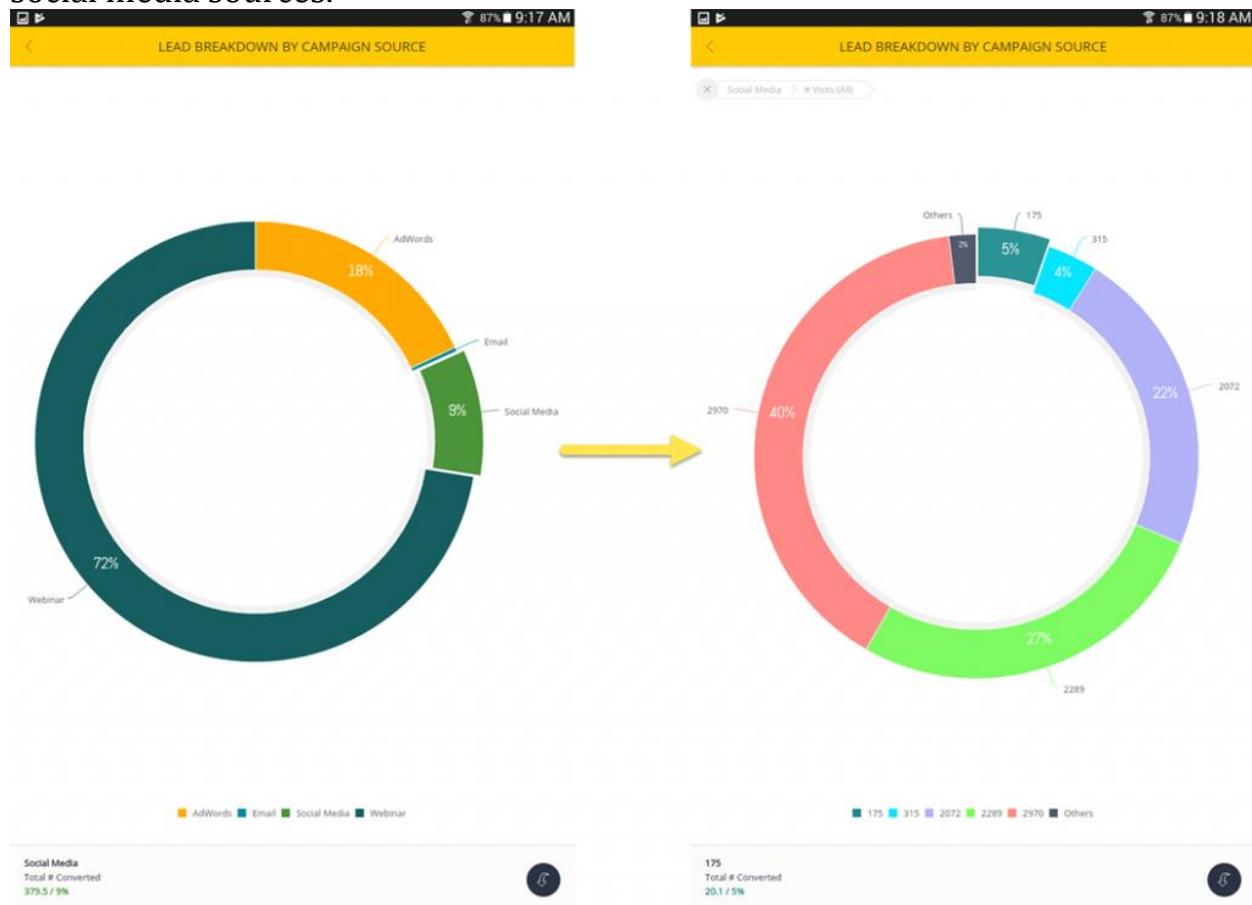
To dive deeper into your filters, tap the relevant filter to open all of its available options. Then, tap any of the filter options to toggle the filters to your data.



## Drilling into your Data

You can drill down in most widgets to get an in-depth view of a selected value. For example, tap on a slice in a pie chart to see the absolute value in addition to the percentage.

