THOUGHTSPOT

Administration Guide

Version 5.1 February 08, 2019

ThoughtSpot, Inc. 1 Palo Alto Square
Building 1, Suite 200
Palo Alto, CA 94306

Table of Contents

Introduction to administration

ThoughtSpot enables you to access and analyze your data through a search-based user interface. You can create your searches on the fly by typing into a search bar, like you do when using an internet search engine. ThoughtSpot makes it easy to see your data, get your questions answered, create interactive graphs, and customize pinboards. You do not need to understand how the data is stored or know SQL to do these things.

ThoughtSpot gives administrators the ability to modify data properties to meet business needs, for example by providing search synonyms for common terms, boosting the importance of a column in search results, or formatting how the data appears. Collaboration and security features make it easy for you to protect sensitive data and for users to share information safely with others.

To perform the actions in this guide, you need administrative access.

Login credentials for administration

Summary: You need administrative access to perform the actions discussed in this guide.

You can access ThoughtSpot via SSH at the command prompt and from a Web browser.

Administrative access

Each ThoughtSpot appliance comes pre-built with three default users. You should talk with a ThoughtSpot Customer Success Engineer or ThoughtSpot support, to get the password for each user. The default users are:

Type	Username	Description
Shell user	`admin`	Used for work that requires sudo or root privileges. Does not exist for application login. Logs for this user are found in `/usr/local/scaligent/logs` logs
Shell user	`thoughtspot`	Used for command line work that does not require sudo or root privileges. For example, these users can use `tsload` , `tql` , and check the cluster status. This user cannot login to the application. Logs for this user are found under `/tmp` .
Application user	`tsadmin`	Access through a Web browser.

Both the `admin` and `thoughtspot` user can SSH into the appliance. Once on the appliance, either user can do any of the following:

- tscli
- tsload
- tql

The `thoughtspot` user is restricted to `tscli` commands that do not require `sudo` or root privileges.

SSH to the appliance

To perform basic administration such as checking network connectivity, starting and stopping services, and setting up email, log in remotely as the Linux administrator user “admin”. To log in with SSH from any machine, you can use the command shell or a utility like Putty.

In the following procedure, replace `<hostname_or_IP>` with the hostname or IP address of a node in ThoughtSpot. The default SSH port (22) will be used.

1. Log in to a client machine and open a command prompt.
2. Issue the SSH command, specifying the IP address or hostname of the ThoughtSpot instance:

```
ssh admin@<hostname_or_IP>
```

3. Enter the password for the admin user.

Log in to the ThoughtSpot application

To set up and explore your data, access the ThoughtSpot application from a standard Web browser using a username and password.

Before accessing ThoughtSpot, you need:

- The Web address (IP address or server name) for ThoughtSpot.
- A network connection.
- A Web browser.
- A username and password for ThoughtSpot.

  Supported Web browsers include:

Browser	Version	Operating System
Google Chrome	20 and above	Windows 7 or greater, Linux, MacOS
Mozilla Firefox	14 and above	Windows 7 or greater, Linux, MacOS
Internet Explorer	11	Windows 7 or greater

> ☑ **Tip:** While Internet Explorer is supported, using it is not recommended. Depending on your environment, you can experience performance or UI issues when using IE.

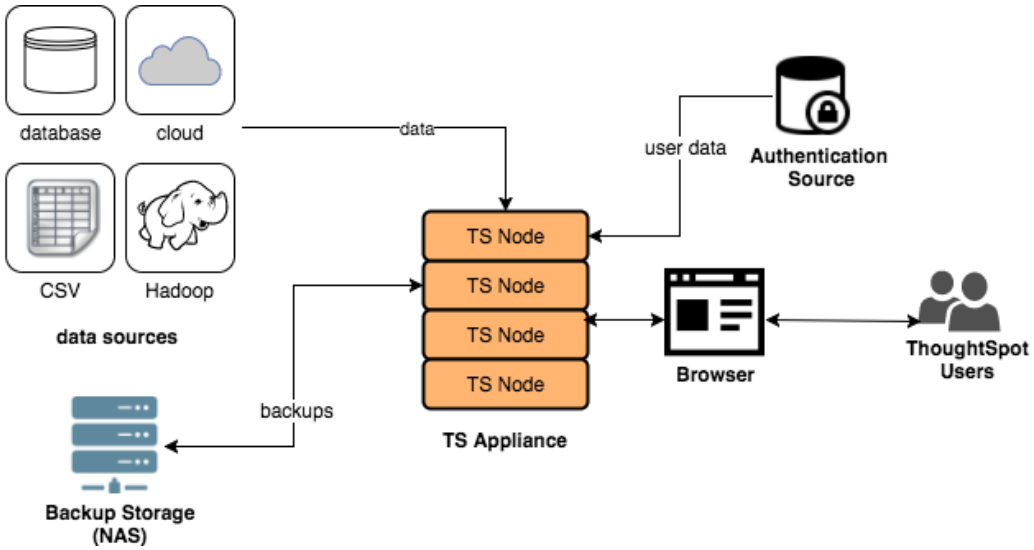
To log in to ThoughtSpot from a browser:

1. Open the browser and type in the Web address for ThoughtSpot:
   `http://<hostname_or_IP>`
2. Enter your username and password and click **Enter Now**.

Architecture components

Summary: To implement ThoughtSpot it is important to understand where it sits within your overall analytics architecture and how it provides data to end users.

ThoughtSpot consists of a cluster of one or more nodes, acting together to provide analytic answers to business questions. As such, there are only a few integration points with ThoughtSpot on your network. The major components in the a ThoughtSpot cluster are:



ThoughtSpot can handle a wide variety of different data sources. ThoughtSpot does all analysis against data in memory to help achieve fast results across millions and billions of records of data. ThoughtSpot caches the data in order to process it.

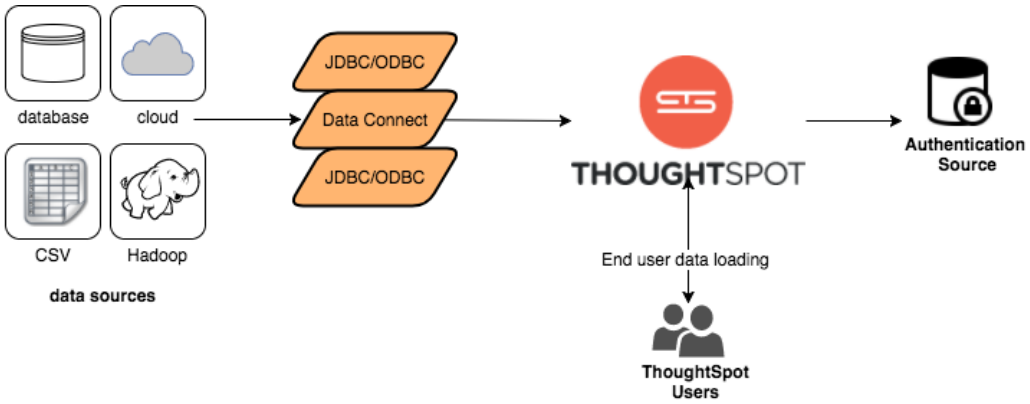
The ThoughtSpot appliance can be a physical appliance that ThoughtSpot ships, one or more AWS instances that are clustered together, or one or more VMware instances that are clustered together. From an external interface, regardless of the appliance type, the appliance appears to be a single instance.

For authentication (logging in), some source of user information is required. These define the login requirements and access control groups. Users will access the data from a supported browser to view saved content or perform searched-based analytics. Finally, it is recommended that you have some sort of networked attached storage for storing backups in case of hardware failure.

Data Caching

> **Summary:** ThoughtSpot does all analysis against data in memory to help achieve fast results across millions and billions of records of data.

ThoughtSpot caches data as relational tables in memory. The tables can be sourced from different data sources and joined together. ThoughtSpot has four ways to get data into the cluster:



ThoughtSpot provides a JDBC and ODBC driver that can be used to write data to ThoughtSpot. This is useful for customers who already have an existing ETL process or tool and want to extend it to populate the ThoughtSpot cache.

Data Connect is a ThoughtSpot add-on that connects to a wide variety of data sources and pulls data into ThoughtSpot.

You can use the `tsload` command line tool to bulk load delimited data with very high throughput. Finally, individual users can upload smaller (< 50MB) spreadsheets or delimited files.

Which approach you use depends on your environment and data needs.

The following table shows the tradeoffs between different data caching options. Many implementations use a variety of approaches. For example, a solution with a large amount of initial data and smaller daily increments might use tsload to load the initial data and then use the JDBC driver with an ETL tool for incremental loads.

JDBC/ODBC	Data Connect	tsload

- Have an ETL load, for example, Informatica, SSIS, and so forth.
- Have available resources to create and manage ETL.
- Have smaller daily loads.

- Purchased as an add-on.
- Source data is well formed for ThoughtSpot or it can be modified prior to being loaded.
- Have smaller daily loads.

- Initial data load.
- When JDBC/ODBC and Data Connect are not options.
- When there are large recurring daily loads.
- Higher throughput but can add I/O costs.

Authentication

Summary: ThoughtSpot provides LDAP/AD, SAML, and ThoughtSpot login to authenticate users.

ThoughtSpot provides three ways to authenticate users LDAP/AD, SAML, and ThoughtSpot login. In general, ThoughtSpot recommends that you use LDAP/AD or SAML if possible since ThoughtSpot provides only basic authentication with no restrictions on passwords, timeouts, failed logins, etc.

The table below shows each of the options and the items to consider for each.

SAML	LDAP/AD	ThoughtSpot
• Use SAML for single sign-on authentication. • Can redirect from ThoughtSpot to SAML logins. • Recommended for portal integration. • Option to sync users and groups if stored in LDAP/AD.	• Configuration. • Users authenticate against LDAP or AD. • Option to sync users and groups with ThoughtSpot to manage group membership.	• User created and managed in ThoughtSpot. • No enterprise password control (expiration, password strength, etc.). • Only recommended when SAML and LDAP are not options.

All users and groups must be known to ThoughtSpot. If you are using LDAP/AD or SAML and don't create users in ThoughtSpot, a user is created when the user first logs in. However, this user is assigned to the `All` group and will only see content available for all users.

Groups are the primary way that security is managed. Groups are not automatically created. You can create groups and users manually or you need to automate the assignment from a source system. ThoughtSpot has an assignment script that works with most LDAP / AD stores. It also has public APIs that you can use to sync users and groups between source systems and your ThoughtSpot appliance.

Data and object security

Summary: Understand how to secure your data and other key information in ThoughtSpot.

ThoughtSpot provides these features for protecting data security:

- Object security
- Row level security
- Column level security
- System privileges

Object Security

Object security is the ability for users to see content within ThoughtSpot. Objects can be tables, columns in tables, worksheets, pinboards, and saved answers.

Users gain access to objects when an object owner share-answers with them. Owners can share with individual users or with entire groups, giving access to anyone within that group. Owners can share with edit or view options.

Currently, you cannot restrict someone who has had content shared with them from sharing with others. Also, a user who belongs in a group can automatically share with anyone else in the group. This has implications on setting up privileges and applying row level security.

Row level security (RLS)

Row level security controls what data a user can see in each shared piece of content. Even if a user has access to a worksheet, for example, they can only see rows from the tables they have been given permission to see.

RLS is applied at the table level and automatically applied every time. Also, in queries where there are tables with table filters, all joins are always enforced, to avoid accidentally allowing users access to data they shouldn't see. RLS requires three things:

- A table filter with a column (possibly in a joined table) that can be used to determine who can see a row, for example, account id or tenant id.
- A group that can be associated with the row of data by name. For example, if the column is `account_id` and has values of `1`, `2`, `3`, users can be assigned to groups `group_1`, `group_2`, `group_3` and then only see their data.
- Users must be assigned to the given group. If they are not assigned to a group that has access, they do not see any data.

Administrative users can always see all rows of data since RLS is not applied for these users.

RLS supports a hierarchy of groups allowing you to give access to some users across multiple groups.

Keep in mind that users within a group can share with one another group. This means that putting everyone into a company group for RLS means they can share with anyone in the company.

Column level security (CLS)

Column level security means only allowing users to see certain columns in a table. This can be accomplished by only sharing certain columns with groups of users from a table.

However, most of the time users are given access to worksheets instead of columns. There is currently no way to only share certain worksheet columns with certain groups. If you need this capability, you must create different worksheets with the columns you want.

Also, note that because someone can share with anyone in a group they belong to, that means they could potentially share restricted columns. For example, assume that HR has a column with salary information in a worksheet that only HR has access to. An HR person could create an answer with the salary information and share with someone outside of HR. That person would now have access to the salary information.

System privileges

System privileges refer to what a user can do in ThoughtSpot. For example, can they upload or download data or share with all users. These privileges are defined on a group level and inherit downwards. So, if Group A had child groups Group B and Group C, then any privilege given to Group A is also available to Group B and Group C. What this often means is that separate sets of groups are required to manage privileges.

Performance considerations

Summary: Make sure you understand the performance considerations in your installation.

ThoughtSpot is configured and licensed by memory availability. However, there are other considerations that will impact the performance of your solution. It's important to understand these considerations prior to implementation, since some solutions will perform better than others.

Each node in a ThoughtSpot cluster has been found to perform ideally with less than 250GB of data and fewer than 0.5 billion total rows of data. For schemas that are particularly complex, performance is increased with even fewer rows of data per node. Ways to reduce the total amount of data and rows of data include limiting the amount of data (number of years, etc.) or combining long, but narrow tables together.

The performance information related on this page apply to a typical 1TB 4 node cluster.

Data Boundaries

Total rows in a result of a join can also have an impact. In general, it is recommended that you have fewer than 10 billion rows in a many-to-many join. Keep in mind these other boundaries:

Description	Boundary
Max number of rows that can be downloaded	10M (default is 1M)
Size in CSV format	1 TB per appliance
Total number of rows across all tables	1B per appliance
Many-to-Many (Generic) join cardinality	10B per appliance
Load frequency	Once every hour

Worksheet Boundaries

Worksheets must have less than 1000 columns. For aggregated worksheets, you should keep in mind the following:

- Number of columns should be less than 50
- Number of rows should be less than 10 millions

You can use an ETL process to circumvent these limitations. Speak with ThoughtSpot Customer Support to learn more.

Aggregated worksheets and joins

To be able to join an aggregated worksheet with a base table, your installation must be configured to allow the behavior. The aggregated worksheet cannot have more than 5 tables involved. Moreover, the number of rows in the final aggregated worksheet cannot be greater than 1000.

Chasm Trap worksheets

For chasm trap scenarios (two or more fact tables joined via a shared dimension) the following boundaries are recommended:

Description	Boundary
Max number of fact tables in a worksheet	5
Max number of shared dimensions	2
Max number of rows in *non* co-sharded shared dimension table of chasm trap	1B
Max number of rows in co-sharded shared dimension table of chasm trap	1B

Row level security Boundaries

Max number of unique RLS rules with search data suggestions should not exceed 15K.

Data Connect Boundaries

The maximum number of connections should be less than or equal to 25.

Scheduled pinboards

For scheduled pinboards, ideal performance is to have 50 or fewer scheduled pinboard jobs.

About installation and upgrades

Your ThoughtSpot application software is already installed for you in a ThoughtSpot appliance (this is true for both physical and virtual appliances). The ThoughtSpot software is updated by ThoughtSpot Support. ThoughtSpot Support will contact you to schedule an update when one becomes available.

As administrator, you are responsible for setting up and configuring ThoughtSpot. This guide explains how. It will also assist you in troubleshooting some common problems, finding additional resources, and contacting ThoughtSpot.

Display your current configuration

To perform the set up and configuration, you must first learn how to gain administrative access.

1. Log into the ThoughtSpot cluster as the `admin` user.
2. Use the `tscli feature` subcommand to display your current configuration.

```
 $ tscli feature get-all-config
 +---------------------------------+----------
+---------------+
 |              NAME               |  STATUS  | CONFIGUR
ATION |
 +---------------------------------+----------
+---------------+
 | Firewall                        | Disabled
|               |
 | Saml                            | Disabled
|               |
 | Ldap                            | Disabled
|               |
 | CustomBranding                  | Disabled
|               |
 | CustomBrandingFontCustomization | Disabled
|               |
 | DataConnect                     | Disabled
|               |
 | RLS                             | Enabled
|               |
 | Callhome                        | Enabled
|               |
 | SSHTunnel                       | Enabled
|               |
 | Fileserver                      | Disabled
|               |
 +---------------------------------+----------
+---------------+
```

Related information

The following tasks are available needed:

- Set your locale
- Test connectivity between nodes
- Set the relay host for SMTP
- Set up a fiscal calendar year
- Configure SSL
- Configure SAML
- Integrate LDAP
- Configure NAS file system
- Set up monitoring
- Configure support services
- Network ports
- Configure load balancing and proxies
- Customize look and feel

- Add the Slack integration

Set your ThoughtSpot locale

Summary: You can change the language displayed in the application.

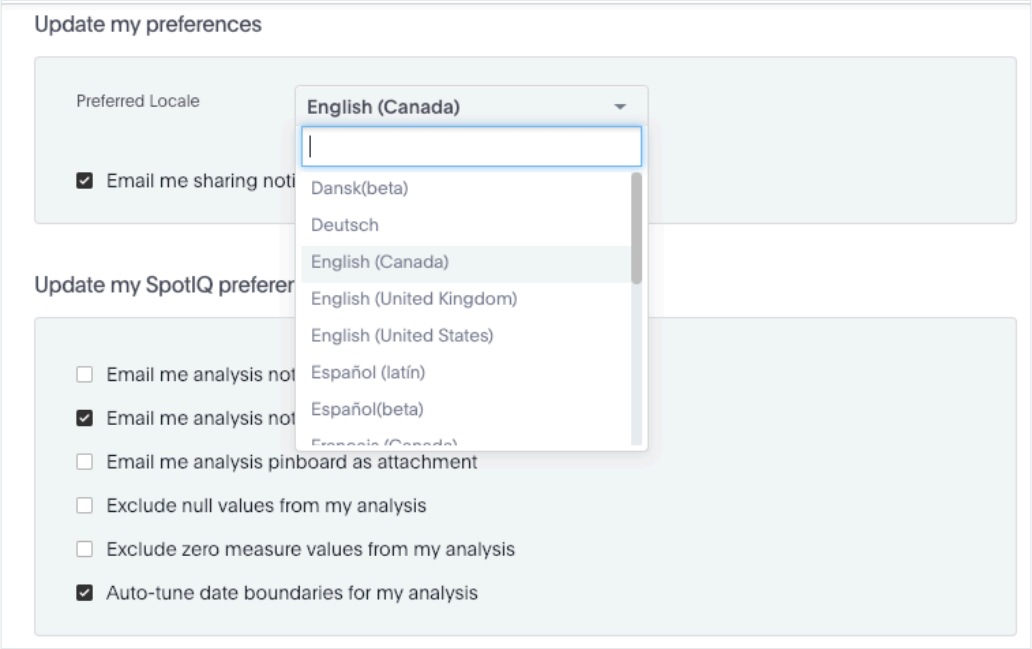
The language the ThoughtSpot UI displays is based off of the locale in a user's profile. The locale preferences control the language and data formats (date and number formats) by geographic locations. In addition to American English (*en-US*), ThoughtSpot supports:

Locale	Language
da-DK	Dansk (beta)
de-DE	Deutsche
en-CA	English (Canada)
en-GB	English (United Kingdom)
en-US	English (United States)
es-US	Español (latín)
es-ES	Español (España - beta)
fr-CA	Français (Canada)
fr-FR	Français (France)
it-IT	Italiano (beta)
nl-NL	Nederland (beta)
nb-NO	Norsk (beta)
pt-BR	Português (Brazil)
pt-PT	Português (Portugal - beta)
fi-FI	Suomi (beta)
sv-SE	Svenska (beta)
zh-CN	中文(简体)
ja-JP	日本語

Date and number formats change to reflect your locale. So, if you set Japanese as your default locale in your profile settings, then the interface will update to reflect that after you refresh your page.

Keywords, operators, and error messages are included in the translated material. (A keyword reference for all supported languages is included in this documentation under "Keywords in Other Languages".)

Formulas, however, are *not translated*. Also, all metadata remains as user inputted.



For example, if you are using ThoughtSpot in the US, the number formatting should look like this: `xxx,xxx.xx`. And in Europe, it should look like this: `xxx.xxx,xx`.

Test network connectivity between nodes

Summary: Verify your network is properly configured for the application.

This procedure tests the network connectivity between the ThoughtSpot nodes, and to the LAN. If you can perform these steps successfully, the network settings on ThoughtSpot are correct.

1. Log in to the Linux shell using SSH.
2. Ping each of the other nodes in the cluster.
3. Ping another machine that exists outside of the cluster, for example, a machine that you will use to stage data to be loaded.

If you cannot perform these tests successfully, there is a problem with the network setup. If the tests fail, check Network connectivity issues.

Set the relay host for SMTP (email)

Summary: To enable alert emails, you'll need to set up a relay host for SMTP traffic.

ThoughtSpot uses emails for sending critical notifications to ThoughtSpot Support. A relay host for SMTP traffic routes the alert and notification emails coming from ThoughtSpot through an SMTP email server.

Set up the relay Host

To set up a relay host:

1. Log in to the Linux shell using SSH.
2. Issue the setup command, providing the IP address of the relay host:

```
$ tscli smtp set-relayhost <IP_address>
```

3. Verify your settings:

```
$ tscli smtp show-relayhost
```

4. Verify that email is working.

Configure an email to receive alerts

ThoughtSpot sends alerts to the email address specified during installation. If no email address was entered, no alerts are sent. You should add an email to receive alerts by issuing:

```
$ tscli monitoring set-config --email <your_email>
```

To send to multiple emails, provide a comma-separated list with no spaces.

Verify the relay with an email

Check if the email settings are working properly by using this procedure.

1. Log in to the Linux shell using SSH.
2. Try sending an email to yourself by issuing:

```
$ echo | mail -s Hello <your_email>
```

3. If you receive the email at the address(es) you supplied, email is working correctly.

Set up a fiscal calendar year

Summary: You can customize your fiscal calendar to start in month other than January.

By default, the application's fiscal calendar in January. If your company's calendar year starts in another month, setting a fiscal calendar quarter makes the ThoughtSpot date searches reflect your fiscal year.

Date formulas with the `fiscal` option specified will also reflect the fiscal year you set here.

When you set a custom fiscal year, you will designate the month on which your company's fiscal year begins. All the date language will then reflect your change, so if someone searches for **this quarter** or **q3**, the answer will conform to the fiscal quarter in use. When you make this change, existing pinboards also change to reflect the custom fiscal calendar. Because of this, if you make this change after your users have been using ThoughtSpot for any period of time, you should alert them of the change you will be making and how it affects previous saved searches.

Contact ThoughtSpot Support, so they can help you set the custom fiscal year.

Configure SSL

Summary: SSL provides authentication and data security

You should use SSL (secure socket layers) for sending data to and from ThoughtSpot. SSL provides authentication and data security. This section applies to both SSL to enable secure HTTP and secure LDAP.

About SSL

Many IT departments require SSL for their applications that access data. To use SSL with ThoughtSpot, you'll need your company's own SSL certificate. The certificate is issued per domain, so if you want to use SSL for both HTTP and LDAP, you will need two separate certificates - one for the HTTP domain and one for the LDAP domain.

If you do not have an SSL certificate:

- Check with your IT department to see if they already have an SSL certificate you can use.
- If not, you will need to obtain the certificate from an issuing authority.
- Alternatively, you may disable SSL if you don't want the security it provides by using the command `tscli ssl off`.

There are many SSL vendors to choose from. Check with your existing Web hosting provider first, to see if they can provide the certificate for you.

When you apply for the SSL certificate, you may specify a SAN, wildcard, or single domain certificate. Any of these can work with ThoughtSpot.

Configure SSL for web traffic

This procedure shows how to add SSL (secure socket layers) to enable secure HTTP (HTTPS) in ThoughtSpot. To set up SSL, you will need:

- The SSL certificate
- The private key

To install the SSL certificate:

1. Follow the instructions from your certifying authority to obtain the certificate. This is usually sent via email or available by download.
2. Copy the certificate and key files to ThoughtSpot:

```
$ scp <key> <certificate> admin@<IP_address>:<path>
```

3. Log in to the Linux shell using SSH.
4. Change directories to where you copied the certificate:

```
$ cd <path>
```

5. Issue the `tscli` command to install the certificate:

```
$ tscli ssl add-cert <key> <certificate>
```

6. To test that the certificate was installed correctly, Log in to the ThoughtSpot application.

   You should see that the application's URL begins with `https://`.

Set the recommended TLS version

There are a couple of security vulnerabilities due to SSL certificates supporting older versions of TLS (Transport Layer Security). This procedure shows you how to set the recommended TLS version to avoid these vulnerabilities.

The PCI (Payment Card Industry) Data Security Standard and the FIPS 140-2 Standard require a minimum of TLS v1.1 and recommends TLS v1.2.

ThoughtSpot supports SSL v3, TLS v1.0, and TLS v1.1 for backwards compatibility. However, the recommended version is TLS v1.2. Therefore, to set the recommended TLS version:

1. Enable your web browser to support TLS v1.2. This can be done in your browser's advanced settings.
2. Log in to the Linux shell using SSH..
3. Issue the following command:

```
tscli ssl set-min-version 1.2
```

   This will block all usage of older versions.

Configure SAML

Summary: You can use the Security Assertion Markup Language (SAML) to authenticate users

You can set up SAML through the shell on ThoughtSpot using a `tscli` based configurator.

Prerequisites

Before configuring SAML, you will need this information:

- IP of the server where your ThoughtSpot instance is running.
- Port of the server where your ThoughtSpot instance is running.
- Protocol, or the authentication mechanism for ThoughtSpot.
- Unique service name that is used as the unique key by IDP to identify the client.

  It should be in the following format: `urn:thoughtspot:callosum:saml`

- Allowed skew time, which is the time after authentication response is rejected and sent back from the IDP. It is usually set to 86400.
- The absolute path to the `idp-meta.xml` file. This is needed so that the configuration persists over upgrades.
- This configurator also checks with the user if internal authentication needs to be set or not. This internal authentication mechanism is used to authenticate `tsadmin`, so set it to true if you do not know what it does.

Use tscli to configure SAML

Use this procedure to set up SAML on ThoughtSpot for user authentication. Note that this configuration persists across software updates, so you do not need to reapply it if you update to a newer release of ThoughtSpot.

1. Log in to the Linux shell using SSH.
2. Execute the command to launch the interactive SAML configuration:

```
tscli saml configure
```

3. Complete the configurator prompts with the information you gathered above.
4. When the configuration is complete, open a Web browser and go to the ThoughtSpot login page. It should now show the Single Sign On option.

Enable Active Directory based access

Summary: ThoughtSpot supports enabling Active Directory (AD) based access individually on each node where the commands are run.

Enable Active Directory based access on a ThoughtSpot node

ThoughtSpot supports enabling Active Directory (AD) based access individually on each node where the commands are run. There is no provision to enable AD access for the whole cluster with a single command. To enable AD access on a cluster, you need to run these commands on each individual node and on any additional nodes added to the cluster.

The command to enable system AD user access is:

```
tscli sssd enable --user <USER> --domain <DOMAIN>
```

You will then be prompted for password credentials.

Note: The user must have permission to join a computer or VM to the domain.

Set sudoers AD Group on a local node

Just like enabling AD based access on a node, setting `sudo` AD group applies only on the node where the command is run, and is not set for the whole cluster.

The command to allow `sudo` permissions for AD group:

```
tscli sssd set-sudo-group <ACTIVE_DIRECTORY_GROUP_NAME>
```

Clear sudoers AD Group on a local node

Clearing `sudo` AD group only applies on the node where command is run, and is not set for the whole cluster.

The command to clear `sudo` permissions for the AD group:

```
tscli sssd clear-sudo-group <ACTIVE_DIRECTORY_GROUP_NAME>
```

Disable AD based access on a local node

Currently ThoughtSpot supports disabling AD based access individually on each node where the commands are run. There is no provision to disable AD access for the whole cluster with a single command. To disable AD access on a cluster, run these commands on each individual node and any additional nodes added to the cluster.

Command to disable system AD user access is:

```
tscli sssd disable
```

Note: Running this command will also remove the AD group from sudoers list.

Related Information

- sssd in the `tscli` command reference

About LDAP integration

Summary: You authenticate users against an LDAP server.

Some companies use LDAP (Lightweight Directory Access Protocol) to manage user authentication. Using LDAP provides security and makes user management more centralized. You can choose to authenticate users against an LDAP server, against ThoughtSpot, or both.

ThoughtSpot supports both anonymous and non-anonymous LDAP integration. Non-anonymous LDAP binding is more rigorous than anonymous authentication, but it should help you track what your users are querying and keep a log trace for auditing purposes.

If you have been using ThoughtSpot with users you created manually, and you now want to transition to LDAP, please contact ThoughtSpot Support. They can assist you in migrating existing users to their LDAP equivalents.

ThoughtSpot supports LDAP with Active Directory.

Configure LDAP for Active Directory

Summary: Use this procedure to set up integration with LDAP using Active Directory.

Before you configure LDAP for Active Directory, collect this information:

- URL to connect to Active Directory.

  For example, `ldap://192.168.2.48:389`

- Default LDAP domain.

  The default domain is the domain under which users who want to be authenticated against Active Directory reside. When a user logs in with a username, the default domain is added to the username before sending it to the LDAP server. If users reside in multiple domains, you can still designate one of them as the default. Users belonging to a non-default domain will have to explicitly qualify their username when they log in, for example: `username@ldap1.thoughtspot.com`.

- Whether you will use SSL.

  If yes, you'll need the certificate from the issuing authority.

- LDAP search base.

  This prompt adds the search base information that allows ThoughtSpot to find user properties such as email and displayname from LDAP.

- Automatically add LDAP users in ThoughtSpot?

  If you choose 'yes' for this, when a user is authenticated against LDAP, if that user does not exist in ThoughtSpot, then the user is automatically created. When users are created in this way, their passwords exist only in LDAP and are not stored in ThoughtSpot.

  In order to log in to ThoughtSpot, the user has to exist in ThoughtSpot independent of whether that user is authenticated against LDAP or against ThoughtSpot's internal authentication. If you choose 'no' for this, users who will authenticate against LDAP have to be manually created with a dummy password as a placeholder in ThoughtSpot before they can log in. The username you specify when creating the LDAP authenticated user manually in ThoughtSpot has to be domain qualified, for example: `username@ldap1.thoughtspot.com`.

- Also use ThoughtSpot internal authentication?

  If you choose 'yes' for this, when a user logs in, ThoughtSpot will first attempt to authenticate the user against LDAP. If that attempt fails, it will then attempt to authenticate the user against ThoughtSpot. If either of these succeed, then the user is successfully logged in. This option is useful in scenarios where some users are not in LDAP and are created only in ThoughtSpot.

You do not need to create a user called `tsadmin` on your LDAP server. Internal authentication can be used for `tsadmin`. To configure LDAP:

1. Log in to the Linux shell using SSH.
2. Run the command to configure LDAP:

```
$ tscli ldap configure
```

3. Answer the prompts using the information you collected. For example:

```
 Choose the LDAP protocol:
 [1] Active Directory
 Option number: 1

 Configuring Active Directory

 URL to connect to Active Directory. (Example: ldap://19
2.168.2.100:389): ldap://192.168.2.100:389

 Default domain (Example: ldap.thoughtspot.com): ldap.th
oughtspot.com

 Use SSL (LDAPS) (y/n): n

 LDAP search base (Example: cn=Users): cn=Users

 Automatically add LDAP users in ThoughtSpot (y/n): y

 Also use ThoughtSpot internal authentication (y/n): y
```

4. If you are using SSL, Add the SSL certificate for LDAP.
5. If you want to remove the LDAP configuration, issue:

```
$ tscli ldap purge-configuration
```

Add the SSL certificate for LDAP

Summary: Install the certificate to support LDAPS

When you set up LDAP, you specified whether or not to use SSL for LDAP (LDAPS). If using SSL, you must install the LDAP SSL certificate. Before you can add the SSL certificate, you must Configure LDAP for Active Directory.

You must have the SSL certificate before you start. For more information on obtaining an SSL certificate, see Configure SSL (secure socket layers).

To add the SSL certificate for LDAP:

1. Follow the instructions from your certifying authority to obtain the certificate. This is usually sent via email or available by download.
2. Copy the certificate to ThoughtSpot:

```
$ scp <certificate> admin@<IP_address>:<path>
```

3. Log in to the Linux shell using SSH.
4. Change directories to where you copied the certificate:

```
$ cd <path>
```

5. Run the command to configure SSL for LDAP, designating an alias for this certificate using the `<name>` parameter:

```
$ tscli ldap add-cert <name> <certificate>
```

Test the LDAP configuration

Summary: This procedure allows you to test the LDAP connection you created.

After configuring LDAP, you can test to make sure it is working by issuing a command.

1. Log in to the Linux shell using SSH.
2. Issue the LDAP testing command, supplying the information for the LDAP server you configured, as in this example:

```
 $ ldapsearch -x -h 192.168.2.61 -p 389 -D "testuser@lda
p.thoughtspot.com" -W -b "dc=ldap,dc=thoughtspot,dc=co
m" cn
```

3. Supply the LDAP password when prompted.
4. If the connection works, you'll see a confirmation message.

Sync users and groups from LDAP

Summary: Use this procedure to synchronize your ThoughtSpot system with an LDAP server.

Before synchronizing users and groups, you will need this information:

- IP address and port of the server where your ThoughtSpot instance is running. This hostport is needed in the following format `http(s)://<host>:<port>` or `http(s)://<domain>`.
- Administrator login username and password for your ThoughtSpot instance.
- URL of the LDAP server, or hostport.

  For example, `ldap://192.168.2.48:389`

- Login username and password for the LDAP system.

  An example username would be `moo_100@ldap.thoughtspot.com`

- Distinguished Name (DN) for the base to start searching for users in the LDAP system.

  For example, `DC=ldap,DC=thoughtspot,DC=com`

- Location of the Python synchronization script, in case you want to modify it or create your own: `/usr/local/scaligent/release/callosum/utilities/ldap_sync_python_api/syncUsersAndGroups.py`

There are two ways for you to fetch users and groups from LDAP and populate them into your ThoughtSpot system:

- Run the synchronization script in interactive mode, which will walk you through the process (shown here).
- Create your own Python script by using the ThoughtSpot Python APIs. If you need details on the Python APIs, contact ThoughtSpot Support. If you choose this method, you can run the script periodically using a cron job.

To run the LDAP sync script in interactive mode:

1. Log in to the Linux shell using SSH.
2. Run the command to start the script:

```
python syncUsersAndGroups.py interactive
```

3. Answer the prompts using the information you collected above. For example:

```
 Complete URL of TS server in format "http(s)://<hos
t>:<port>": http://10.77.145.24:8088
 Disable SSL authentication to TS server (y/n): y
 Login username for ThoughtSpot system: admin
 Login password for ThoughtSpot system: 12345
 Complete URL of server where LDAP server is running in
format ldap(s)://<host>:<port>: ldap://192.168.2.48:389
 Login username for LDAP system: moo_100@ldap.thoughtspo
t.com
 Login password for LDAP system: 12345
 Syncs user and groups between LDAP and TS systems (y/
n): y
 Delete entries in ThoughtSpot system that are not curre
ntly in LDAP tree being synced (y/n): n
 Distinguished name for the base to start searching grou
ps in LDAP System: DC=ldap,DC=thoughtspot,DC=com
 Scope to limit the search to (choice number)
 0:base Searching only the entry at the base DN
 1:one Searching all entries on level under the base DN
- but not including the base DN
 2:tree Searching of all entries at all levels under an
d including the specified base DN: 2
```

```
 Filter string to apply the search to: (|(CN=TestGroupAl
pha)(CN=TestGroupBeta))
```

Answering this prompt is optional. If left blank, the default value of `'(CN=*)'` will be used.

```
 Apply sync recursively, i.e. Iterates through group mem
bers and creates member groups, users and relationships
in a recursive way. (y/n): n
```

This prompt is asking if you would like to include group members even if they do not belong to the current sub tree that is being synced.

4. Alternatively, to input your own shorthand script commands:

   Issue the Python script commands, supplying all of the above information, following this format example:

```
 python syncUsersAndGroups.py script \
 --ts_hostport <ts_hostport> \
 --disable_ssl \
 --ts_uname <ts_username> \
 --ts_pass <ts_password> \
 --ldap_hostport '<ldap_hostport>' \
 --ldap_uname '<ldap_username>' \
 --ldap_pass '<ldap_password>' \
 --sync \
 --purge \
 --basedn 'DC=ldap,DC=thoughtspot,DC=com' \
 --filter_str '(|(CN=TestGroupAlpha)(CN=TestGroupBet
a))' \
 --include_nontree_members
```

The bottom half of the above command example targets sub trees under the DC called TestGroupAlpha and TestGroupBeta, and iterates through them recursively to create/sync users, groups, and their relationships in the ThoughtSpot system. It also deletes any other entities created in the ThoughtSpot system from this LDAP system that are not currently being synced.

Configure NAS file system

Summary: You can use network attached storage to support backup/restore and data loading.

Some operations, like backup/restore and data loading, require you to either read or write large files. You can mount a NAS (network attached storage) file system for these operations. Currently, ThoughtSpot does not have an option for direct attached storage. Your NAS storage can use whichever drive format you would like.

This procedure shows you how to mount a NAS file system for storing or accessing large files. The file system will be mounted at the same location on each node in the cluster automatically. When any node is restarted, the file system will be mounted again automatically, if it can be found.

When supplying a directory for writing or reading a backup, you can specify the `mount` point as the directory to use. Likewise, you can stage data there for loading.

Backups are written by the Linux user `admin`. If that user does not have permission to write to the NAS file system, you could write the backups to disk (for example `/export/sdc1`, `/export/sdd1`, `/export/sde1`, or `/export/sdf1`) and then set up a cron job that executes as root user and copies the backup to the NAS device every night, then deletes it from the directory.

Do not send the periodic backups or stage files on `/export/sdb1` since it is a name node. It is used internally by Hadoop Distributed File System (HDFS) and if this drive fills up, it can cause serious problems. Do not allow backups or data files to accumulate on ThoughtSpot. If disk space becomes limited, the system will not function normally.

1. Log in to the Linux shell using SSH.
2. Mount the directory to the file system, by issuing the appropriate command:
   - For an NFS (Network File System) directory:

```
tscli nas mount-nfs
   --server <server_NFS_address>
   --path_on_server <path>
   --mount_point <target>
   --options vers=<version>, sec=<security schem
e>, <OPTIONS>
```

> **Note:** Other command-line options are available to forward to the command (default: `noexec`).

   - For a CIFS (Common Internet File System) directory:

```
tscli nas mount-cifs
   --server <server_CIFS_address>
   --path_on_server <path>
   --mount_point <target>
   --username <user>
   --password <password>
   --uid <uid>
   --gid <gid>
   --options <OPTIONS>
```

Note: Other command-line options are available to forward to the `mount.cifs` command (default: `noexec`).

3. Use the mounted file system as you wish, specifying it by referring to its mount point.
4. When you are finished with it, you may optionally unmount the NAS file system:

```
tscli nas unmount --dir <directory>
```

Set up monitoring

Summary: Setting up monitoring is a one time operation.

To configure monitoring of your cluster, set up the frequency of heartbeat and monitoring reports and an email address to receive them.

1. Log in to the Linux shell using SSH.
2. Issue the `tscli` command to set up monitoring:

```
tscli monitoring set-config
   --email <email>
   --heartbeat_interval <heartbeat_interval>
   --report_interval <report_interval>
```

The parameters are:

- `--email <email>` is a comma separated list (no spaces) of email addresses where the cluster will send monitoring information.
- `--heartbeat_interval <heartbeat_interval>` is the heartbeat email generation interval in seconds. Must be greater than 0.
- `--report_interval <report_interval>` sets the cluster report email generation interval in seconds. Must be greater than 0.

3. To view your settings and verify that they have been applied, issue:

```
tscli monitoring show-config
```

You should see information like:

```
Monitoring Configuration:

  Alert Email: dev-alerts@thoughtspot.com

  Heartbeat Interval: 900 sec

  Report Interval: 21600 sec
```

4. After the heartbeat interval has passed, check your email to verify that emails are being delivered.
5. If you don't receive any emails, verify that email is working.

Configure support services

Summary: Set up and configure ThoughtSpot support services for your installation.

There are several configurations you can set up in your installation to ensure your company's support from ThoughtSpot works smoothly.

Set up a reverse tunnel for support

You can set up a reverse tunnel to allow ThoughtSpot Support to get access to your ThoughtSpot instance, to perform support-related activities. This setup is a much simpler, more secure, and scalable than the alternative option of using a virtual meeting room.

Granting remote support access can streamline troubleshooting activities, since it enables your support agent to work directly in a secure setting. The remote tunnel enables SSH and HTTP access to your by ThoughtSpot Support. This access can be granted and revoked easily, so you can enable it for a troubleshooting session, and then disable it again. Before doing this procedure, make sure your company's security policies allow reverse tunneling.

Before you can do this procedure, your networking team needs to open port `22` in your firewall outgoing rules.

To enable remote support:

1. Contact ThoughtSpot and open a support ticket for making the appropriate reverse tunnel settings on our end. Provide the cluster name of the cluster for which you want to enable remote support.
2. After the ticket is completed, continue with the remaining steps in this procedure to make the settings on your side.
3. Log into the Linux shell using SSH.
4. Issue the command to configure the destination for the remote tunnel.

   You only need to do this once, when you are enabling the tunnel for the very first time. After that, this setting persists when you start and stop the remote tunnel.

   ```
   $ tscli support set-remote --addr tunnel.thoughtspot.com --user ubuntu
   ```

5. Test that the setting is configured:

   ```
   $ tscli support show-remote
   ```

6. Enable the remote tunnel:

```
$ tscli support start-remote
```

7. Contact ThoughtSpot and test the setup with your ThoughtSpot Support contact.
8. After your remote session with ThoughtSpot Support, turn the remote tunnel off, until you need to use it again:

```
$ tscli support stop-remote
```

You can repeat the steps to start and stop the remote tunnel as needed for future support operations.

9. Ensure that the remote tunnel is disabled:

```
$ tscli support show-remote
```

Configure a secure file server

ThoughtSpot Support uses a secure file server to provide new releases and to receive logs and troubleshooting files that you upload. The secure server connection is also required if you want to enable the optional statistics collection using the call home feature.

Before you can upload a file to the secure file server, obtain your user name and password for logging in to the secure file server. You can get these from ThoughtSpot Support.

Configuring the connection to the file server is a one time operation. You do not need to reconfigure the connection unless your password changes. Note that you can do a one time override of the user and password you used to configure the connection. This is done by passing a different user and password on the command line when uploading or downloading a file.

To configure the connection to the secure file server:

1. Log in to the Linux shell using SSH.
2. Issue the command to configure the file server:

```
 $ tscli fileserver configure --user <user_name> [--pass
word <password>]
```

If you do not supply the `--password` parameter, you are prompted to enter it.

Call home with cluster usage data

"Call home" data is metadata and usage data from your ThoughtSpot cluster. This data allows ThoughtSpot's Support team to troubleshoot your cluster. They use the data to see basic usage information over time for your ThoughtSpot instance. ThoughtSpot's "call home" functionality intermittently sends a call home bundle of statistics to a ThoughtSpot S3 server via HTTPS. The data is encrypted at rest on the server.

By default, call home is enabled on your cluster. You can disable call home in by doing the following:

1. Log into the ThoughtSpot server as `admin` user.
2. Use the `tscli` command to disable.

```
$ tscli callhome disable
```

Designate a support contact

A support contact person can answer questions for about data and searching at your company. If the person can't answer a question that person should submit system and software-related questions to ThoughtSpot Support. Your designated support contact should have an available email and phone number.

To designate the custom support contact:

1. Log in to the Linux shell using SSH.
2. Issue the `tscli` command to set the email address:

```
$ tscli support set-admin-email <email_address>
```

3. Issue the `tscli` command to set the phone number:

```
$ tscli support set-admin-phone <phone_number>
```

4. If you need to reset both of these to the default (ThoughtSpot Support), issue:

```
$ tscli support rm-admin-email
$ tscli support rm-admin-phone
```

Manage the feedback contact

Users in ThoughtSpot may be asked for feedback for new or BETA features in the system. By default, feedback goes directly to ThoughtSpot support. Alternatively, you can send feedback to someone in your company, this is useful if you are using ThoughtSpot embedded.

Your designated feedback contact should have an available email. To designate the custom feedback contact, do the following:

1. Log in to the Linux shell using SSH.
2. To set the feedback email address, do the following:

```
$ tscli support set-feedback-email <email_address>
```

3. Verify the email address was set:

```
$ tscli support show-feedback-email
```

If you need to reset the email to the default (ThoughtSpot support), issue:

```
$ tscli support rm-feedback-email
```

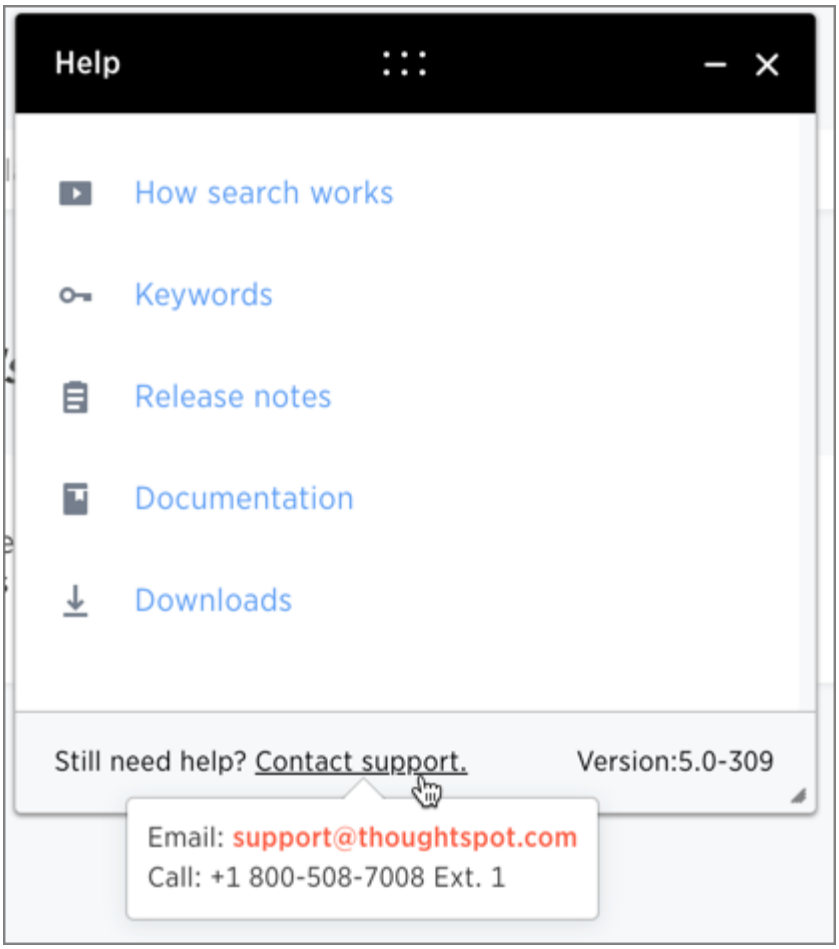
You can also choose not to send feedback out of your system at all by doing the following:

```
$ tscli support set-feedback-email ' '
```

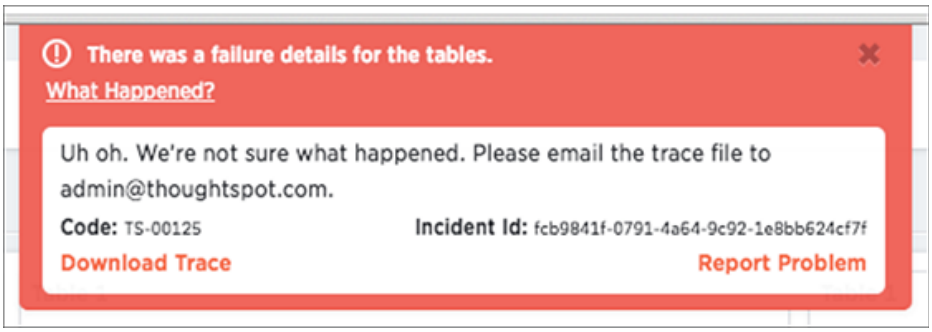
How users find your company's support contact

After you set the custom support contact information, here's where your users will see it:

- In the Help Center, when a user selects **Contact Support**.



- In error messages, when a user selects **What Happened?**



Network ports

Summary: Lists the required and optional ports for an installation.

For regular operations and for debugging, there are some ports you will need to keep open to network traffic from end users. Another, larger list of ports must be kept open for network traffic between the nodes in the cluster.

Required ports for operations and debugging

The following ports need to be opened up to requests from your user population. There are two main categories: operations and debugging.

Port	Protocol	Service Name	Direction	Source	Destination	Description
22	SSH	SSH	bidirectional	Administrators IP addresses	All nodes	Secure shell access. Also used for scp (secure copy).
80	HTTP	HTTP	bidirectional	All users IP addresses	All nodes	Hypertext Transfer Protocol for website traffic.
443	HTTPS	HTTPS	bidirectional	All users IP addresses	All nodes	Secure HTTP.
12345	TCP	Simba	bidirectional	Administrators IP addresses	All nodes	Port used by ODBC and JDBC drivers when connecting to ThoughtSpot.
2201	HTTP	Orion master HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the cluster manager.
2101	HTTP	Oreo HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the node daemon.
4001	HTTP	Falcon worker HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the data cache.
4251	HTTP	Sage master HTTP	bidirectional	Administrator IP addresses	All nodes	Port used to debug the search engine.

Network Ports

This reference lists the potential ports to open when setting up your security group.

Required ports for inter-cluster operation

Internally, ThoughtSpot uses static ports for communication between services in the cluster. Do not close these ports from inter-cluster network communications. In addition, a number of ports are dynamically assigned to services, which change between runs. The dynamic ports come from the range of Linux dynamically allocated ports (20K+).

Port	Protocol	Service Name	Direction	Source	Dest.	Description
80	TCP	nginx	inbound	All nodes	All nodes	Primary app HTTP port (nginx)
443	TCP	Secure nginx	inbound	All nodes	All nodes	Primary app HTTPS port (nginx)
2100	RPC	Oreo RPC port	bidirectional	All nodes	All nodes	Node daemon RPC
2101	HTTP	Oreo HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Node daemon HTTP
2181	RPC	Zookeeper servers listen on this port for client connections	bidirectional	All nodes	All nodes	Zookeeper servers listen on this port for client connections
2200	RPC	Orion master RPC port	bidirectional	All nodes	All nodes	Internal communication with the cluster manager
2201	HTTP	Orion master HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the cluster manager
2210	RPC	Cluster stats service RPC port	bidirectional	All nodes	All nodes	Internal communication with the stats collector
2211	HTTP	Cluster stats service HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the stats collector

Port	Protocol	Service Name	Direction	Source	Dest.	Description
2230	RPC	Callosum stats collector RPC port	bidirectional	All nodes	All nodes	Internal communication with the BI stats collector
2231	HTTP	Callosum stats collector HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the BI stats collector
2240	RPC	Alert manager	bidirectional	All nodes	All nodes	Port where alerting service receives alert events
2888	RPC	Ports used by Zookeeper servers for communication between themselves	bidirectional	All nodes	All nodes	Ports used by Zookeeper servers for communication between themselves
3888	RPC	Ports used by Zookeeper servers for communication between themselves	bidirectional	All nodes	All nodes	Ports used by Zookeeper servers for communication between themselves
4000	RPC	Falcon worker RPC port	bidirectional	All nodes	All nodes	Port used by data cache for communication between themselves
4001	HTTP	Falcon worker HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the data cache
4021	RPC	Sage metadata service port (exported by Tomcat)	bidirectional	Admin IP addresses and all nodes	All nodes	Port where search service contacts metadata service for metadata
4201	HTTP	Sage auto complete server HTTP interface port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the search service
4231	HTTP	Sage index server HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the search service

Port	Protocol	Service Name	Direction	Source	Dest.	Description
4232	RPC	Sage index server metadata subscriber port	bidirectional	All nodes	All nodes	Port used for search service internal communication
4233	RPC	Sage index server RPC port	bidirectional	All nodes	All nodes	Port used for search service internal communication
4241	HTTP	Sage auto complete server HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Port used to debug the search service
4242	RPC	Sage auto complete server RPC port	bidirectional	All nodes	All nodes	Port used for search service internal communication
4243	RPC	Sage auto complete server metadata subscriber port	bidirectional	All nodes	All nodes	Port used for search internal communication
4251	RPC	Sage master RPC port	bidirectional	All nodes	All nodes	Port used for search service internal communication
4405	RPC	Diamond (graphite) port	bidirectional	All nodes	All nodes	Port used for communication with monitoring service
4500	RPC	Trace vault service RPC port	bidirectional	All nodes	All nodes	Trace collection for ThoughtSpot services
4501	HTTP	Trace vault service HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Debug trace collection
4851	RPC	Graphite manager RPC port	bidirectional	All nodes	All nodes	Communication with graphite manager
4852	HTTP	Graphite manager HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Debug graphite manager

Port	Protocol	Service Name	Direction	Source	Dest.	Description
4853	RPC	Elastic search stack (ELK) manager RPC port	bidirectional	All nodes	All nodes	Communication with log search service
4853	HTTP	Elastic search stack (ELK) manager HTTP port	bidirectional	Admin IP addresses and all nodes	All nodes	Debug log search service
5432	Postgres	Postgres database server port	bidirectional	All nodes	All nodes	Communication with Postgres database
8020	RPC	HDFS namenode server RPC port	bidirectional	All nodes	All nodes	Distributed file system (DFS) communication with clients
8080	HTTP	Tomcat	bidirectional	All nodes	All nodes	BI engine communication with clients
8787	HTTP	Periscope (UI) service HTTP port	bidirectional	All nodes	All nodes	Administration UI back end
8888	HTTP	HTTP proxy server (tinyproxy)	bidirectional	All nodes	All nodes	Reverse SSH tunnel
11211	Memcached	Memcached server port	bidirectional	All nodes	All nodes	BI engine cache
12345	ODBC	Simba server port	bidirectional	All nodes	All nodes	Port used for ETL (extract, transform, load)
50070	HTTP	HDFS namenode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS metadata
50075	HTTP	HDFS datanode server HTTP port	bidirectional	All nodes	All nodes	Debug DFS data

Required ports for inbound and outbound cluster access

ThoughtSpot uses static ports for inbound and outbound access to a cluster.

Port	Protocol	Service Name	Direction	Source	Dest.	Description
22	SCP	SSH	bidirectional	ThoughtSpot Support	All nodes	Secure shell access.

Port	Protocol	Service Name	Direction	Source	Dest.	Description
80	HTTP	HTTP	bidirectional	ThoughtSpot Support	All nodes	Hypertext Transfer Protocol for website traffic.
443	HTTPS	HTTPS	bidirectional	ThoughtSpot Support	All nodes	Secure HTTP.
12345	TCP	Simba	bidirectional	ThoughtSpot Support	All nodes	Port used by ODBC and JDBC drivers when connecting to ThoughtSpot.

Port	Protocol	Service Name	Direction	Source	Destination	Description
443	HTTPS	HTTPS	outbound	All nodes	208.83.110.20	For transferring files to thoughtspot.egnyte.com (IP address 208.83.110.20).
25 or 587	SMTP	SMTP or Secure SMTP	outbound	All nodes and SMTP relay (provided by customer)	All nodes	Allow outbound access for the IP address of whichever email relay server is in use. This is for sending alerts to ThoughtSpot Support.
389 or 636	TCP	LDAP or LDAPS	outbound	All nodes and LDAP server (provided by customer)	All nodes	Allow outbound access for the IP address of the LDAP server in use.

Required ports for IPMI (Intelligent Platform Management Interface)

ThoughtSpot uses static ports for out-of-band IPMI communications between the cluster and ThoughtSpot Support.

Port	Protocol	Service Name	Direction	Source	Dest.	Description
80	HTTP	HTTP	bidirectional	ThoughtSpot Support	All nodes	Hypertext Transfer Protocol for website traffic.

Related information

EC2 Best Practices

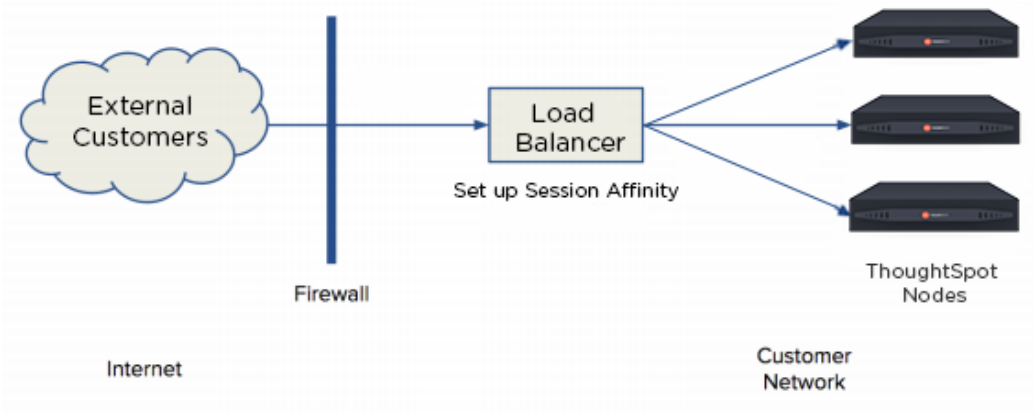
Configure load balancing and proxies

Summary: A load balancer is needed in front of a server group in order to direct traffic to individual servers in a way that maximizes efficiency.

Here are some of the best practices and guidelines for a typical implementation with ThoughtSpot. Your experience may differ depending on your environment and preference.

Load balance across ThoughtSpot nodes

The following shows a network architectural diagram which includes a load balancer for ThoughtSpot nodes.



The load balancer is an appliance in your infrastructure that routes traffic automatically to nodes to provide failover. You can also place a load balancer or proxy in front of the ThoughtSpot appliance if you'd like external network users to access the system.

The best way to load balance across all ThoughtSpot nodes in a cluster is to map one domain name (FQDN) to all the IPs in the cluster in a round robin fashion.

For example, if you want to use a DNS server based load balancing, then you can define multiple "A" resource records (RR) for the same name.

Below is an example of how you could set that up

```
thoughtspot.customer.com IN A 69.9.64.11
thoughtspot.customer.com IN A 69.9.64.12
thoughtspot.customer.com IN A 69.9.64.13
thoughtspot.customer.com IN A 69.9.64.14
```

The example indicates that IP addresses for the domain thoughtspot.customer.com are 69.9.64.11, 69.9.64.12, 69.9.64.13, and 69.9.64.14.

Session Affinity

Session Affinity refers to directing requests to the same application server for the time it takes to complete a task.

In order for session affinity to work on ThoughtSpot, HTTPS (an SSL certificate) has to be installed on the load balancer level. If it is installed outside of the load balancer, session affinity may not occur and the ThoughtSpot system will fail.

Web proxies

You can access ThoughtSpot through any standard web proxy server. Web proxies are fairly universal regardless of the application they are proxying. However, ThoughtSpot doesn't use any new protocols, like SPDY or HTTP/2, which may have a dependency on the proxy. Instead, ThoughtSpot is commonly placed behind a web HTTP/HTTPS proxy.

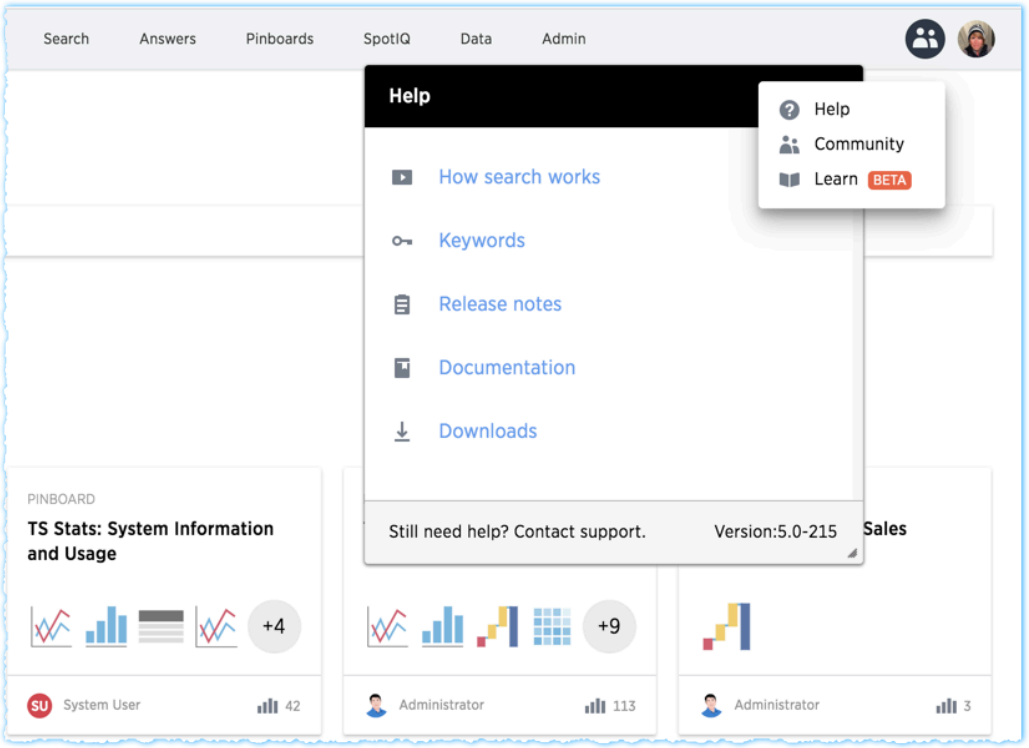
Additionally, the proxy can round robin across multiple nodes in the ThoughtSpot backend. You can essentially use the web proxy as a load balancer. Therefore, your session will carry over if the proxy round robins between the ThoughtSpot backends as long as the URL doesn't change.

Customize ThoughtSpot Help

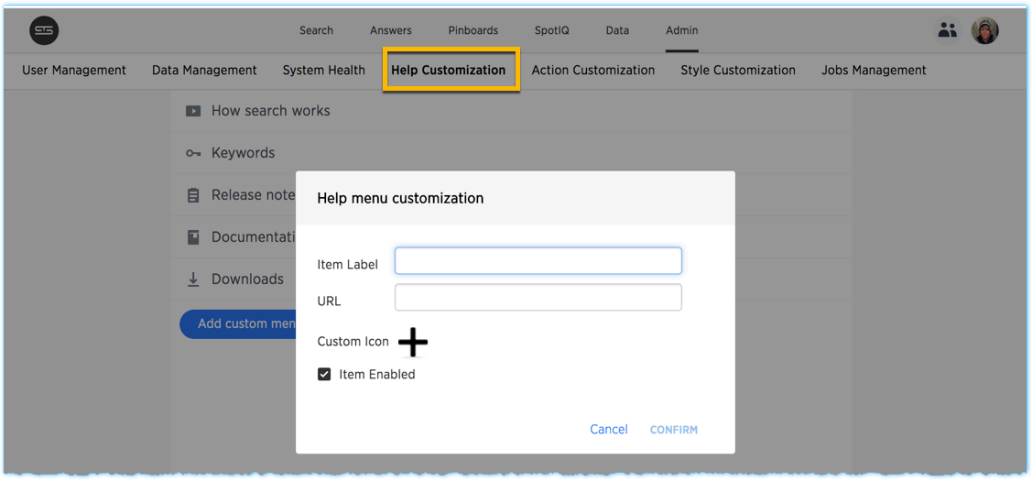
Summary: You customize ThoughtSpot Help to be specific to your data, examples, and documentation.

You can customize the Help for your ThoughtSpot application to tailor it to your organization. Configuring these Help settings sets system-wide defaults for all your users.

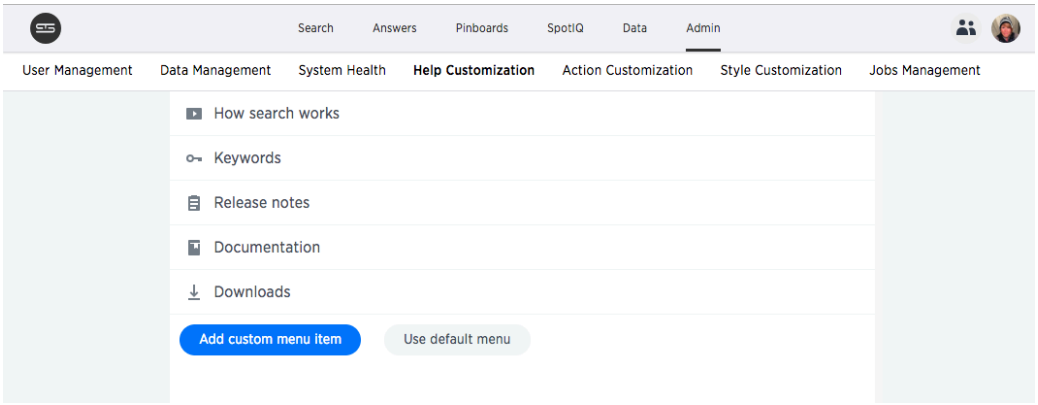
When your ThoughtSpot users click the Help icon, they see a list of links.



As an administrator, you can add your own links to this list. This allows you to include documentation specific to your company, such as information about the data available in ThoughtSpot, where to get support internally, or company-specific training.



You can also edit existing links, change icons, or remove items from the help listing altogether.



Customize look and feel

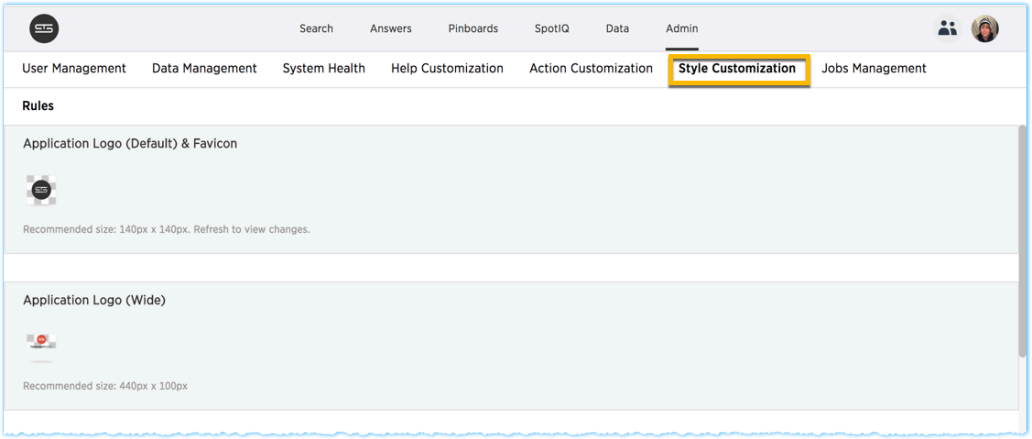
> **Summary:** You can brand the ThoughtSpot application for your company.

You can customize the look and feel of the ThoughtSpot application for your company. Configuring these settings sets system-wide defaults for all your users.

Style customization is enabled by default. You can configure your cluster to disable this functionality. Contact support@thoughtspot.com for information about disabling this feature.

Where to customize styles

A user with administrative rights can view and access the customization on the **Admin** page.



Use the **Style Customization** page to access the configuration settings.

General guidelines for customization

Your changes take effect either immediately or with browser refresh. You can revert your changes by using the **Reset** button which displays when your cursor moves to the right of any setting.

The table below lists the style customizations you can configure.

Setting	Description

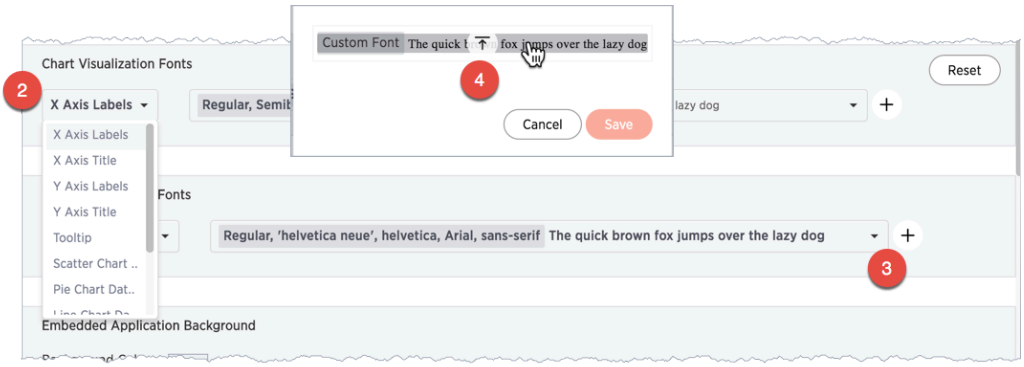
Application Logo (Default) & Favicon	Sets a default application and favicon logo. This should be 140 pixels square.
Application Logo (Wide)	This logo should be 440 x 100 pixels.
Chart Visualization Fonts	Set a font for chart labels. You can specify any Web Open Font Format (`WOFF`) file.
Table Visualization Fonts	Set a font for table labels. You can specify any Web Open Font Format (`WOFF`) file.
Embedded Application Background	Set the background for an embedded ThoughtSpot instance. This is only used if you are embedding ThoughtSpot in another application.
Chart Color Palettes	Set the default palette for all charts. To set a value, however over a color value and enter a HEX value or select one from the chart. Individual users can still customize their own chart colors. They can use the **Reset colors** option on a chart to clear their changes.
Footer text	Define a footer to appear with the ThoughtSpot application.
Page title	Sets the title for the browser tab.

How to specify fonts

You can set your systems default fonts by specifying either or both of the **Chart Visualization Fonts** and **Table Visualization Fonts** values. To set this values, your font must be defined in a Web Open Font Format (`WOFF`) file.

Changes to chart and table defaults apply only to charts and features created *after* you configure a value. If you change table fonts, older tables retain their previous fonts. Some settings can be overridden on a per-object level. Users cannot configure their own defaults.

To set a new font:



1. Make sure you have a WOFF file available for your font.
2. Select the label you want to change.
3. Press the + button.

   The system displays the **Custom Font** dialog.
4. Press the **Custom Font** field.

   The system displays the file finder.
5. Add the `WOFF` file you want.

   The file appears in the font dialog
6. Press **Save** to change the font.

How to specify the behavior or clickable links in data

There is a system-wide setting which determines what happens when a user clicks on a link within the data. When your data includes URLs, they display as clickable links in ThoughtSpot tables. By default, clicking on a link opens the URL in a separate tab. But there is a system-wide setting that can be changed to open the links within the context in which they appear.

Changing this setting opens the links:

Link type	Opens in
Link in search result table in ThoughtSpot	Same browser tab as ThoughtSpot application
Link in table embedded in an iFrame	Same iFrame that contains the table
Link in full ThoughtSpot application embedded in an iFrame	Same iFrame that contains the application

Enable SearchIQ

> **Summary:** Enable SearchIQ to provide natural language search for some or all of your users.

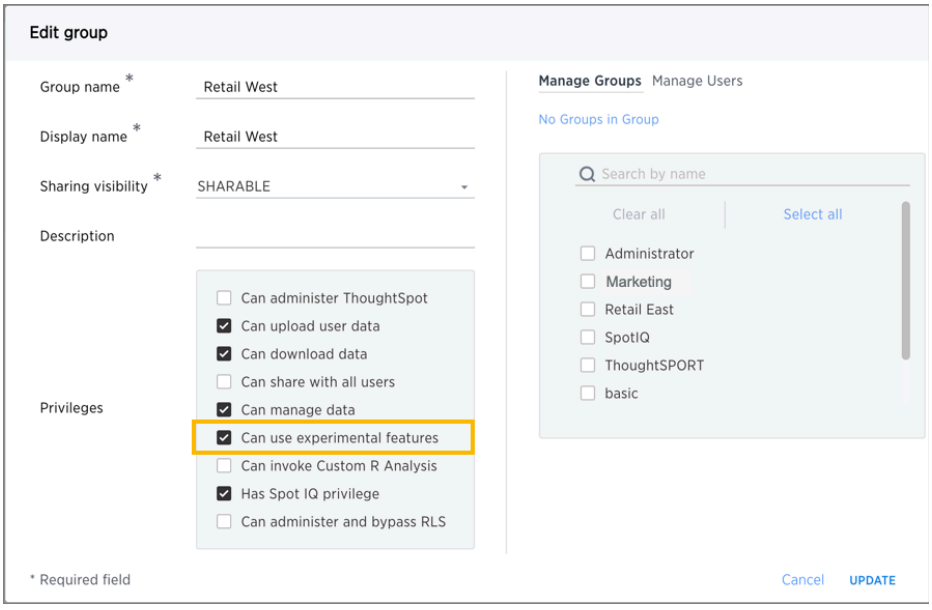
SearchIQ is a different search experience that understands more natural, speech-like search language than the original ThoughtSpot search or Search+. For example, you can search for `What was my top selling product last month?`, instead of having to type `top 1 product by sales last month`. You can also speak your search using the voice-to-text capability of your operating system.

> **Note:** SearchIQ is turned off by default, you can have ThoughtSpot Support enable it for you.

Users can easily teach SearchIQ their language as they use it. Over time, SearchIQ gets smarter about how to interpret terms that don't come directly from keywords, column names, or data values. In order to give it time to learn, the best practice recommendation is to enable SearchIQ for a small group of users initially. Over time you can roll it out to larger groups as SearchIQ learns what your users search for the most.

To enable SearchIQ on a ThoughtSpot instance:

1. Call ThoughtSpot Support and ask them to turn on SearchIQ for your ThoughtSpot instance.
2. Grant the **Can use experimental features** privilege to any group that should have access to the SearchIQ search experience.



> **Note:** Because the SearchIQ search experience is very different from the original ThoughtSpot search experience, you should give your new users some orientation on it, so they know what to expect.

Related information

- About SearchIQ
- Use SearchIQ
- Teach SearchIQ your language

Slack integration

Working with ThoughtSpot, you can configure your installation to work with Slack. Users can use the integration, called **Spot**, to make queries or view charts. Any users with administrative rights can apply a `spot` sticker to specific objects. Then, the object is available through Slack.

The first time a user messages Spot, it returns a link to log into ThoughtSpot. Once a user logs in, the user's Slack and ThoughtSpot account are associated. Actions a user makes from Slack are tied to the user's permissions and authorization.

Spot workflow for administration

Here are the high level steps:

1. Work with support@thoughtspot.com to install the Spot Slack bot on your cluster.
2. Log into ThoughtSpot.
3. Label answers, pinboards, and other objects with the spot sticker.
4. Start Spot Bot.
5. Register Spot bot with your company's Slack instance.
6. Register your Spot Slack account to ThoughtSpot.

Related Information

Relevant `tscli` commands are here, but these will not work until Spot is enabled by ThoughtSpot Support. Support will work with you to install Spot, and then provide the rest of the workflow to you, including `tscli` command usage.

Load and manage data

Summary: There are several methods of loading data into ThoughtSpot. This section describes each method and why you might choose it above the others. Consider setting up data-cleansing tasks alongside data-load tasks to make the most of your ThoughtSpot cluster's capacity

The fastest and easiest way to load a new table is by importing it using the Web browser. This is best for one time data loads of small tables which do not have complex relationships to other tables. This method is limited to tables that are under 50 MB (megabytes) in size.

Using ThoughtSpot Loader, you can script recurring loads and work with multi-table schemas.

If your data already exists in another database with the schema you want to use in ThoughtSpot, you can pull the schema and data in using the ODBC or JDBC driver.

These are the methods you can use to load data, along with the benefits of each method:

Method	Description	Benefits
Load data from the ThoughtSpot UI	Use the ThoughtSpot Web interface to upload an Excel or CSV (comma separated values) file from your local machine.	Easy way to do a one-time data load of a small file (under 50MB).End users can upload their own data and explore it quickly.
Use ThoughtSpot Data Connect. For details, see the ThoughtSpot Data Connect Guide	This is a premium feature, available at additional cost. Use ThoughtSpot Data Connect to connect directly to external data sources and pull in tables and columns from them. You can also set up recurring loads to keep the data fresh.	Easy way to connect to multiple sources of data directly and set up recurring loads. You won't need to define a schema to accept the data loads, because this is done automatically for you.
Import with the ThoughtSpot Loader (tsload)	Use TQL and tsload to load data directly into the back end database that ThoughtSpot uses.	Best way to load large amounts of data or a schema with multiple tables.Can be scripted and used for recurring data loads, such as monthly sales results or daily logs. Can be integrated with an ETL solution for automation.
Use the ODBC/JDBC driver to connect to ThoughtSpot	Use the ODBC or JDBC client with your ETL tool. For information, see the ThoughtSpot Data Integration Guide.	Make use of an established ETL process and tool(s).Connect to ThoughtSpot using third party tools like SSIS. You don't need to define a schema to accept the data load.

If you're uploading data through the Web interface, you can use a native Excel file. If you want to use a CSV (comma separated values) or delimited file, or you are loading using ThoughtSpot Loader, you'll need to create CSV files with the data to be loaded first.

Note: End users will almost always work with worksheets and data they upload.

Related Information

- Load CSV files with the UI
- Append data through the UI
- Schema planning concepts
- Overview of schema building
- Import CSV files with tsload
- How to view a data schema

Configure casing

Summary: You can set the type of case sensitivity you would like to see reflected in the ThoughtSpot display.

Before you load your data, you should consider the type of casing you would like your data to reflect. The case sensitivity for source data strings is preserved in the display. So, the visual display of results is identical to the input case that is loaded.

> **Note:** The casing will remain lowercase in other parts of the application, such as when you ask a question or filter.

It is important to note that string casings aren't applied globally, but by column. So datasets will have different string casings as long as they're in different columns. Tables that are already compacted will keep their lowercase format. In these cases, to get the specific string case that you want, you would have to truncate related tables and reload them.

To take advantage of case configuration, you need to have ThoughtSpot Support enable it on your cluster for you. In addition, title casing should be disabled for string casing to properly work.

Load CSV files with the UI

Summary: The simplest way to load data is to upload a CSV or Excel file from the ThoughtSpot Web interface.

Loading data through the Web browser is recommended for smaller tables (under 50MB) with simple relationships between them. This method is recommended for small, one time data loads. Using this method, the data schema is created for you automatically.

Any user who belongs to a group that has the privilege **Has administration privileges** or **Can upload user data** can upload their own data from the browser.

Your data should be in a CSV (comma separated values) before you load it. A CSV file is a text file made up of data fields separated by a delimiter and optionally enclosed with an enclosing character. If your data contains multiple tables, you'll have a separate CSV for each table.

Formatting the CSV

Your ETL (extract, transform, load) process will typically generate CSV files. You can also create a CSV file from a Microsoft Excel spreadsheet by opening the spreadsheet in Excel, choosing **Save As** and selecting CSV.

A CSV file contains a delimiter that marks the separation between fields in the data. The delimiter is usually comma, but it can be any character. The file also contains fields optionally enclosed with double quotes. Use these guidelines when creating the CSV file:

- If the CSV contains column headers, they must match the column names in the database exactly.
- Often a `|` (pipe) or tab is used as the delimiter, because it may be less likely to occur within the data values.
- When a field contains a double quote, it must be escaped with the character specified in the escape character argument in `tsload`.
- When a field contains the delimiter, the field must be enclosed in double quotes.

ThoughtSpot supports a wide range of date and timestamp formats in the CSV file. Blank values in user uploaded CSV files are interpreted as NULL values. These include the values (case insensitive):

- `NULL`
- `\N`
- `NA`
- `N/A`
- [space]

If you are appending data to an existing schema or table, columns in the CSV file must be in the same order as defined in the target table.

If you are loading a fact table that joins to dimension tables, you must load the fact table first, and then the dimension tables. The joining key must be a single column of unique values in the dimension table. `NULL` values in the fact table cannot be joined.

Create a CSV file

The first step in loading data is to obtain or create one or more CSV files that contain the data to be loaded into ThoughtSpot. CSV is a common format for transferring data between databases. ThoughtSpot requires this format.

Most applications such as Microsoft Excel or Google Sheets can output CSV formatted files. If your source is an Excel spreadsheet or Google Sheet:

1. Save, export, or download the file in CSV format. The exact procedure you use will depend on the source application.
2. Review the file's format before uploading it to ThoughtSpot.

   Your source data may be in another database. If this is the case, your company's ETL (extract, transform, load) process will typically generate CSV files. If your source is another database:

3. Connect to the source database.
4. Extract each table you wish to import into ThoughtSpot as a CSV file.

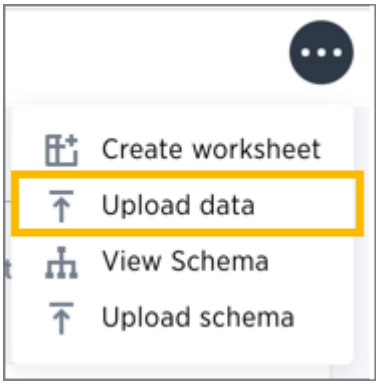
   The column delimiter should be a `,` (comma), `|` (pipe), or tab.

For general information about CSV files and the rules for creating them, see the Comma-separated_values on Wikipedia.

Load the CSV File

Any user who belongs to a group that has the privilege **Has administration privileges** or **Can upload user data** can upload their own data from the browser. To load the CSV or Excel file into ThoughtSpot:

1. Log into ThoughtSpot from a browser.
2. Click **Data**, on the top navigation bar.
3. Click the the ellipses icon (3 dots) ••• , in the upper right corner, and select **Upload Data**.



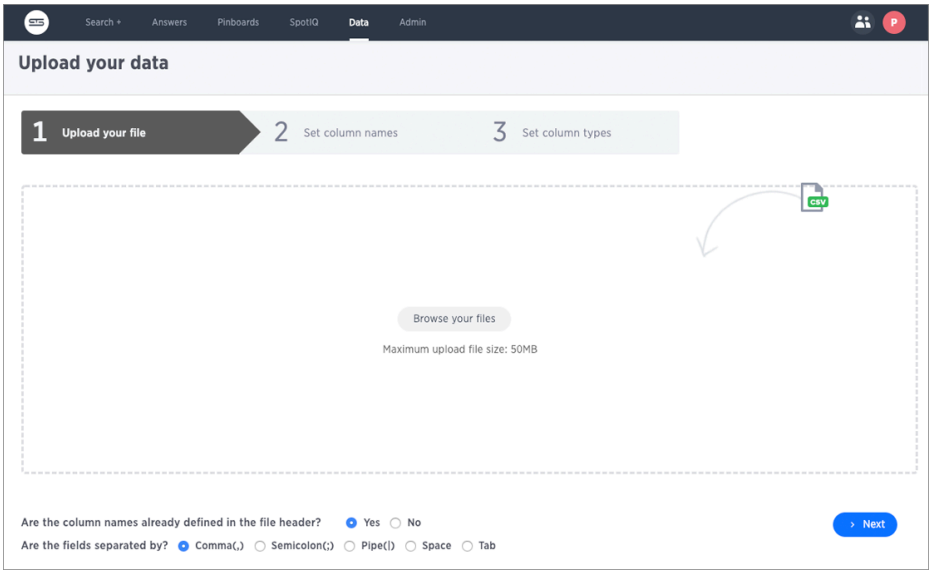
4. Upload the CSV or Excel file by doing one of these options:
   a. Click on **Browse your files** and select the file.

b. Drag and drop the file into the drop area.

5. Answer the question **Are the column names already defined in the file header?**
6. Answer the question **Are the fields separated by?** Click **Next**.
7. Click on the column header names to change them to more useful names, if you'd like. Click **Next**.
8. Review the automatically generated data types for each column, and make any changes you want.

   There are four data types: Text, Integer, Decimal, and Date.
9. Click **Import**.

   When an upload is complete, the system reports the results and offers you some further actions.



- Click **Link to Existing Data** if you want to link the data you uploaded to the data in another table or worksheet.
- Click **Search** if you want to begin a new search.
- Click **Auto analyze** if you want to use the SpotIQ feature to find insights in your new data.

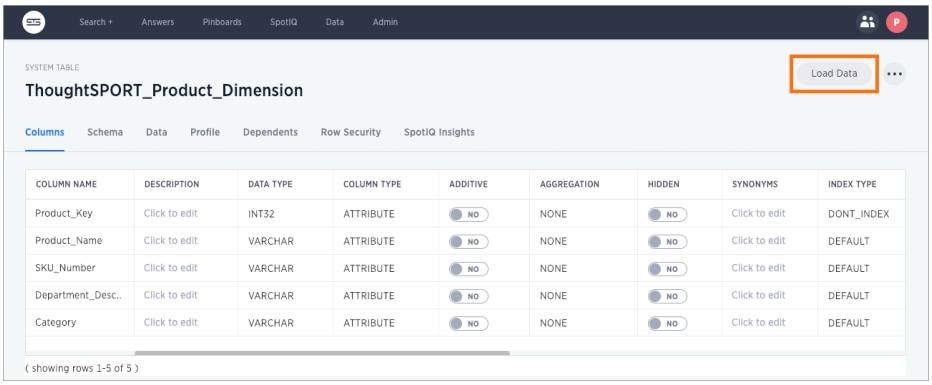
Append to an existing table

You can append data to your existing system tables through the ThoughtSpot application, even if the tables were initially loaded using Data Connect or `tsload`. The CSV file must have the same structure as the table it is being loaded into, including number and type of columns, in the same order as the target table.

To append data into ThoughtSpot:

1. Log in to ThoughtSpot from a browser.
2. Click **Data** on the top navigation bar.

3. Click the name of the table you would like to append data to.
4. Click the **Load Data** button.



5. Upload the CSV or Excel file by doing one of these options:
    - Click **Browse your files** and select the file.
    - Drag and drop the file into the drop area.

6. Answer the question **Are the column names already defined in the file header?**.
7. For the question **Do you want to append to the existing data or overwrite it?**, select **Append**.
8. Answer the question **Are the fields separated by?**, and click **Next**.
9. Click **Upload**.
10. Click **Link to existing data** if you want to link the data you uploaded to the data in another table or worksheet. Or click **Ask a question** if you want to begin a new search.

How to view a data schema

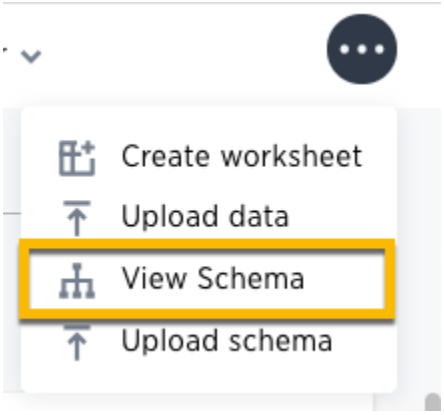
Summary: Use the schema viewer to see tables and worksheets and their relationships.

ThoughtSpot offers a schema viewer that lets you see your database schema in the web browser. The Schema Viewer is interactive, so you can configure it to show just what you want to see.