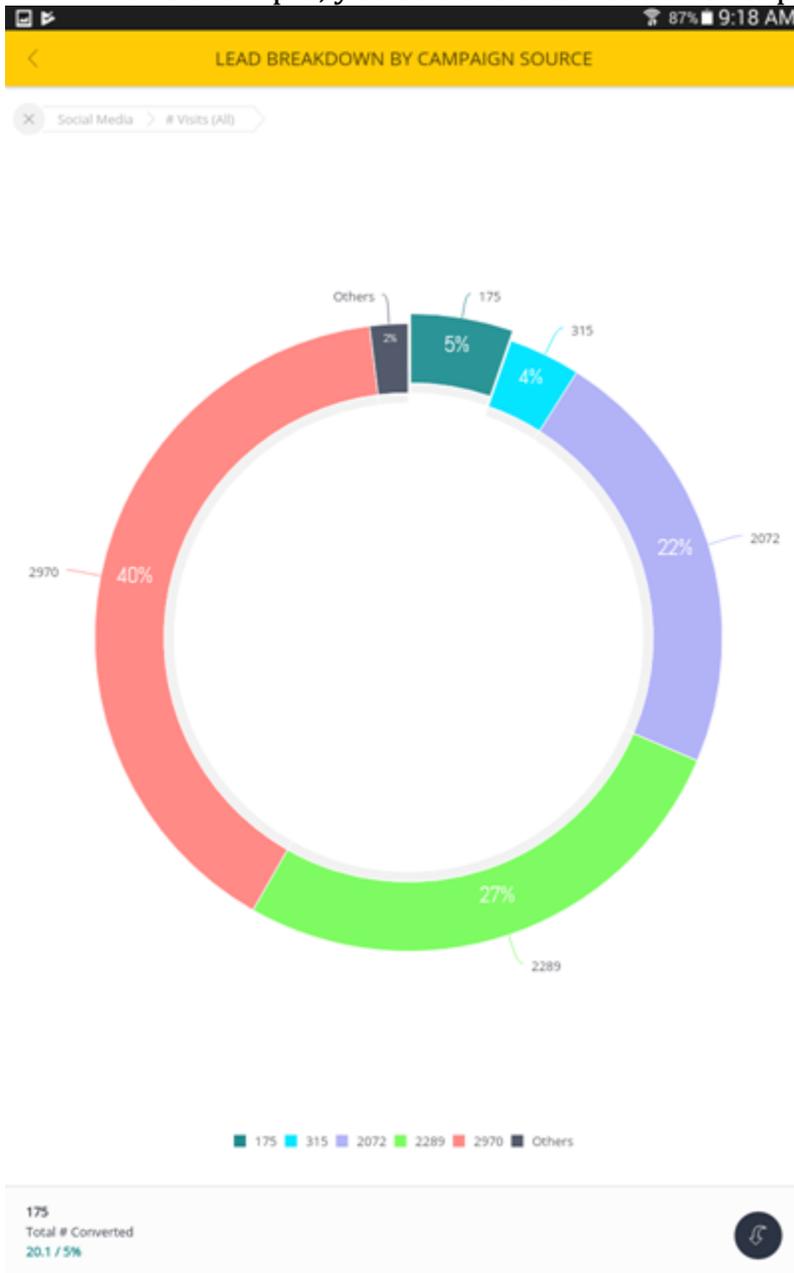
In the following example, the left chart shows the lead breakdown by campaign source, and on the right side, the chart shows a breakdown of visits from various social media sources.



Any time you select a value, a tooltip is displayed at the bottom of your screen, which provides a description of the selected value and the drill down

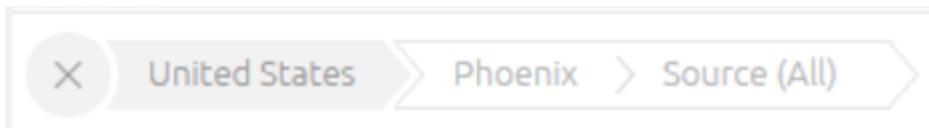


icon. In this example, you see the total lead and its percentage.