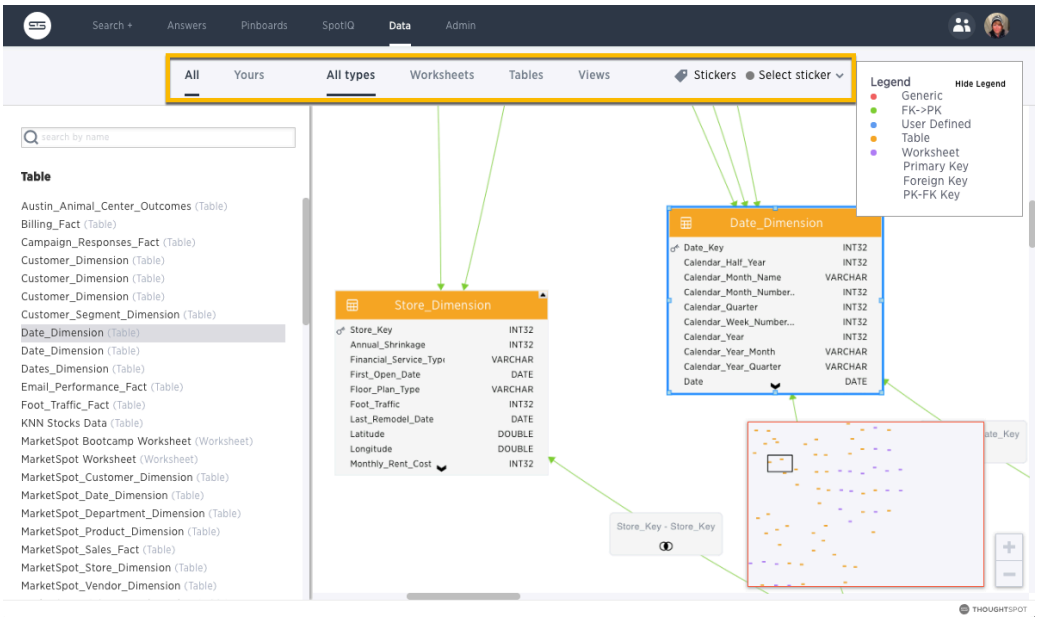
You need **Admin** privileges to use the **Schema Viewer**.

Bringing up the Schema Viewer

You can bring up the Schema Viewer from the **Data** screen. Click the ellipses icon (3 dots) , and select **View Schema**.



When viewing the schema, you can filter the tables shown similarly to how you filter data sources. The list of tables, worksheets, and imported data on the left includes only those objects you want to see. Clicking on one of the objects brings it to the middle of the viewer and highlights it. You can drag the objects around in the viewer.



Why to use the Schema Viewer

You can use the Schema Viewer to find out information like:

- What is the relationship between two tables? What tables make up this
- worksheet, and how are they joined?

The schema viewer shows joins between tables, join directionality, and join type (whether they are Foreign Key to Primary Key, relationship joins, or joins defined by users through the web interface). Use the **Table** list to find a specific table or worksheet.

How the Schema Viewer shows joins

You can use the Schema Viewer to review your schema and ensure that it was modeled using best practices. For example, joins are shown in separate colors that indicate their type:

- Red is used for generic relationships
- Green is used for primary key/foreign key joins

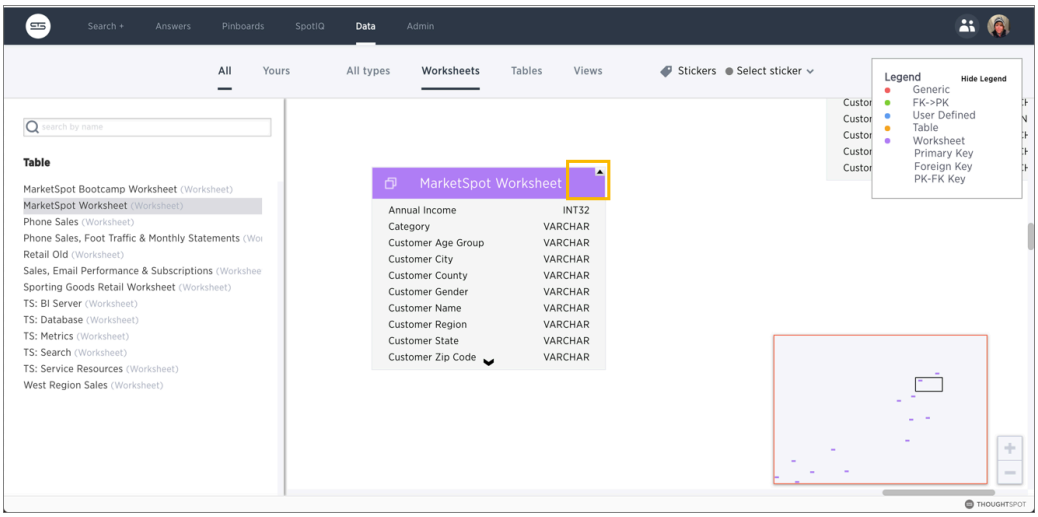
When viewing a worksheet, you'll also be able to see whether an inner, left outer, right outer, or full outer join was used between each of the joined tables.

A good rule to follow is "Keep it Green". This means that you'll get better results from PK/FK joins rather than from using generic relationships. You should only use generic relationships when the tables being joined have a many-to-many rather than a PK/FK structure. If you find tables that have been joined using a generic relationship, but could have used a PK/FK join, you should drop the relationship and create a PK/FK join instead. To do this, you need to use the ALTER TABLE…DROP RELATIONSHIP statement in TQL. Then use ALTER TABLE…ADD FOREIGN KEY to create the PK/FK join.

Worksheet view

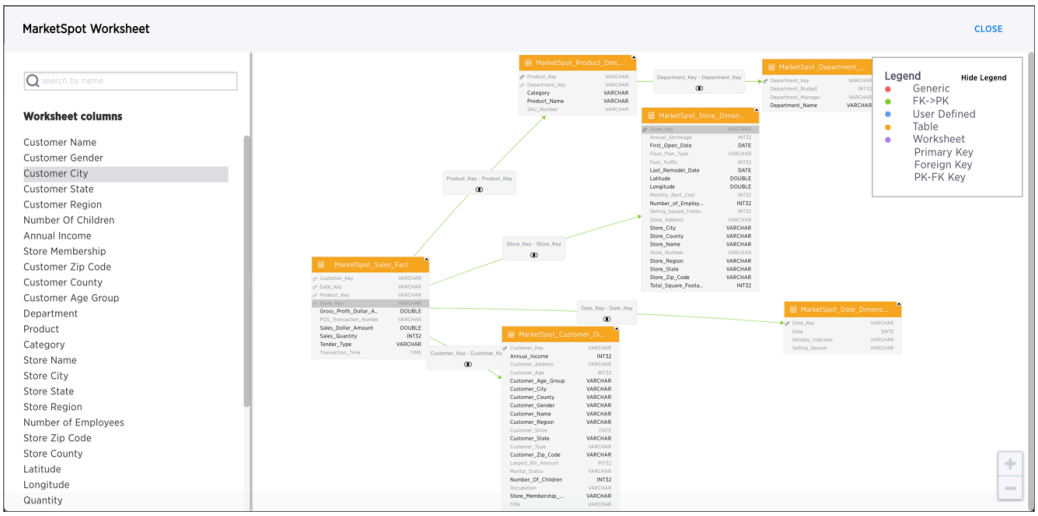
Worksheets are often based on more than one table. The worksheet schema will show schemas for the tables behind the worksheet, as well as the joins between tables *that were created as a part of the worksheet*.

Click on a worksheet, to see it in the Schema Viewer. If the schema view is not showing the schema behind the worksheet, double click the tab on the top right of the worksheet object.



The worksheet view shows the following information:

- All tables in the worksheet, and the relationships between these tables.
- Source columns for all columns of a worksheet.
- Keys and definitions for each relationship, as well as join paths and types.
- Columns that are derived from formulas.
- Correct join paths for newly created chasm trap worksheets. Existing chasm
- trap worksheets created prior to ThoughtSpot version 4.4 will not show the correct join paths.



Related Information

- Worksheet joins
- Change the schema using TQL
- Constraints

Schema planning concepts

Summary: Considerations in creating a schema for the ThoughtSpot Loader.

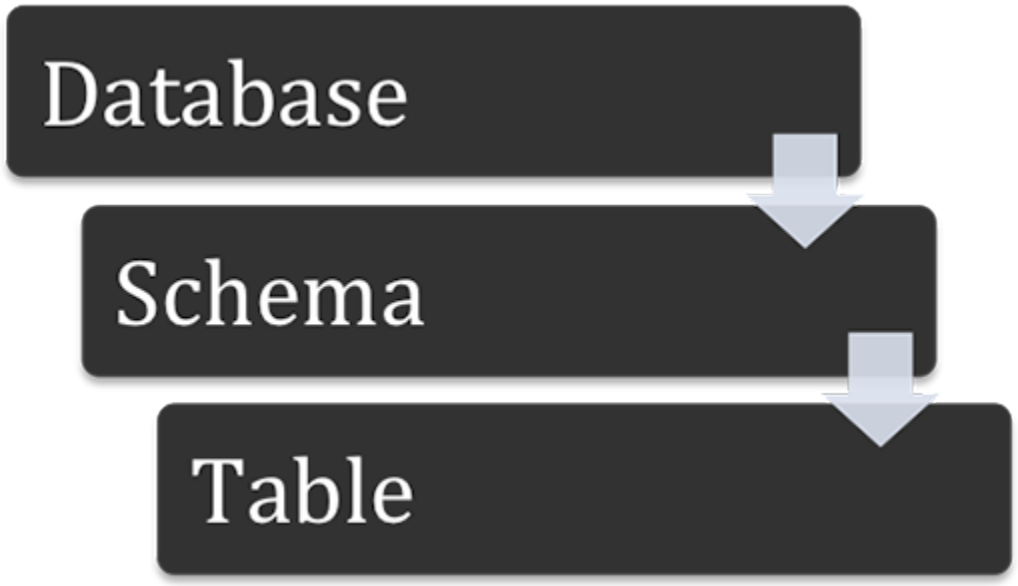
Before you can load data with ThoughtSpot Loader, you must create a schema to receive it, using the SQL command line interface (TQL).

The TQL syntax is similar to the SQL used in other relational databases, but with some important differences. You'll use DDL (data definition language) to create the schema into which you'll load the data. You'll probably want to put all your DDL statements into a text file, which you'll use as a script for creating the schema.

Before writing your TQL script, you need to understand some basic ThoughtSpot concepts.

About databases and schemas

ThoughtSpot organizes objects in a hierarchical namespace. Databases contain schemas, which contain tables.



ThoughtSpot can contain one or more databases, and each database can have multiple schemas. If you do not specify a schema, the default schema (`falcon_default_schema`) is used automatically. This makes it easier to add tables to the database without the need to explicitly create a schema.

If you do create an additional schema, you must refer to its objects using the syntax `<schemaname>.<objectname>` . If you do not qualify the schema when referencing its objects, the default schema (`falcon_default_schema`) will always be assumed.

By default, ThoughtSpot creates an internal database to host tables corresponding to data that is imported by users from a Web browser.

Review the structure of your data

The schema you create to hold the data needs to be a good fit for your data. First, familiarize yourself with the tables you want to load, and understand their structure. Make note of this information for each table:

- The column names and data types
- Type of table (fact or dimension)
- Primary key column(s)
- The size of the table on disk
- Any other tables it can be joined with (foreign keys)

Here's what you'll need to take into account in your TQL for creating each table, based on these properties:

Table type	Table size	To be joined with	Schema recommendations
Fact	Any	Small dimension table(s)	Sharded. Foreign key references the primary key in the dimension table.
Fact	Any	Large dimension table(s)	Sharded on the same distribution key as the dimension table it will be joined with. Foreign key references the primary key in the dimension table.
Fact	Any	Another fact table	Sharded on the same distribution key as the fact table it will join with. Many-to-many relationship defines how the tables will be joined.
Dimension	under 50MB	Fact table(s)	Replicated (not sharded). Has a primary key.
Dimension	over 50MB	Fact table(s)	Distributed dimension table, sharded on the same distribution key as the fact table it will be joined with. Primary key must be the same as the distribution key.

Where to go next

- Data types
  ThoughtSpot supports the common data types. Compare these with the data types you want to load, and do any necessary conversion ahead of loading the data.
- Constraints
  Constraints include primary keys, foreign keys, and relationships. Relationships allow you to create a generic relationship for use when you want to join tables that don't have a primary key/foreign key relationship.
- Sharding
  For the best performance, you should split (or shard) very large tables across nodes. If you have a large dimension table, you might choose to co-shard it with the fact table it will be

joined with.

- Chasm traps
  In a complex schema, you may have a fact table with no relationship to another fact table, except that each contains a foreign key to a shared dimension table. This is known as a chasm trap, and ThoughtSpot can handle it!

Data types

Summary: ThoughtSpot supports the common data types.

Before you import data, compare the data types you want to load with these supported data types. Then, convert your data before loading it. Typically, you would export the data, transform it to meet these type rules, and then load the data. This is known as an extract-transform-load process.

Supported data types

The tables you create to receive the data must have the same number of columns and data types as the data you will be loading. Choose a data type for each column from the list of supported data types:

Data	Supported data types	Details
Character	`VARCHAR(*n*)`	Specify the maximum number of characters, as in `VARCHAR(255)`. The size limit is 1GB for `VARCHAR` values.
Floating point	`DOUBLE` or `FLOAT`	`DOUBLE` is recommended. `DOUBLE` has a range of `1.7E +/- 308` (15 digits).
Boolean	`BOOL`	Can be `true` or `false`.
Integer	`INT` or `BIGINT`	`INT` holds 32 bits. `BIGINT` holds 64 bits. `INT` has a range of `-2,147,483,648` to `2,147,483,647`. `BIGINT` range is `-9,223,372,036,854,775,808` to `9,223,372,036,854,775,807`.
Date or time	`DATE`, `DATETIME`, `TIMESTAMP`, `TIME`	`DATETIME`, `TIMESTAMP`, and `TIME` are stored at the granularity of seconds. `TIMESTAMP` is identical to `DATETIME`, but is included for syntax compatibility.

Warning: There is a 1GB limitation on the number of characters for `VARCHAR`. If you have any `VARCHAR` data that exceeds this limit, the entire load will fail.

Geographical data types

For geographical data types, use VARCHAR. For latitude and longitude, you can use either `VARCHAR` or `DOUBLE`. After loading the data, designate it as a geographical data type when you Edit the system-wide data model. Wherever abbreviations or codes are used, they are the same as what the USPS (United States Postal Service) recognizes.

These data types can be designated as geographical data, which enables them to be visualized using the Geo chart types:

- Countries, for example:
    - United States
        - `long name` : United States
        - `name_sort` : United States of America
        - `abbreviation` : U.S.A.
        - `adm0_a3` : USA
        - `adm0_a3_is` : USA
        - `adm0_a3_us` : USA
        - `admin` : United States of America
        - `brk_a3` : USA
        - `brk_name` : United States
        - `formal_en` : United States of America
        - `iso_a2` : US
        - `iso_a3` : USA
        - `iso_n3` : 840

- `COUNTY` for counties in the United States, for example:
    - santa clara county
    - pike county, ohio
    - pike county, OH

- `STATE_PROVINCE` for states in the United States, for example:
    - `name` : California
    - `US Postal Service abbreviation` : CA

- `LATITUDE` which must be used with `LONGITUDE`, for example:
    - 37.421023
    - 1.282911

- `LONGITUDE` which must be used with `LATITUDE`
    - 122.142103
    - 103.848865

- `ZIP_CODE` for zip codes and zip codes +4 in the United States
    - `po_name` : MT MEADOWS AREA
    - `ZIP` : “00012”
    - `zip2` : 12

- Other Sub-nation Regions which are administrative regions found in countries other than the United States, for example:
    - bremen
    - normandy
    - west midlands

> ⚠ **Important:** You cannot upload your own custom boundaries.

Constraints

Summary: Constraints allow you to build relationships and join tables.

Constraints include primary keys, foreign keys, and relationships. Relationships allow you to create a generic relationship for use when you want to join tables that don't have a primary key/foreign key relationship.

Primary keys

When a primary key is selected for a table, it impacts data loading behavior. When a new row is added:

- If another row already exists with same primary key, it is updated with the values in the new row.
- If a row with the same primary key does not exist already, the new row is inserted into the table.

This behavior is referred to as "upsert" because it does an `INSERT` or an `UPDATE`, depending on whether a row with the same primary key already exists.

Note that ThoughtSpot does not check for primary key violations across different shards of the table. Therefore, you need to shard the table on the primary key columns if you require this "upsert" behavior.

Foreign key relationships

Foreign key relationships tell ThoughtSpot how two tables can be joined. These relationships are only used for joining the tables, and not for referential integrity constraint checking.

The directionality of primary key - foreign key relationships is important. The foreign key relationship is defined on the fact table and references the primary key(s) in the dimension table. So you can think of the fact table as the source and the dimension table as the target. In the schema viewer, you'll notice that the arrow that represents a PK/FK join points to the dimension table.

If you use primary and foreign keys, when users search the data from the search bar, tables are automatically joined. For example, assume there are two tables:

- revenue, which is a fact table
- region, which is a dimension table

There is a foreign key on the fact table on `regionid` which points to the id in the region dimension table. When a user types in "revenue by region", the two tables will be joined automatically.

Foreign keys have to match the primary key of the target table they refer to. So if there are multiple columns that make up the primary key in the target table, the foreign key must include all of them, and in the same order.

Generic relationships (many-to-many)

You may have a schema where there is a fact table that you want to join with another fact table. If there isn't a primary key/foreign key relationship between the tables, you can use many-to-many to enable this. You can do this by using the RELATIONSHIP syntax to add a link between them, that works similarly to the WHERE clause in a SQL join clause.

Note: Using generic relationships is not a best practice. In cases where you have two fact tables you want to join, it is better to find a way to create a bridge table between them, so you have a chasm trap. Look at your two fact tables to see if they share some common data that you could use to create a dimension table between them. For example a date or product dimension could be use to join an inventory fact table with a sales fact table. This is best done in your ETL process, before bringing the data into ThoughtSpot.

Note: A many-to-many implementation does not protect from over counting in some searches. If you plan to use it, make sure your searches don't include aggregation or count searches that will count one value multiple times, because it satisfies the join condition for multiple rows.

This is a special kind of relationship, that applies to specific data models and use cases. For example, suppose you have a table that shows wholesale purchases of fruits, and another table that shows retail fruit sales made, but no inventory information. In this case, it would be of some use to see the wholesale purchases that led to sales, but you don't have the data to track a single apple from wholesale purchase through to sale to a customer.

In a many-to-many relationship, the value(s) in a table can be used to join to a second table, using an equality condition (required) and one or more range conditions (optional). These conditions act like the WHERE clause in a SQL JOIN clause. They are applied using AND logic, such that all conditions must be met for a row to be included.

To use a many-to-many relationship, you need to follow a few rules:

- There must be one equality condition defined between the two tables.
- Each table must be sharded on the same key that will be used for the equality condition.
- There can optionally be one or more range conditions defined.

This example shows the TQL statements that create the two fact tables and the relationship between them.

```
TQL> CREATE TABLE "wholesale_buys" (

  "order_number" VARCHAR(255),

  "date_ordered" DATE,

  "expiration_date" DATE,

  "supplier" VARCHAR(255),

  "fruit" VARCHAR(255),

  "quantity" VARCHAR(255),

  "unit_price" DOUBLE

  )  PARTITION BY HASH (96) KEY ("fruit");


TQL> CREATE TABLE "retail_sales" (

  "date_sold" DATE,

  "location" VARCHAR(255),

  "vendor" VARCHAR(255),

  "fruit" VARCHAR(255),

  "quantity" VARCHAR(255),

  "sell_price" DOUBLE

  )  PARTITION BY HASH (96) KEY ("fruit");


TQL> ALTER TABLE "wholesale_buys" ADD RELATIONSHIP WITH "retai
l_sales" AS "wholesale_buys"."fruit" = "retail_sales"."fruit" a
nd ("wholesale_buys"."date_ordered" < "retail_sales"."date_sol
d" and "retail_sales"."date_sold" < "wholesale_buys"."expiratio
n_date");
```

Sharding

Summary: Sharding partitions very large tables into smaller, faster, more easily managed parts called data shards.

ThoughtSpot tables can be replicated or sharded. Replicated tables exist in their entirety, the complete data set, on each node. Sharded tables consist of a single data set divided into multiple tables or shards. The shards have identical schemas but different sets of data.

When to use sharding

By default, ThoughtSpot tables are replicated, you must explicitly shard tables. Sharding your tables impacts the total amount of memory used by the table as well as its performance.

For example, you might shard a large table of sales data. So, you could divide a single sales table into shards each of which contains only the data falling within a single year. These shards are then distributed across several nodes. Requests for sales data are dispersed both by the year and the location of the shard in the node cluster. No single table or node is overloaded, and so the performance of a query and the system load are both improved.

To optimize ThoughtSpot performance, you should *shard* very large fact tables whenever possible. If you have a large dimension table, you might choose to shard it along with the fact table it is joined with. Sharding both the fact and dimension table is known as *co-sharding*.

Table sizes and sharding recommendations

Number of rows per shard	5-10 million
Maximum	10 million rows per shard
Maximum number of shards	~ 80% of CPU cores

Example

Number of rows in table	1.1 billion
CPUS in cluster	256
HASH (128)	~50% of total CPUs
	8.6 million rows per shard

How to shard

Sharding is a type partitioning and is sometimes called *Horizontal partitioning*. The term sharding is particular to situations where data is distributed not only among tables but across nodes in a system. To create a sharded table add the add `PARTITION BY HASH ( )` clause to your `CREATE TABLE` statement.

```
TQL> CREATE TABLE ...
...PARTITION BY HASH (96) KEY ("customer_id");
```

The `HASH` parameter determines the number of shards and the `KEY` parameter the sharding key. The recommended number of shards depends upon the number of nodes in your cluster:

Number of Nodes	Number of Shards
1	32
2	64
3	96
4-12	128
13-24	256
25-36	384
37-48	512
49-60	640
61-72	768

If you omit the `PARTION BY HASH` statement or if the `HASH` parameter is 1 (one), the table is unsharded. This also means the table physically exists in its entirety on each node.

If you want to use the primary key for sharding, specify that the table is to be partitioned by `HASH` on the primary key, as in this example:

```
TQL> CREATE TABLE "supplier" (
  "s_suppkey" BIGINT,
  "s_name" VARCHAR(255),
  "s_address" VARCHAR(255),
  "s_city" VARCHAR(255),
  "s_phone" VARCHAR(255),
  CONSTRAINT PRIMARY KEY ("s_suppkey")
  )  PARTITION BY HASH (96) KEY ("s_suppkey");
```

The system does not use primary keys as sharding keys by default. If you specify the `PARTION BY HASH` statement with a `HASH` greater than 1 (one) *but omit the* `KEY` *parameter* ThoughtSpot shards the table randomly. This is not recommended; avoid this by always ensuring you specify the `KEY` parameter with a HASH greater than 1 (one).

How to choose a shard key

When you shard a large table, you select a *shard key* from the table. This key exists in every shard. You can use any data type that is valid for use as the primary key as the shard key. Choosing a shard key plays an important role in the number of shards and the size of any single shard.

A shard key should contain a value that has a good distribution (roughly the number of rows with each value in that column). This value is typically part of the primary key, but it can include other columns. For example:

```
CREATE TABLE "sales_fact"
  ("saleid" int,
  "locationid" int,
  "vendorid" int,
  "quantity" int,
  "sale_amount" double,
  "fruitid" int,
  CONSTRAINT
PRIMARY KEY("saleid,vendorid"))
PARTITION BY HASH(96)
KEY ("saleid");
```

Notice the shard key contains the `saleid` value that is also part of the primary key. When creating a shard key use these guidelines.

- Include one or more values of the table's primary key in the shard key.

  This prevents scenarios where the data with the same primary key ends up in different shards and nodes because the shard key changed.

- If you expect to join two tables that are both sharded, make sure both tables use the same shard key.

This guideline ensures better join performance. So, for example, if you have two tables and the primary keys are:

`PRIMARY KEY("saleid,vendorid")` on A `PRIMARY KEY("saleid,custerid")` on B

You should use `saleid` for your shared key when you shard both table A and B.

- Choose a shard key so that the data is distributed well across the keys.

  For example, suppose the table you want to shard has a primary key made up of `saleid`, `custid` ,and `locationid`. If you have 10K sales but 400 locations, and 2000 customers, you would not want to use the `locationid` in your shard key if 5k sales were concentrated in just 2 locations. The result would be data in fewer shards and degrade your performance. Instead, your shard key may be `custid` ,and `locationid`.

- Choose a shard key that results in a wide variety of keys.

  For example, suppose the table you want to shard has a primary key made up of `saleid`, `productid` ,and `locationid`. Suppose the table has 10K sales, 40 locations, and 200 products. If the sales are evenly distributed across locations you would not want to use the `locationid` in your shard key. Instead, `saleid` and `productid` would be the better choice as it results in a wider variety of keys.

As mentioned in the previous section, it is possible to simply use the primary key as a shard key. It isn't a good idea to use shard keys outside of the primary key. The reason is that it, with a non-primary shard key, it is possible to get two versions of a record if the shard key for a record changes, but the primary key doesn't. A second version reults because, in the absence of a unique shard key, the system create a secondary record rather than doing a SQL MERGE (`upsert`).

Sharded dimension tables

In a typical schema, you'd have a sharded fact table with foreign keys to small dimension tables. These small dimension tables are replicated in their entirety and distributed on every node. This works best where dimension tables under 50MB in size.

If you have a large dimension table, replicating it and distributing it can impact the performance of your ThoughtSpot System. In this case, you want to shard the dimension tables and distribute it the same way as the fact table it joins to.

When sharding both a fact and its dimension table (known as co-sharding) keep in mind the guidance for creating a shard key. Only shard dimension tables if the dimension table is large (over 50MB) and the join between the fact and dimension tables use the same columns. Specifically, the tables must:

- be related by a primary key and foreign key
- be sharded on the same primary key/foreign key
- have the same number of regions (or shards)

If these requirements are met, ThoughtSpot automatically co-shards the tables for you. Co-sharded tables are always joined on the sharding key. Data skew can develop if a very large proportion of the rows have the same sharding key.

This example shows the `CREATE TABLE` statements that meet the criteria for sharding both a fact table and its dimension table:

```
TQL> CREATE TABLE products_dim (
  "id" int,
  "prod_name" varchar(30),
  "prod_desc" varchar(100),
  PRIMARY KEY ("id")
)
PARTITION BY HASH (96) KEY ("id")
;

TQL> CREATE TABLE retail_fact (
  "trans_id" int,
  "product_id" int,
  "amount" double,
  FOREIGN KEY ("product_id") REFERENCES products_dim ("id")
)
PARTITION BY HASH (96) KEY ("product_id")
;
```

If a dimension table is joined to multiple fact tables, all of the fact tables must be sharded in the same way as the dimension table. Self-joins are not supported.

Joining two sharded fact tables

You can also join two sharded fact tables with different shard keys, this is known as *non co-sharded* tables. It may take a while to join two tables sharded on different keys since a lot of data redistribution is required. Therefore, ThoughtSpot recommends that you use a common shard key for two fact tables.

You are not limited by the column connection or relationship type.

Chasm traps

> **Summary:** A chasm trap occurs when two many-to-one joins converge on a single table.

In a complex schema, you may have a fact table with no relationship to another fact table, except that each contains a foreign key to a shared dimension table. This is known as a chasm trap, and ThoughtSpot can handle it!

Understand how chasm traps occur

A fact table, just as it sounds, stores facts about your business. If you are selling apples, the sales fact table has facts about these apples.

SaleID	AppleTypeID	StoreID	Units Sold
4	55	2	12
8	34	33	3
10	09	09	1

Dimension tables describe the attributes that are interesting to analyze. For example, the apple table might look like this.

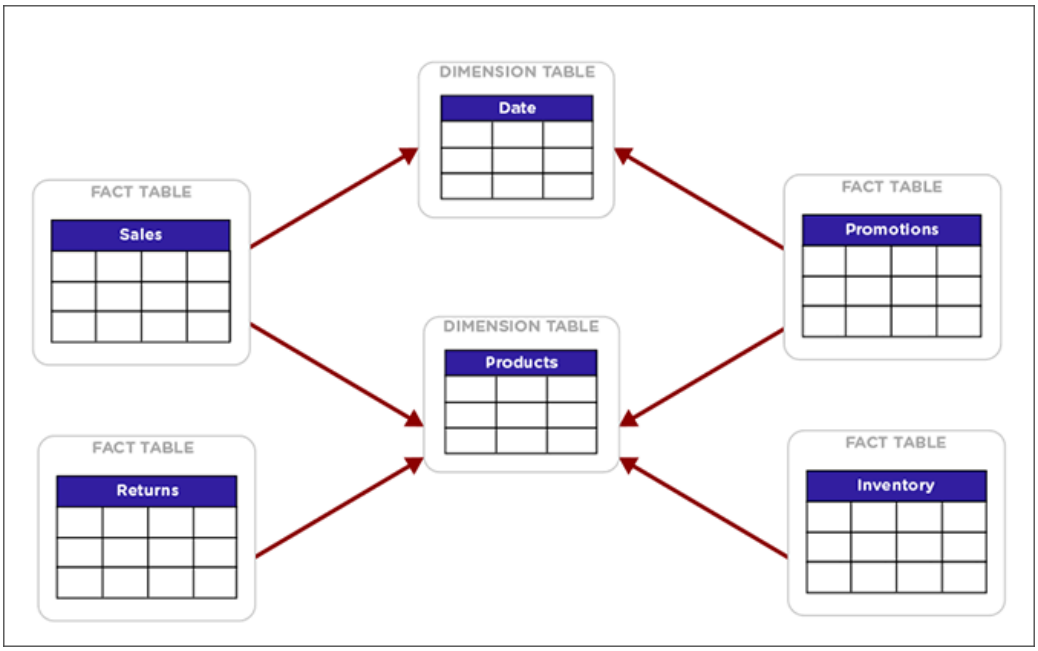
AppleTypeID	Color	Name	Use
55	Red	Red Delicious	Snack
34	Green	Granny Smith	Cooking
09	Yellow	Golden	Snack

As you can imagine, in a business you might have several fact tables that access dimension tables. So, an apple business may record waste as well as sales.

TimeID	AppleTypeID	StoreID	Units Wasted
4	55	2	2
8	34	33	43
10	09	09	11

Both the sales and waste tables are facts that reference the apple dimension table.

A chasm trap in a data schema can introduce problems of over counting if you join the two fact tables through their shared dimension table. This diagram shows a typical complex schema with several tables that are related over a chasm trap:



Examples of use cases where a chasm trap could occur when attribution analysis compare campaign data with purchase data, where all they have in common is that both contain a customer identifier that is a foreign key to a customer dimension table. Chasm traps also occur, for example, in cost of sales analysis when wholesale orders data is only related to the retail sales data through a shared products dimension table.

In many databases, joining tables across a chasm trap creates a *Cartesian product* or *cross join*. That is each row from the first fact table is joined to each row from the second table. A Cartesian product causes over counting when computing counts and aggregates. ThoughtSpot protects you from this kind of over counting.

There are still just a few things to look out for when using a schema that contains chasm traps:

- The tables should be joined to the dimension table via an equi-join (i.e. a primary key/foreign key relationship). They cannot be joined using a range of values.
- Review the column setting called Attribution Dimension. You may need to change this setting if some of the columns in the shared dimension table should not be used for attribution when combining fact tables.
- Tables that will be joined across a chasm trap do not need to be co-sharded. They will be joined appropriately automatically in the most efficient way.

Chasm trap limitations

Join information in **What am I Looking At?** does not appear for searches on a worksheet containing a chasm trap or on base tables that are related over a chasm trap.

Overview of schema building

Summary: Before you can load data into ThoughtSpot, you must build a database schema to receive it.

You can build a schema by writing a SQL script that creates the objects in your schema. ThoughtSpot provides the ThoughtSpot SQL Command Line (TQL) for creating, viewing, and managing a schema using SQL. Your script can use any statements that are supported in ThoughtSpot SQL Command Line (TQL). The TQL syntax is similar to the SQL used in other relational databases, but with some important differences.

You'll use DDL (data definition language) to create the schema into which you'll load the data. You'll probably want to put all your DDL statements into a text file, which you'll use as a script for creating the schema.

Upload the script through the browser

You can upload an your SQL script directly through the browser in the ThoughtSpot application. You can edit the script or add to it right within the browser, too. The steps to build a schema through the browser are:

1. Write a SQL script to create the schema
2. Import a schema (use the SQL editor)

Use TQL on the SQL command line

You can choose to run your SQL script within the Linux shell instead. You can run TQL in interactive command line mode, or you can write a script and use TQL to run it. The SQL syntax in ThoughtSpot is called TQL for ThoughtSpot SQL. The ThoughtSpot SQL Command Line (TQL) runs in an interactive mode. To invoke TQL Log in to the Linux shell using SSH and type `tql`. At the prompt, type `h` or `help` to see a list of supported commands.

The steps to build a schema using TQL include:

1. Connect to the database with the ThoughtSpot SQL Command Line (TQL).
2. Write a SQL script to create the schema.
3. Type your SQL commands on the command line, terminating each command with a semicolon (;).

   Commands can span multiple lines. ThoughtSpot supports a limited number of SQL commands, plus some custom SQL extensions. For example, you can specify the number of shards and the distribution key as part of the CREATE TABLE syntax. A full list of supported SQL in TQL is available in the TQL reference.

Where to go next

- **Connect with TCL and create a schema**
  Having examined the structure of the data to be loaded and become familiar with the ThoughtSpot SQL Command Line (TQL), you are now ready to create the schema.
- **Write a SQL script to create the schema**

Using a SQL script to create your schema is a recommended best practice. This makes it easier to adjust the schema definitions and recreate the schema quickly, if needed.

- Schema creation examples
  These examples demonstrate the steps involved in creating a schema using the ThoughtSpot SQL Command Line (TQL). After the schema is created, you can load data into it with ThoughtSpot Loader.
- Upload and run a SQL script)
  You can run a SQL script to create your database schema through the browser, without having to log in to the shell on the ThoughtSpot instance. You can edit the script and run it directly in the browser to create the schema.

Connect with TCL and create a schema

To perform administrative tasks directly in the database, you will use the ThoughtSpot SQL Command Line (TQL). TQL supports many, but not all, common SQL commands.

Connect with TQL

Before connecting with TQL, you will need:

- Access to your ThoughtSpot instance Linux shell from a client machine.
- The administrator OS login.

To connect to TQL:

1. Log in to the Linux shell using SSH.
2. Invoke TQL:

```
$ tql

TQL>
```

3. Enter your SQL command, followed by a semicolon (`;`).

Enter a SQL script

Having examined the structure of the data to be loaded and become familiar with the ThoughtSpot SQL Command Line (TQL), you are now ready to create the schema.

This method is a good way to get familiar with TQL and how to create database objects, but when creating a schema in a production system, you will most likely Write a SQL script to create the schema.

To create the schema directly in TQL:

1. Connect to the database with the ThoughtSpot SQL Command Line (TQL).
2. If the database you will be using does not exist, create it now:

```
TQL> CREATE DATABASE my_database;
```

3. Connect to the database:

```
TQL> USE my_database;
```

4. If you wish to use a schema other than the default one, create it now:

```
TQL> CREATE SCHEMA my_schema;
```

5. Issue a `CREATE TABLE` command for each table you will create, using the information in Plan the schema.

> ☑ **Tip:** Foreign key declaration within a `CREATE TABLE` will show the table created even if there are problems with the foreign key. Therefore, it is good practice to also issue a separate `ALTER TABLE ADD CONSTRAINT FOREIGN KEY` command.

How to write a SQL script

Summary: Using a SQL script to create your schema is a recommended best practice. This makes it easier to adjust the schema definitions and recreate the schema quickly, if needed.

The schema creation script is a text file that contains all the SQL commands to create your schema. Comments should be enclosed in the comment tags /* and */.

Enclose all object names (schema, table, and column) in double quotes and any column values in single quotes in your scripts. Object names that are also reserved words in SQL, or that contain special characters (any character other than alphanumeric or _), must be surrounded by double quotes. If you see the error message “Error parsing SQL. Check SQL input.”, you should check for object names without double quotes in your script.

If you are working in a schema other than the default schema, object names must be fully qualified, as in `"<schema_name>"."<object_name>"`.

If your schema includes constraints to define relationships between tables (foreign key, or the `RELATIONSHIP` syntax), it is recommended that your script first creates all the tables, and then at the end, creates the relationships between them using the `ADD CONSTRAINT` syntax. This makes it easier to troubleshoot the script and make changes.

If TQL is run using the flag `--allow_unsafe`, your statements will always execute without this warning. Note that when running TQL from a script, you will need to decide what behavior you want if the script contains changes that affect dependent objects. If you want the script to run even if objects with dependencies are affected, run it using this flag, for example:

```
cat safest_script_ever.sql | tql --allow_unsafe
```

1. Open a new file in a text editor.
2. Type in the command to create the database, if it does not already exist:

   ```
   CREATE database <db_name>;
   ```

3. Type in the command to specify the database to use:

   ```
   USE database <db_name>;
   ```

4. Type in the command to create the schema, if you don’t want to use the default schema:
5. Type in each of the CREATE TABLE statements, with its column definitions, primary key constraints, and sharding specification (if any).
6. At the end of your script, optionally type in the `ALTER TABLE` statements to add foreign keys to use in joining the tables.
7. Save the file.

8. Run the script using one of these methods:

    - Import a schema (use the SQL editor).
    - Log in to the shell, copy your script to your ThoughtSpot instance using scp, and pipe it to TQL:

      ```
      $ cat create-schema.sql | tql
      ```

Schema creation examples

Summary: Simple examples that illustrate how to use the TQL and the ThoughtSpot Loader.

These examples demonstrate the steps involved in creating a schema using the ThoughtSpot SQL Command Line (TQL). After the schema is created, you can load data into it with ThoughtSpot Loader.

Simple schema creation example

The example creates a database (`tpch`) with two tables (`customer` , `transaction`). The example does not create a schema explicitly. So it will use the default schema (falcon_default_schema).

In this example:

- The table `customer` has a primary key called customer_id. The table `customer_transactions` has a primary key called transaction_id.
- The `customer` table is unsharded.
- The `customer_transactions` table is sharded into 96 shards using the `transaction_id` column.
- Both tables have referential integrity on `customer_id` .

```
$tql

TQL> CREATE DATABASE tpch;

TQL> USE tpch;

TQL> CREATE TABLE customer (
      name  VARCHAR(100),
      address VARCHAR(255),
      zipcode INT,
      customer_id INT,
      CONSTRAINT PRIMARY KEY (customer_id)
      );

TQL> CREATE TABLE customer_transactions (
     transaction_id INT,
     customer_id INT,
     amount DOUBLE,
     transaction_date DATETIME,
     CONSTRAINT PRIMARY KEY (transaction_id),
     CONSTRAINT FOREIGN KEY (customer_id) REFERENCES
     customer(customer_id)
     )  PARTITION BY HASH (96) KEY (transaction_id);
```

More complex schema creation example

The example uses a custom schema called sample_schema to hold the tables. Because of this, every table reference has to be schema qualified.

```
$ tql


TQL> CREATE DATABASE "sample_db";

TQL> USE "sample_db";

TQL> CREATE SCHEMA "sample_schema";

TQL> CREATE TABLE "sample_schema"."customer" (

  "c_custkey" BIGINT,

  "c_name" VARCHAR(255),

  "c_address" VARCHAR(255),

  "c_city" VARCHAR(255),

  "c_nation" VARCHAR(255),

  "c_region" VARCHAR(255),

  "c_phone" VARCHAR(255),

  CONSTRAINT PRIMARY KEY ("c_custkey")

);

TQL> CREATE TABLE "sample_schema"."supplier" (

  "s_suppkey" BIGINT,

  "s_name" VARCHAR(255),

  "s_address" VARCHAR(255),

  "s_city" VARCHAR(255),

  "s_nation" VARCHAR(255),

  "s_region" VARCHAR(255),

  "s_phone" VARCHAR(255),
```

```
  CONSTRAINT PRIMARY KEY ("s_suppkey")

);

TQL> CREATE TABLE "sample_schema"."lineorder" (

  "lo_orderkey" BIGINT,

  "lo_linenumber" BIGINT,

  "lo_custkey" BIGINT,

  "lo_partkey" BIGINT,

  "lo_suppkey" BIGINT,

  "lo_orderdate" DATE,

  "lo_orderpriority" VARCHAR(255),

  "lo_shippriority" VARCHAR(255),

  "lo_quantify" BIGINT,

  "lo_extendprice" BIGINT,

  "lo_ordtotalprice" BIGINT,

  "lo_discount" BIGINT,

  "lo_commitdate" DATE,

  CONSTRAINT PRIMARY KEY ("lo_orderkey","lo_linenumber"),

  CONSTRAINT FOREIGN KEY ("lo_custkey") REFERENCES "sample_sche
ma"."customer" ("c_custkey"),

  CONSTRAINT FOREIGN KEY ("lo_suppkey") REFERENCES "sample_sche
ma"."supplier" ("s_suppkey")

  )  PARTITION BY HASH (96) KEY (lo_orderkey);
```

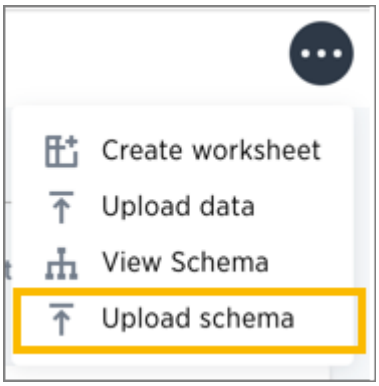
Upload and run a SQL script

Summary: Importing a schema through the Web browser makes it possible to run your SQL script without needing to have a Linux login.

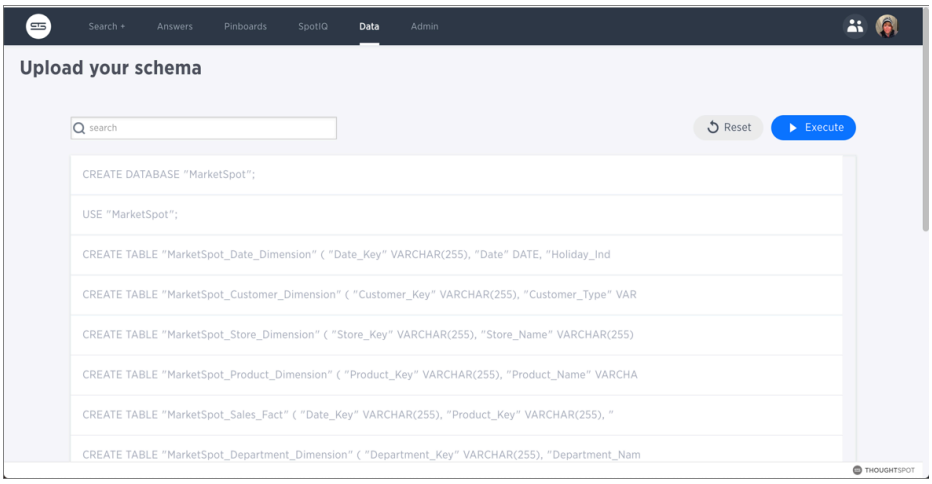
You can run a SQL script to create your database schema through the browser, without having to log in to the shell on the ThoughtSpot instance. You can edit the script and run it directly in the browser to create the schema. You can use this capability in any of these ways:

- Create the SQL script ahead of time, and use the browser to run it.
- Use the editor to type your SQL directly into the browser.
- Use the browser SQL interface as an interactive SQL editor, for example to test an existing script or make changes to an existing schema.

1. Log into ThoughtSpot from a browser.
2. Click **Data**, on the top navigation bar.
3. Click the ellipses icon (3 dots) , and select **Upload schema**.



4. Drag and drop your SQL file into the browser, or choose **Browse Your Files** to locate it.
5. You're now in the SQL editor. Use it to view your script and make any changes.



6. When ready, run your script by clicking the **Execute** button.
7. If there are any errors, correct them and run the script again.

How to change a schema

Summary: After you've created a schema and loaded data, you may find yourself wishing you'd set things up a little differently. You can make changes to the schema, such as changing the primary key, relationships to other tables, and sharding.

Making changes to a schema after data has been loaded and users have created worksheets or pinboards on the tables requires care, so that you don't lose the relationship between the objects created in ThoughtSpot and the underlying tables. If you follow the procedures here, your tables will retain their relationships to the objects created on top of them.

☑ **Tip:** Always take a snapshot of your database before making any schema changes. This snapshot allows you to revert back to the prior state if you make an error, or something doesn't work as you expected after the schema change.

Change the primary key for a table

Use this procedure to change the primary key for a table. But use it with caution, particularly if you are changing to a primary key for which values are not unique.

You can change the primary key of a table without having to `TRUNCATE` it first and reload the data. However, changing the primary key could result in data deletion. This is because of the upsert behavior which is applied when multiple rows have the same primary key. This is very important to understand ahead of time, if you are considering changing to a primary key for which values are not unique.

To change the primary key, first remove any existing primary key, and then define a new one (if any). You do not need to truncate the tables to do this operation beginning in version 3.2. Any dependent objects (pinboards or worksheets) will remain intact.

To change the primary key of a table:

1. Create a manual snapshot.
2. Connect to the database with the ThoughtSpot SQL Command Line (TQL).
3. Drop the existing primary key (if any), by issuing a command like this example:

```
TQL> ALTER TABLE "cart"
     DROP CONSTRAINT
     PRIMARY KEY;
```

   Dropping a primary key can impact existing worksheets, answers, and pinboards. The system warns you if dropping a primary key impacts other objects. To continue, use the `--allow_unsafe` flag.

4. Add a new primary key, if desired:

```
TQL> ALTER TABLE "cart"
     ADD CONSTRAINT
     PRIMARY KEY ("owner_id");
```

5. Test that any dependent objects (pinboards, worksheets, etc.) are still working correctly.
6. Delete the snapshot you created earlier using the command:

```
tscli snapshot delete <name>
```

Change a relationship between tables

Use this procedure to remove a relationship between tables or define a new one. This operation works for both kinds of relationships: foreign key or generic relationship.

To change a relationship between two tables, first remove any existing relationship, and then define the new relationship (if any). You do not need to truncate the tables to do this operation. Any dependent objects (pinboards or worksheets) will remain intact.

To change the relationship between tables:

1. Create a manual snapshot.
2. Connect to the database with the ThoughtSpot SQL Command Line (TQL).
3. Issue the command to drop the existing relationship

   Before dropping a relationship TQL checks for and then warns of any dependent objects. To continue with the drop any way, use the `--allow_unsafe` flag. The following examples illustrate several different types of drop operations.

   Drop a foreign key by name, if it was given a name when it was defined:

```
TQL> ALTER TABLE
     "sales_fact"
     DROP CONSTRAINT
     "FK_PO_number";
```

   Drop a relationship by name, if it was given a name when it was defined:

```
TQL> ALTER TABLE "fruit_dim"
     DROP RELATIONSHIP "REL_dates";
```

   Drop the foreign key relationship explicitly, if it doesn't have a name, by referencing the two tables that are joined. This drops all foreign keys between the two tables:

```
TQL> ALTER TABLE "shipments"
     DROP CONSTRAINT
     FOREIGN KEY "orders";
```

Drop all generic relationships between two tables:

```
TQL> ALTER TABLE "wholesale_buys"
     DROP RELATIONSHIP
     WITH "retail_sales";
```

4. Define a new relationship, if you want to, using `ALTER TABLE...ADD CONSTRAINT...`
5. Test that any dependent objects (pinboards, worksheets, etc.) are still working correctly.
6. Delete the snapshot you created earlier using the command:

```
tscli snapshot delete <name>
```

Change sharding on a table

You can change the sharding on a table or remove it altogether (creating a replicated table) using this procedure. This procedure preserves the data within the table.

This procedure reshards a table. This is also called redistributing or repartitioning. You can use this method to reshard a table without losing its data or metadata. This means that worksheets and pinboards built on top of the table will continue to work.

You can use these steps to do any of these operations:

- shard a table that was previously replicated.
- change a replicated table to a sharded table.
- change the number of shards to use for a sharded table.

To change the sharding on a table:

1. Create a manual snapshot.
2. Connect to the database with the ThoughtSpot SQL Command Line (TQL).
3. Issue the command to change the sharding using this syntax:

```
TQL> ALTER TABLE <table>
     [SET DIMENSION | SET FACT
     [PARTITION BY HASH
     [(<shards>)]
     [KEY(<column>)]]]
```

For example:

- To make a sharded table into a dimension table (replicated on every node), use:

```
ALTER TABLE "products"
   SET DIMENSION;
```

- To make a dimension table into a sharded (fact) table or change the number of shards, use:

```
ALTER TABLE "sales"
   SET FACT PARTITION BY HASH (96)
   KEY ("productID");
```

4. Test that any dependent objects (pinboards, worksheets, etc.) are still working correctly.
5. Delete the snapshot you created earlier using the command:

```
tscli snapshot delete <name>
```

Convert column data type

Summary: You can convert the data in a column from one data type to another by issuing a TQL command.

There are some details you should be aware of when doing a data type conversion.

Data type conversion behavior

When converting from one data type to another, any values that can not be converted will be set to NULL. If errors occur during data type conversion, the operation is aborted. However, you may choose to force the conversion despite the errors. You can start TQL in allow_unsafe mode to continue with the data conversion, at your own risk, of course! To start TQL in unsafe mode, issue this command:

```
tql ––allow_unsafe
```

Multiple columns of a single table can be converted using a single TQL command. The behavior is transactional. So for example, you would issue a command like this example:

```
ALTER TABLE products
   MODIFY COLUMN product_id int,
   MODIFY COLUMN supplier VARCHAR(4);
```

Also note that changing data type has implications on the primary key and sharding enforcement. For example, changing the data type of a column that is part of the sharding key would lead to a redistribution of data. Then imagine that the sharding key column contained the text values `00100`, `0100`, and `100`, which all map to same integer value. If this type of a column is changed from a `VARCHAR` to an `INT`, then it would be subject to the upsert behavior on primary keys. So, in this example, only one of the three rows would be preserved.

Be aware that data type conversion will preserve the data in the underlying database table, but there is no guarantee that any objects built on top of it (worksheets or pinboards) will be preserved. This is because you might make a data type change that makes a chart built on top of the table invalid (for example a growth chart would be invalidated if the date column it depends on were changed to a varchar column).

Supported data type conversions

In general, the data type conversions that make logical sense are supported. But there are a few nuances you should be aware of:

- When you convert from `INT` to `BOOL`, zero is converted to false, and all non-zero values are converted to true.
- When you convert from `BOOL` to `INT`, true gets converted to 1, and false gets converted to

0.
- When you convert from `DOUBLE` to `INT`, the value gets rounded.
- When you convert from `INT` to `DOUBLE`, the value gets rounded.
- When you convert from `DATETIME` to `DATE`, the date part of value is preserved and the time part is dropped.
- When you convert from `DATE` to `DATETIME`, the time gets added as `00:00:00`. The date part of the value is preserved.
- When you convert from `DATETIME` to `TIME`, the time part of the value is preserved.
- Conversion from TIME to `DATETIME` is not supported.

Date and time conversions

Some data type conversion require a format string. These include:

- conversion from `DATE` / `TIME` / `DATETIME`
- conversion to `DATE` / `TIME` / `DATETIME`

For these types of conversions, you'll use a special syntax using parsinghint and the date format specifications supported in the strptime library function.

For the example, first create a table with a timestamp stored as a `VARCHAR`:

```
CREATE TABLE fruit_sales
   (time_of_sale VARCHAR(32));

INSERT INTO fruit_sales
   VALUES ('2015-12-29 13:52:39');
```

Now, convert the column from a `VARCHAR` to `DATETIME`, using the format `%Y-%m-%d %H:%M:%S`:

```
ALTER TABLE fruit_sales
   MODIFY COLUMN time_of_sale DATETIME
   [parsinghint="%Y-%m-%d %H:%M:%S"]
```

Finally, convert the column back to VARCHAR:

```
ALTER TABLE fruit_sales
   MODIFY COLUMN time_of_sale VARCHAR(32);
```

String to boolean conversions

String to boolean conversions have format strings, too. You'll use `parsinghint` as you do for date and time conversions. You can choose among these approaches:

OPTION 1: Specify string values for both true and false. Any non-matching values get converted to null. In this example, “100” gets converted to true, and “0” gets converted to false. “-1” gets converted to null.

```
ALTER TABLE db
   MODIFY COLUMN s bool [parsinghint="100_0"];
```

OPTION 2: Specify a string value for true. Any non-matching value gets converted to false. In this example, “100” gets converted to true, “-1” and “0” get converted to false.

```
ALTER TABLE db
   MODIFY COLUMN s bool [parsinghint="100_"];
```

Option 3: Specify a string value for false. Any non-matching value get converted to true. In this example, “-1” and “100” get converted to true, and “0” gets converted to false.

```
ALTER TABLE db
   MODIFY COLUMN s bool [parsinghint="_0"];
```

String to boolean conversions

When converting from a string to a boolean, you must specify a string for true and false. By default, a string to boolean conversion generates `true` for `true`, `false` for `false`.

```
ALTER TABLE db
   MODIFY COLUMN b varchar(32);
```

But you may override the default strings that get generated by using `parsinghint`, as in this example:

```
ALTER TABLE db
   MODIFY COLUMN b varchar(32) [parsinghint="tr_fa"];
```

Change the Data Type of a Column

When you issue the TQL command to convert a column from one data type to another, the conversion is handled automatically. However, you’ll need to ensure that any visualizations built on top of the table display correctly.

You should always take a snapshot of your database before making any schema changes. This will allow you to revert back to the prior state if you make an error, or something doesn't work as you expected after the schema change.

When changing a data type in an existing table, be aware that answers and pinboards created on top of that table (or worksheets that include it) may change. This is because charts and aggregations depend upon the data type. So for example changing from `INTEGER` to `VARCHAR` could break charts that used the numeric data type `INTEGER` to calculate an average or a total. Because of this, use caution, and check all dependent objects before and after changing the data type, to ensure that they display as intended.

To change the data type of a column:

1. Connect to the database with the ThoughtSpot SQL Command Line (TQL).
2. Issue the command to change the data type using this syntax:

```
TQL> ALTER TABLE <table>
     MODIFY COLUMN <column> <new_data_type>;
```

For example:

```
ALTER TABLE fact100
   MODIFY COLUMN product_id int;
```

Import CSV files with tsload

Summary: The tsload command is a common way to import data from a CSV file.

Use ThoughtSpot Loader (`tsload`) to load data from a CSV text file into an existing table in ThoughtSpot. ThoughtSpot Loader (`tsload`) is a common way to import data. When using `tsload` , you can load larger datasets and make the loading process repeatable through scripting. The `tsload` command accepts flags that enable you to specify column and row separators, date or timestamp formats, null value representations, and similar parameters. Many of these options have defaults that you can override.

Before importing data, you need to Build the schema.

To use ThoughtSpot Loader, type the command `tsload` followed by the appropriate flags. You can see the list of the flags it accepts in the ThoughtSpot Loader flag reference or by issuing `tsload -help` .

tsload supports both full and incremental data loads. For incremental loads, an upsert (insert or update) is performed. If an incoming row has the same primary key as an existing row, it updates the existing row with the new values.

You can integrate tsload into your ETL environment for more automated data loads. Most ETL tools provide the ability to write target data into files and support scripted post-transformation actions that can include loading data into ThoughtSpot. This procedure describes manually loading data, but the tsload commands could be saved as a script:

1. Log in to the Linux shell using SSH.
2. Change to the directory where your CSV files are staged.
3. Use the following syntax to invoke `tsload` , specifying the appropriate flags and your data source file:

```
$ tsload --target_database=my_database
       --target_table=my_table --alsologtostderr
       --empty_target --source_file=my_file.csv --v 1
       --field_separator="separator_char"
```

This example imports the CSV file `ssbm_customer.csv` into the table CUSTOMER:

```
$ tsload --target_database=SAMPLE_DB
       --target_table=CUSTOMER --alsologtostderr
       --empty_target --source_file=ssbm_customer.csv
       --v 1 --field_separator "|"
```

4. Once the processing begins, you'll see messages to indicate the progress and then two summary messages after the load is complete.

```
 Started processing data row
 Source has 32 data rows, ignored row count 0
 Waiting for rows to commit...(please wait)

 Source summary
 --------------
 Data source:                     ssbm_customer.csv
 Source data format:              csv
 Header row?:                     no
 Tokenizer Options:               escape_char: "" fiel
d_separator: "|" enclosing_char: "\""
 Date format:                     %Y%m%d
 Date time format:                %Y%m%d %H:%M:%S
 Flexible mode?:                  no

 Load summary
 ------------
 Target table:                    CUSTOMER
 Should empty target?:            yes
 Status:                          Successful
 Rows total:                      32
 Rows successfully loaded:        30
 Rows failed to load:             0
 Rows duplicate/omitted:          2
 % of Rows successfully loaded:   93.75 %
 Load Rate (MB/s):                0.00 MB/s
 Load Rate (Rows/s):              3.53 Rows/s
 Start time (Wallclock):          Tue Jan 29 09:09:07
 End time (Wallclock):            Tue Jan 29 09:09:08
 Total load time = 1.13 seconds = 0.02 minutes = 0.00 ho
urs
 Data size = 50 bytes = 0.06 KB = 0.00 MB
```

5. In the load summary, be sure to check the **Rows duplicate/omitted** number. This indicates the number of rows (if any) that were omitted from loading because they did not satisfy the table constraints. A common cause of this would be a duplicate primary key. If any rows were omitted, review your CSV file, make the required adjustments, and then load it again.
6. Once your file has been loaded properly, repeat this process to load data from any additional CSV files.

Use a script to load data

Summary: Learn how to load one or files in bulk with a script.

If you need to load data from multiple CSV files, create a script to automate the process. You can also use a similar script to automate recurring data feeds.

Understand how to create a script

The data loading script is a text file that contains all the calls to `tsload` for loading the data from your CSV files.

The example script shown here uses the `cat` command to read the data file, and pipes it to `tsload`. When creating and testing your script, you may wish to replace each `cat` with `cat -10`, to load only the first ten lines of each file. This allows you to quickly run a test of your script. When the test succeeds for all the data files, you can then remove each `-10`, so the complete files will load when you run the script again.

1. Log in to the Linux shell using SSH.
2. Navigate to the directory that contains your CSV files and open a new file in a text editor.
3. Type in the commands to load the data.

   This example shows commands to load three files:

```
 cat Players.csv | tsload
    --target_database baseball --target_table "players"
    --empty_target --field_separator ","
    --max_ignored_rows 10 --bad_records_file bad_record
s.txt
    --has_header_row --alsologtostderr --null_value ""

 cat AllstarFull.csv | tsload
    --target_database baseball --target_table "allstarfu
ll"
    --empty_target --field_separator ","
    --max_ignored_rows 10 --bad_records_file bad_record
s.txt
    --has_header_row --alsologtostderr --null_value ""

 cat Appearances.csv | tsload
    --target_database baseball --target_table "appearanc
es"
    --empty_target --field_separator ","
    --max_ignored_rows 10 --bad_records_file bad_record
s.txt
    --has_header_row --alsologtostderr --null_value ""
```

4. Save the file.
5. Run the script:

```
$ ./load_baseball_data.sh
```

Loading data efficiently

If you have a very large data file that takes a long time to load, you can reduce the load time by splitting it up into multiple files and loading them in parallel using multiple invocations of `tsload`. If the size of any of your data files is greater than 50 million rows, running `tsload` in parallel can reduce the load time significantly.

- Split up your large data file into multiple smaller files.
- Stage the data files in a location accessible to the node on which you'll run the script. Usually you'll use an NAS mounted file system.
- Create a script to load the files (see example below).
- Run the script to load the files. You will make your script multi-threaded by invoking multiple loader threads (between 1 and 5 are recommended).

To optimize the load time even further, determine what the bottleneck is and adjust your process accordingly. If the disk I/O for reading the data files is the bottleneck, you can stage the data files on separate NAS mounted file systems and reference them accordingly in your script. If the CPU on the machine you're using to run the load script is the bottleneck, you can split the load script into the same number of parts as you have nodes in your ThoughtSpot instance, place one script on each node, and run them in parallel. Make sure the other nodes are able to access the data files where they are staged. Running the load script on separate nodes will put the data on all the nodes, just as when you run the script on a single node. Running the script on all the nodes at the same time just lets you take advantage of CPU power of each node for hashing data files.

For example, suppose you have 30 days of data in 30 files, one for the data collected on each day. Each day's data file contains 10 million rows, for a total of 300 million rows of data. You want to load the whole month of data. For this example we'll have 5 loader processes, each one handling 6 days of data. Here is a sample script you could use the load the data files in parallel:

```
/* Script load_script.sh, loads 30 days of data in parallel */

#!/bin/bash

pidlist=""

cat day1.csv day2.csv day3.csv day4.csv day5.csv day6.csv | tsl
oad
--target_database sales --target_table SALES_FACT --max_ignore
d_rows 10
--bad_records_file ./SALES_FACT.bad --date_format %Y-%m-%d
--date_time_format "%Y-%m-%d %H:%M:%S" --source_data_format del
imited
--field_separator "|" --null_value "" --enclosing_character
"\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &

cat day7.csv day8.csv day9.csv day10.csv day11.csv day12.csv |
tsload
--target_database sales --target_table SALES_FACT --max_ignore
d_rows 10
--bad_records_file ./SALES_FACT.bad --date_format %Y-%m-%d
--date_time_format "%Y-%m-%d %H:%M:%S" --source_data_format del
imited
--field_separator "|" --null_value "" --enclosing_character
"\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &

cat day13.csv day14.csv day15.csv day16.csv day17.csv day18.cs
v | tsload
--target_database sales --target_table SALES_FACT --max_ignore
d_rows 10
--bad_records_file ./SALES_FACT.bad --date_format %Y-%m-%d
--date_time_format "%Y-%m-%d %H:%M:%S" --source_data_format del
imited
--field_separator "|" --null_value "" --enclosing_character
"\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &
```

```
cat day19.csv day20.csv day21.csv day22.csv day23.csv day24.cs
v | tsload
--target_database sales --target_table SALES_FACT --max_ignore
d_rows 10
--bad_records_file ./SALES_FACT.bad --date_format %Y-%m-%d
--date_time_format "%Y-%m-%d %H:%M:%S" --source_data_format del
imited
--field_separator "|" --null_value "" --enclosing_character
"\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &

cat day25.csv day26.csv day27.csv day28.csv day29.csv day30.cs
v | tsload
--target_database sales --target_table SALES_FACT --max_ignore
d_rows 10
--bad_records_file ./SALES_FACT.bad --date_format %Y-%m-%d
--date_time_format "%Y-%m-%d %H:%M:%S" --source_data_format del
imited
--field_separator "|" --null_value "" --enclosing_character
"\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &

wait $pidlist
```

Call your script using a command like:

```
nohup bash ./load_script.sh > master_log.txt &

tail -f master_log.txt
```

Constructing your script in this way will execute all the commands in the background, and output to the file `master_log.txt`. You'll see a running status as the commands in the script execute. After the script completes, you can check the log file for detailed information, such as the number of rows that loaded successfully.

Delete a data source

Summary: How to prepare for and delete a data source using the ThoughtSpot application.

There are two separate ways to delete a data source, through the browser or through TQL describes the dependency checking that occurs when deleting or changing a table using TQL. When you want to delete a data source, you first need to handle any dependent objects that have been built on top of it. You can easily see these dependencies, and choose how to handle them before deleting the data source.

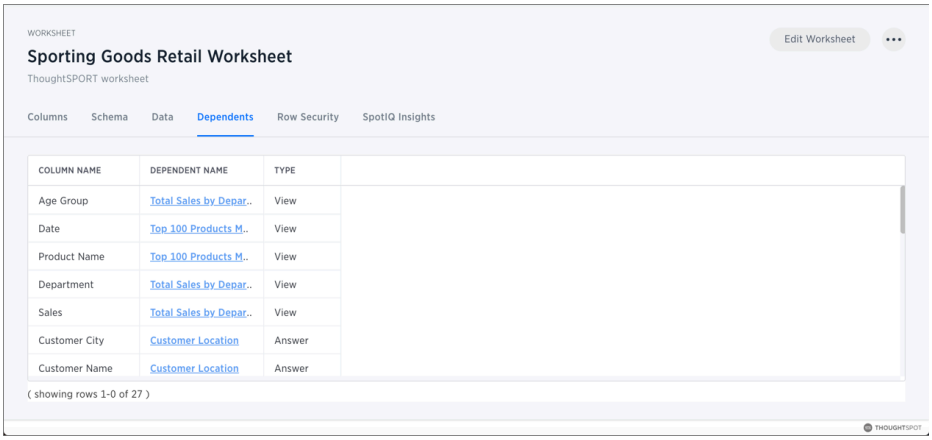
Check data source dependencies

You can see all of the dependencies for any data source (worksheet or table) on the **Data** page.

To view dependent objects for a data source:

1. Click **Data** on the top navigation bar.
2. Click the name of the data source whose dependencies you want to view.
3. Click **Dependents**.

   You will see a list of the names of the dependent objects (worksheets and pinboards), and the columns they use from that data source. You can use this information to determine the impact of changing the structure of the data source or to see how widely it is used.



4. Click on a dependent object to modify or delete it.

   If you want to remove the dependency by modifying the dependent object, you'll need to remove all search terms or columns that refer back to the data source you are trying to delete.

5. When all dependencies have been removed, you will be able to go back and delete the data source.

Delete a data source

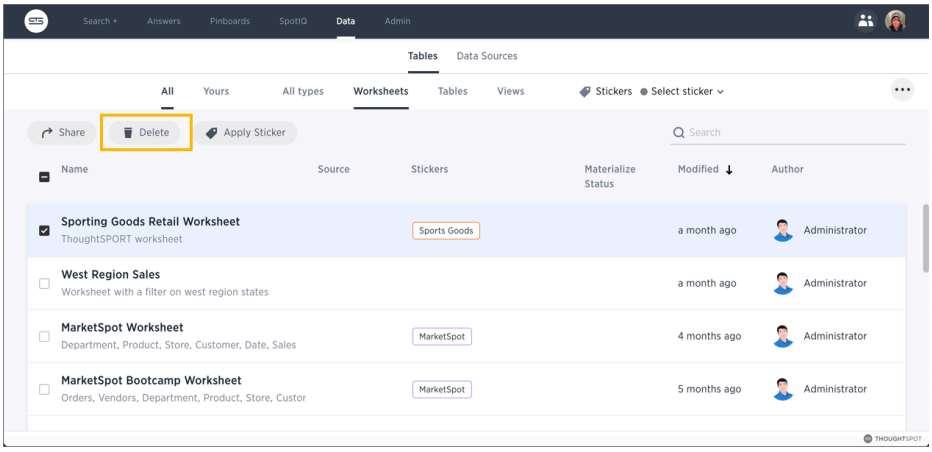
You can delete data sources from the browser, as long as they were not created by an administrator through `tsload` or Data Connect.

You can delete data sources from the browser if they were created from the browser. These types of data sources include:

- Data imported from the browser.
- Worksheets.

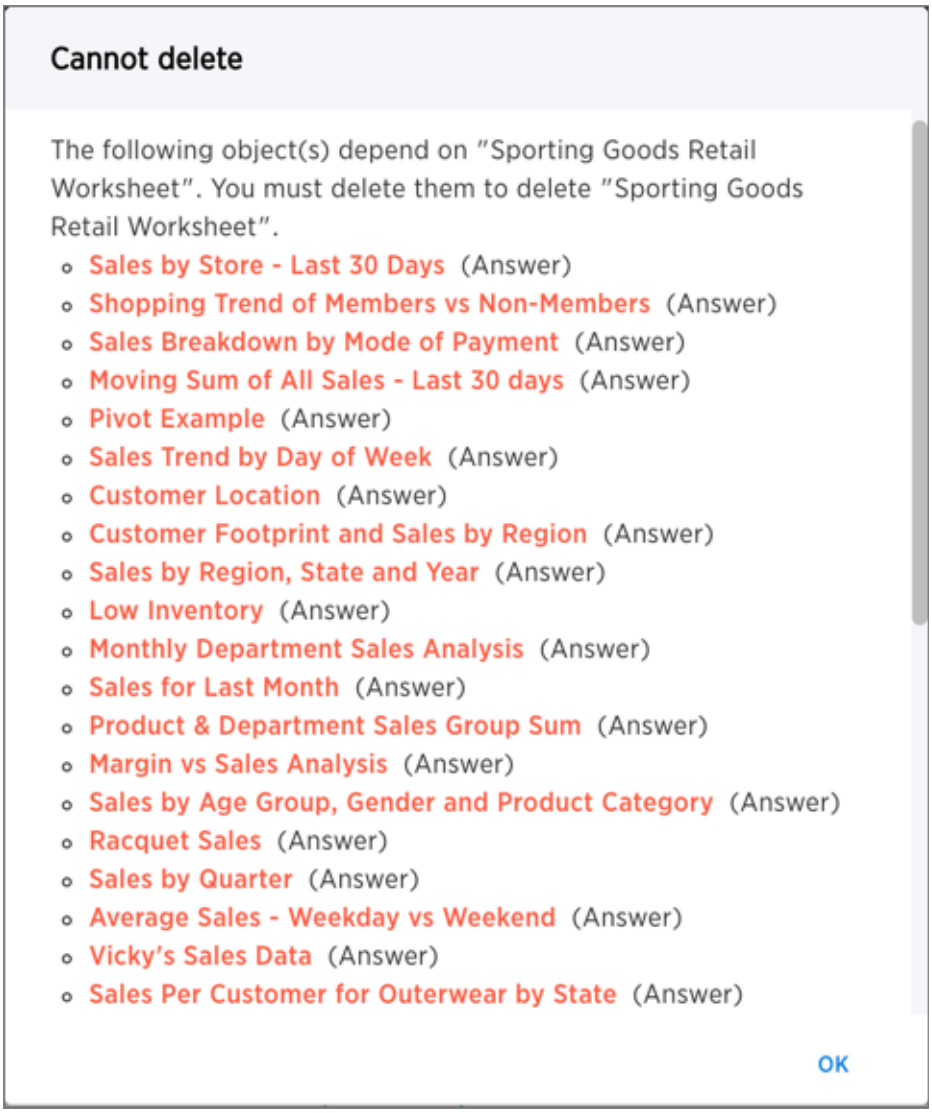
ThoughtSpot checks for dependencies whenever you try to delete a table or worksheet.

1. Click **Data** on the top navigation bar.
2. Check the box next to the name of the data source you want to delete.
3. Click the delete icon.



4. If you attempt to delete a data source with dependent objects, the operation will be blocked.

   You will see a list of dependent objects with links.



5. Click on a dependent object to modify or delete it.

   If you want to remove the dependency by modifying the dependent object, you'll need to remove all search terms or columns that refer back to the data source you are trying to delete.

6. When all dependencies have been removed, you will be able to go back and delete the data source.

Delete or change a table in TQL

Summary: You can delete a data source in the web browser or using ThoughtSpot SQL Command Line (TQL).

When you enter a TQL statement, the system warns you of possible dependency consequences with a prompt asking if you'd like to proceed. This should make you feel safe issuing TQL commands, even commands like dropping a table.

If TQL is run using the flag `--allow_unsafe`, your statements will always execute without this warning. Note that when running TQL from a script, you will need to decide what behavior you want if the script contains changes that affect dependent objects. If you want the script to run even if objects with dependencies are affected, run it using this flag, for example:

```
cat safest_script_ever.sql | tql --allow_unsafe
```

If you do not run the script using the flag, it will fail if any of its commands might cause problems with dependent objects.

TQL actions with possible dependency consequences include:

- Change, add, or remove a primary key.
    - When changing or adding a primary key, if the key in question is not unique in the data it may cause deletion of rows, because of upserts occurring when duplicate primary keys are found.
    - When changing or removing a primary key, incoming foreign key relationships will be broken.
- Change a column datatype.
- Add a relationship or foreign key.
- Drop a relationship or foreign key constraint.
- Change or add a sharding key.
- Drop a table, schema, or database.

When issuing one of the above commands, you will see a warning message similar to this:

```
TQL> ALTER TABLE table1
     DROP CONSTRAINT PRIMARY KEY;

WARNING: This operation will break the Foreign Key relationshi
p “products”
with table “sales”, which will break 34 user-visible visualizat
ions and
2 Worksheets. We recommend taking a snapshot before performing
this operation.
Do you wish to proceed? (yes/no).
```

About data modeling

Summary: Modeling, tagging, and adding links between your data sources can make the data even easier to search.

Data modeling allows you to define metadata and other aspects of your data. For example, you can give data columns search friendly names or predefine how they can be explored and aggregated. Metadata include such information as **Column Names**, **Column Visibility**, **Column** and **Data Definition**, **Column Rank** and so forth.

When you load data, ThoughtSpot has defaults for data modeling metadata. After loading data, you can start searching your data without doing any data modeling, creating relationships, or tagging. However, since you know your data best, you can customize the modeling settings. Putting some thought into these will make the data even easier and more intuitive to search for your end users.

User interfaces for modeling data

Data modeling is a very lightweight process compared to what you may have experienced in other tools. You can configure the model for an individual data table or you can view and configure all the system data using a modeling file. Editing the data model file requires that you have administrative privileges.

The model file contains a row for each column in your data set. It isn't unusual to have tens of thousand of rows in this file. This means editing this file is equivalent to editing all the tables at once. When you add new data to your system, this file expands to accommodate the new data columns you have added.

Both of these methods, have the same effect, they improve search. Moreover, while they have different effects of scale, they use the same mechanisms to accomplish these effects.

Modeling topics

The following topics explain how to model your data:

- Change a table's data model
  Explains how to make modeling settings for a table you've just loaded, or to make a quick change to existing settings.
- Edit the system-wide data model
  Explains how to define a default data model to use for data system-wide.
- Data modeling settings
  Explains the possible data model settings and their accepted values. These are the same for a table or the system.
- Link tables using relationships
  Linked tables can be searched together or combined into a worksheet for easy searching. Tables that have no relationship between their columns can not be combined in a single search.
- About stickers
  You can create stickers to make it easier for people to find data sources and pinboards.

Change a table's data model

Summary: You can adjust the data model for a newly loaded table.

To make modeling settings for a data source you've just loaded, or to make a quick change to existing settings, use the ThoughtSpot web interface. You can adjust the **Columns** settings from the data management listing.

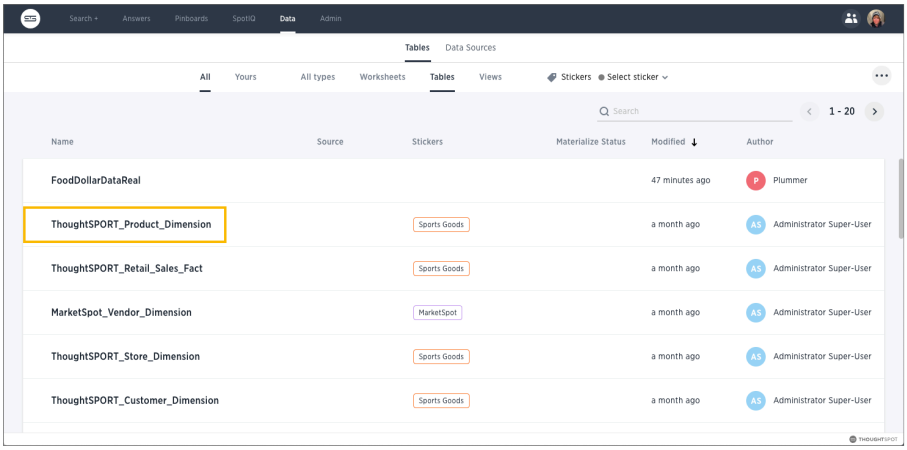
You can change all the same data model settings here as in the model file. This method is easier and faster, unless you need to make many settings in bulk. In that case, using the model file is recommended.

About data sources

You can change the data modeling settings for base **Tables**, **Worksheets**, and Views. Worksheets will inherit the data modeling settings from the tables upon which they are based. However, if you make further changes to a base table *after* you've created worksheets on it, the new data model changes will not propagate up. You will need to make any new data model changes directly to the worksheets (if you want them).

Change the data model for a data source

1. Click **Data** on the top navigation bar.
2. Click on a data source you own or can edit.

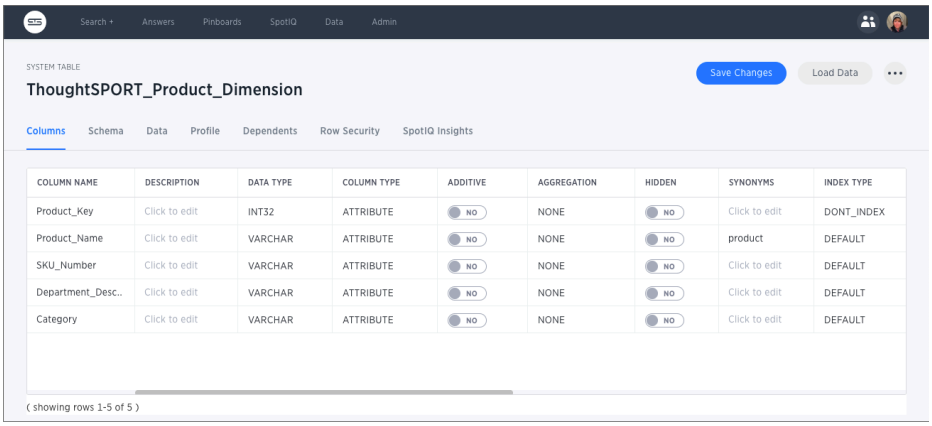


   This brings up the **Columns** screen, where you'll make your modeling settings.

3. Modify one or more column settings.

   Descriptions of the possible settings are listed in Data modeling settings.

4. Save your changes.



5. To check your changes, use the **SEARCH** page to search for across the changed data.

Related information

- Data modeling settings
- Edit the system-wide data model
- Understand data sources

Edit the system-wide data model

Summary: Edit the modeling file to edit your data settings.

When you load data, ThoughtSpot uses defaults for data modeling metadata. You change these defaults using the data modeling file if you have access to the **ADMIN** > **Data Management** page. This means editing this file allows you to navigate and edit all your system's data columns at once, in bulk. When you (or your users) add new data to your system, this file changes as it expands to accommodate new data columns.

> ☑ **Tip:** If you just want to change a subset of your data, use the Change a table's data model instead.

Overview of the modeling process

The data formats you use in your system are controlled by the modeling file, an Excel file. To make these changes you download the model file, change the model, and upload your changes back into the system.

In each row of the modeling file, all the data properties corresponding to a column from your data are listed. You can modify many of these properties by typing in the new value. Remember these important guidelines when editing the model file:

- Do not modify any value in a column which contains **DoNotModify** in the field under the column heading.
- Make sure to keep the file in the same format as it had when you downloaded it.

The model file contains a row for each column in your data set. It isn't unusual to have tens of thousand of rows in this file. You can change all or a subset of rows. You can edit the file to leave the heading rows and only those rows you want to change. This can make the file more convenient to work with.

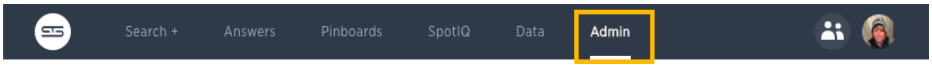
The model file must be saved as UTF-8 encoded. If your model file includes multi-byte characters, make sure you save it in the correct format or you won't be able to upload it after making your changes.

Download the model file

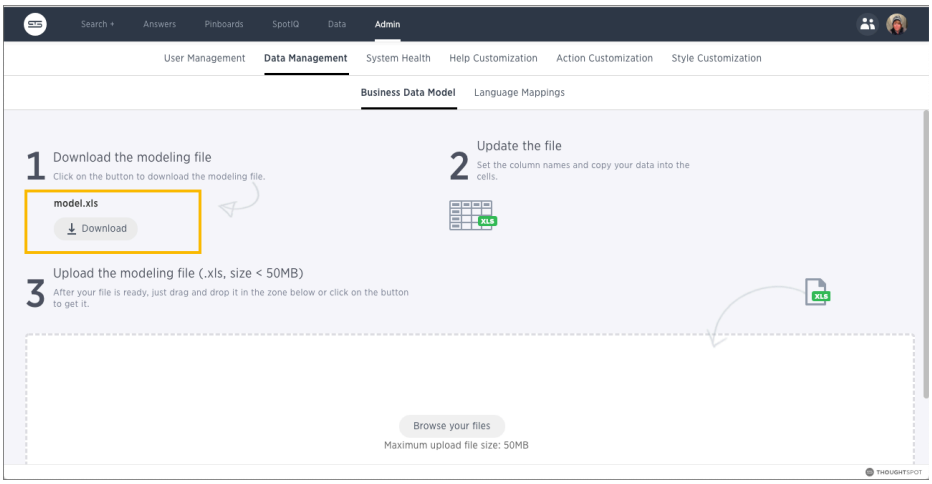
Before you can make changes to the model file, you need to download it. Then, you edit it using Microsoft Excel, vi/vim, or a similar text editing tool.

To obtain the model file:

1. Log in to ThoughtSpot from a browser as an Administrator user.
2. Click on the **admin** tab in the top navigation bar.



3. Click on **Data Management**.
4. Click **Download model.xls**.

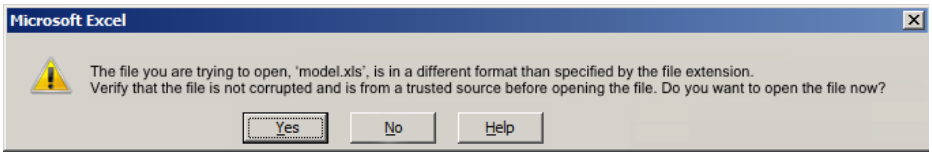


Edit the file and change the settings

You'll make changes to the settings using this procedure. To see a list of the changes you can make, see Data modeling settings. You can edit any of the values in the model file, except for those where the words **DoNotModify** appear below the column header. To make changes in the model file:

1. Open the model file you downloaded (`model.xls`) in Excel, vi/vim, or a text editor.

   If you are using Excel, you may see a warning message.

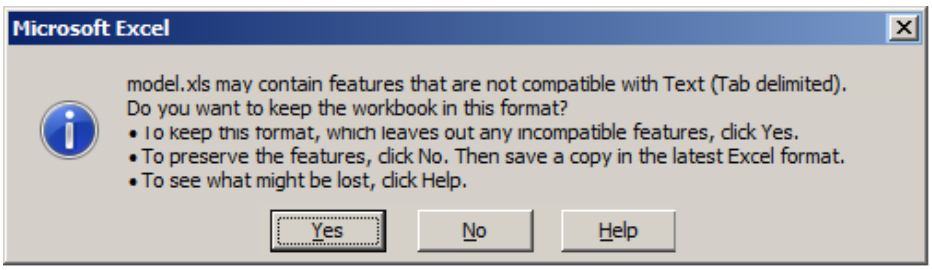
   Click `YES` to proceed.

2. Find the column you want to modify.

   Descriptions of the meanings of the columns are listed in Data modeling settings.

3. Select the value you want to change.
4. Type in the new value.
5. After making all your changes, save the model file.

   If you are using Excel, you will see a message. Click `YES` to save the file.

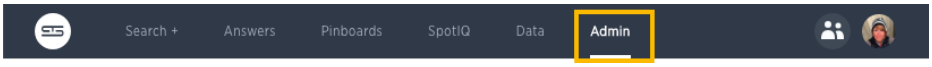
   

The model file must be saved as UTF-8 encoded. If your model file includes multi-byte characters, edit the file using vi or vim to ensure the file is saved in the correct format. Otherwise, you won't be able to upload it after making your edits.

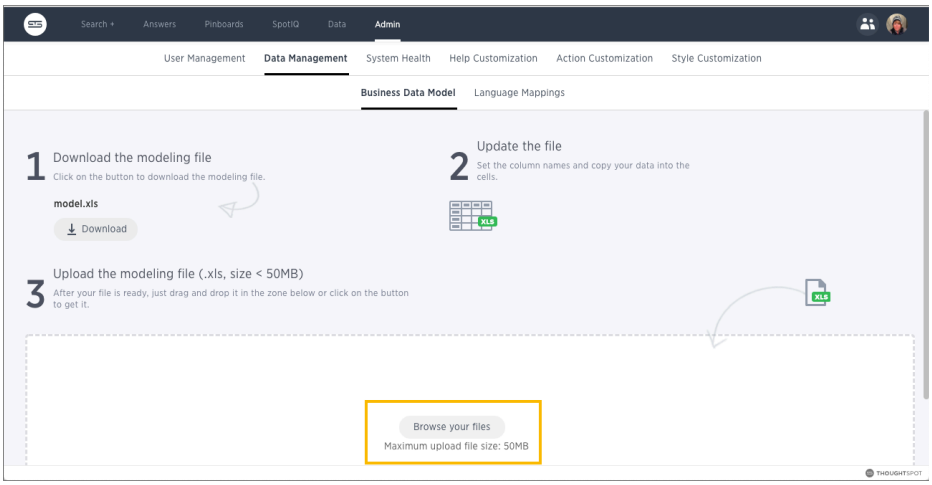
Upload the edited file

After you have made changes to the modeling file, you must upload it back to ThoughtSpot before the changes will take effect. To upload the model file:

1. Log in to ThoughtSpot from a browser as an Administrator user.
2. Click on the **Admin** icon, on the top navigation bar.



3. Click on **Data Management**.
4. Click **Browse your files** to upload the model.xls file, or drag and drop it in the zone.



If you receive an error message upon uploading the file, check that it does not include any multi-byte characters (for example, Japanese or other multi-byte language characters). If it does, you'll need to download the file again and make your edits using vi or vim.

If you choose to remove all the rows you have not changed from the model file before uploading it. If you upload a model file that includes only the changed rows, you won't lose any of the pre-existing model file settings. This is a good option if your model file is causing an error on upload, but you aren't sure where in the model file the problem is.

As soon as the file is uploaded, ThoughtSpot performs any necessary re-indexing for you automatically. Your new settings will be reflected within a few minutes.

Related information

- Data modeling settings
- Change a table's data model

Overview of data modeling settings

You can change these settings in two ways, both of which change the model. If you want to make a few small changes, you should make them in the ThoughtSpot application . If you want to make many changes you should edit the modeling file. Whether you are changing data modeling settings using the modeling file or the Web interface, the settings and their accepted values are the same.

Modeling settings

This index lists the editable data modeling settings.

Setting name	Description
Column Name	Sets the name of the column to be used in searches.
Description	Adds a text description of what the column contains.
Data Type	Read only. Shows the column's data type.
Column Type	Sets the type of column, either `ATTRIBUTE` or `MEASURE` .
Additive	Controls the type of aggregations that will be available for a column.
Aggregation	Sets the default aggregation type for a column.
Hidden	Sets the visibility of a column.
Synonyms	Adds synonyms that can be used in the search bar to refer to a column.
SpotIQ Preference	Excludes specified columns from SpotIQ analyses. By Default, all columns are included in SpotIQ.
Index Type	Sets the type of index that will be created for a column.
Geo Config	Enables a column to be used in GeoMap visualizations.
Index Priority	Changes the priority of a column in search suggestions.
Format Pattern	Specifies the format to use for numeric values or dates that show in the column.
Currency Format	Specifies the format to use when showing the currencies in a column.
Attribution Dimension	Only applies to tables that join over a Chasm Trap. Designates whether the tables depend on this column for attribution.

Setting name	Description
Entity Category	Specifies how to categorize the data in the column by entity type. By default, `ENTITY TYPE` is not set. Entity categories support SearchIQ so that when you type a natural question, ThoughtSpot better knows how to interpret it. For example, if you ask "who are the top performers?" ThoughtSpot will first choose columns set with `PERSON` from which to return answers. If you ask "when was the movie Jurassic Park released?", columns set to `TIME` will be used to answer the "when" part of the question, and so forth.