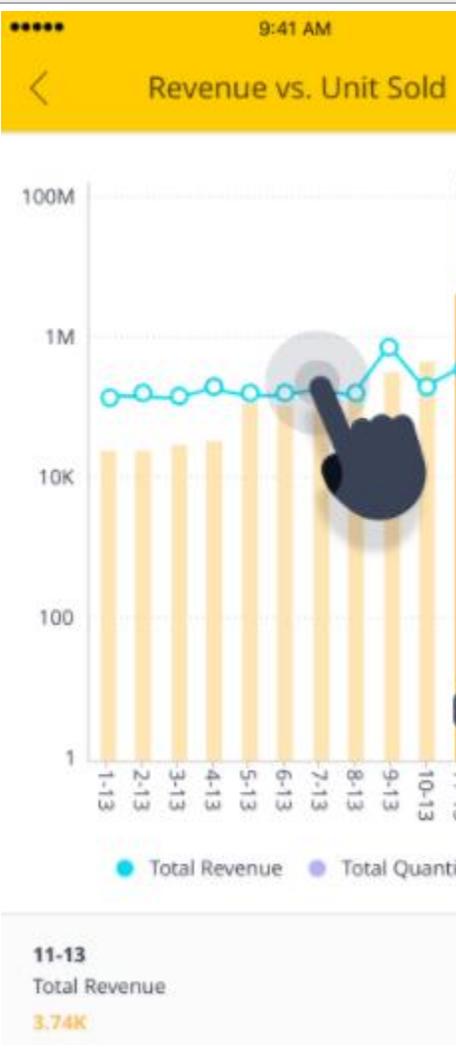
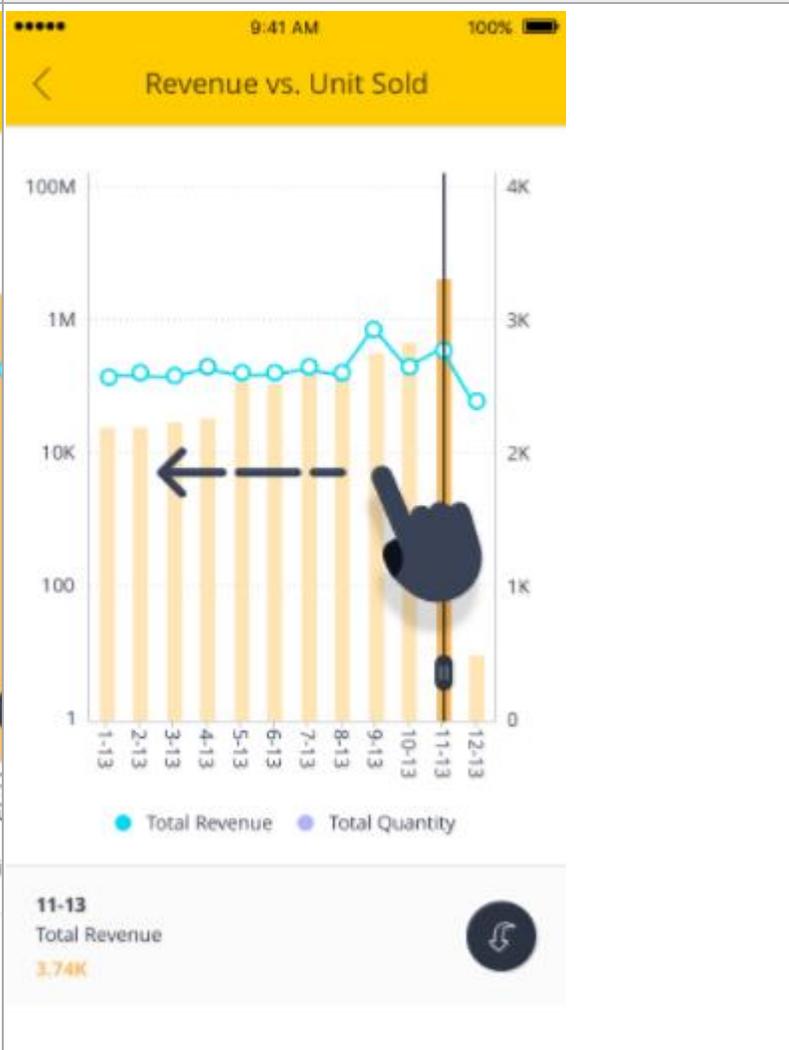
**Note:** You cannot drill down while in landscape mode.