Data modeling for worksheets

For worksheets, only some of the settings can be modified, whether you are using the modeling file or the Web interface. The editable settings for worksheets are:

- Name
- Description
- Synonyms

If you want to change any of the settings that cannot be modified in a worksheet, you need to make your changes to the underlying table instead, and they will be reflected in all worksheets that use the table.

Related information

- Model the data for searching
- Add a geographical data setting for a column

Set column name, description, and type

Summary: Modeling includes setting basic information for a data column such as its name, description, and type.

Basic information for a data column is its **NAME**, **DESCRIPTION**, and **TYPE**. All of these can influence how a user experiences your data. For example, the **DESCRIPTION** appears as a "tip" when a user hovers over a column. So it is the means for helping users understand where the data comes from.

Change the column name

Column Name (UI)/**ColumnName** (model file) is the name that displayed to users for that column in ThoughtSpot. The column name is what users type to add that column to their search. Change the text that is shown for the column names in ThoughtSpot to make the names more meaningful to users.

The model file contains a row for each column in your data set. It isn't unusual to have tens of thousand of rows in this file. You can change all or a subset of rows. You can edit the file to leave the heading rows and only those rows you want to change. This can make the file more convenient to work with.

The default is the name you gave the column when you defined the table in the database or imported the CSV file from the browser.

1. Find the column name you want to change.
2. Type in the new column name.
3. Save your changes.

Change column description

Description (UI)/**ColumnDescription** (model file) an optional description for the corresponding column. You can provide a description for a specific column, to provide additional information for users about the data it contains. When a user hovers over the column, a tooltip will show this description.

To create a column description:

1. Find the column description you want to change.
2. Enter a new description.
3. Repeat for all columns where you want to add a description.
4. Save your changes.

Change column type

Column Type (UI)/**ColumnType** (model file) describes the kind of data a column stores. This is set automatically upon defining the table, but in some cases, you may want to change the type. There are two types of columns:

- `ATTRIBUTE` contains a property, like name, address, or id number.
- `MEASURE` contains a numeric value that can be compared in a meaningful way using math, such as a count or measurement.

When a new table is created, the default column type is set according to the **Data Type** (UI)/**DataType** (model file) defined for each column. By default, columns with the numeric data types (`FLOAT` , `DOUBLE` , `INT` , or `BIGINT`) are assigned the type `MEASURE` . Columns with `VARCHAR` , `BOOL` , or date/time data types are assigned the type `ATTRIBUTE` .

Usually the default setting for column type works fine. But occasionally you'll need to change a `MEASURE` to an `ATTRIBUTE` . Examples of numeric values for which mathematical operations are not meaningful include:

- ID numbers
- Key values that are primarily used for joining tables
- Product number or SKU
- Sports team member jersey number
- Year, when separate from a date (e.g. 1999, 2000)

To change the column type:

1. Find the column type you want to change.
2. Change it to either `MEASURE` or `ATTRIBUTE` .
3. Save your changes.

Related information

- Model the data for searching
- Hide column or define a column synonym

Set ADDITIVE or AGGREGATION

Summary: You can allow aggregate on MEASURE columns and some ATTRIBUTE columns.

Aggregation is grouping many units or parts into a new value. In data, aggregation gathers multiple input values and calculates an summary value from them. You then use this aggregated value to do an analysis.

Every summary value results from a data aggregation function. An example aggregation function would be average or minimum. You can control how aggregation works in your data.

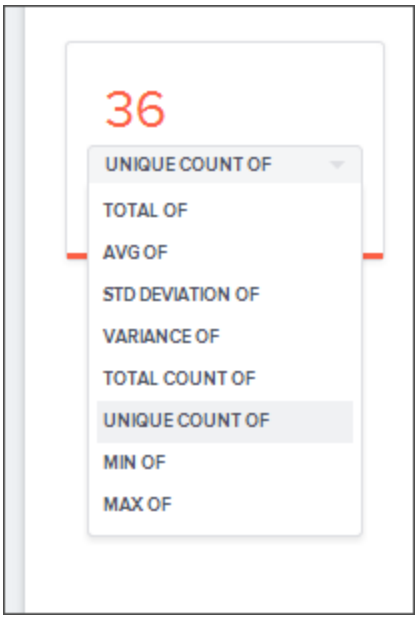
Making an ATTRIBUTE column ADDITIVE

Your data may contain a column with a numeric data type that you have defined as an `ATTRIBUTE` rather than a `MEASURE`. For example, you may have `ATTRIBUTE` column with an `INTEGER` data type that represents age. Typically, these columns have an **ADDITIVE** setting of `NO`. Within a search result that contains data from this column, the options for each column on the left side of the screen includes:

- UNIQUE COUNT OF
- TOTAL COUNT OF

To display extended aggregate view options, you must set **ADDITIVE** to `YES` on these `ATTRIBUTE` columns. This option is only possible on columns that have a numeric data type (`FLOAT`, `DOUBLE` or `INTEGER`) or a date data type (`DATE`, `DATETIME`, `TIMESTAMP`, or `TIME`). After you make this change, these additional view options area-charts offered:

- TOTAL OF
- AVG OF
- STD DEVIATION OF
- VARIANCE OF
- TOTAL COUNT OF
- UNIQUE COUNT OF
- MIN OF
- MAX OF



To change this setting:

1. Find the column whose **ADDITIVE** setting you want to change
2. Select the **ADDITIVE** toggle.
3. Change the value to one of these:
    - `YES` or `NO`, if using the Web interface.
    - `TRUE` or `FALSE`, if using the model file.
4. Save your changes.

Change Aggregation

Both `MEASURE` columns and `ATTRIBUTE` columns support **AGGREGATION** operations. To aggregate a column without having to enter the aggregation type explicitly in your searches every time, you can set a default **Aggregation** for that column. Setting this default can make combining data more intuitive and faster.

`ATTRIBUTE` columns have **AGGREGATION**(UI)/**AggregationType** (model file) values with default aggregate type of **NONE**. You can change **AGGREGATION** to one of the supported aggregation types. To extend the available aggregation actions, set **ADDITIVE** on these columns to `YES` (`TRUE`).

Aggregate type	Description
NONE	Does no aggregation. This is the default for `ATTRIBUTE` type columns.
SUM	Adds the values together and returns the total. This is the default for `MEASURE` type columns.
AVERAGE	Calculates the average of all the values.
MIN	Calculates the minimum value.
MAX	Calculates the maximum value.

Aggregate type	Description
STD_DEVIATION	Calculates the standard deviation of all the values.
VARIANCE	Calculates the variance of all the values.
COUNT	Calculates the total number of values.
COUNT_DISTINCT	Calculates the total number of distinct values.

Keep in mind that not all `MEASURE` data should be aggregated. Consider a table containing data about athletes on a sports team. The data contains some numerical values, including points scored, salaries, and jersey numbers for each of the players. Because jersey number is an INTEGER, it would become a column of type `MEASURE` (not `ATTRIBUTE`). So it will aggregate, by default. But you may want to make its aggregation type **NONE** instead. This ensures that search results that include jersey number will not attempt to compare or aggregate those values in a way that is not meaningful.

To set this value.

1. Find the column whose default aggregation type you want to change
2. Select its **Aggregation**. If using the modeling file, use the **AggregationType** setting.
3. Select the new default aggregation type.
4. Save your changes.

Related information

Model the data for searching

Hide a column or define a synonym

Summary: Hide a column from users or make it easier to find by assigning a synonym.

You can hide columns from users in ThoughtSpot without dropping them from the database. It is common to hide a column when its data contains identifier columns that are used to join tables, but which do not have any meaning to users.

Alternatively, rather than hiding a column, you can make it easier to find by creating synonyms for it. This is helpful, for example, when different departments refer to the data using different terminology.

Hide a column

As the number of columns in the dataset increases, the search experience requires more effort. Users have to navigate through larger numbers of columns to choose the correct one. There might also be some columns in the dataset that you don't want to expose to the users.

Change the **HIDDEN** (UI)/**Hide** (model file) setting to hide a column. By default, all columns in a data source ware shown in ThoughtSpot. To hide these columns, set the **HIDDEN** setting to `YES`.

1. Find the **HIDDEN** (UI)/**Hide** (model file) setting for a column.
2. Set its value to `YES`.
3. Save your changes.

Create synonyms for a column

When users search a data source, they might try typing different words to try to retrieve a particular column. This could be due to different groups in your organization using different terms for the same data. Or maybe your users just intuitively use different words when searching for that item. Using synonyms allows them to access the data even if the term they choose isn't the same as the actual column name.

You can set column synonyms for columns in tables, user imported data, and worksheets. The returned table or chart uses the *actual column name*, but the search bar reflects the term the user typed in (the synonym).

To create a column description:

1. Find the column for which you want to add synonyms.
2. Select its **Synonyms**.
3. Type in a comma separated list of the synonyms you want to add.

   If a synonym is more than one word, it must be enclosed in double quotes. If you are using the Web interface, you would type:

   ```
   profit,"gross profit"
   ```

If you are using the model file, the list of synonyms must be enclosed in square brackets. For example:

```
[profit, "gross profit"]
```

4. Save your changes.

Related information

Model the data for searching

Set columns to exclude from SpotIQ analyses

Summary: You can specify columns to exclude from SpotIQ analyses.

SpotIQ is a ThoughtSpot feature that provides users with insights about their data by automatically surfacing interesting characteristics (trends, correlations, outliers, and so on).

If you have access to tables, worksheets, and views for data modeling purposes, you can specify columns to exclude from SpotIQ analyses. By default, all columns are *included* in SpotIQ analyses.

Exclude columns from SpotIQ analyses

To specify columns to exclude from SpotIQ analyses:

1. Click **Data** in the top menu, and choose **Tables**, **Worksheets**, or **Views**.
2. Click on the name of your data source.
3. On the **Columns** tab, find the COLUMN NAMES you want to exclude from SpotIQ analyses, and scroll to the right to find `SPOTIQ PREFERENCE`.
4. Use the drop-down menu to set the `SPOTIQ PREFERENCE` to **EXCLUDE** for each column you want to exclude.
5. Click **SAVE CHANGES** in the upper right.

Include columns in SpotIQ analyses

By default, all columns are included in SpotIQ analyses.

If you have previously set some columns to EXCLUDE and you want to re-set these to be included, do the following.

1. Click **Data** in the top menu, and choose **Tables**, **Worksheets**, or **Views**.
2. Click on the name of your data source.
3. On the **Columns** tab, find the COLUMN NAMES you want to set back to include in SpotIQ analyses, and scroll to the right to find `SPOTIQ PREFERENCE`.
4. Use the drop-down menu to set the `SPOTIQ PREFERENCE` to **DEFAULT** for each column you want to include.
5. Click **SAVE CHANGES** in the upper right.

Related information

- SpotIQ tutorial
- Overview of data modeling settings

Manage suggestion indexing

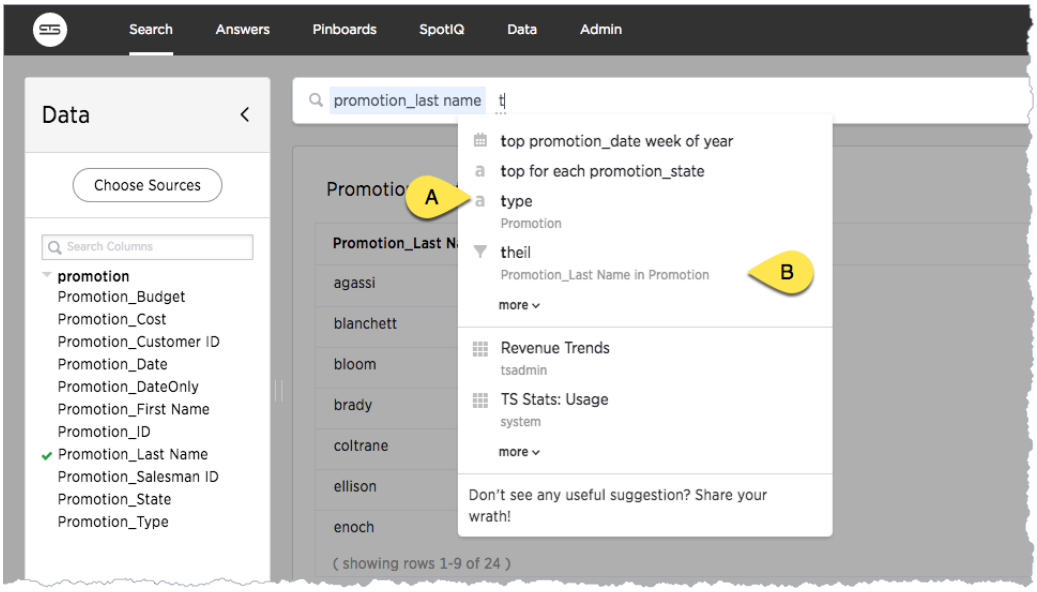
Summary: ThoughtSpot dynamically indexes Search bar suggestions for column names and values.

When a user searches in the **Search** bar, ThoughtSpot supplies the user with suggestions for column names and their column values. The **COLUMN NAME** and any **SYNONYMS** appear in **Search** suggestions. A column's **INDEX TYPE** controls whether and how ThoughtSpot suggests column values.

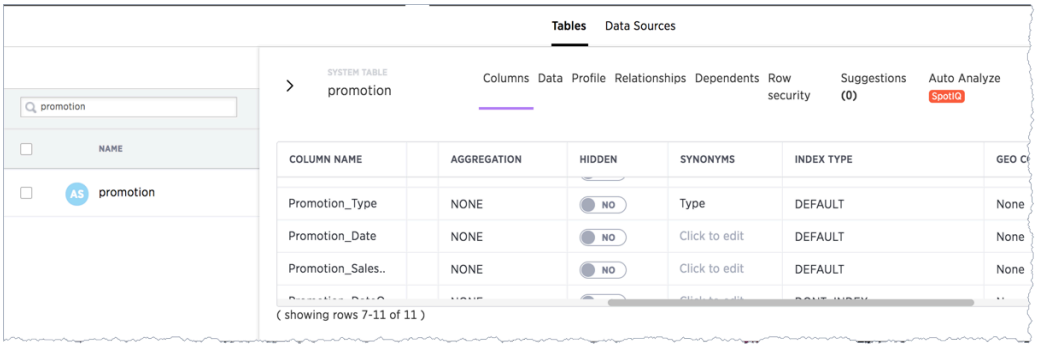
Additionally, ThoughtSpot uses a column's **INDEX PRIORITY** value to determine where to rank a column's name and values in the search suggestions. These values impact the dynamically calculated *usage based ranking (UBR)*,

Example of Search suggestion behavior

The example below illustrates how searching for `promotion_last_name t` causes the system to suggest several ways of completing the `t` in the search:



The system is suggesting the synonym `type` (callout A) for a column in the `Promotion` table. It is also suggesting a value of `theil` (callout B) for the `Promotion_Last Name` column. If you look in the **Data** > **Tables** page, you'll see there is a `type` synonym for the `Promotion_Type` column which is using default indexing.



Managing search suggestions through **INDEX TYPE** and **INDEX PRIORITY** is important. Properly configured suggestions can decrease “noise” in the suggestion list. Increasing the visibility of important columns is helpful for new or intermittent ThoughtSpot users.

Understand the default indexing behavior

ThoughtSpot has a system default **INDEX TYPE** behavior for search suggestions. This system default is configured on your cluster and applies to all worksheets and tables. You can override this default behavior on a per-column basis.

The system behavior when the **INDEX TYPE** is **DEFAULT** is as follows:

- With two exceptions, the system indexes all columns using their **COLUMN NAME** value. The exceptions are columns with **COLUMN TYPE** of `MEASURE` and columns with **DATA TYPE** of `DATE`.
- Columns that contain data values with large amount of free-form strings, that is, a length is greater than 50 words, are indexed as `PREFIX_ONLY` by default.

  > ❗ **Warning:** If a column has a very large free text values, ThoughtSpot recommends you keep `DEFAULT` or set `DONT_INDEX`. Other settings indexing on these values may generate confusing suggestions.

- Short strings (like a `firstname` column) are indexed using `PREFIX_AND_SUBSTRING` by default, which indexes both prefix and substrings.
- If a column is using has a *cardinality* – the number of unique column values – greater than 10 million, it is not indexed.

If a column’s **INDEX TYPE** is *not* **DEFAULT** and the column’s cardinality is greater than 30 million, ThoughtSpot does not index the column.

High cardinality and performance

A column’s cardinality can impact indexing. If you have a column with a very high cardinality and a very high number of rows, indexing these values can impact your ThoughtSpot performance. ThoughtSpot Support recommends you turn off indexing of primary key columns on extremely large tables (> 10 million rows) in your cluster.

High cardinality is relative to other considerations. In some cases, columns with fewer than 10 million rows but with columns containing long strings can cause performance problems with memory. If you have concerns or questions, your ThoughtSpot Customer Success Engineer can help you determine appropriate cardinality thresholds for your ThoughtSpot installation.

Configure your own cluster defaults

If you need to, you can work with ThoughtSpot Support or your Customer Success Engineer to configure new cluster defaults.

Override the system default on a column

You can change a column's **INDEX TYPE** in the **Data** > **Tables** > **Columns** page or in the **Index** value in the modeling file.

The values you can set for **INDEX TYPE** are:

Index type	Description
`DEFAULT`	The default behavior applies to all `ATTRIBUTE` columns that are not `DATE` types. `PREFIX_AND_SUBSTRING` for short values and `PREFIX_ONLY` for long values and free-form text.
`DONT_INDEX`	Prevents indexing on the column values. The column doesn't appear in search suggestions.
`PREFIX_AND_SUBSTRING`	Allows full indexing such that prefix and sub-string search both work for the column values.
`PREFIX_AND_WORD_SUBSTRING`	Allows indexing such that only prefix search works for each word of a multi-word string, for the column values.
`PREFIX_ONLY`	Allows indexing such that only prefix search works for the column values.

Consider a column in which there are four values `ThoughtSpot`, `Thought`, `Spot` and `Thought Spot`. If you search for `sp`, depending on the setting for indexing, the column value search result suggestions will vary:

Index field value	Search bar suggestions
`DEFAULT`	`ThoughtSpot`, `Spot` and `Thought Spot`
`DONT_INDEX`	No suggestions.
`PREFIX_AND_SUBSTRING`	`ThoughtSpot`, `Spot` and `Thought Spot`
`PREFIX_ONLY`	`Spot`
`PREFIX_AND_WORD_SUBSTRING`	`Spot` and `Thought Spot`

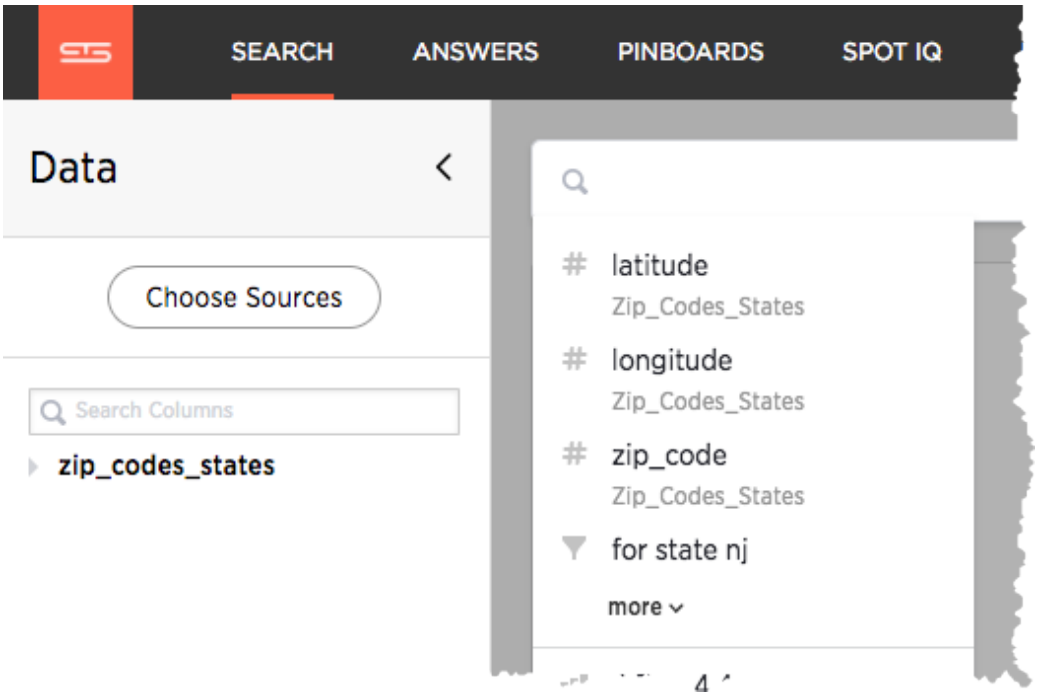
To change a value in the application UI:

1. Open a worksheet or table from the **Data** page.
2. Find the column whose index type you want to modify.
3. Set its **INDEX TYPE**.
4. Save your changes.

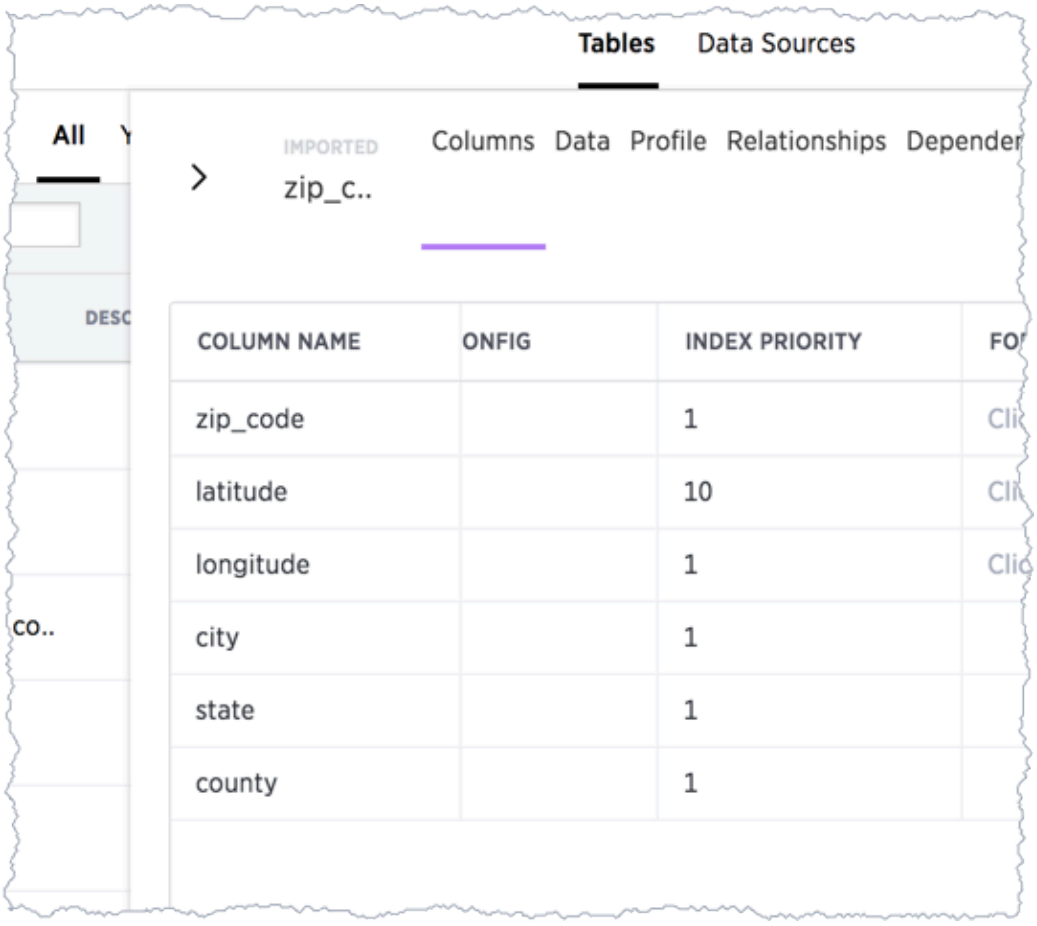
If you are using the model file, locate the **Index** cell, and enter the **INDEX TYPE** you want to use.

Change a column's suggestion priority

A column's **INDEX PRIORITY** determines the order or rank in which it and its values appear in the search dropdown.



By default, the **INDEX PRIORITY** value is set to `1` for all columns. You can push a column up in the order (increase the rank) by increasing its **INDEX PRIORITY** value. A higher value (like `2`) will cause the corresponding column and its values to appear higher up in the search dropdown than columns with lower value (like `1`).



You should only use numbers between 1-10 in the **INDEX PRIORITY** field.Use a value between `8-10` for important columns to improve their search ranking. Use `1-3` for low priority columns.

To change a value in the application UI:

1. Open a worksheet or table from the **Data** page.
2. Find the column whose index type you want to modify.
3. Change the **INDEX PRIORITY** to a number between 1 and 10.
4. Save your changes.

If you are using the model file, locate the **Index** cell, and enter the priority you want to use.

Related information

- Model the data for searching
- Usage based rankings (UBR).

Add a geographical data setting

Certain attribute columns that contain location data can be used to create GeoMaps. ThoughtSpot supports Latitude, Longitude, Zip Code, US States, US Counties, Countries, and select international sub-nation regions.

You can designate a column as `Geo` by editing the **GEO CONFIG** column in the table **Columns** page. You cannot edit the geo configuration column information in the `model.xls` file.

Guidelines for geographic columns

Columns that can be designated as `Geo` columns need to contain text (`VARCHAR`) data unless they contain latitude/longitude data. Latitude and longitude columns can contain numeric data (`DOUBLE`) or text.

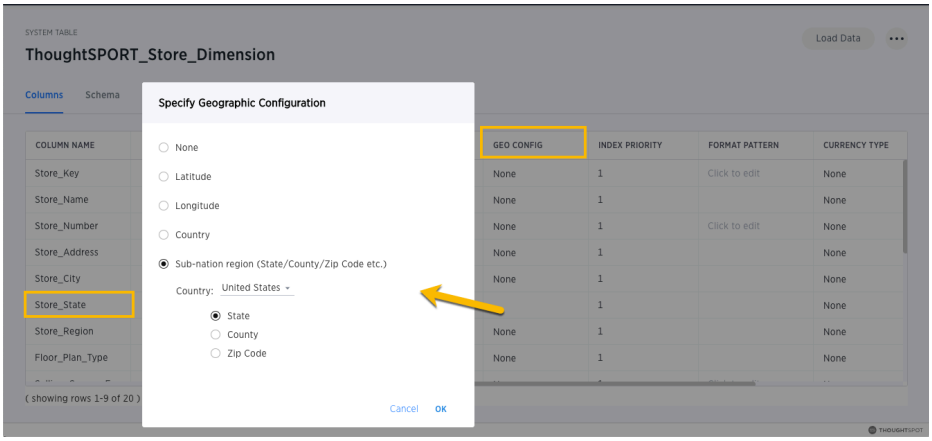
If you are using a column with the data type `DOUBLE` for latitude and longitude, you will also need to change the following settings for those columns:

- set **Column Type** to `ATTRIBUTE`
- set **Additive** to `NO`
- set **Aggregation Type** to `NONE`

For information these settings, see Set ADDITIVE or AGGREGATION.

How to edit geographic columns

1. Find the **GEO CONFIG** for the column that contains the geographical data.
2. Select the column to display the **Specify Geographic Configuration** dialog.



3. Change the value to the appropriate **GEO CONFIG**, depending on the kind of geo data the column contains.

   If your data includes latitude and/or longitude columns that are stored as a numeric data type (`DOUBLE`), make these changes for those columns:

   a. Change the **Type** to `ATTRIBUTE` .
   b. Change **ADDITIVE** to `NO` / `FALSE` .

4. Save your changes.

List of geotypes

GeoType	Description	Type: Example
`COUNTRY_REGION`	Countries	• name: United States • long name: United States • name_sort: United States of America • abbreviation: U.S.A. • adm0_a3: USA • adm0_a3_is: USA • adm0_a3_us: USA • admin: United States of America • brk_a3: USA • brk_name: United States • formal_en: United States of America • iso_a2: US • iso_a3: USA • iso_n3: 840
`COUNTY`	Counties in the United States	• santa clara county • pike county, ohio • pike county, OH
`STATE_PROVINCE`	States in the United States	• name: California • US Postal Service abbreviation: CA
`LATITUDE`	Must be used with `LONGITUDE`	• 37.421023 • 1.282911
`LONGITUDE`	Must be used with `LATITUDE`	• –122.142103 • 103.848865
`ZIP_CODE`	Zip codes and zip codes +4 in the United States	• po_name: MT MEADOWS AREA • ZIP: "00012" • zip2: 12
Other Sub-nation Regions	Administrative regions found in countries other than the United States	• bremen • normandy • west midlands

Related information

Model the data for searching

Set number, date, and currency formats

Summary: Explains how to set key formats for column values.

You can set number, date, and currency display formats. These formats define how these value types display in tables and charts.

Number formats

You can set a format for how numbers are displayed in tables and charts. For example, you can display numbers with a different number of digits after the decimal point, based on the data modeling setting **Format Pattern**. You can use any of the supported number formats for delimiters and number of digits to show using Java Decimal Notation. Currency symbols are not supported.

The system has default values which are:

`#,###`	For integer data types `INT` and `BIGINT`. As you can see, these can only contain numbers, alpha characters are not permitted.
`#,###.00`	for decimal data types `DOUBLE` and `FLOAT`.

These are some examples of formats you can use:

Stored Value	Format Pattern	Display Value
12345.6789	`#,##0.##`	12,345.68
12345.6789	`#,##0.###`	12,345.679
12345.6789	`#,##0.00000`	12,345.68
12345.6789	`#,##0`	12,345
12345.6789	`#,##0.00`	12,345.68
12345	`#,##0.##`	12,345
12345	`#,##0.00`	12,345.00

You can change the date format used to display a column's values for a single table or, by editing the data model, for the entire ThoughtSpot instance. Editing the data model file requires that you have administrative privileges.

1. Decide if the change is for a table or the entire instance.
2. Find the **Format Pattern** for the column.

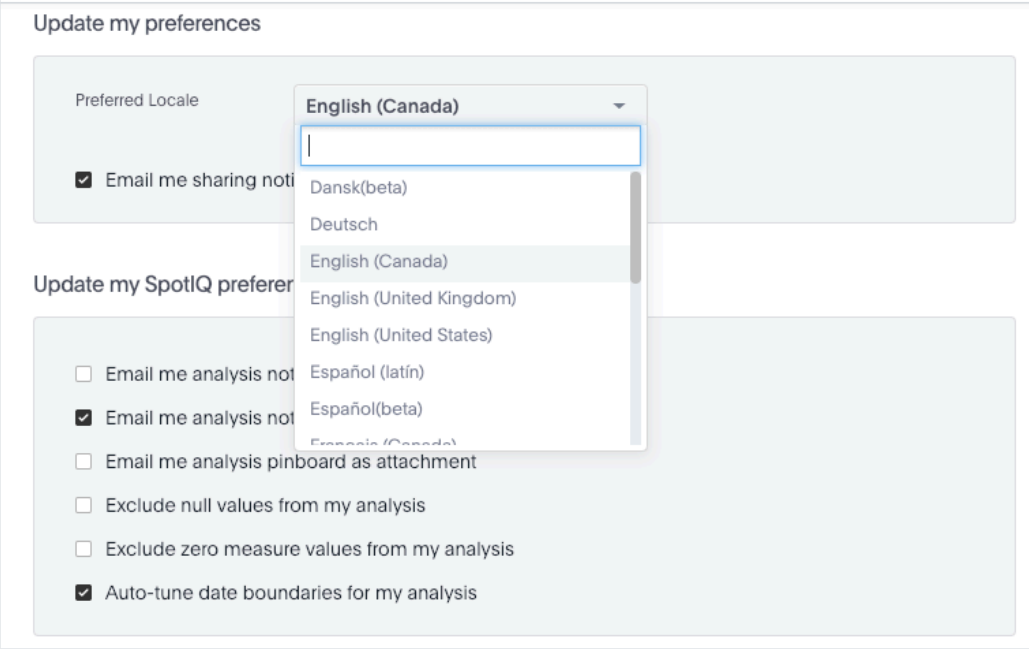
   This is either a column in a single table or a column in the data modeling file.

3. In the column, enter the format you want to use.
4. Save your changes.

   If you are using a data-modeling file you'll need to upload the new file to your installation.

Profile-based number formatting

Number formatting is set by default based on your ThoughtSpot profile's **Preferred locale** setting. You can set this value to accommodate your geographic locations.



For example, if you are using ThoughtSpot in the US, the number formatting should look like this: `xxx,xxx.xx`. And in Europe, it should look like this: `xxx.xxx,xx`.

Date formats

Format Pattern (UI)/ **Format Pattern** (model file) formats for how dates are displayed in tables and charts. For example, you can display dates in a standard European or US format based on the data modeling setting **Format Pattern**. These are some examples of formats you can use:

Format mask	Description
`YYYY` or `yyyy`	four digit year such as `2017`
`YY` or `yy`	last two digits of year such as `17`
`M`	month with no leading zero `1` - `12`
`MM`	Two digit month `01` - `12`
`MMM`	Three letter month such as `Jan`

Format mask	Description
`D`	Day of year without a leading zero `0` - `365`
`DD`	Day of year with up to one leading zero `01` - `365`
`DDD`	Day of year with up to two leading zeroes `001` - `365`
`d`	Day of month with no leading zero `1` - `31`
`dd`	Two digit day of month `01` - `31`
`HH`	Two digit `24` hour representation of hour `00` - `23`
`hh`	Two digit `12` hour representation of hour `01` - `12`
`H`	24 hour representation of hour with no leading zero `0` - `23`
`h`	12 hour representation of hour with no leading zero `1` - `12`
`mm`	Minutes `00` - `59`
`m`	Minutes with no leading zero `0` - `59`
`ss`	Seconds `00` - `59`
`s`	Seconds with no leading zero `0` - `59`
`a`	AM/PM indicator

Valid delimiters include most non-alphabet characters. This includes but is not limited to:

- `\` (forward slash)
- `/` (backward slash)
- `|` (pipe symbol)
- `:` (colon)
- `-` (dash)
- `_` (underscore)
- `=` (equal sign)

Examples of valid format masks you can produce for display are as follows:

- `MM/dd/yyyy`
- `MMM`
- `DD/MM/yyyy`
- `MM/dd/yyyy HH:mm`
- `DD/MM/yyyy HH:mm`

To change the date format used to display a column's values for a single table or, by editing the data model, for the entire ThoughtSpot instance.

1. Decide if the change is for a table or the entire instance.
2. Find the **Format Pattern** for the column.

   This is either a column in a single table or a column in the data modeling file.

3. In the column, enter the format you want to use.
4. Save your changes.

   If you are using a data-modeling file you'll need to upload the new file to your installation.

Set currency format

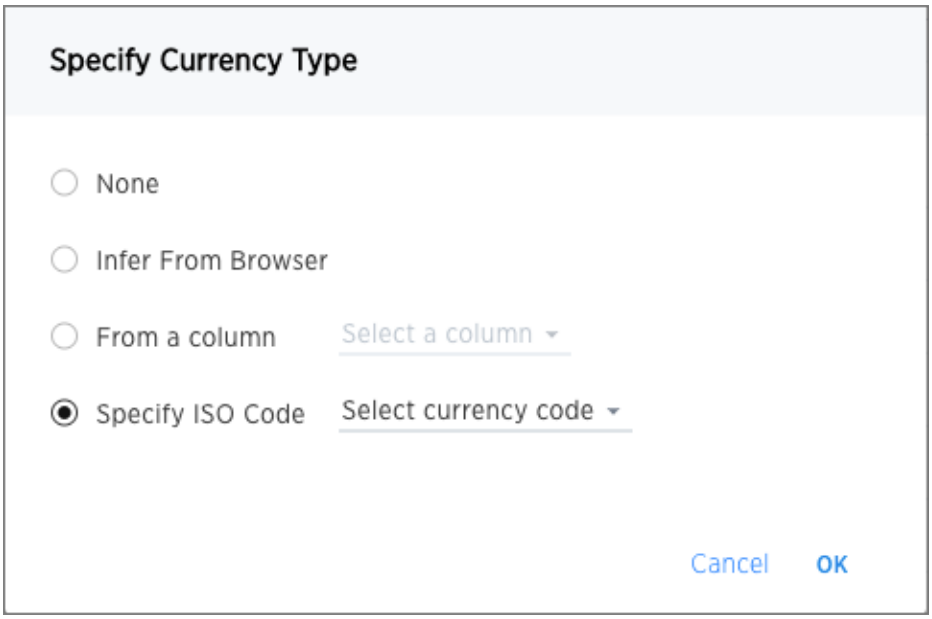
You can set a format for how currencies are displayed in tables and charts when using the ThoughtSpot Data API or embedding. For example, you can display currencies in a standard European Euro or US Dollar format based on the data modeling setting **Currency Type**.

You can change the currency format used to display a column's values for a single table. When you specify the currency type of your data in the **Columns** settings, your currency data will only display the correct format and currency code in the embedded use case. Currency specific symbols are available in the non-embedded use case as well, but they are not localized.

All users are treated as if they are in `en-US` locale unless they are in embed mode and their browser configuration tells ThoughtSpot that they are in some other locale. For example, `100 Polish Zloty` appears as `100zł` to a user in Poland, but without localization enabled, it appears as `PLN 100`.

This subtle difference can be seen when you use the REST API. See the ThoughtSpot Application Integration Guide for more information on the API.

1. Find the **Currency Type** for the column whose display format you want to change.
2. Click on it to open the **Specify Currency Type** menu.



3. Select one of the following ways you would like to change the format.

Option	Description
Infer From Browser	Your currency data will be modeled upon the locale of your browser setting.

Option	Description
From a column	Your currency data will be modeled upon the existing currency information in the selected column This option is disabled if there is no `VARCHAR` column to choose from.
Specify ISO Code	Your currency data will be modeled upon your selection from the available currency code choices.

4. Click **Ok** to save your changes.

Related information

Model the data for searching

Change the Attribution Dimension

> Summary: The **Attribution Dimension** setting applies only to tables that are related through a chasm trap. If your schema does not include these, you can ignore this setting.

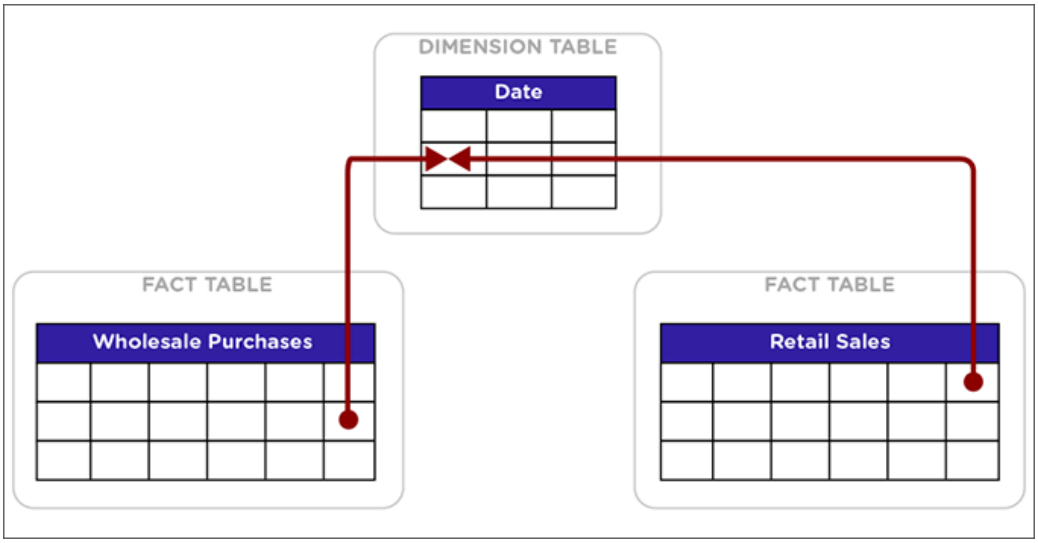
The **Attribution Dimension** setting only applies to tables that join over a Chasm Trap. By default, the attribution dimension setting will be set to `YES`, but you can override that by setting the column's attribution dimension property to `NO`, as described here.

Understand chasm traps and attribute dimension

In the classic chasm trap, two fact tables are related through a shared dimension table. When the two fact tables are joined, the shared column(s) in the dimension table are used to attribute rows in one fact table to match with rows in the other fact table.

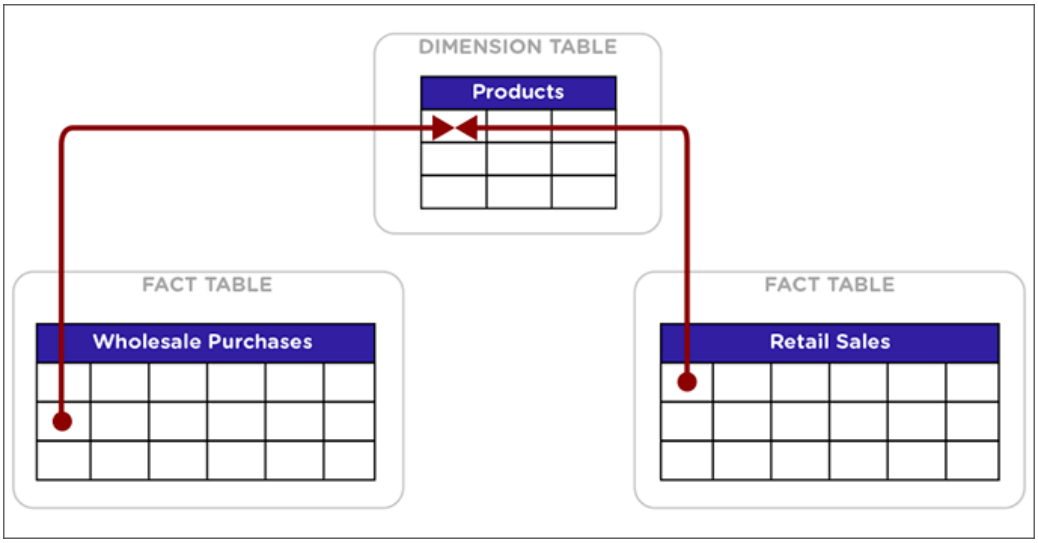
Usually, all goes well using this method. But sometimes an incorrect or illogical attribution can result. This can happen because the column chosen is not meaningful for performing this attribution. If you are seeing unexpected results in searches that include tables across a chasm trap, this setting is for you.

Below is an example of a column that is not an attribution dimension. Suppose you have two fact tables, Wholesale Purchases and Retail Sales, that share a common dimension Date.



In this example, the date column in the Date dimension should not be used for attribution, since unrelated rows in both of the fact tables could share the same row in the Date table. Why? Because if Sally bought oranges wholesale on April 25, 2005 and made a retail sale of apples on the same day, there is no logical relationship between those two events. Combining the two events using the date they share will not create any meaningful information.

If matching rows in two fact tables over a chasm trap depends on the values in a column contained in a dimension table, that column is known as an attribution dimension.



In this example, the Product ID column in the Products dimension table is an attribution dimension. For rows where the Product ID in the Wholesale Purchases and in the Retail Sales tables is a match, those rows are logically related in a meaningful way. They can be combined in charts and reports to produce a logical, expected outcome.

How to set attribute dimension

You cannot configure this setting in the model file. You can only configure it on a table-by-table basis. To designate a column as not being an attribution dimension (i.e. not producing any meaningful attribution across a chasm trap):

1. Find the column that is not an attribution dimension.
2. Select its **Attribution Dimension**.
3. Set the value to `NO`. If you're using the modeling file, set it to **FALSE**.
4. Save your changes.

Related information

Model the data for searching

Set entity categories for SearchIQ

Summary: You can specify a per column entity category to help SearchIQ.

SearchIQ is a search experience that allows you to ask more natural questions, similar to the way you might talk to a person.

If you have access to tables and worksheets for data modeling purposes, you can specify columns entity types for different columns in your data sources which support the SearchIQ user experience.

About Entity Categories

Category	Description
PERSON	Contains data that represents a person, relevant to questions about “who?”
PLACE	Contains data that represents a location, relevant to questions about “where?”
TIME	Contains data that represents a date or time, relevant to questions about “when?”
PRODUCT	Contains data that represents a product
ZIP_CODE	Contains zip code data, relevant to questions like “where?” or “what zip code?”
LAT_LONG	Contains data that represents geographical positioning, relevant to questions like “where?”
COMPANY_ORG	Contains data that represents a company or organization
NUM_TYPES	Contains numerical data

Set Entity Categories

To specify entity categories:

1. Click **Data** in the top menu, and choose **Tables** or **Worksheets**.
2. Click on the name of your table or worksheet.
3. On the **Columns** tab, find the COLUMN NAMES for which you want to specify entity categories, and scroll to the right to find `ENTITY CATEGORY`.
4. Use the drop-down menu to set the `ENTITY CATEGORY` to the type you want.
5. Click **SAVE CHANGES** in the upper right.

Related information

- SearchIQ
- Overview of data modeling settings

Link tables using relationships

You can link tables by creating relationships between their columns. Linked tables can be searched together or combined into a worksheet for easy searching. Tables that have no relationship between their columns can not be combined in a single search.

There are two ways to create relationships between tables:

1. Create a constraint using TQL.
2. Create a relationship through the web interface.

The two methods create the same kind of relationship both from an end user perspective and an administrative perspective. When creating a relationship between two tables, the columns that form the link must be the exact same data type. For example, a column of type `INT32` to another `INT32` column.

Both types of relationships exist within the database. You can also generate a script through TQL that contains all relationships, whether create via the web interface or in TQL.

Relationships created through either method can be managed either via TQL or by going to the **Relationships** page when viewing data in the **Date Modeling** section in the ThoughtSpot application. You can view, modify, or delete relationships in either place.

You may create relationships using a mixture of TQL and the web interface, but the relationships you create cannot form a circular relationship, or “cycle”. If you attempt to create a relationship that would complete a cycle, you will see a message saying that the relationship could not be added because it conflicts with another existing relationship.

Join a table or view to another data source

> **Summary:** Learn how to define joins between a table or view and another table, view, or worksheet

Joining a table or view to another table, view, or worksheet creates a relationship that allows them to be searched together. Choose a column to join on that both tables contain (e.g. employee ID or product key). This process creates a generic join between the table or view and the other table, view, or worksheet on the column you specify.

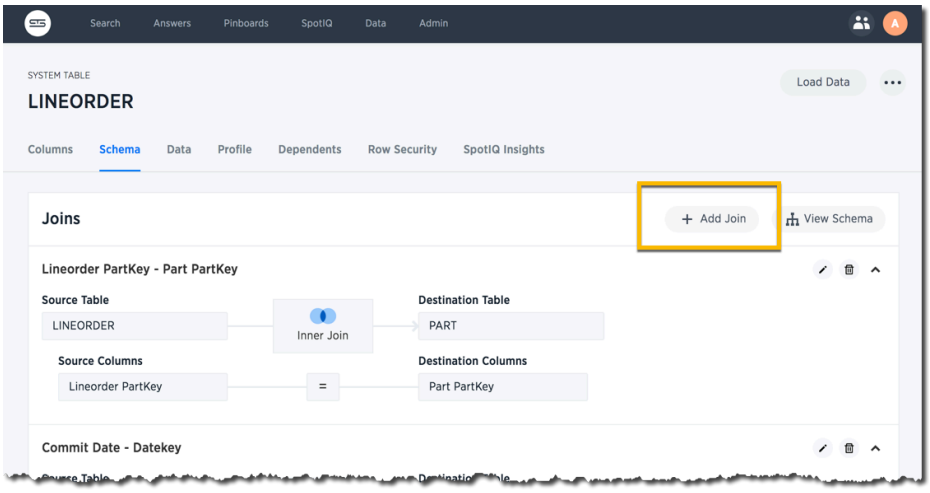
If you want to create a primary key/foreign key relationship, you need to use TQL rather than the web interface.

You must have either the Can administrator ThoughtSpot or the Can manage data privilege to create a join relationship. If you're not an administrator, you also need edit permissions on the table, view, or worksheet.

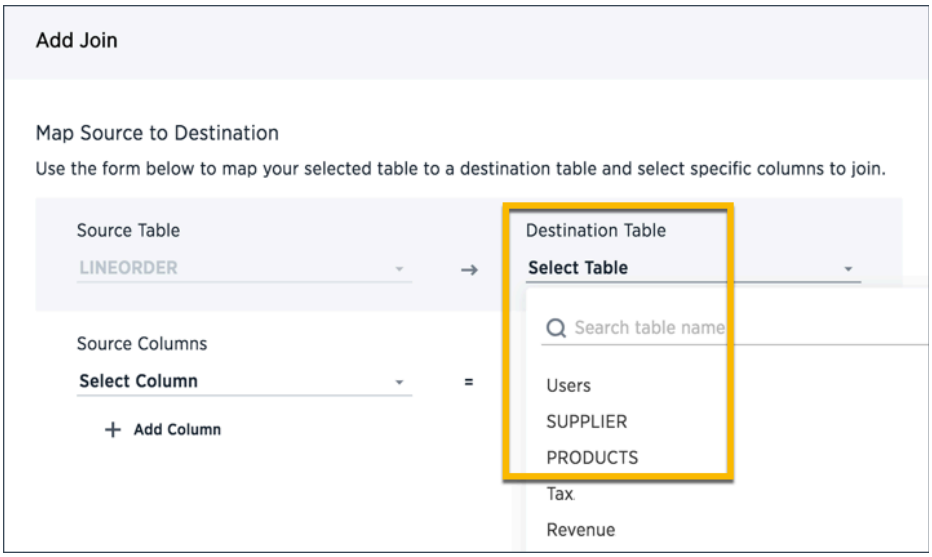
When creating a join between the columns in two data sources, the columns being linked must have the same data type, with the same meaning. That is, they must represent the same data. Normally, you'll make this kind of link from a fact table column to a column in a dimension table that uniquely identifies a logical entity in your data such as Employee ID for a person, Product ID for a product, or Date Key for a specific date in a date lookup table.

To create a relationship through the Web interface:

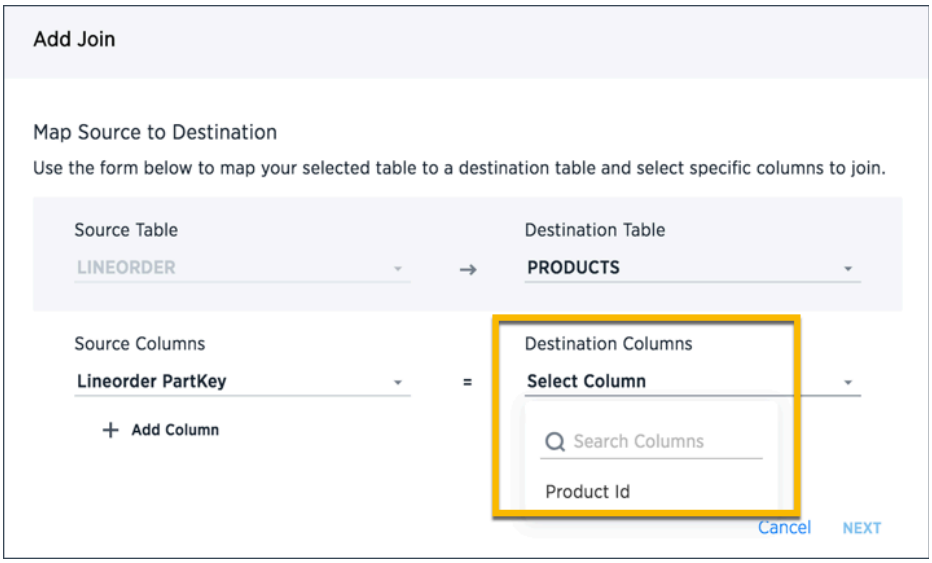
1. To find your table or view, click **Data** in the top menu, and choose **Tables** or **Views**.
2. Click on the name of your table or view.
3. Click **Schema**. You will see the list showing existing joins.
4. Click the + **Add Join** button on the upper right side of the screen.



5. Use the **Map source to destination** dialog to choose the destination table, view, or worksheet you want to join to.



6. Choose the columns you want to join on from the table or view (source) and destination table, view, or worksheet. Click **Next**.



7. Give your join a name and description and click **ADD JOIN**.
8. Repeat these steps until all the joins you want to make have been created.

After creating the join, you may change its name and description by clicking the edit icon. If you want to change the data source or column being joined, you'll need to delete the join and create a new one.

Related Information

- Constraints

Delete a relationship

Summary: You can delete relationship (link) between tables through the application or TQL.

You must have either the Can administrator ThoughtSpot or the Can manage data privilege to delete a relationship. If you're not an administrator, you also need edit permissions on the table, view, or worksheet.

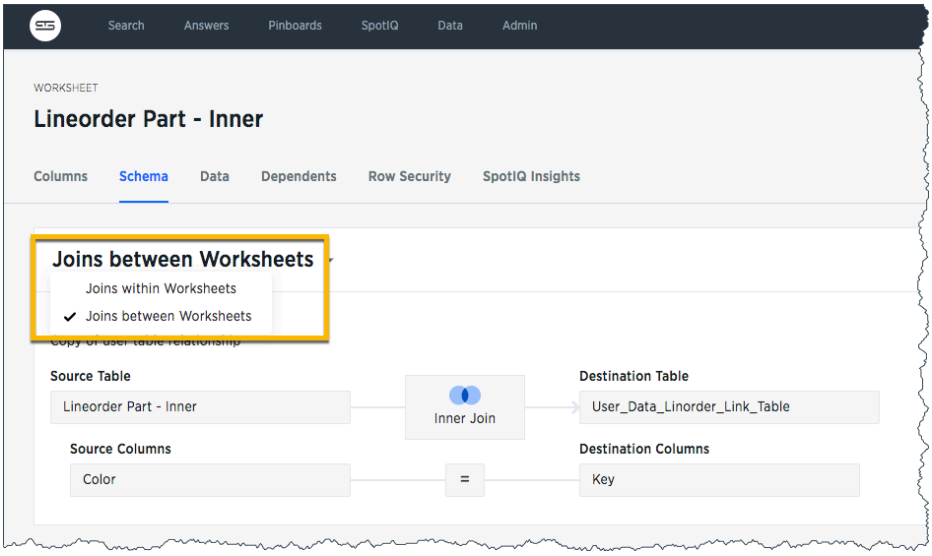
If you created a relationship (join or link) between tables using the Web interface, you can also delete it from the Web interface. But if the relationship was created using TQL, you must also use TQL to delete it.

To delete a relationship using TQL, use an `ALTER TABLE...DROP CONSTRAINT...` or `ALTER TABLE...DROP RELATIONSHIP...` statement.

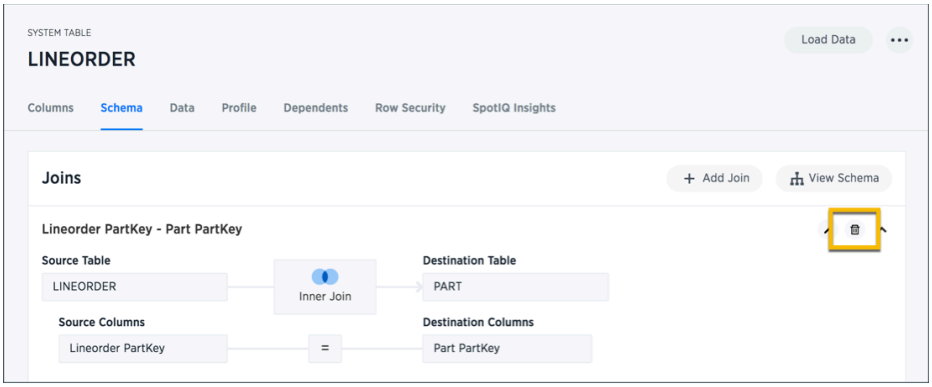
To delete a relationship from the Web interface:

1. Click **Data** on the top navigation bar.
2. Click on the name of the data source you from which you want to remove the relationship.
3. Click **Schema**. You will see the list showing existing joins.

   If this is a worksheet, you will need to click on **Joins within worksheets** and choose **Joins between worksheets**.



4. Find the relationship you want to delete, and click the **Delete icon**.



5. Repeat these steps until all the joins you want to remove have been deleted.

Related Information

- Constraints

About stickers

Summary: Stickers enable you to create categories for classification of objects, including pinboards, answers, data sources, and worksheets.

You can create stickers to make it easier for people to find data sources and pinboards. Stickers are global in scope. This means that everyone can see the stickers and use them to tag objects. They can also filter lists of objects by sticker. Stickers are often used to designate subject areas, such as sales, HR, and finance, but you can use them any way you like.

Keep in mind these permissions when working with stickers:

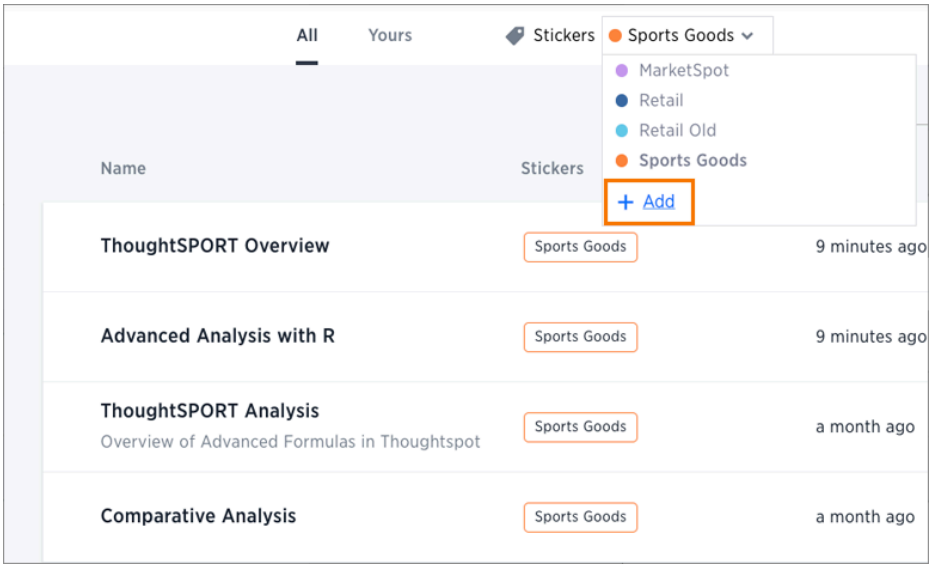
- Only administrators can create stickers.
- Anyone can apply a sticker.
- Anyone can filter by a sticker.

Create a sticker

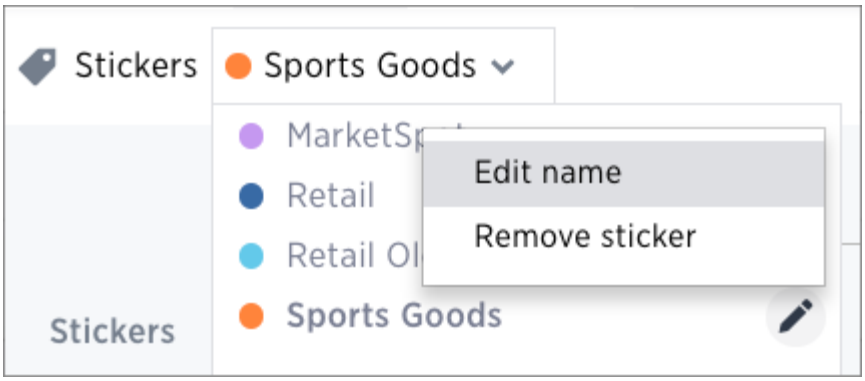
Only administrator users can create stickers. Anyone can apply the stickers you create, or use them as filters when selecting from a list of sources or pinboards.

To create a sticker:

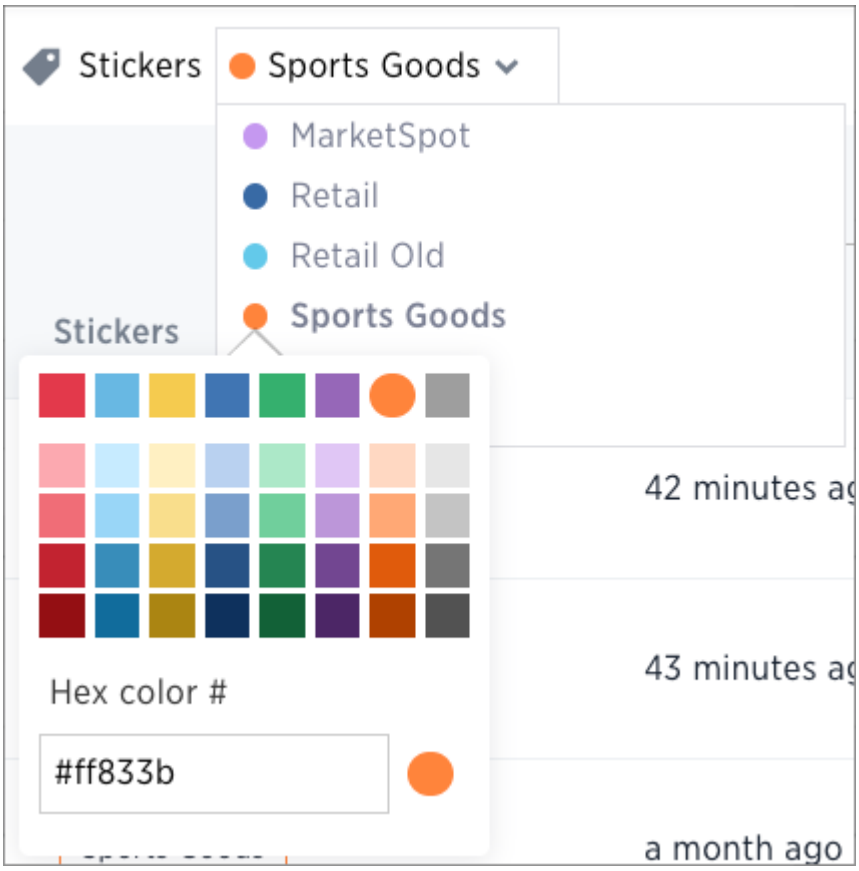
1. Navigate to the **Manage Data** or **Pinboards** screen using the icons in the top navigation bar.
2. Choose the currently selected sticker, scroll to the bottom of the list, and click + **Add**.



3. Type the name for the new sticker.
4. You can change the name of a sticker by clicking the edit icon next to its name.



5. You can change the color of a sticker by clicking the color circle next to its name.

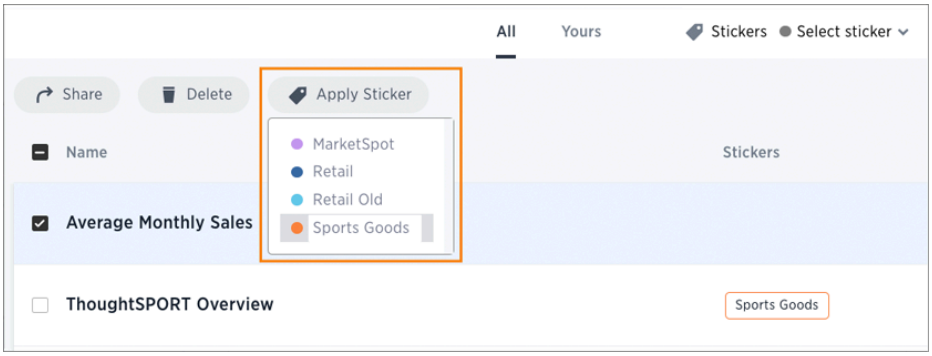


Apply a sticker

Only administrators create stickers, but anyone with edit privileges can tag an object with a sticker.

To tag an object with a sticker:

1. From the top menu, choose Answers, Pinboards, or Data.
2. Find the item(s) you want to tag in the list, and check the box next to its name.
3. Click the apply sticker icon and choose one from the list. You can apply as many stickers as you like to an object.

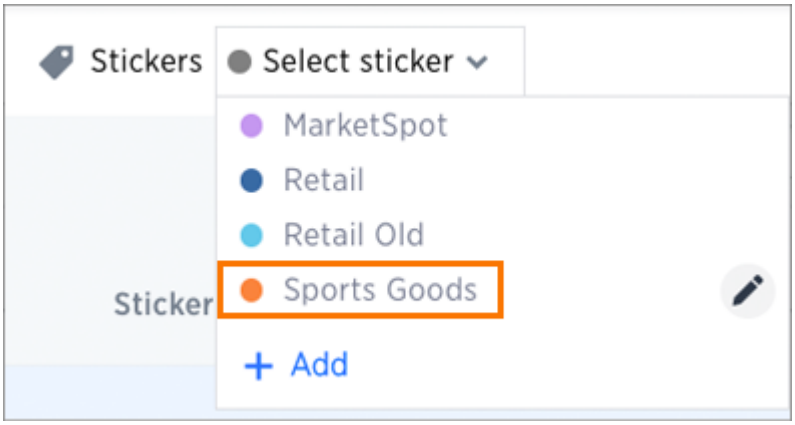


Filter by stickers

Whenever you are selecting objects from a list, you can filter by sticker to find what you're looking for. Anyone can use stickers to filter lists of pinboards or data sources. You can also filter by sticker when selecting data sources.

To filter by sticker:

1. From the top menu, choose **Answers**, **Pinboards**, or **Data**.
2. Click **Select sticker**, and select the name of the sticker you want to filter by.



Simplify search with worksheets

Summary: Worksheets are flat tables created by joining columns from a set of one or more tables or imported datasets.

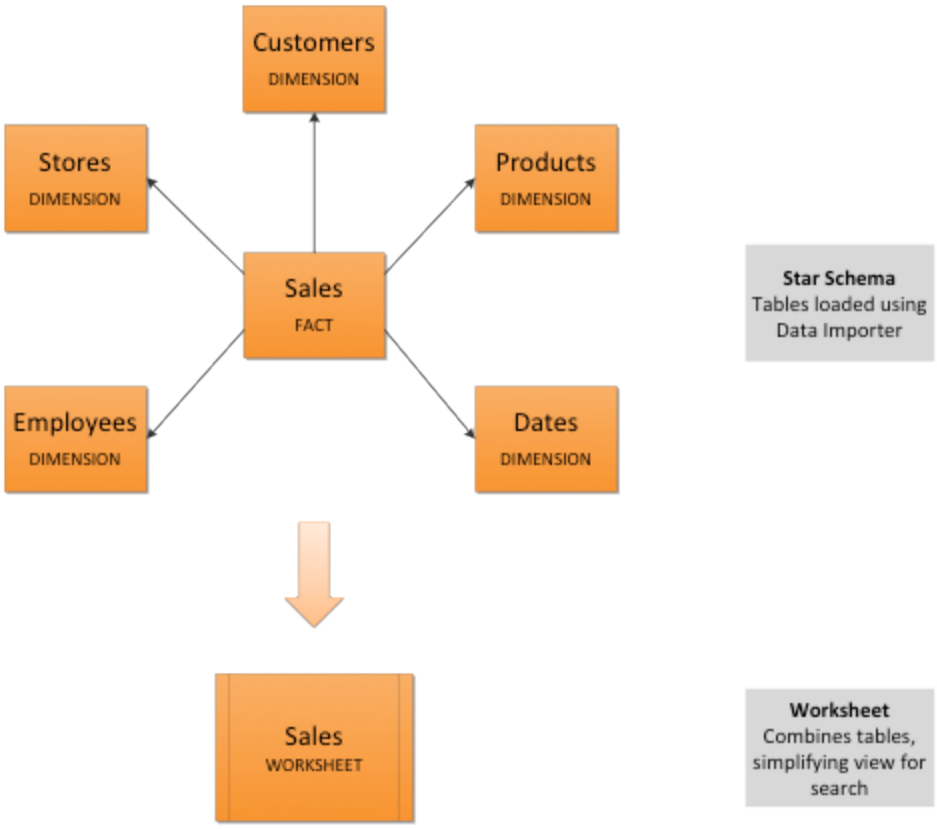
After modeling the data, create worksheets to make searching easier. For example, a sales executive might need to search for information about retail sales. The required data could be contained in several tables (sales, customers, products, stores, etc.), with foreign key relationships between them. An administrator who is familiar with the data model can create a retail sales worksheet, that combines all of the related fact and dimension tables into a single, easy-to-use view, and share it with the sales executive. This provides access to the data without requiring an understanding of how it is structured.

Guidelines for worksheets

Users are often unfamiliar with tables and how they are related to one another. A worksheet groups multiple related tables together in a logical way. You might use a worksheet for these reasons:

- To pre-join multiple tables together.
- To give a user or group access to only part of the underlying data.
- To include a derived column using a formula.
- To rename columns to make the data easier to search.
- To build in a specific filter or aggregation.
- To give users a filtered set of data to search.

Typically, you create one worksheet for each set of fact and dimension tables. For example, you may have a sales fact table and an inventory fact table. Each of these fact tables shares common dimensions like date, region, and store. In this scenario, you would create two worksheets: sales and inventory. The following diagram depicts the workflow for creating the sales worksheet.



The process for creating a worksheet is:

1. Decide which tables to use for the worksheet.
2. Create a new worksheet.
3. Add sources (tables) to the worksheet.
4. Choose the worksheet join rule.
5. Select the columns to include.
6. Optionally modify the join types within the worksheet.
7. Optionally create formulas.
8. Optionally create worksheet filters.
9. Save the worksheet.
10. Share the worksheet with groups or users.

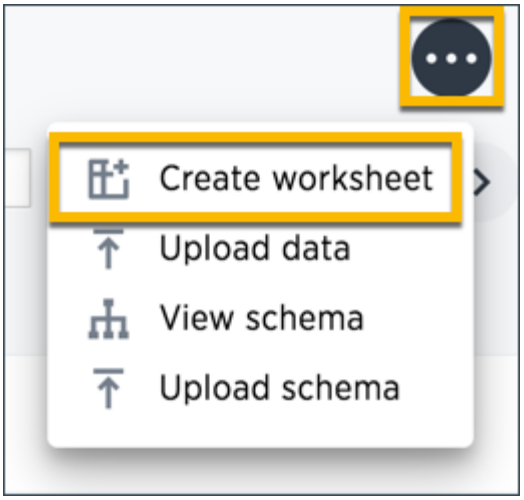
Create a worksheet

Create a worksheet to make the data easy for users to search. This process includes adding a new worksheet, after which you will choose the data sources to include in it.

To create a new worksheet:

1. Click on **Data**, on the top navigation bar.

2. Click the the ellipses icon (3 dots) , and select **Create worksheet**.

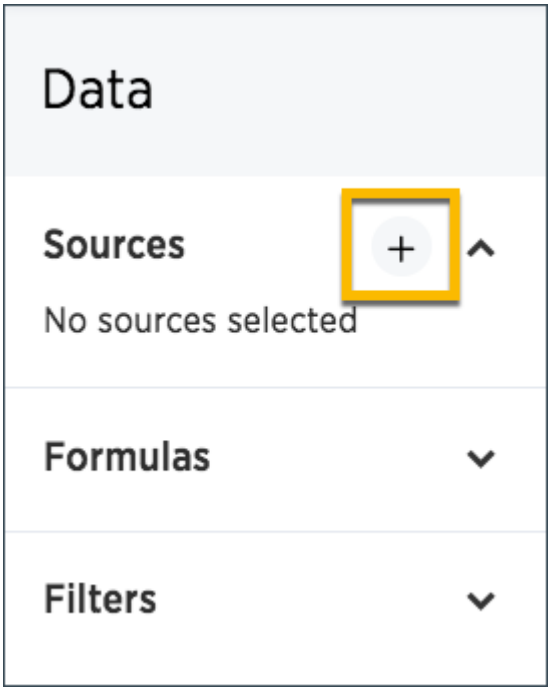


Add sources and columns to a worksheet

After creating a worksheet, you need to add the sources that contain the data. Sources is another name for tables. The sources you choose are typically related to one another by foreign keys.

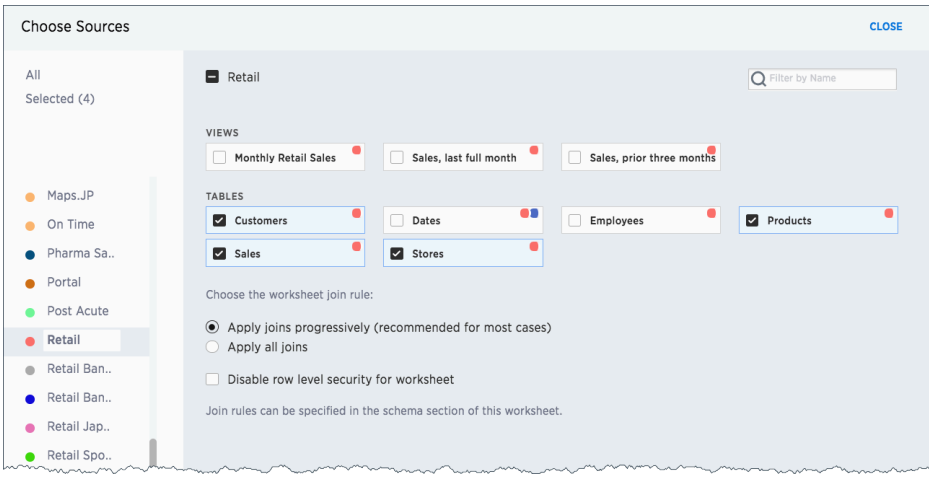
To add the sources to the worksheet:

1. Click on the **+ icon**.



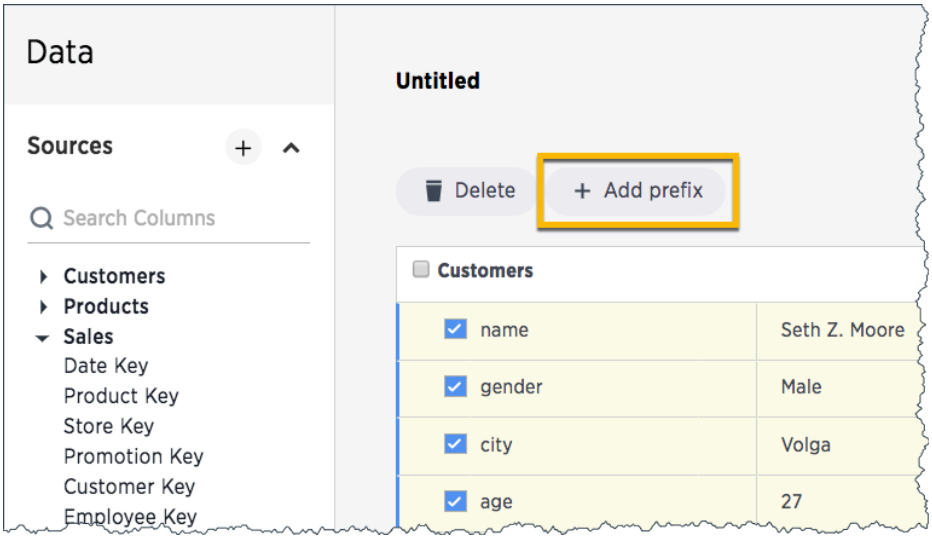
2. Check the box next to each of the sources you want to include in the worksheet.

Note that the list of sources only shows the data sources on which you have view or edit privileges.

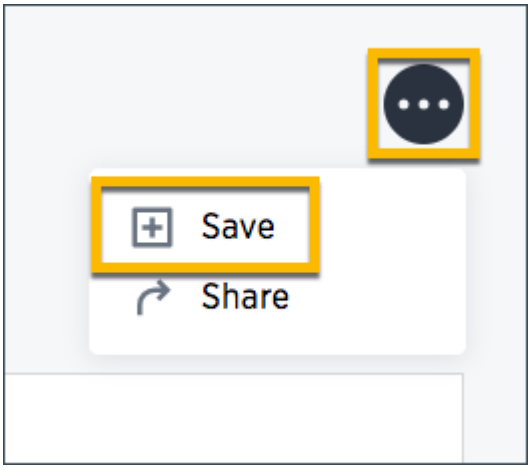


3. If you want to see what the data inside the sources looks like, click **Explore all data**.
4. Choose the worksheet join rule.
5. If you want to disable Row Level Security, for this worksheet, check the checkbox to disable it.
6. Click **CLOSE** to save your changes.
7. Expand the table names under **Columns** to select the columns to add to the worksheet.
    a. To add all of the columns from a table, click on the table name and click **+ Add Columns**.
    b. To add a single column, double click on its name.
    c. To add multiple columns, Ctl+click on each column you want to add and click **+ Add Columns**.

    Note that once you add a column, non-related tables (i.e. those without a primary/foreign key relationship) become hidden. If you are working with two tables that should be related, but are not, you can add a relationship between them.
8. Optionally modify the join types within the worksheet.
9. Optionally create formulas.
10. Optionally create worksheet filters.8. Click on the worksheet title to name it, and then **Save** it.
11. Click on each column name to give it a more user-friendly name for searching. You can tab through the list of columns to rename them quickly.
12. If you want to add a prefix to the name of several columns, select them, click the **Add Prefix** button, and type in the prefix.



13. Click the the ellipses icon (3 dots) , and select **Save**.



14. Share your worksheet, if you want other people to be able to use it.

Where to go next

- **How the worksheet join rule works**
  Use the worksheet join rule to specify when to apply joins when a search is done on a worksheet. You can either apply joins progressively, as each search term is added (recommended), or apply all joins to every search.

Edit or rename worksheet

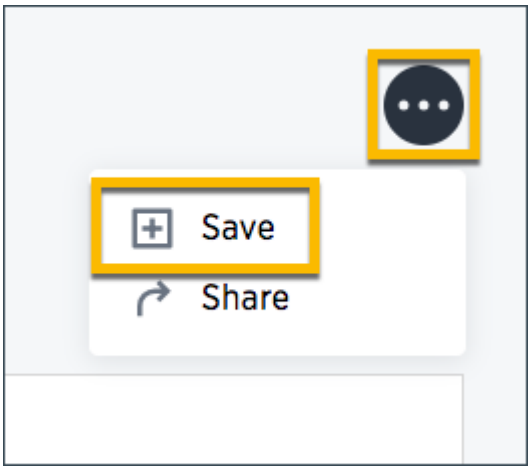
Summary: A worksheet can be edited by anyone with the proper permissions.

If you created a worksheet, or you have edit permissions on it, you can make changes such as adding sources and columns, adding or editing formulas, changing relationships, and changing column names. You can also rename a worksheet or change its description.

Edit a worksheet

To edit a worksheet:

1. Click on **Data** on the top navigation bar.
2. Click on the name of the worksheet you want to edit.
3. Click the **Edit** button in the upper right hand side of the screen.
4. Make your changes to the worksheet.
5. Click the the ellipses icon (3 dots) , and select **Save**.



Rename a worksheet or table

You can change a worksheet or table name from the ThoughtSpot application.

To change the name of a worksheet or table:

1. Click on **Data**, on the top navigation bar.
2. Find the worksheet or table you want to rename and click on its name.
3. Click the current name, and enter a new name.

Related information

- Change the join rule for a worksheet
- Add joins between a worksheet and other data
- Modify table joins within a worksheet

Create a formula in a worksheet

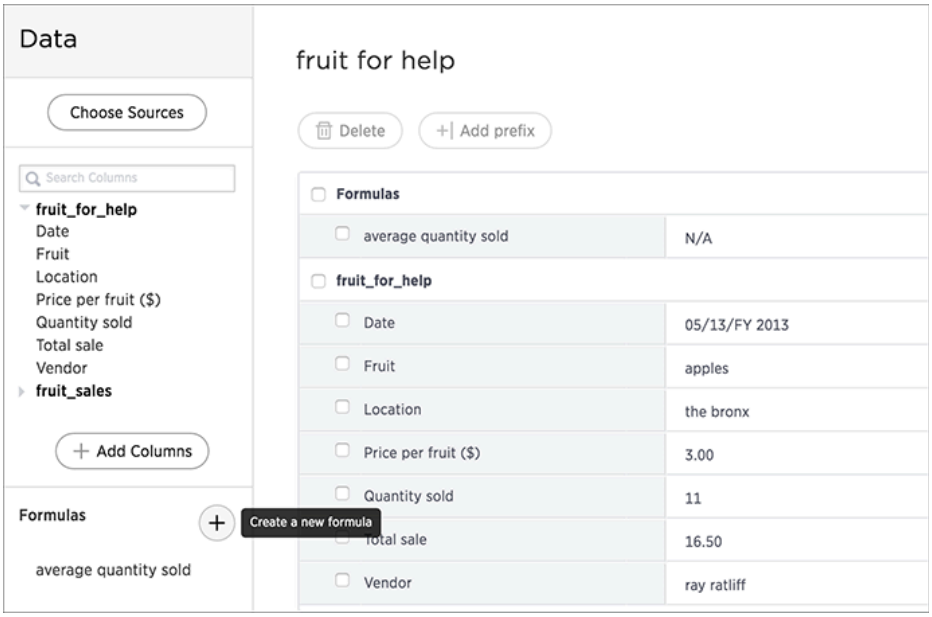
Summary: You can define formulas and use them to create derived columns in worksheets.

You create formulas by combining standard functions and operators, column names, and constant values.

Anyone who can create a worksheet can add a formula to it. Formulas are not reusable; the formula you create is associated only with the worksheet it belongs to. A complete list of available formulas and examples of each is available in the Formula function reference.

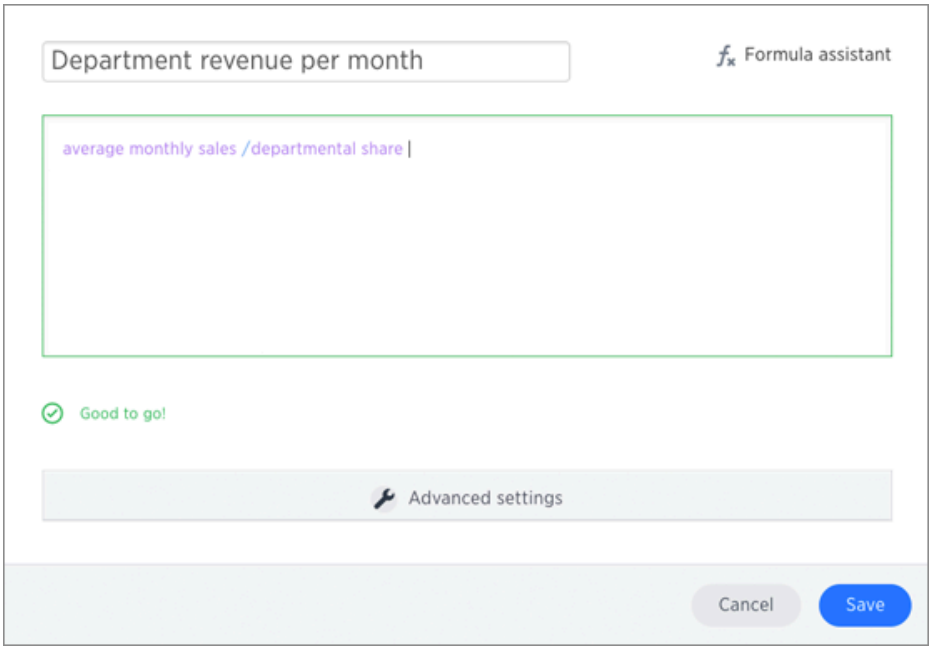
You can create a formula in a worksheet by using the Formula Builder. When you do this, the result of the formula gets added to the worksheet as a column. Use these steps to create a formula:

1. Create a new worksheet, or edit an existing one.
2. Click the + button next to **Formulas**.



d

3. Type your formula in the Formula Builder.

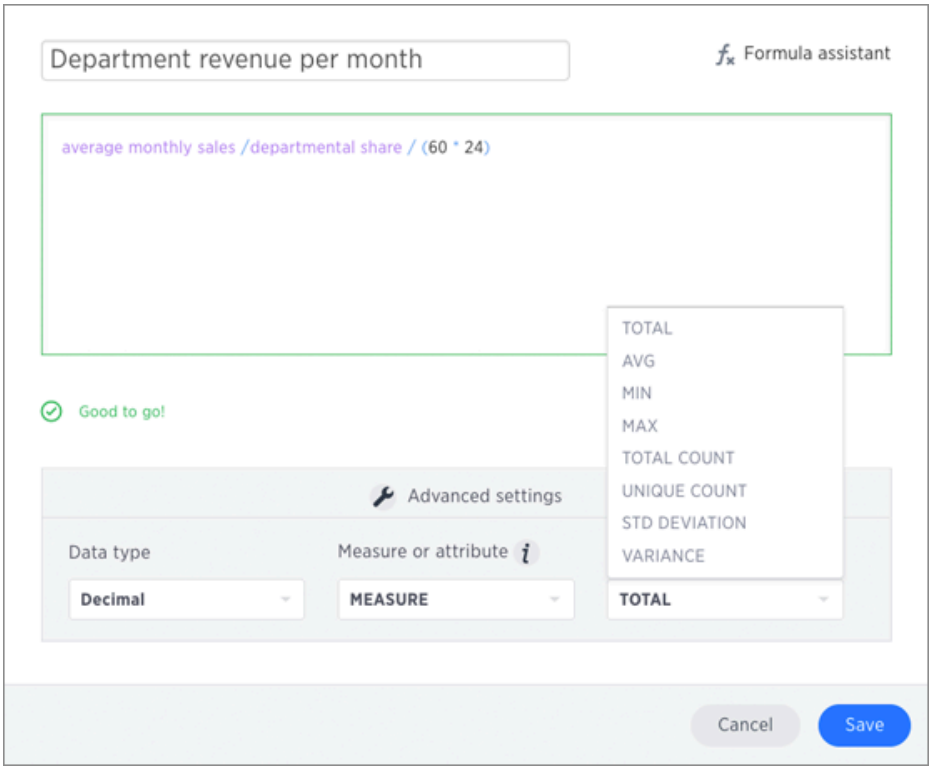


Note: Formulas elements are color coded by type and can include the formula operators and functions (blue), the names of columns (purple), and/or constants (black).

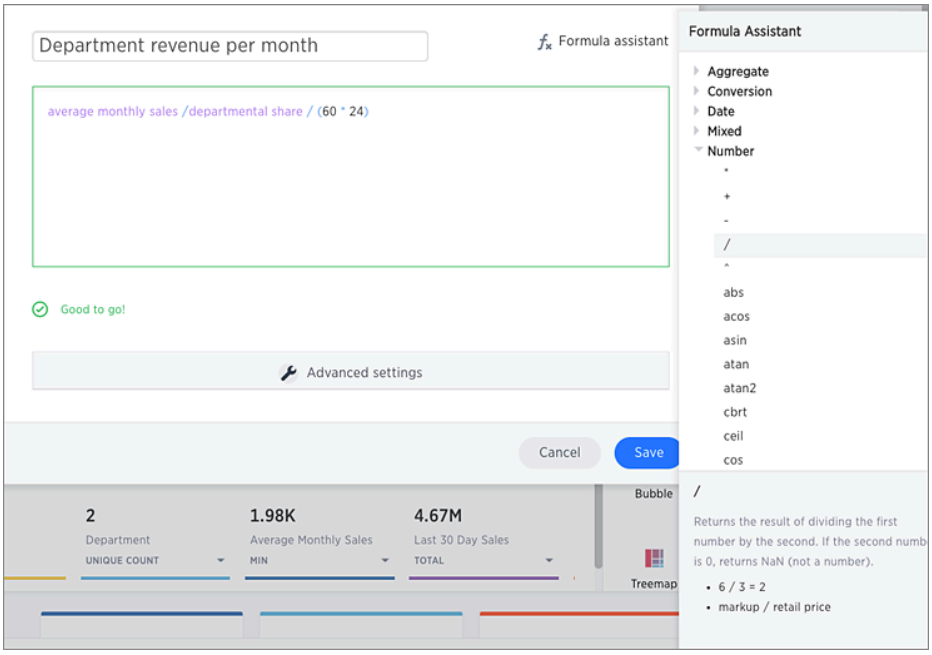
4. If you want to change what your formula returns, use the **Advanced settings**.