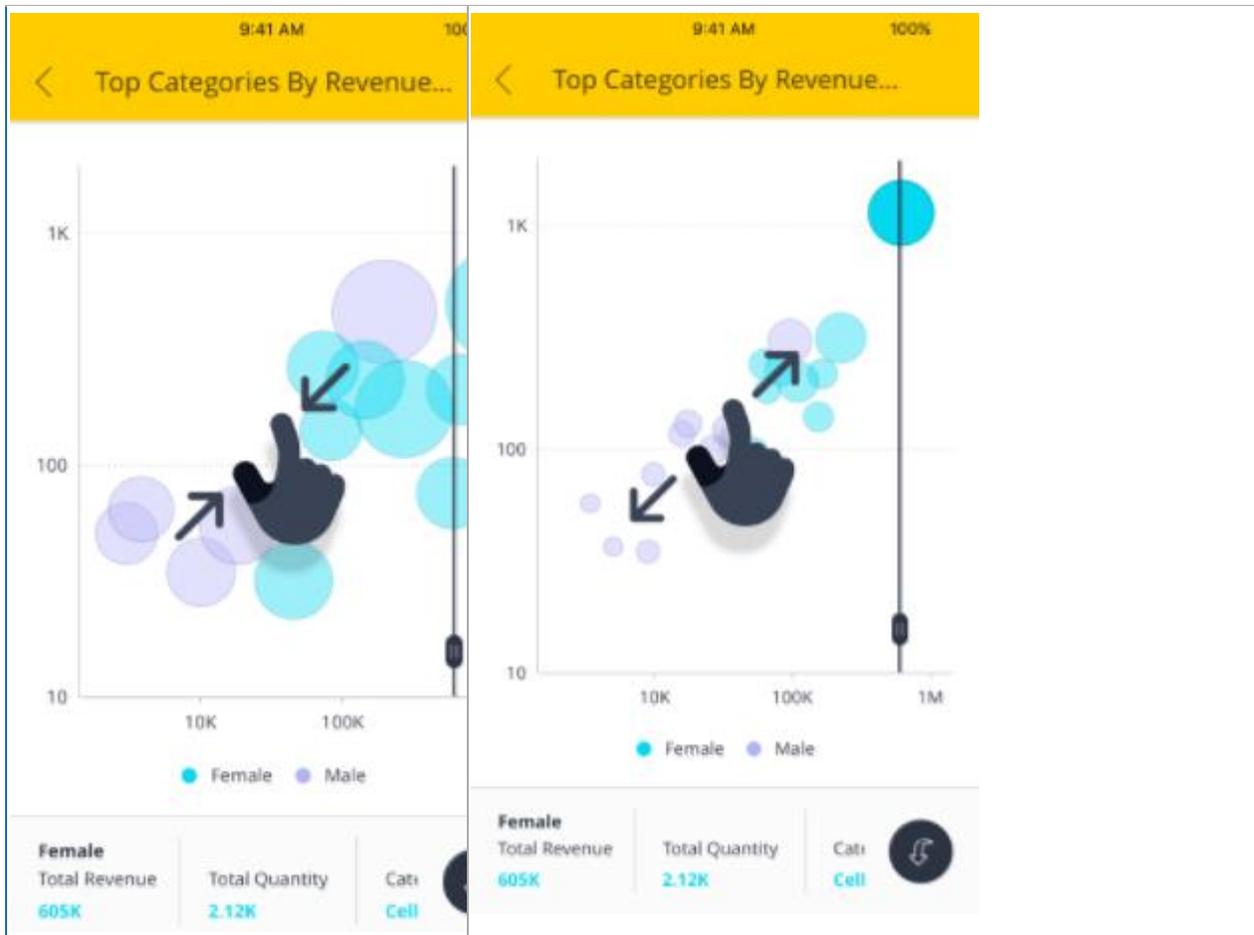
To return to a higher level, click on a breadcrumb. To drill all the way up, click on the X icon.



# Viewing More Details

Sisense Mobile supports a variety of gestures that let you dive deeper in your data and see more details.