   Depending on your formula, you may be able to change:

   - Data type
   - ATTRIBUTE or MEASURE
   - Aggregation type



5. You can see a list of formula operators with examples by clicking on **Formula Assistant.**



6. Name the formula by clicking on its title and typing the new name. Click **Save**.

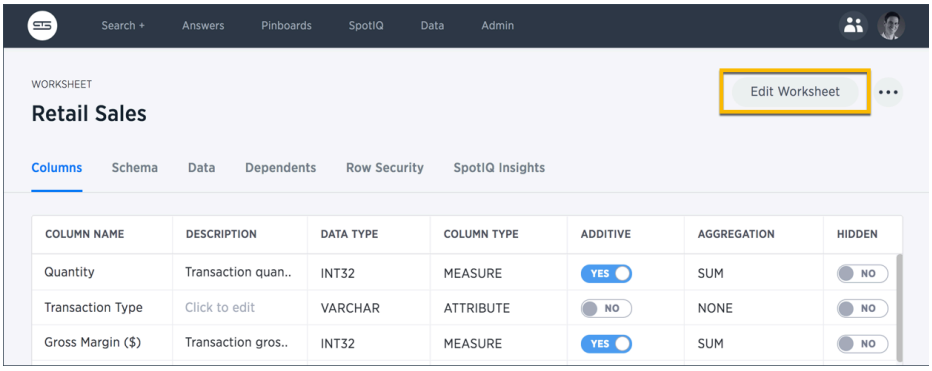
Add a filter to a worksheet

Summary: You can add a filters to a worksheet to limit the data users can access from the worksheet.

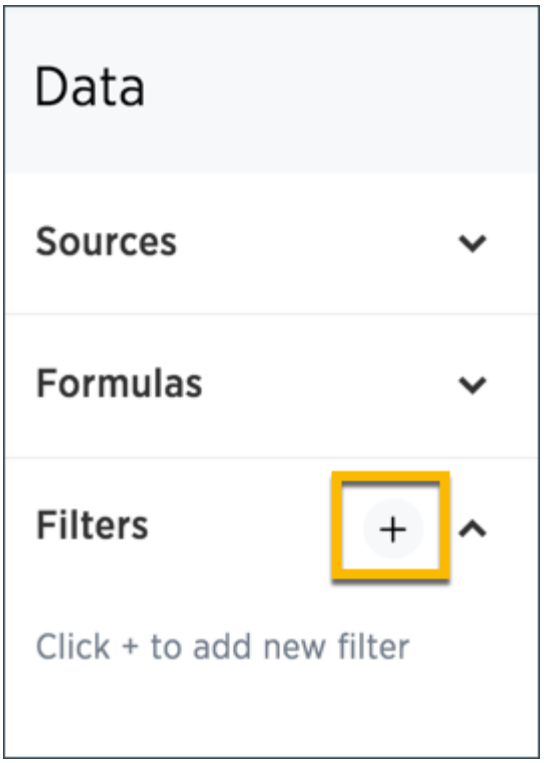
Beginning in ThoughtSpot version 5.0, you can add filters to a worksheet to limit the data it contains. This is useful when you have underlying tables that store more data than is necessary for the types of analyses the worksheet is intended for. You can also use worksheet filters to provide data security, when you want different groups of users to be able to see different data without relying on row level security.

To add a filter to a worksheet:

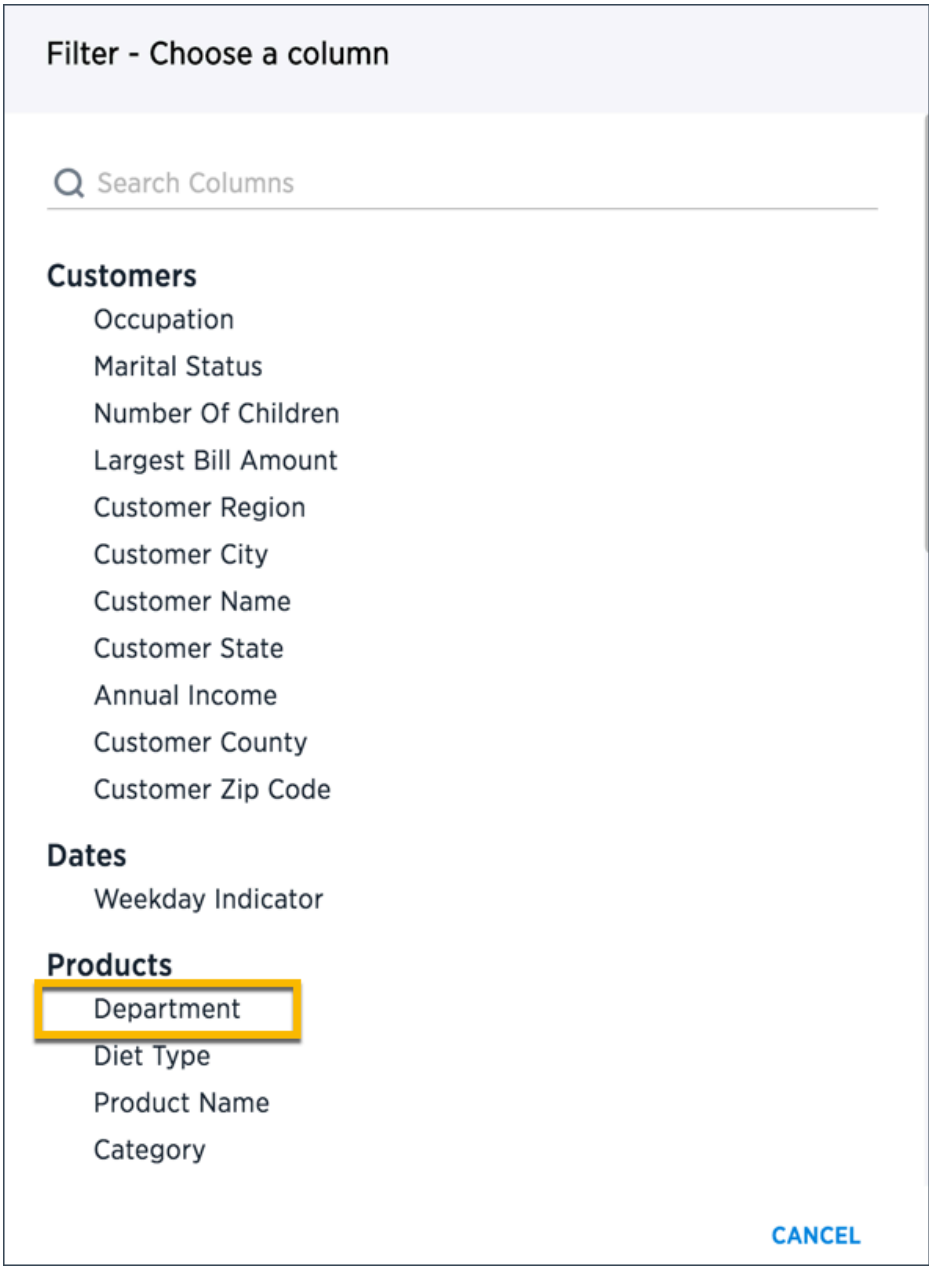
1. Click **Data** in the top menu bar, find your worksheet, and click on its name.
2. Click the **Edit Worksheet** button.



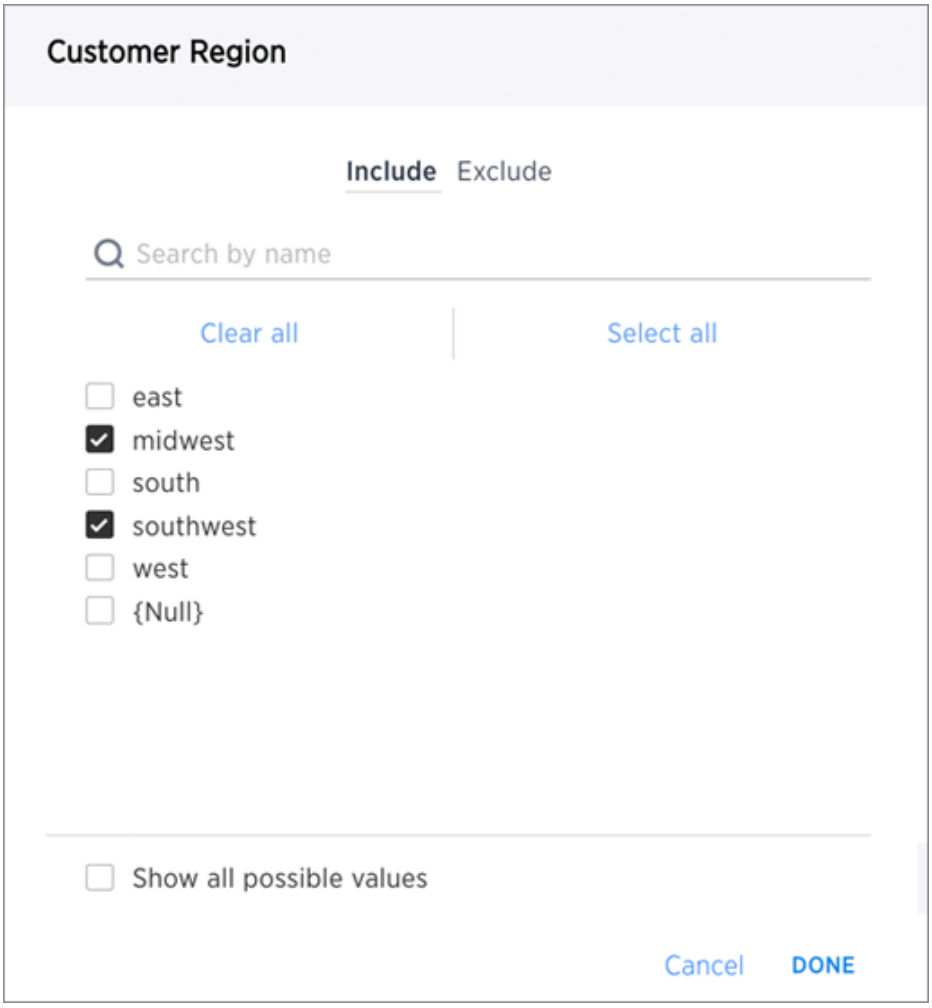
3. Click **Filters** on the left menu and click +.



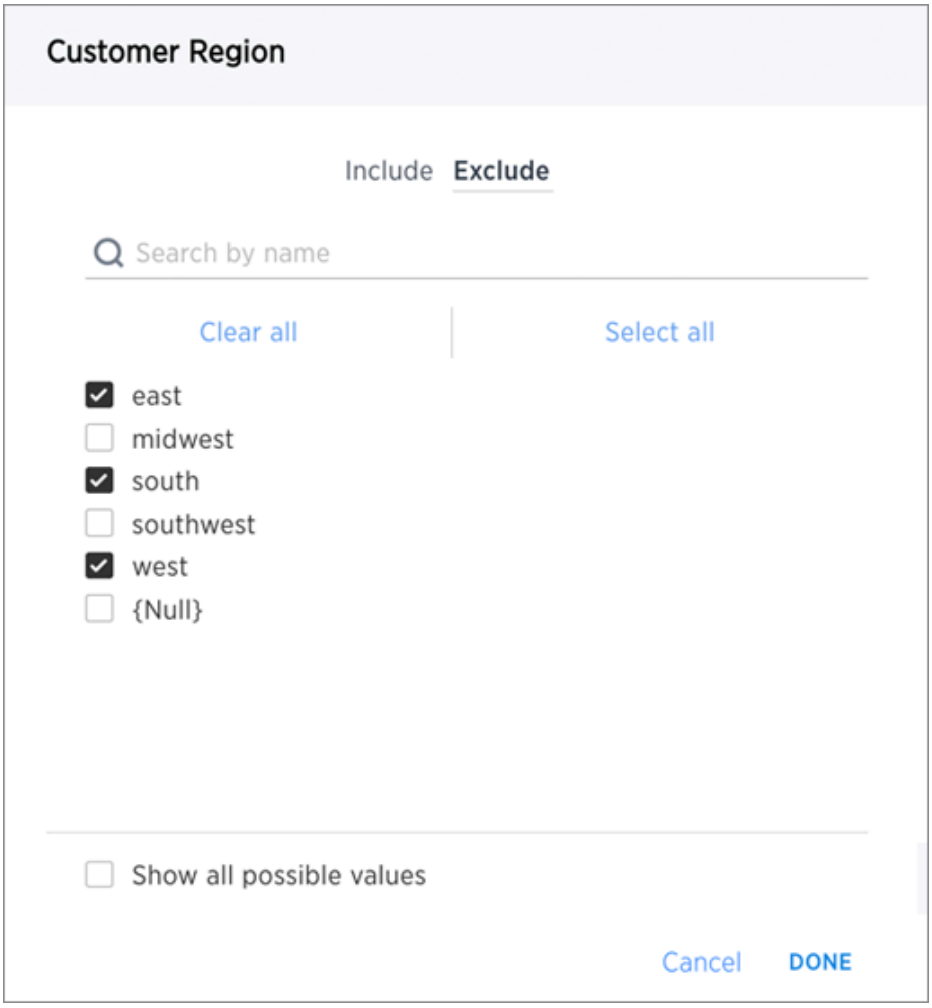
4. Choose the column you want to filter on.



5. Select the values to include in your answer.



6. If you want to exclude values, click **Exclude** and choose values to exclude.



7. Click **ADD FILTER**.

   If there are too many values, you can use the filter search bar to find the ones you want.

How the worksheet join rule works

Use the worksheet join rule to specify when to apply joins when a search is done on a worksheet. You can either apply joins progressively, as each search term is added (recommended), or apply all joins to every search.

Understand progressive joins

Often, a worksheet includes several dimension tables and a fact table. With progressive joins, if your search only includes terms from the fact table, you'll see all of the rows that satisfy your search. But as you add terms from dimension tables, the total number of rows shown may be reduced, as the joins to each dimension table are applied. It works like this:

- If you choose **Apply joins progressively (recommended for most cases)**, joins are only applied for tables whose columns are included in the search.
- If you choose **Apply all joins**, all possible joins are applied, regardless of which tables are included in the search.

When using **Apply joins progressively**, the number of rows in a search using the worksheet depends on which tables are part of the search. The worksheet acts like a materialized view. This means that it contains the results of a defined query in the form of a table. So if a particular dimension table is left out of the search, its joins are not applied.

Rule-Based Row Level Security (RLS) with worksheets

With Rule-Based RLS, you need to protect every table that contains any sensitive data. To do this, you'll grant access by creating explicit row level security rules on each of the underlying tables which contain data that row level security should apply to.

When creating the row level security rules for a table that's part of a worksheet, you aren't limited to referencing only the columns in that table. You can specify columns from other tables in the worksheet as well, as long as the tables are joined to the table you're creating the rule on. Then, when creating a worksheet on top of them, the behavior is consistent regardless of the worksheet join rule you choose. Users will never be able to see data they should not, regardless of what their search contains.

Imagine you have a worksheet that contains a `Sales` fact table, and `Customer` and `Product` dimensions that are joined on `Customer SSN` and `Product Code` columns. In order to secure the `Sales` table, you can use `Customer Name` from the `Customer` column to create a row level security rule.

How joins are applied with chasm traps

When working with worksheets and row level security, you need to understand how joins are applied. This is particularly important with chasm trap schemas. For chasm trap schemas, if row level security is only set on one of the tables, people could see data they should not see if the scope of their search does not include that table. (this protects them from having people see the wrong things if they have chasm trap).

For chasm trap *worksheets*, progressive and non-progressive joins do not apply. There is an entirely different methodology for how worksheet joins on a chasm trap schema work with row level security. So you can safely ignore that setting.

Change the join rule or RLS setting for a worksheet

Summary: As long as you have permissions to edit a worksheet, you can always go into it and set a different join or RLS rule.

If you find that the charts and tables built on a worksheet contain a large number of null values (which display as `{blank}` in the web browser), you can fix this by changing the internal joins for the worksheet.

If you have the **Can administer ThoughtSpot** privilege or the **Can manage data** plus edit privilege on a worksheet, you can edit the worksheet and change its RLS or other key settings.

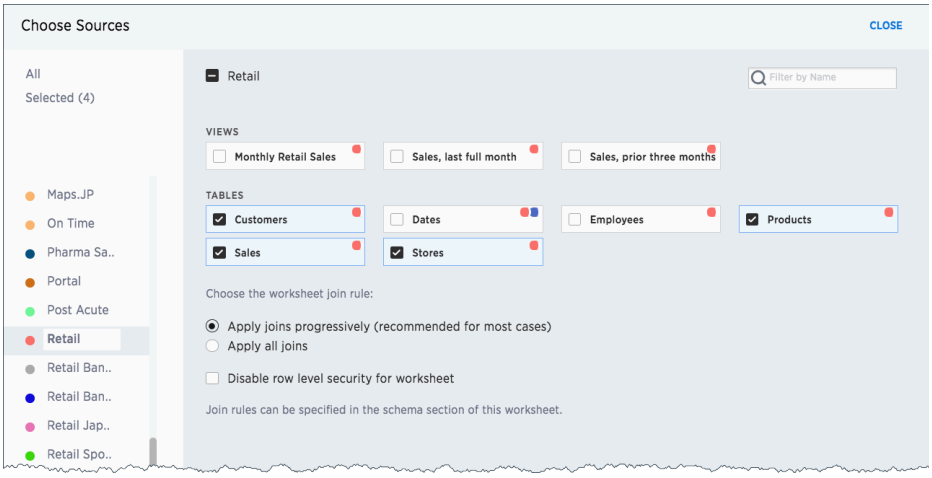
Change/configure a worksheet

Before working through this procedure, make sure you are familiar with how the following affect data:

- internal worksheet joins
- worksheet join rule
- row level security (RLS)

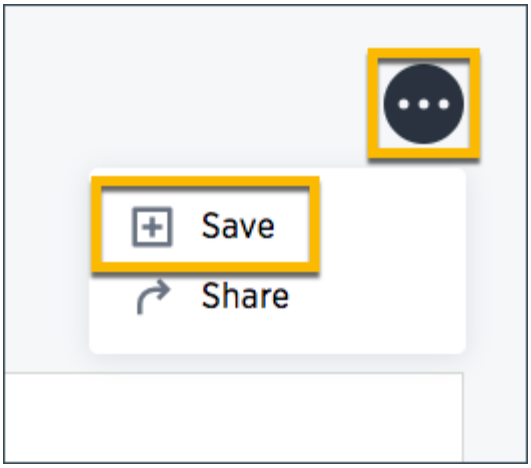
To configure these values for a worksheet:

1. Click on **Data** on the top navigation bar and then on **Worksheets**.
2. Click on the name of the worksheet you want to edit from the list.
3. Click the **Edit Worksheet** button in the upper right hand side of the screen.
4. Click on the **+ icon** next to **Sources**.
5. Scroll to the bottom of the page.
6. Configure the worksheet join rule and RLS setting as needed.



7. Click **CLOSE**.

8. Click the the ellipses icon (3 dots) , and select **Save**.



Join a worksheet to another data source

Summary: Learn how to define joins between a worksheet and a table, view, or other worksheet

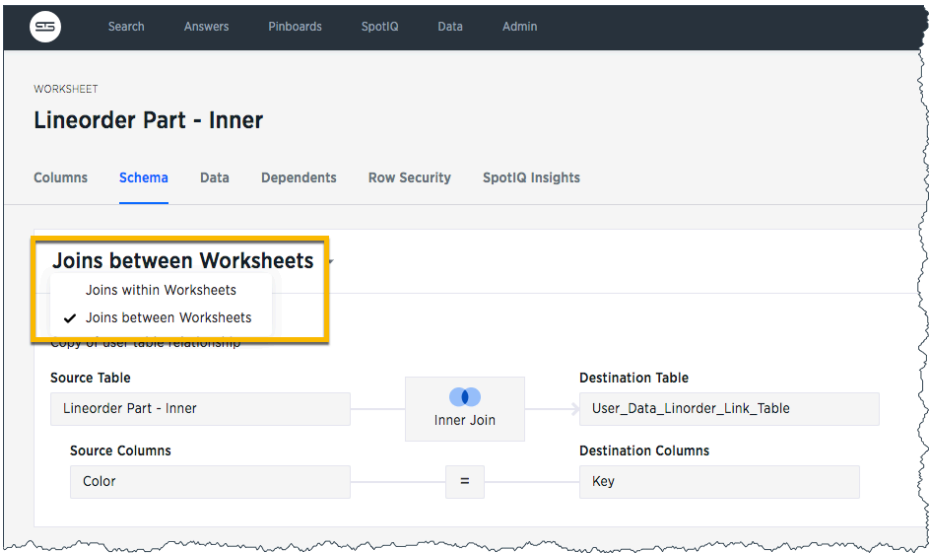
Joining a worksheet to a table, view, or another worksheet creates a relationship that allows them to be searched together. Choose a column to join on that both data sources contain (e.g. employee ID or product key). This process creates a generic join between the worksheet and the other table, view, or worksheet on the column you specify.

You must have either the **Can administer ThoughtSpot** privilege or the **Can manage data** privilege to create a join relationship. If you're not an administrator, you also need edit permissions on the table, view, or worksheet.

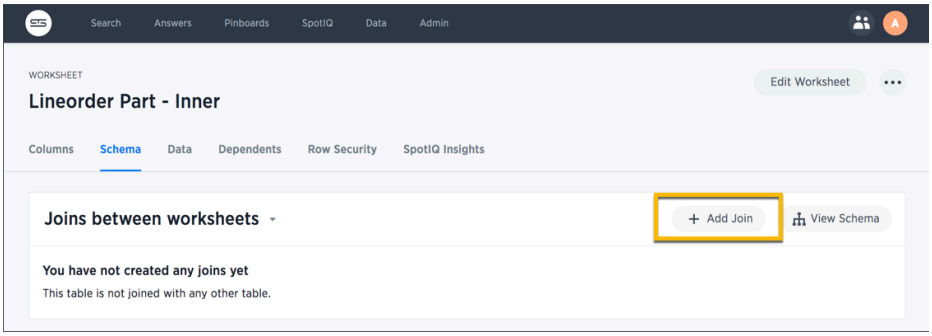
When creating a join between the columns in two data sources, the columns being linked must have the same data type, with the same meaning. That is, they must represent the same data.

To create a relationship through the Web interface:

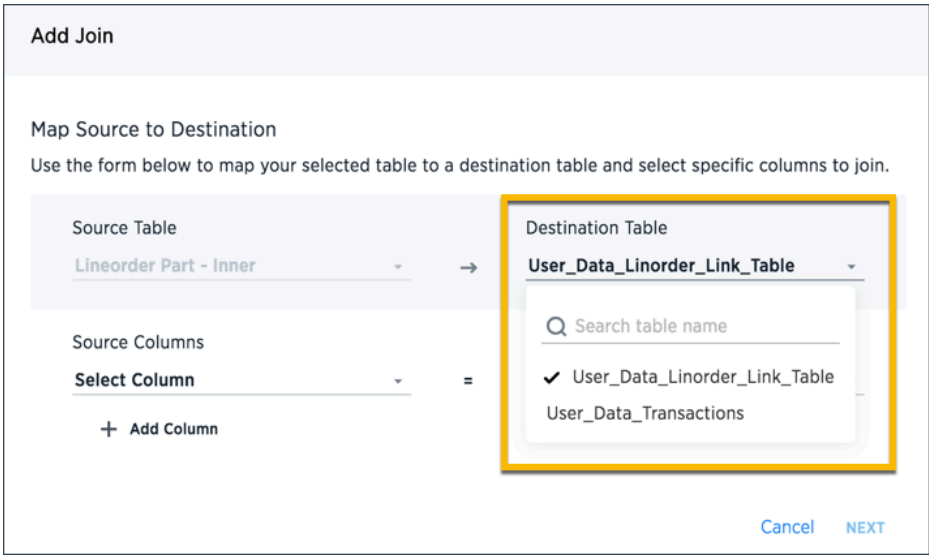
1. To find your worksheet, click **Data** in the top menu, and choose **Worksheets**.
2. Click on the name of your worksheet.
3. Click **Schema**. You will see the list showing existing joins within the worksheet.
4. To view the joins between the worksheet and other data sources, click on **Joins within worksheets** and choose **Joins between worksheets**.



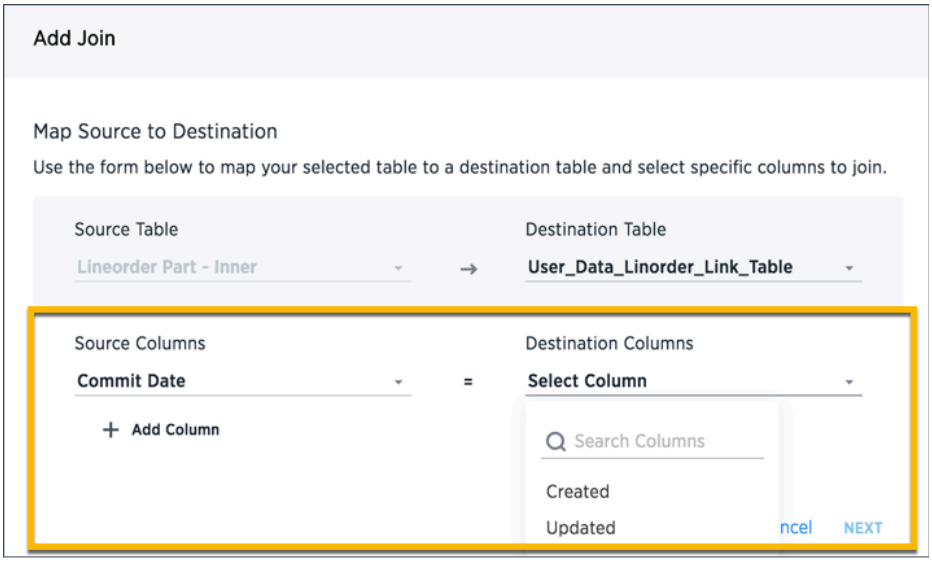
5. Click the + **Add Join** button on the upper right side of the screen.



6. Use the **Map source to destination** dialog to choose the destination table, view, or worksheet you want to join to.



7. Choose the columns you want to join on from the worksheet (source) and destination data source. Click **Next**.



8. Give your join a name and description and click **ADD JOIN**.
9. Repeat these steps until all the joins you want to make have been created.

After creating the join, you may change its name and description by clicking the edit icon. If you want to change the data source or column being joined, you'll need to delete the join and create a new one.

Related Information

- Constraints

Modify joins within a worksheet

> **Summary:** Learn how to change the join type between the tables within a worksheet

When you create a worksheet, you select a join rule. The join rule works together with the joins defined within the worksheet determine how the tables that make up the worksheet are joined, and how those joins behave when searching on the worksheet.

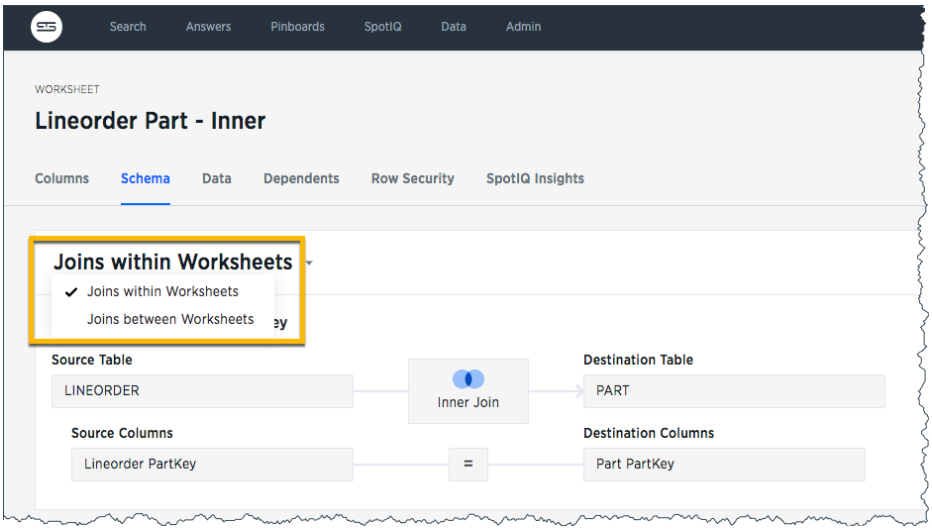
Beginning in ThoughtSpot version 5.0, you aren't limited to just one join rule for the entire worksheet. You can define different types of joins for each join between tables in a worksheet. Be default, each of these individual table joins uses an inner join. But you can override this at the individual join level.

You must have either the **Can administer ThoughtSpot** privilege or the **Can manage data** privilege to modify joins within worksheets.

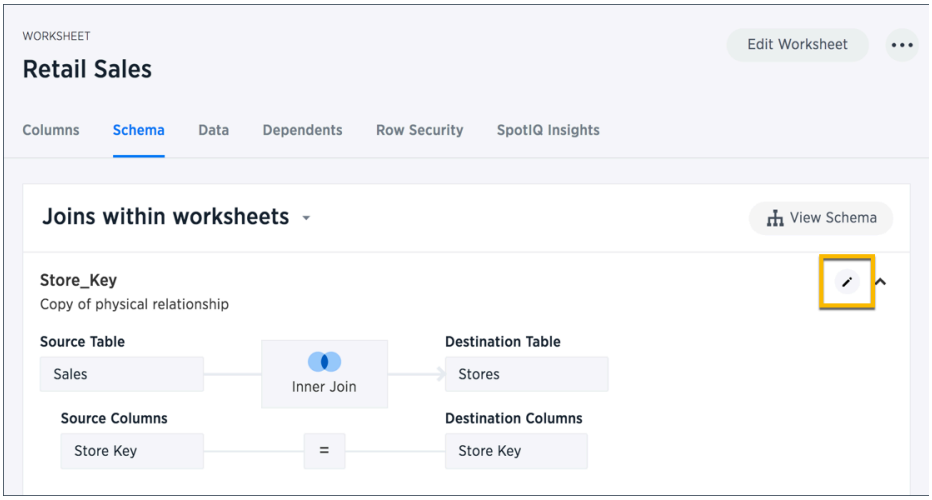
To modify the join types within a worksheet:

1. To find your worksheet, click **Data** in the top menu, and choose **Worksheets**.
2. Click on the name of your worksheet.
3. Click **Schema**. You will see the list showing existing joins within the worksheet. The joins shown here include all the joins between the underlying tables, whether created using TQL or in the browser.

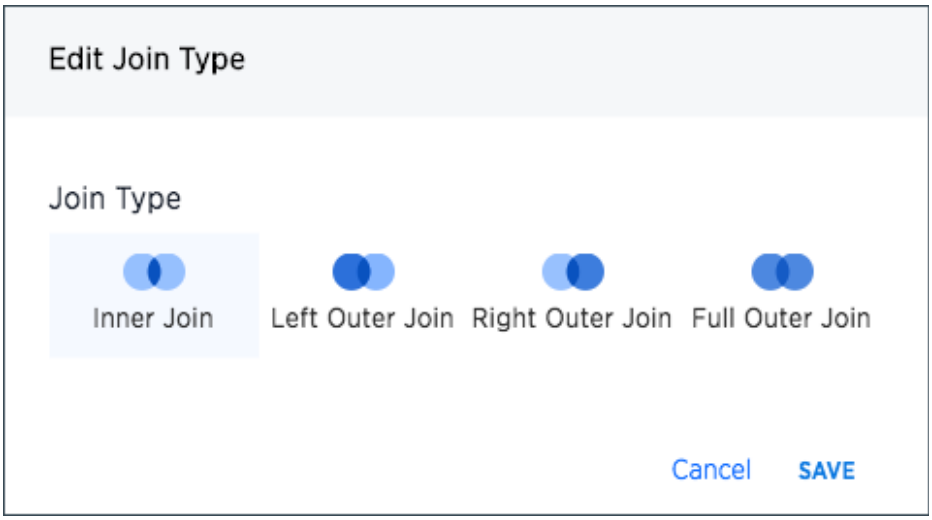
   If you don't see the joins within the worksheet, make sure **Joins within worksheets** is chosen.



4. Find the join you want to modify and click the **Edit** icon. Note that the fact table is always the left table, and it is shown on the left side.



5. Select the join type to use for this relationship in the worksheet and click **Save**.



Now these two tables will be joined using the type you selected, in the context of this worksheet.

Related Information

- Join rule
- Create joins using TQL
- Create join relationships in the browser

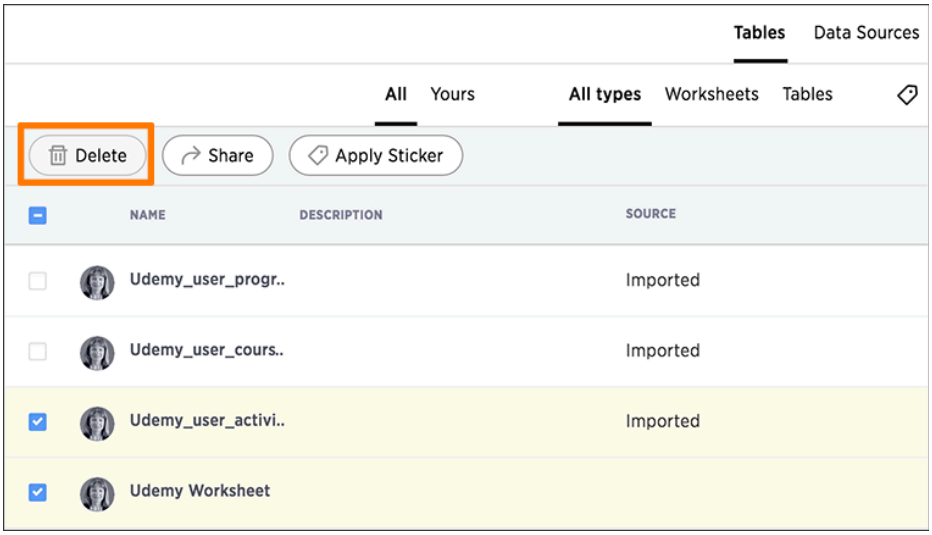
Delete a worksheet or table

Summary: When you try to delete a worksheet or table, you'll see a message listing any dependent objects that must be removed first.

ThoughtSpot checks for dependencies whenever you try to remove a table or worksheet. A list of dependent objects is shown, and you can click on them to delete them or remove the dependency. Then you'll be able to remove the table or worksheet.

To delete a worksheet or table:

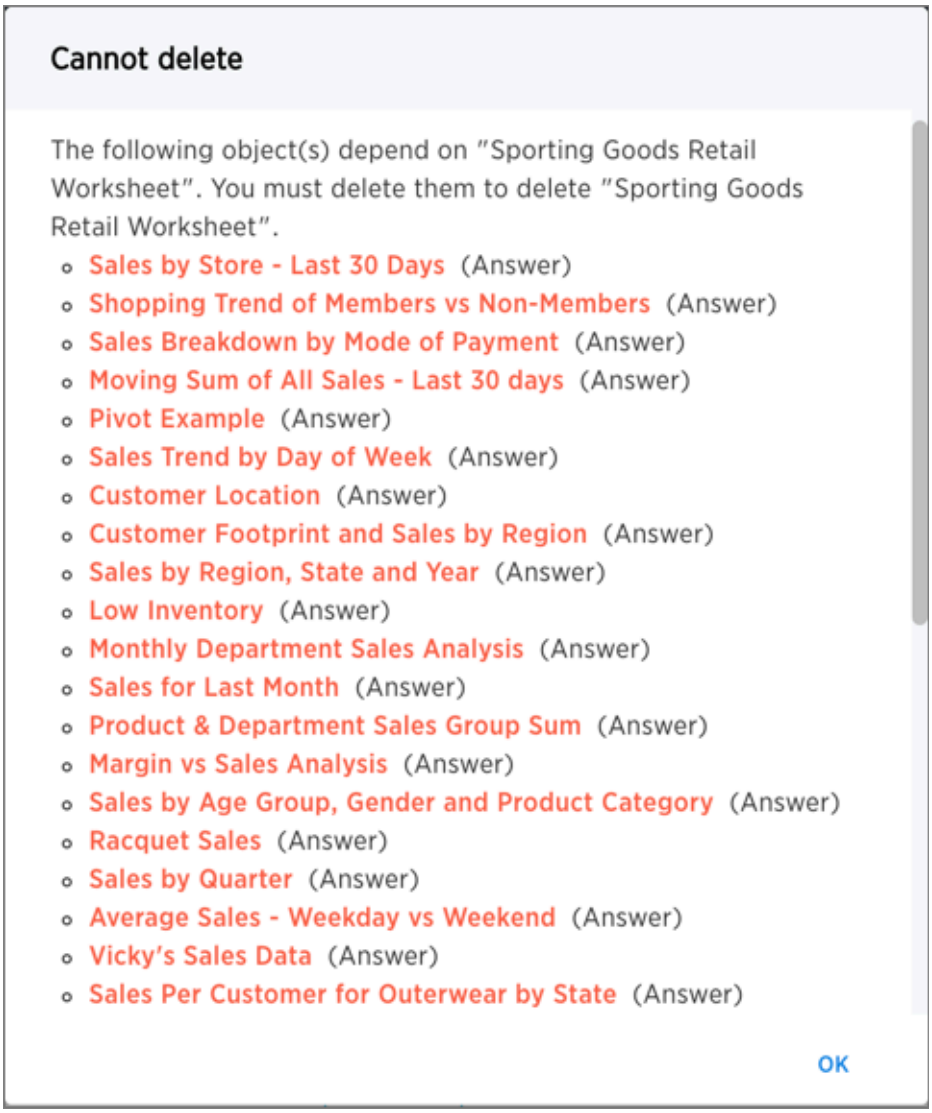
1. Click on **Data**, on the top navigation bar.
2. Find the worksheet or table you want to remove in the list, and check the box next to its name.
3. Click the **Delete** icon.



If you are attempting to delete a data source with dependent objects, the operation will be blocked. You will see a warning, with a list of dependent objects with links.

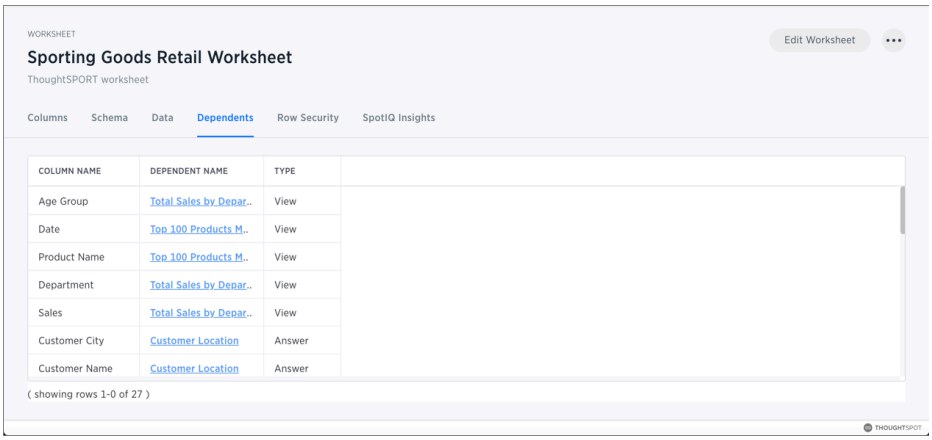
4. Click on the link for an object to modify or delete it.

   When all its dependencies are removed, you will be able to delete the data source.



5. You can also click on the name of a worksheet or table and then click **Dependents**, to see a list of dependent objects with links.

   The **Dependents** list shows the names of the dependent objects (worksheets and pinboards), and the columns they use from that source. You can use this information to determine the impact of changing the structure of the data source or to see how widely used it is. Click on a dependent object to modify or delete it.


WORKSHEET
Sporting Goods Retail Worksheet
ThoughtSPORT worksheet
Edit Worksheet
Columns
Schema
Data
Dependents
Row Security
SpotIQ Insights
COLUMN NAME
DEPENDENT NAME
TYPE
Age Group
Total Sales by Depar..
View
Date
Top 100 Products M..
View
Product Name
Top 100 Products M..
View
Department
Total Sales by Depar..
View
Sales
Total Sales by Depar..
View
Customer City
Customer Location
Answer
Customer Name
Customer Location
Answer
(showing rows 1-0 of 27)
THOUGHTSPOT

Understand groups and privileges

Summary: Creating groups and assigning users to them makes privilege management easier.

Before people can log in and use ThoughtSpot, you need to create a username, a password, and a membership in one or more groups for them.

This page describes manual creation of users, groups, and privileges, but you can also manage users through LDAP or SAML. For information on setting up SAML authentication, see the *ThoughtSpot Application Integration Guide*.

Privileges and groups

Privileges determine what kinds of actions users are allowed to do. You assign privileges to groups. Then, you create users and assign them to groups. This is how you grant users access to different capabilities in ThoughtSpot.

Each group includes a set of privileges for its users. The privileges a group has determine the actions that its members are allowed to do. If a user belongs to more than one group, they will have the highest level of the privileges from all the groups they belong to. Plan your groups so that you can use them to assign a common set of privileges to multiple users. Good planning will pay off in ease of administration and a better search experience.

There is a default group called **All**, which includes every user in ThoughtSpot. When you create a new user, they will be added to the **All** group automatically. You cannot delete the **All** group or remove members from it.

You can also have a hierarchy of groups. That is, groups can belong to (that is, be children of) other groups. When using group hierarchies, permissions are inherited from the parent group. So if you're a member of a sub-group, you would automatically have the privileges of the parent group.

List of privileges

Here are the different privileges, and the capabilities they enable:

Privilege	Description
Can administer ThoughtSpot	Can manage Users and Groups and has view and edit access to all data. Users with this privilege can also download a saved answer.
Can upload user data	Can upload their own data from the application's **Data** page using **Actions** > **Upload data**.
Can download data	Can download data from search results and pinboards.
Can share with all users	Can see the names of and share with users outside of the groups the user belongs to. Members of groups with this privilege can also share with groups marked as **NOT SHAREABLE**.

Privilege	Description
Can manage data	Can create a worksheet. Can also create an aggregated worksheet from the results of a search by selecting **Save as worksheet**. Can also use ThoughtSpot Data Connect, if it is enabled on your cluster.
Can use experimental features	Can access trial and experimental features that ThoughtSpot makes available to early adopters.
Can invoke Custom R Analysis	Can access R scripts to further explore search answers. Includes options to invoke R scripts on visualizations, create and share custom scripts, and share the results of R analysis as answers and pinboards.
Can schedule pinboards	Can create pinboard schedules and edit their own scheduled jobs.
Can administer and bypass RLS	Users in groups with this privilege (directly or via group inheritance): • Are exempt from row-level security (RLS) rules. • Can add/edit/delete existing RLS rules. • Can check or uncheck Bypass RLS on a worksheet. Your installation configuration may enable or disable the availability of this privilege. By default, it is enabled. Administrators or groups with the privilege **Can administer ThoughtSpot** can grant this privilege.

Typically, the **ALL** group has a common set of privileges applies such as the **Can upload user data** and/or **Can download data** privileges.

Privileges are additive, meaning that if a user belongs to more than one group, they will have the highest level of privileges from among the groups they are a member of. They are also inherited from the parent, so that a sub-group gets all the same privileges of its parent, all the way up the group hierarchy.

If you add the privilege **Has administration privileges** to a group, note that all users in that group will be able to see all the data in ThoughtSpot. Administrators can see all data sources, and Row level security does not apply to them.