Highlight a Value	Swipe across your Visualizations
 <p>The screenshot shows a mobile app interface with a yellow header and a chart titled "Revenue vs. Unit Sold". The chart has a logarithmic y-axis with labels 1, 100, 10K, 1M, and 100M. The x-axis shows dates from 1-13 to 10-13. A hand icon is pointing to a data point on the blue line representing Total Revenue. A summary card at the bottom left shows "11-13 Total Revenue 3.74K".</p>	 <p>The screenshot shows the same mobile app interface, but with a hand icon swiping across the chart. A dashed arrow points to the left, indicating the direction of the swipe. The chart is zoomed in on the 11-13 date. A summary card at the bottom right shows "11-13 Total Revenue 3.74K" with a refresh icon.</p>
Pinch in to Zoom	Pinch out to Expand

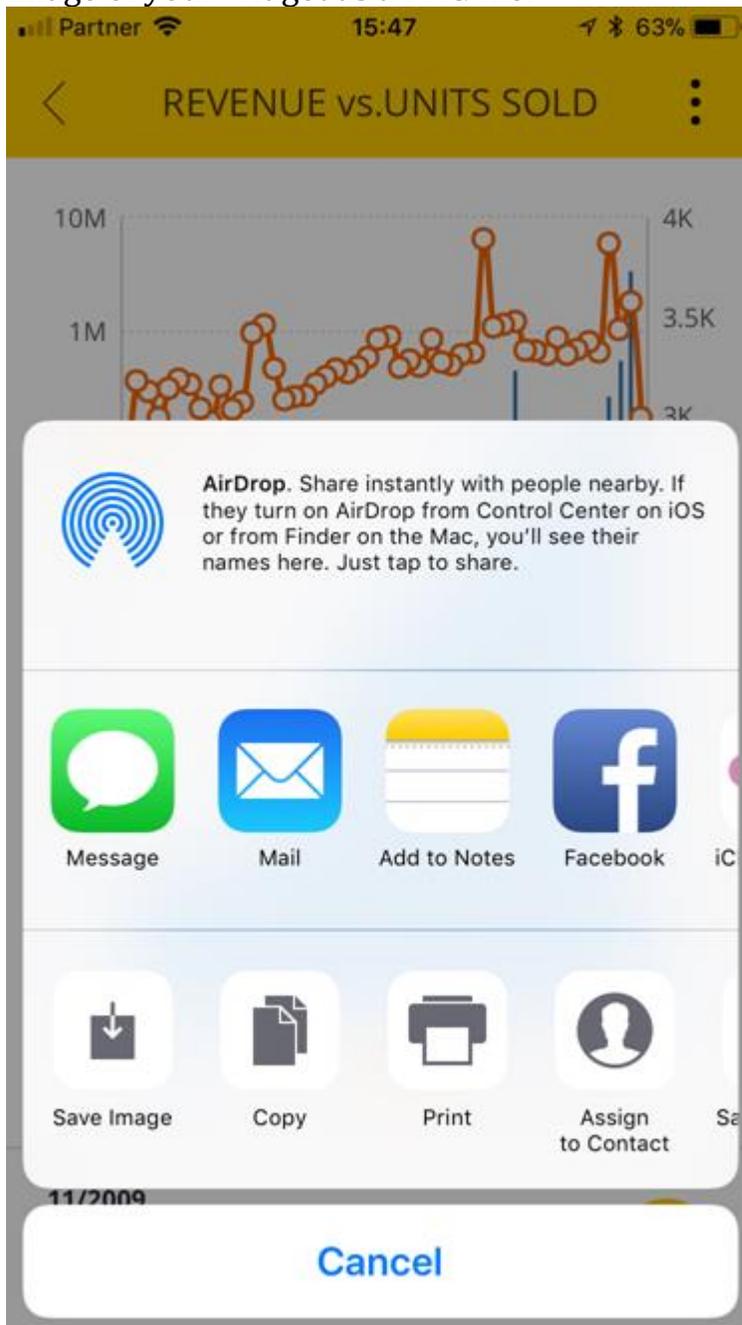


## Sharing Widgets

You can share widgets as PNG images from Sisense Mobile across various apps supported on your device.

In your dashboard, tap a widget to open Widget Mode, then tap the menu icon  and tap Share. Your device will display a list of apps that you can use to send the

image of your widget as a PNG file.

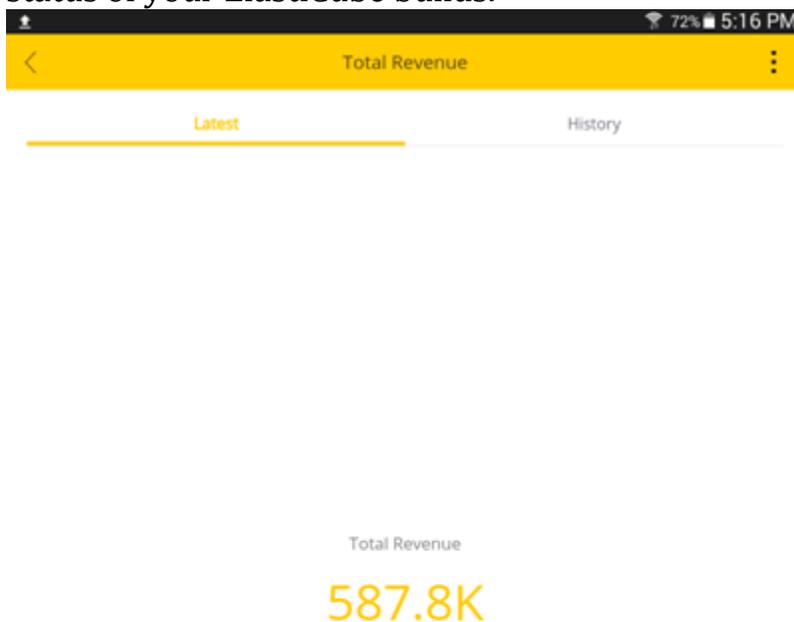


## ***Sisense Pulse***

Sisense Pulse is a centralized location where you can stay on top of your most important KPIs across multiple dashboards or manage your data and build alerts. By adding important KPIs from your dashboards to Sisense Pulse, you can get a comprehensive picture of your data from a single location.

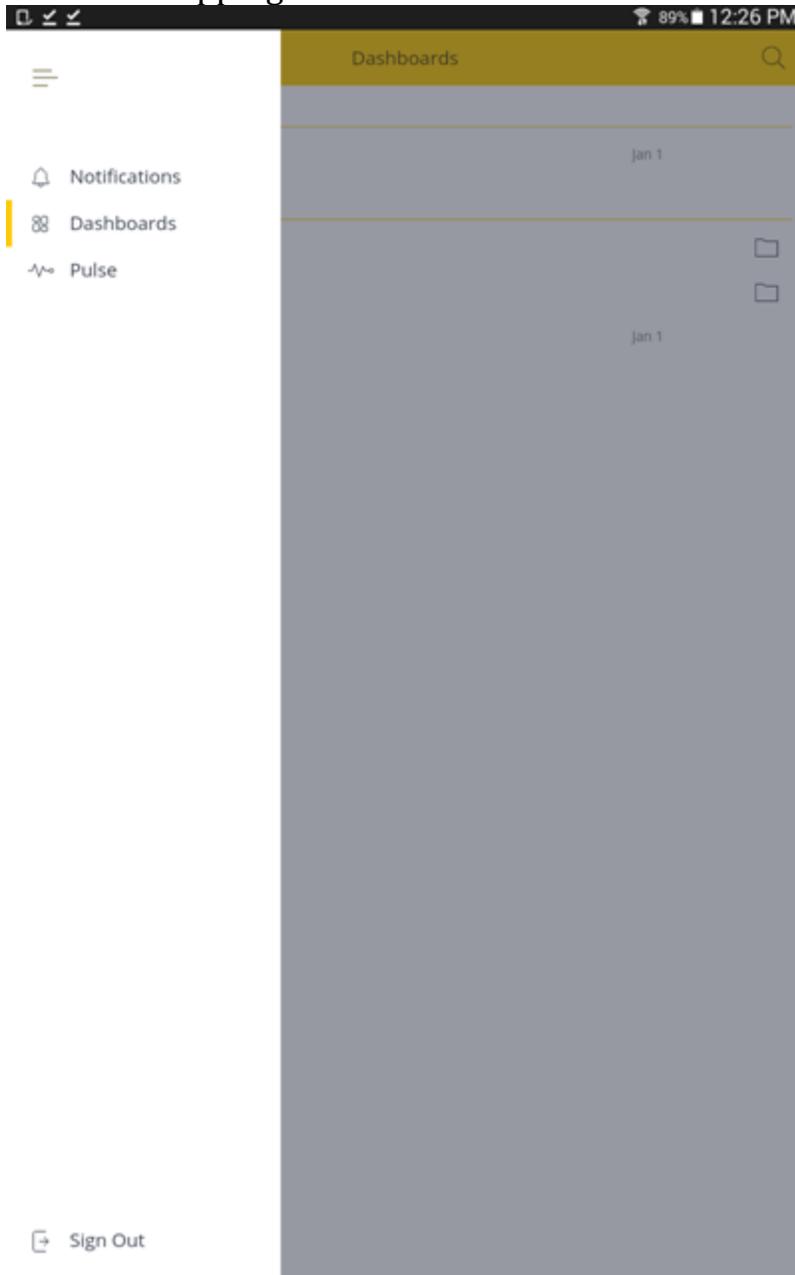
Sisense Pulse contains tiles that display information from your dashboards and the status of your ElastiCube builds.

Within Sisense Pulse, there are two types of tiles, data tiles that display your widgets' current status and recent history and build tiles that display the latest status of your ElastiCube builds.



In Sisense Mobile you can manage your alerts from Sisense Pulse or create data alerts based on widgets in your dashboards.

**Note:** Build alerts can only be created from the Sisense Web Application. You can access Sisense Pulse from Sisense Mobile by opening the navigation menu and tapping Pulse.

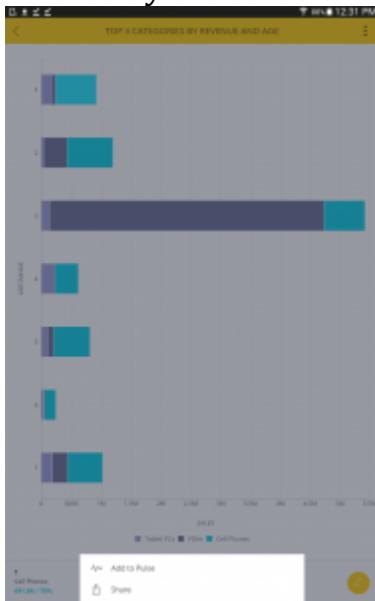


## Creating Data Alerts

Data alerts are triggered by conditions you define for your widgets. For example, if you have an Indicator widget that represents your company's revenue, you can define an alert that notifies you when this revenue reaches or falls below a certain threshold.

For each type of widget you can add to Pulse, you can monitor a single value. For example, for Column charts, you can monitor a single section (value) of a column. To monitor multiple values, you can add each value you want to monitor to Pulse separately.

When you create an alert for a specific value, Sisense applies any filters that were already applied, for example, if you create a data alert for the total revenue of a given age range, the pulse alert created will be called Total Revenue and will be filtered by the value of the age range you selected.



To create a data alert, from your dashboard, tap the relevant widget to open it in Widget Mode. In Widget Mode, tap the value you want to monitor, then tap the menu icon  and tap **Add to Pulse**. This opens the Alert Conditions menu where you select the type of data alert you want to create. For example, if you create a

threshold data alert, the following settings are displayed.

The screenshot shows a mobile application interface for configuring an alert. At the top, there is a yellow header bar with a back arrow on the left, the text "Add to Pulse" in the center, and a "Save" button on the right. Below the header, the screen is divided into several sections with light gray headers:

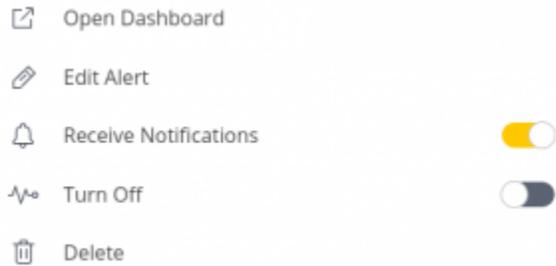
- Alert Name:** A text input field containing "Total Revenue".
- Alert Condition:** A section containing a "Threshold" field with the value "> 691797" and a right-pointing chevron.
- Notification Methods:** A section containing two toggle switches: "Mobile" (turned on) and "Email" (turned on).
- Notification Limitations:** A section containing two toggle switches: "Notify once" (turned on) and "Notify when no longer met" (turned off).

The bottom half of the screen is a large, empty light gray area.

After you have set the conditions, tap **Save** and the new alert will be added to Sisense Pulse.

## Managing Alerts

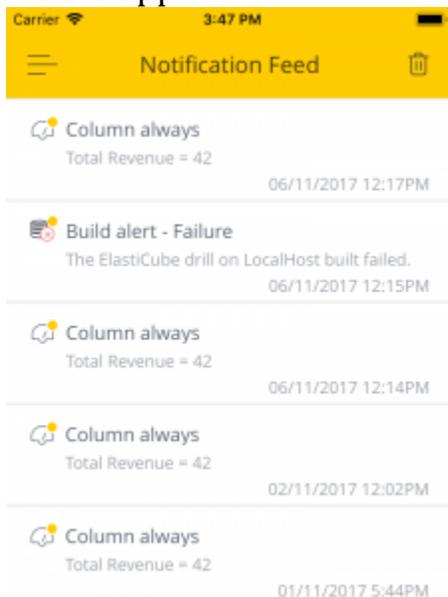
You can manage your alerts from Sisense Pulse. In Sisense Pulse, tap the relevant tile to open Tile Mode. Once you're in Tile Mode, the menu icon  is displayed in the top right corner. Tap it to display a list of alert management options.



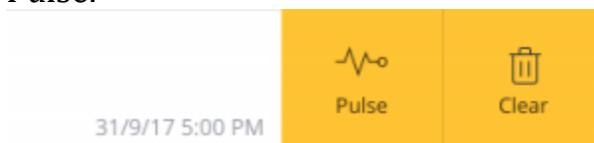
From the options area, you can edit alerts, delete alerts, deactivate alerts, and toggle notifications.

## Pulse Notifications

If your Sisense user has been configured to receive Pulse alerts to your mobile device, when an alert is triggered, you will receive a notification from the Sisense Mobile app.



You can swipe left on an alert to clear the alert or to view the alert in Sisense Pulse.



To clear all alerts, click the trash can icon  at the top of the notification feed. **Note:** You must have Sisense Mobile installed to receive Pulse notifications.

### **Single Sign On**

Sisense supports SSO for Sisense Mobile. If your Sisense server is configured to use SSO, Sisense Mobile will use the same SSO configuration. No additional configuration is required on the Sisense server.

When you log in to Sisense Mobile, you are redirected to the IdP login URL using web view. Following authentication by the IdP, the SSO token is returned to the Sisense Server, which returns a web cookie to Sisense Mobile, and the application

is opened. For more information about how to implement SSO, see [Single Sign On](#).

## ***Plugins***

Sisense Mobile V2.3 and above supports Sisense plugins and supports plugins made with the Sisense JavaScript API.

For information on developers plugins for Sisense Mobile, [click here](#).

## Supported Sisense Plugins

The following table lists Sisense plugins that are supported on Sisense Mobile.

Plugin	Comment
Aggregated Table	
Auto Add Filter Based On User	
Comments	Always use an absolute URL for the form.
Dynamic Buckets	
Embed Images	Always use absolute URLs for the image
Filtered Measure (Parameter)	
Forecasting	
Jump to Dashboard	From Sisense Mobile BI app version 2.5.2

### ***Release Notes***

#### **April 4, 2019**

- Support added for Jump to Dashboard plugin

#### **April 26, 2018**

- Sample dashboards are now available
- Bug fixes

#### **March 3, 2018**

- Support for Hours and Minutes time resolutions in date filters (Sisense 7.0.1 and above)
- Ability to collapse/expand Recent Dashboards
- Improved support for SSO connectivity
- Bug fixes