Permissions to see and edit tables, worksheets, and pinboards are set when you share them with users and groups, as described in the topic Data security.

The following table shows the intersection of user privilege and ability:

		Create/Edit WS	Create View	Modify Col. Props.[1]	Upload Data	Download Data	Share within Group	Share with all users	Manage RLS rules	CrUD Relationships	Read Relationships	See Hidden Cols	Join with Upload Data	Schema Viewer	Use Data Connect	Use Scheduler	Use Auto-Analyze
Can administer ThoughtSpot	Y	Y	Y	Y	Y	Y	Y	Y	Y[2]	Y	Y	Y	Y	Y	Y	Y	
Can upload user data	N	N	N	Y	N	Y	N	N	Y[3]	Y[4]	N	N	N	N	N	N	
Can download data	N	N	N	N	Y	Y	N	N	N	Y[4]	N	N	N	N	N	N	
Can manage data	Y	Y	Y	Y	N	Y	N	N	Y[4]	Y[4]	Y[5]	Y	N	Y	N	N	
Can share with all users	N	N	N	N	N	Y	Y	N	N	Y[4]	N	N	N	N	N	N	
Can Auto-Analyze (SpotIQ privilege)	N	N	N	N	N	N	N	N	N	Y[4]	N	N	N	N	N	Y	
Can Administer and Bypass RLS	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	
None	N	N	N	N	N	Y	N	N	N	Y[4]	N	N	N	N	N	N	

	Create/Edit WS	Create View	Modify Col. Props. [1]	Upload Data	Download Data	Share within Group	Share with all users	Manage RLS rules	CrUD Relationships	Read Relationships	See Hidden Cols	Join with Upload Data	Schema Viewer	Use Data Connect	Use Scheduler	Use Auto-Analyze

Table notes:

1. Applies to non-owners only.
2. Any tables.
3. Author of at least one table in relationship.
4. Only when read permission for columns used in the relationship.
5. With edit permission.

Related information

- Add a group and set security privileges
- Add a user

Create, edit, or delete a group

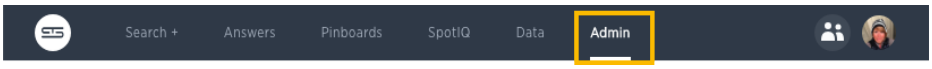
Summary: Good planning when creating groups and assigning privileges will pay off in ease of administration and a better search experience.

Before adding users, create the groups they will belong to. Each group includes a set of privileges for its users.

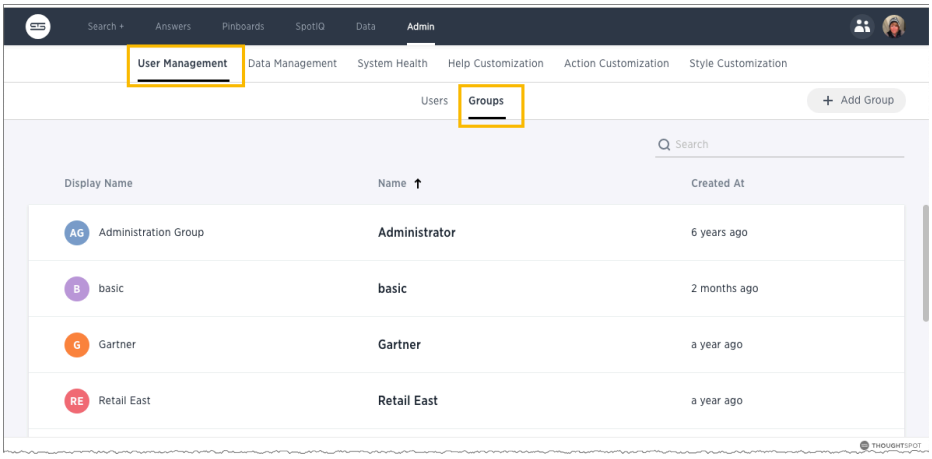
Create a group

To create a group and add privileges for the group:

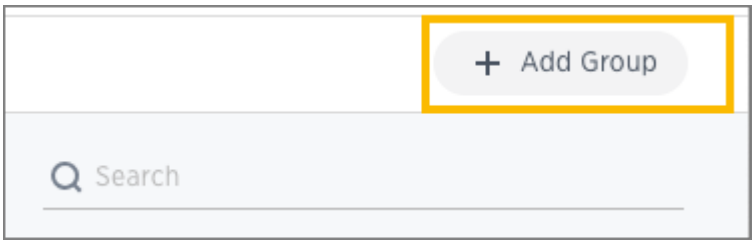
1. Log into ThoughtSpot from a browser.
2. Click on the **Admin** icon, on the top navigation bar.



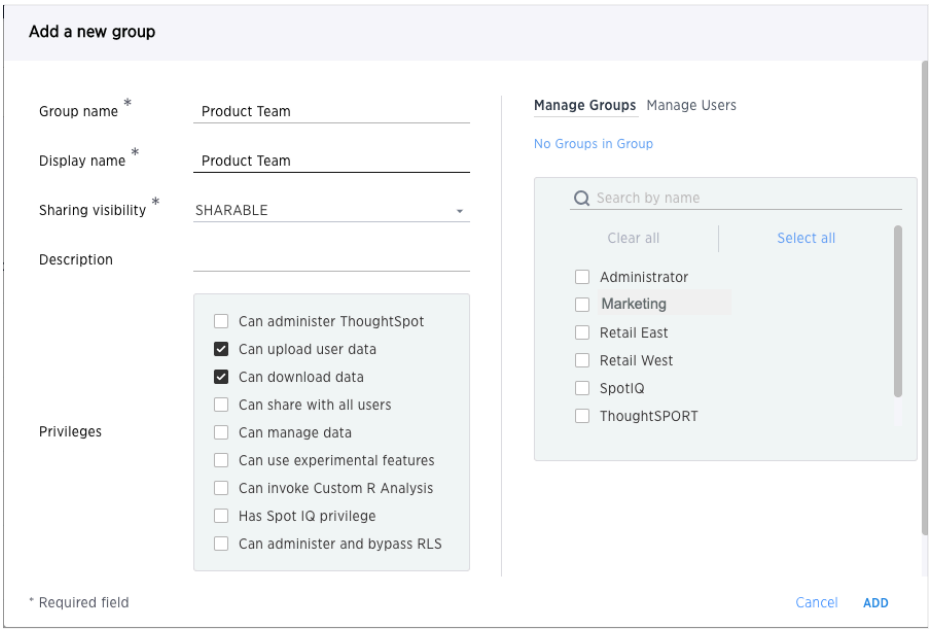
3. In the **Admin** panel, click on **User Management** and **Groups**.



4. Click the + **Add Group** button on the upper right hand side of the list of groups.



5. Enter the details for the new group:



Field	Description
Group name	Enter a unique name for the group.
Display name	Name of the group as it appears in ThoughtSpot.
Sharing visibility	Indicate whether objects can be shared with this group. When set to **SHAREABLE**, this group is an option in the **Share** dialog.
Description	Optionally enter a description.
Privileges	Check the privileges you want to grant to the group. If you add the privilege **Has administration privileges** to a group, all users in that group can see all the data in ThoughtSpot. Administrators can always see all data sources, and Row level security does not apply to them.

6. Click the **Manage Groups** tab if you want to add sub-groups.

   Find the groups you want to add in the list, or search for them by name. Check the box next to each group you want to add to the group.

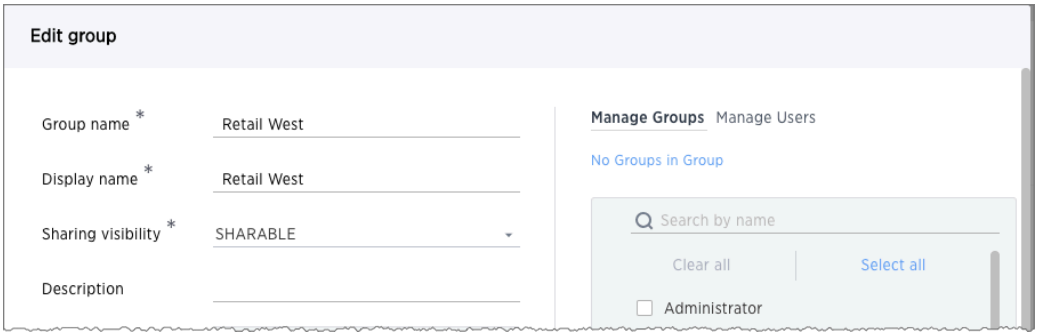
7. Click the **Manage Users** tab if you want to add users.

   Find the users you want to add in the list, or search for them by name. Check the box next to each user you want to add to the group.

8. Click **Add** to create the group.

Edit a group or delete a group

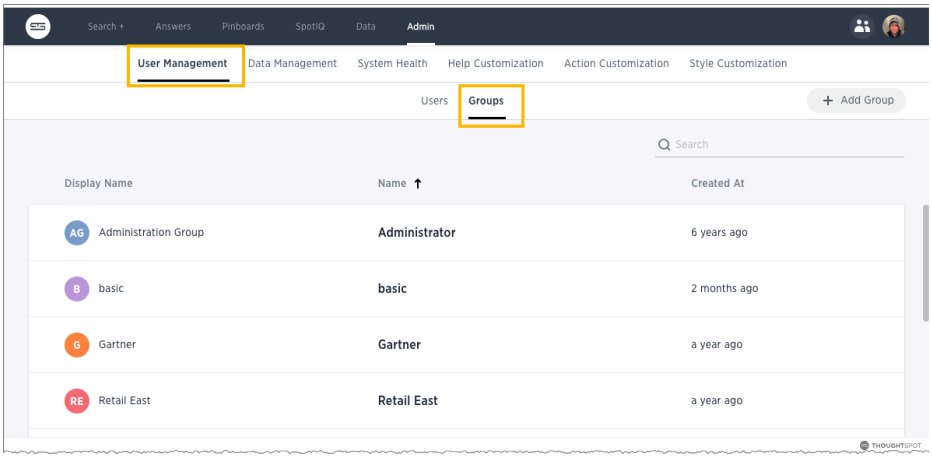
After adding a group, you can always go in and edit its settings to add or revoke privileges. The new settings will apply to all the group members. When editing a group, keep in mind that only sub-groups appear in a group:



The **No Groups in Group** only indicates there are no children in this group's hierarchy. There may be a parent. This group inherits all the privileges of any parent group it may have. Keep this in mind when adding users.

To edit or delete an existing group:

1. Log into ThoughtSpot from a browser.
2. Click on the **Admin** icon, on the top navigation bar.
3. In the **Admin** panel, click on **User Management** and **Groups**.



4. Find the group you want to edit in the list and click its name, or the edit icon  .

   If you don't see the name of the group, try searching for it. You can also delete a group from this page by clicking the **Delete** icon. Deleting a group does not delete its users.

5. Make your changes and click **Update**.

List the group members

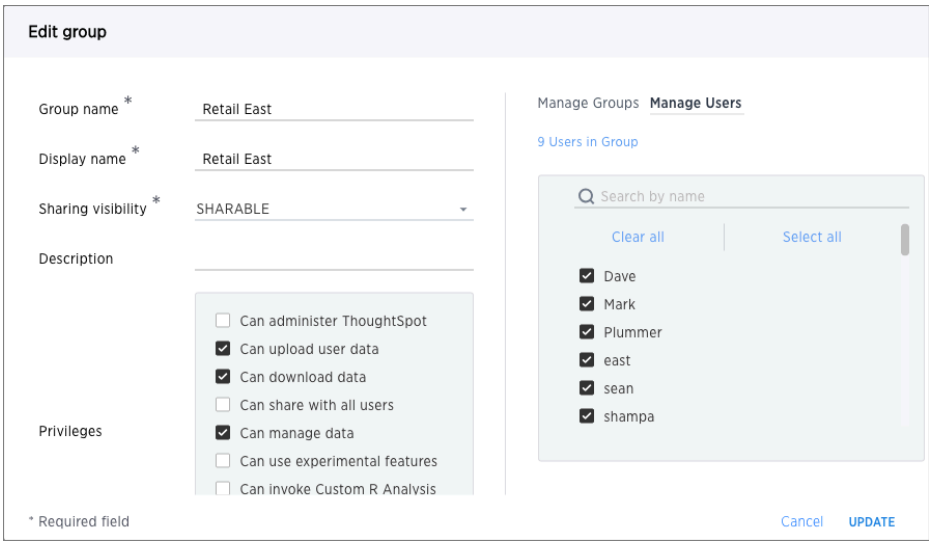
The system shows you the first 15 users in your group. To identify if other users are present, you must search for the specific user name.

1. Click **Admin** from the top navigation bar.

   The system displays the **Admin** panel.

2. Select **User Management**.
3. Click on a group to edit it.

   The system displays the **Edit group** dialog.



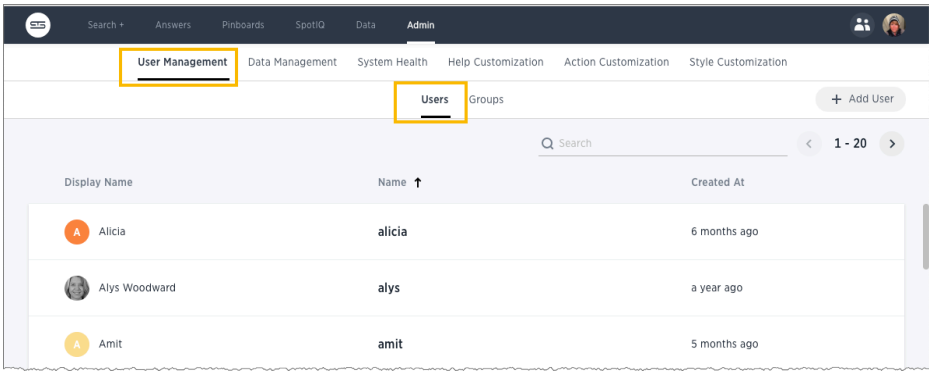
4. Choose **Manage Users**.

   The first 100 users which are in the group are listed. The group could maintain more, you have to search for a specific user to find others.

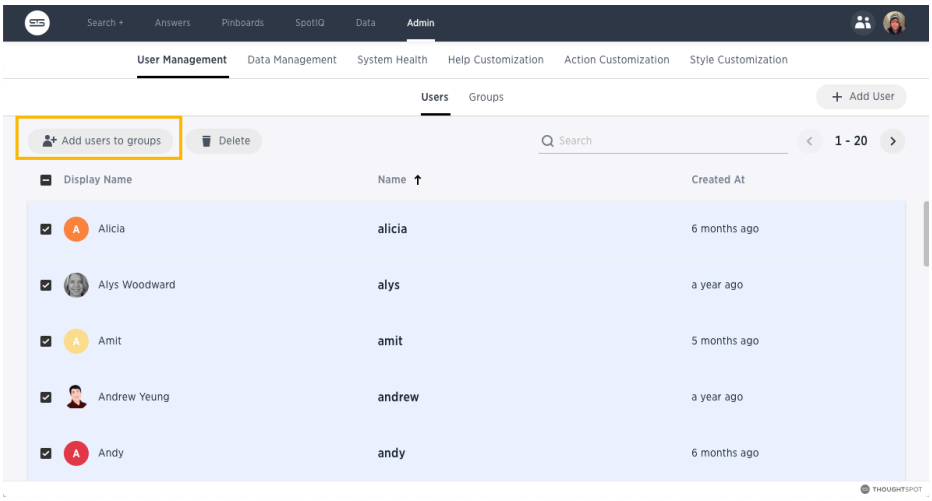
Add multiple users to a group

You can add multiple users to a group using one button. To add multiple users to a group:

1. Log into ThoughtSpot from a browser.
2. Click on the **Admin** icon, on the top navigation bar.
3. In the **Admin** panel, click on **User Management** and **Users**.



4. Select the users you would like to add to the same group from the list.
5. Click the **Add Users to Groups** button on the top of the list of users.



Add, edit, or delete a user

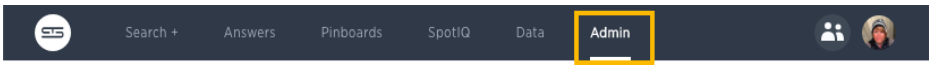
You will create a user account for each unique person who will access ThoughtSpot, either manually or through LDAP. If a user has access through LDAP, that user's information is managed via your LDAP installation. If you create a user manually in ThoughtSpot, you manage that user in ThoughtSpot.

You can edit manually created users through the interface. If a manually-created user forgets their password, you can reset it by editing the user. If you have forgotten the admin password, please call ThoughtSpot Support.

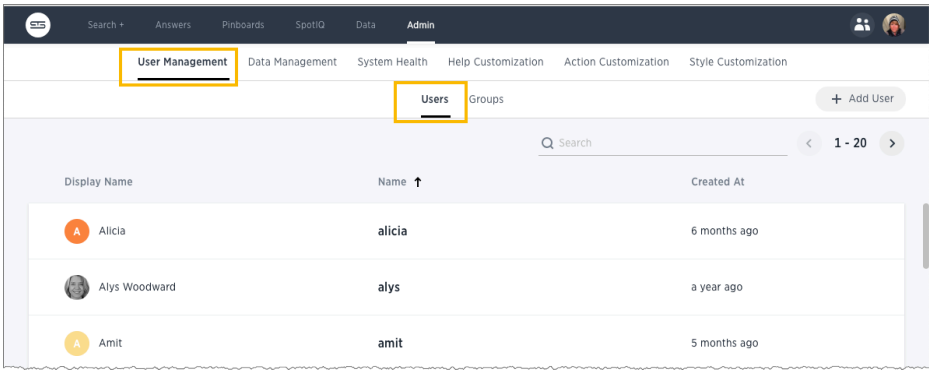
Create a user through the interface

This procedure shows how to creating a user manually. When you create a user, you can assign group memberships. The group's privileges and permissions apply to all of its members. Any user you create will be added to the group **All** automatically.

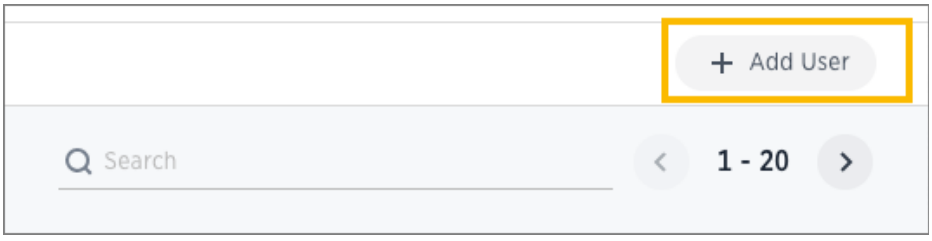
1. Log into ThoughtSpot from a browser.
2. Click on the **Admin** icon, on the top navigation bar.



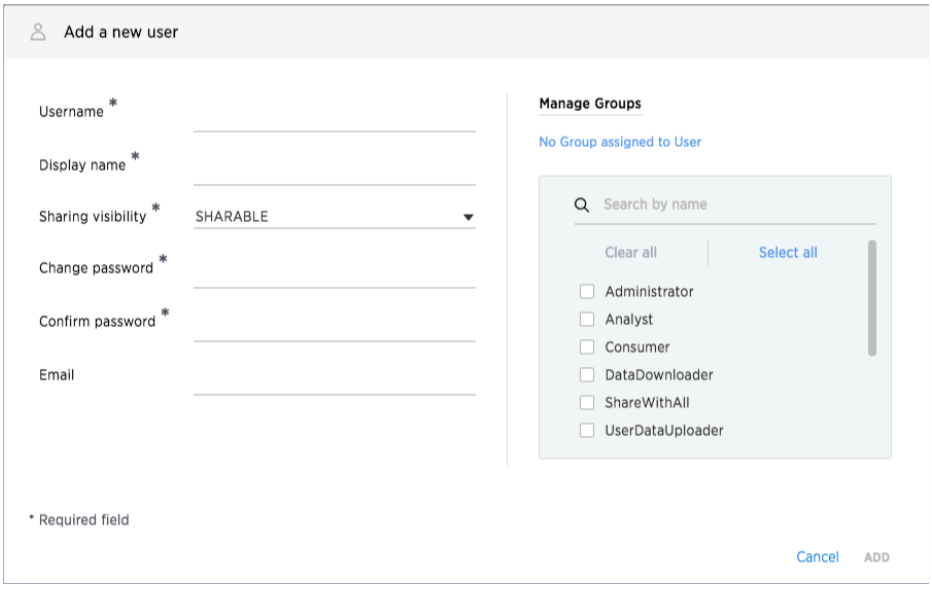
3. In the **Admin** panel, click on **User Management** and **Users**.



4. Click the **+ Add User** button on the upper right hand side of the list of groups.



5. Enter the details for the new user:



Field	Description
Username	A login name for the user. Usernames must be unique and lowercase. If you are using Active Directory to authenticate users, and your LDAP configuration requires users to be created manually (i.e. they are not created automatically in ThoughtSpot upon authentication), the user-name you specify has to be domain qualified (e.g. `user-name@ldap.thoughtspot.com`), and you must enter a dummy password.
Display name	A unique name for the user (usually their first and last name).
Sharing visibility	Indicate whether objects can be shared with this user. When set to **SHAREABLE**, this user is an option in the **Share** dialog.
Change password	A password.
Confirm password	Enter the password again.
Email address	The user's email address. This is used for notification when another user shares something with them.

Manage groups	Select all the groups the user will belong to. If you add the user to a group that has the privilege **Has administration privileges**, note that they will be able to see all the data in ThoughtSpot. When you create a new user, the groups they belong to define the user's: • Privileges, the actions they are allowed to do, which are defined when you Add a group and set security privileges. • Permissions, the data they can access and view, which is defined when you Data security. Administrators can see all data sources, and Row level security does not apply to them.

6. Click **Add** to create the user.

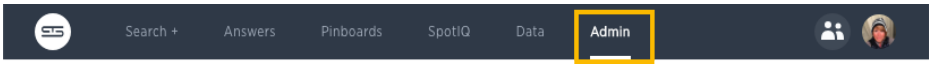
Edit or delete a user

After a user has been created, you can always go back and change their settings, for example to change their group memberships or change their password. You can also change their name as long as it remains unique

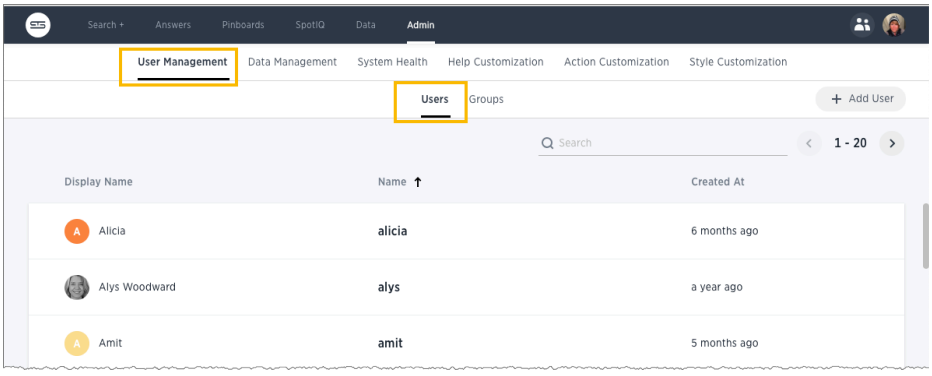
As an administrator, you can edit a user and change the groups the user belongs to. You can also edit a user to reset a user's password by entering and confirming the new password. This is useful if a user has forgotten their password, or to effectively disable an account.

To edit an existing user:

1. Log into ThoughtSpot from a browser.
2. Click on the **Admin** icon, on the top navigation bar.



3. In the **Admin** panel, click on **User Management** and **Users**.



4. Find the user you want to edit in the list and click on its name or the edit icon . If you don't see the name of the user, try searching for it.

You can also delete a user from this page by clicking the **Delete** icon.

5. Make your changes and click **Save**.

Job management (scheduled pinboards)

Summary: All jobs on your cluster will appear on the Jobs Management page. You can also view jobs for individual pinboards under the pinboard Actions dropdown.

The **Jobs Management** page found on the **Admin** section in the ThoughtSpot web application allows you to create and manage jobs, namely scheduled pinboards. Scheduled pinboards should help with preparing for recurrent meetings, when reviewing the same pinboard is necessary. They should also be useful when you have metrics you want to monitor at a consistent interval, like daily or monthly sales targets.

You can get pinboards emailed to you on a regular basis and do analysis offline. This introduces an additional format for you to consume and share pinboards with others, including those who don't have a ThoughtSpot account.

Contact ThoughtSpot Support if scheduled pinboards is not enabled on your cluster, or you can run the command `tscli scheduled-pinboards` to enable it yourself.

Scheduled pinboard creators

Administrators and users with can schedule pinboard privilege can schedule and manage pinboard jobs. These scheduled pinboard creators must have at least edit-only and view-only rights to the pinboard they want to share.

Warning: It is recommended that admins carefully choose who to give can schedule pinboard privilege to, since there is a possible security hole where a user with limited access can get a pinboard email with all access data.

Row level security

The scheduled pinboards respect row level security rules. This means if the recipients are users in ThoughtSpot, then they can only see data based on their own access to the pinboard. If the user does not have at least view-only access to the pinboard, then they will not see anything in the email. However, if the recipients are from outside of the cluster, then they will have access to the dataset of the pinboard based on the sender's permissions.

Scheduled pinboard formats

The pinboard visualizations are attached to the scheduled email as CSV or PDF files. Saved configurations such as pinboard filters are applied to the attachments. Refer to the table to see how the pinboard data is represented in each file format.

CSV	PDF
The CSV file gets data only for table visualizations.	The PDF file gets data for all visualizations.
The email has n CSV attachments, where there are n table visualizations in the pinboard.	The email has only one attachment file, which includes every visualization on its own page.
Table visualizations have all data rows that they’re supposed to have.	Table visualizations include only the first 100 rows.
In the case of a corrupted pinboard: no email is sent. An error message indicating failure to export data is visible on the Admin Jobs Management page.	In the case of a corrupted pinboard: the PDF attachment has empty/error screenshots.
In the case of a corrupted visualization: an email with the visualizations whose data can be exported is sent. An error message indicating visualization export error is visible on the Jobs Management page.	In the case of a corrupted visualization: the PDF attachment has empty/error slots for the corrupted visualizations.

The size of each email is limited to 25 MB, which matches most email services size limitations.

And the total number of recipients for a scheduled pinboard job cannot exceed the default of 1000.

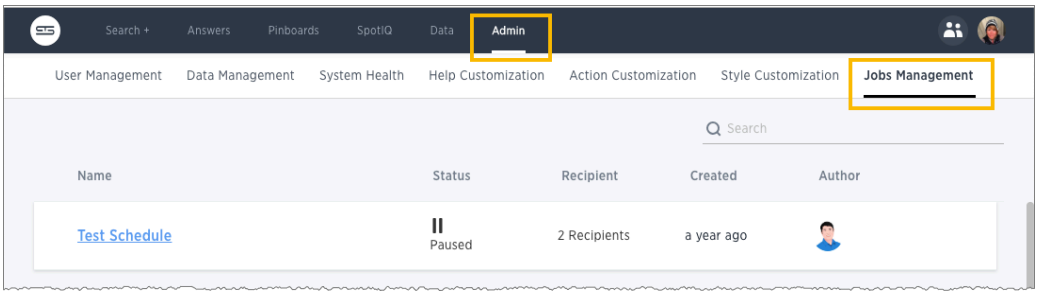
Related information

For information on creating a pinboard, see the Schedule a pinboard job.

Scheduled pinboards management

Summary: You can manage all scheduled pinboards on the Jobs Management page under Admin.

Users who are not admins, but have can schedule pinboard privilege, can only view pinboard schedules they've created. You can select specific jobs and choose to pause, resume, edit, or delete them. You can have up to 50 scheduled jobs on your cluster at time. Contact ThoughtSpot Support if you'd like to increase this limit.

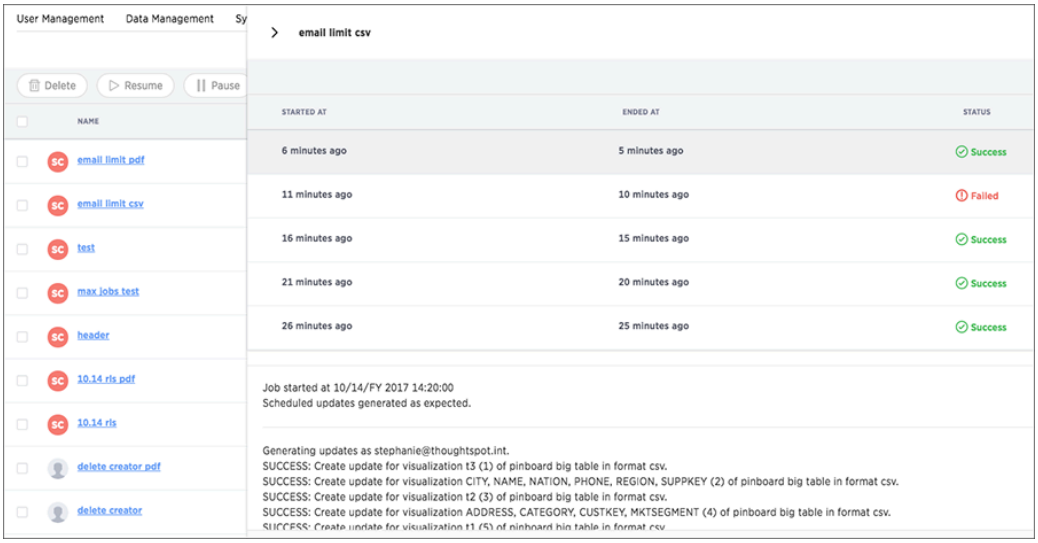


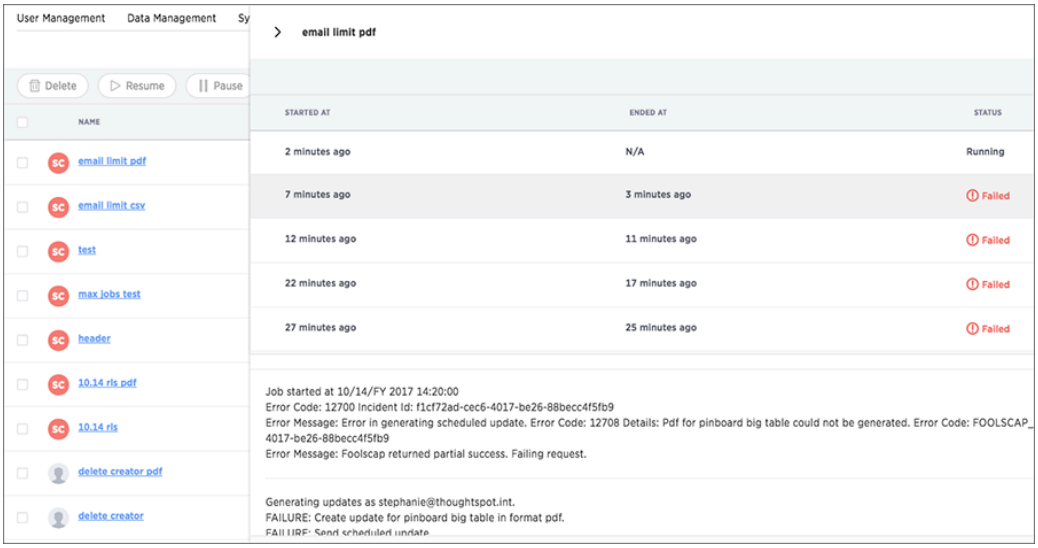
Bulk actions

Select the scheduled pinboards and use the **Delete**, **Resume**, and **Pause** buttons to perform these bulk actions. Deleting a pinboard will also delete any schedules linked to it.

Job statuses

Clicking on the row of a job will open a detailed view of every generated update of that job. You can see the start and end times of the job, as well as the status. Clicking on a job will show more information about the status updates.





Pinboard links

Click the scheduled pinboard name link to jump to a Edit schedule page, where you can edit the schedule configurations.

You can also click on the pinboard link provided in the scheduled pinboard emails to jump to the pinboard in ThoughtSpot. In order to have the link direct you to the correct URL, you must first configure front end host and port access. Contact ThoughtSpot Support to configure these settings.

Overview of security features

There are several aspects of security, including access and permissions, data security and privacy, and security from an IT perspective.

- System Security refers to audit logs and security policies.
- Data Security refers to which users can see which data in the ThoughtSpot application, and includes:
    - Users and Groups
    - Privileges
    - Table and columns sharing
    - Row level security
    - Worksheet sharing
    - Pinboard sharing
- Network Security refers to ports for external traffic and traffic within the cluster. Some ports must remain open for handling network requests from outside the ThoughtSpot instance. To see a list of network ports that must remain open to outside traffic, and for inter-cluster communication, review the information in Network ports.

System security

Summary: System security refers to audit logs and security policies.

ThoughtSpot includes a number of management tools, monitoring applications, and automated processes to support system security. System security includes managing access and privileges, audit logs, security policies, and Linux OS installed package updates.

Audit logs

There are several ways you can view audit log information in ThoughtSpot. You can see recent events in the Control Center or view more detailed audit logs using tscli. Administrators can view audit logs of configuration changes users have made to ThoughtSpot in these ways:

- Monitor events from the Control Center.
- Generate audit log reports through the `tscli` command.

You can access an audit log of cluster events through tscli. You can also access information on cluster updates, configurations, data loading and metadata events.

Use the `tscli event list` command to return an audit list of events from the cluster. The syntax is:

```
tscli event list
   [--include <all|config|notification>]
   [--since <hours,minutes,days>
   | --from <yyyymmdd-HH:MM>
   --to <yyyymmdd-HH:MM>]
   [--detail]
   [--summary_contains
   <'string1'| 'string2' ...>]
   [--detail_contains
   <'string1'| 'string2' ...>]
   [--attributes
   <key1='value1'|
   key2='value2' ...>]
```

Optional parameters are:

Parameter	Description
`--include`	Specifies the type of events to include, and can be `all`, `config`, or `notification`.
`--detail`	Returns the events in a detail format rather than a tabular summary, which is the default.

Parameter	Description
`--summary_contains <'string1' | 'string2' ...>`	Specifies a string to check for in the event summary. Enclose strings in single quotes, and separate multiple strings with &pipe;. Events that match all specified strings will be returned.
`--detail_contains <'string1'| 'string2' ...>`	Specifies a string to check for in the detail. Enclose strings in single quotes, and separate multiple strings with `|` (pipe symbol). Events that match all specified strings will be returned.
`--attributes <key1='value1' &pipe; key2='value2' ...>`	Specifies attributes to match as key=value pairs. Separate multiple attributes with `|` (pipe symbol). Events that match all specified key/value pairs will be returned. Put single quotes around the value(s).

And a time window made up of either:

- `--since <hours,minutes,days>` is a time in the past for where the event audit begins, ending at the present time. Specify a human readable duration string, e.g. 4h (4 hours), 30m (30 minutes), 1d (1 day).

Or both:

- `--from <yyyymmdd-HH:MM>` is a timestamp for where to begin the event audit. It must be of the form: yyyymmdd-HH:MM.
- `--to <yyyymmdd-HH:MM>` is a timestamp for where to end the event audit. It must be of the form: yyyymmdd-HH:MM.

To get audit logs:

1. Log in to the Linux shell using SSH.
2. Issue the `tscli event list` command, with the desired parameters, for example:

```
$ tscli event list
   --include config
   --since 24 hours
```

Security policies

Security policies are the principles and processes ThoughtSpot uses in development to ensure a product that conforms to security standards. Security policies ensure a secure product with each release. When a release is in development, each build is tested using Qualys Network Security and Vulnerability Management Suite. Issues and vulnerabilities are fixed proactively, based on the results.

The ThoughtSpot Engineering and ThoughtSpot Support teams are notified of Common Vulnerabilities and Exposures (CVEs), so they can patch OS packages proactively as well. You can view installed packages along with their version numbers at any time, in order to see if you require an update to ThoughtSpot.

Whenever a CVE is identified, and an OS package needs to be updated, the next patch release will include the patch or update. You can view installed Linux packages at any time, along with the version numbers of the installed packages.

Data security

Summary: Data security refers to which users can see which data in the ThoughtSpot application.

Sharing and security privileges govern what data a user can access and what they can do with the data. Admins can use privileges to regulate access to information and provide a personalized user experience.

Users, groups, and privileges

Data security applies to users and groups. Users can be managed manually or through LDAP. Each user can have membership in one or more groups. Admins can make security settings that determine what users are allowed to do in ThoughtSpot. These settings are applied at the group level.

The following table shows the intersection of user privilege and ability:

The following table shows the intersection of user privilege and ability:

		Create/Edit WS	Create View	Modify Col. Props.[1]	Upload Data	Download Data	Share within Group	Share with all users	Manage RLS rules	CrUD Relationships	Read Relationships	See Hidden Cols	Join with Upload Data	Schema Viewer	Use Data Connect	Use Scheduler	Use Auto-Analyze
Can administer ThoughtSpot	Y	Y	Y	Y	Y	Y	Y	Y	Y[2]	Y	Y	Y	Y	Y	Y	Y	
Can upload user data	N	N	N	Y	N	Y	N	N	Y[3]	Y[4]	N	N	N	N	N	N	
Can download data	N	N	N	N	Y	Y	N	N	N	Y[4]	N	N	N	N	N	N	
Can manage data	Y	Y	Y	Y	N	Y	N	N	Y[4]	Y[4]	Y[5]	Y	N	Y	N	N	
Can share with all users	N	N	N	N	N	Y	Y	N	N	Y[4]	N	N	N	N	N	N	
Can Auto-Analyze (SpotIQ privilege)	N	N	N	N	N	N	N	N	N	Y[4]	N	N	N	N	N	Y	
Can Administer and Bypass RLS	N	N	N	N	N	N	N	N	Y	N	N	N	N	N	N	N	
None	N	N	N	N	N	Y	N	N	N	Y[4]	N	N	N	N	N	N	

Create/Edit WS	Create View	Modify Col. Props. [1]	Upload Data	Download Data	Share within Group	Share with all users	Manage RLS rules	CrUD Relationships	Read Relationships	See Hidden Cols	Join with Upload Data	Schema Viewer	Use Data Connect	Use Scheduler	Use Auto-Analyze
Table owners [2]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Table authors [3]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Can edit [4]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Can view [5]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table notes:

1. Applies to non-owners only.
2. Any tables.
3. Author of at least one table in relationship.
4. Only when read permission for columns used in the relationship.
5. With edit permission.

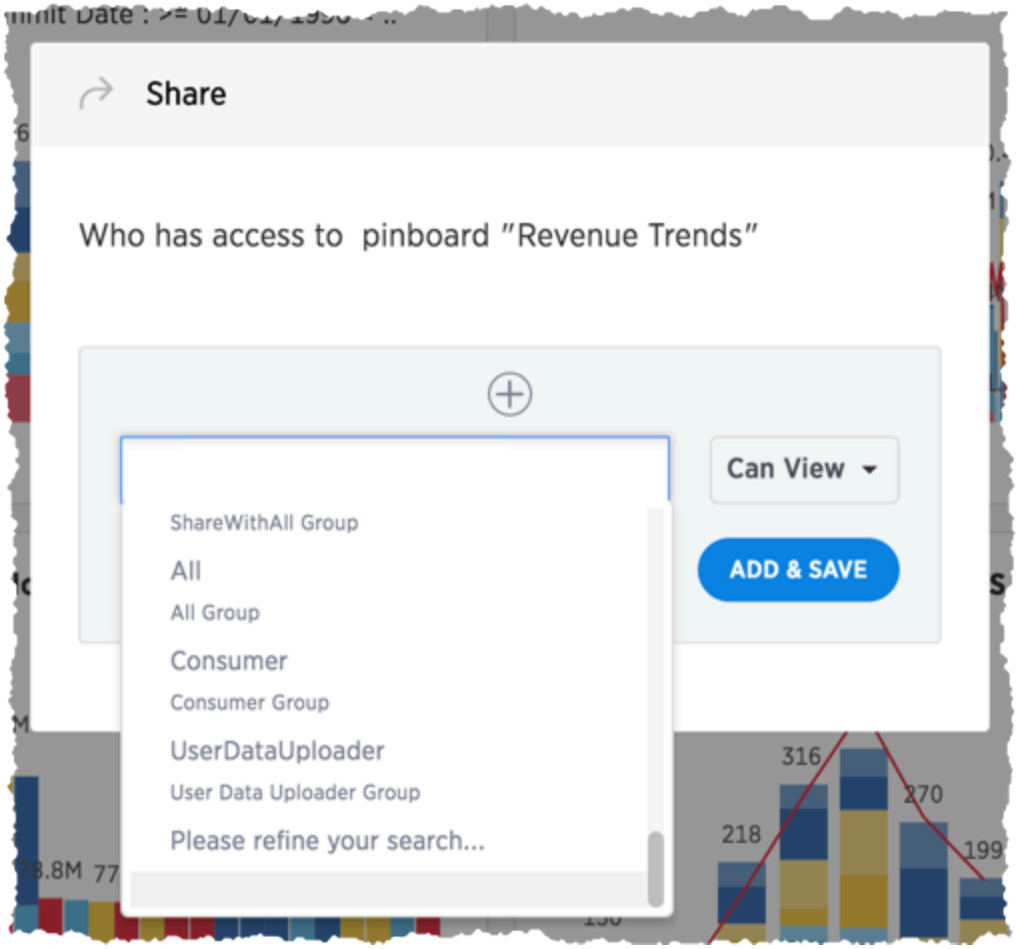
Security model for sharing objects

You can share with groups and with individual users. Sharing of tables can be defined at the table, column, or row level. This provides flexibility in modeling your data security policy. Security and sharing settings apply to several different types of objects, each of which has its own security default settings and rules.

Object type	Description	Default security model
Tables	The source data tables that have been loaded using ThoughtSpot Loader.	Administrator users have access to source tables. They can share a table with other users or groups. See [Share tables and columns](share-source-tables.html#)
Columns	The columns in the source data tables that have been loaded using ThoughtSpot Loader.	Administrator users have access to columns in the source tables. They can share selected columns with other users or groups. See [Share tables and columns](share-source-tables.html#)
Rows	The rows in the source data tables that have been loaded using ThoughtSpot Loader.	All rows in the source tables are shared with all users by default.
Imported data	Data that was imported using a Web browser.	Only the user who imported the data (and any user with administrator privileges) has access to it by default. They can share a table (or selected columns) with other users or groups. See [Share tables and columns](share-source-tables.html#)
Worksheets	A worksheet created using a Web browser.	Only the creator of the worksheet (and any user with administrator privileges) has access to it by default. They can share a worksheet with other users or groups. See [Share worksheets](share-worksheets.html)
Pinboards	A pinboard of saved search results.	Anyone who can view a pinboard can share it. See [Share a pinboard](share-pinboards.html)

Understanding SHAREABLE

When you share an object, only the users and groups that have **SHAREABLE** set for the **Sharing visibility** option appear on the dialog.



Only users in the **Administrators** group or users with **Admin** privileges can share with groups marked as **NOT SHAREABLE**. Members of a group with **Can share with all users** authorization can also share with groups marked as **NOT SHAREABLE**.

Users in groups marked **NOT SHAREABLE** cannot share objects among themselves. In multi-tenant scenarios, admins can create groups that bring together portions of two non-share groups so that they can share. For example, the members of group C can share even if they belong to other groups that cannot.

Row level security

ThoughtSpot includes robust row level security, which allows you to filter all objects users see based on conditions you set at the level of row values in base data tables.

You may find it useful to create groups for RLS. To prevent these groups from appearing in the **Share** dialog, create a **NOT SHAREABLE** group with a single user and an RLS group with another single user (1-to-1).

Related information

- Revoke access (unshare)
- Row level security

Share tables and columns

Summary: You can share an entire table, or only some of its columns.

By default, when data is loaded using the ThoughtSpot Loader, ODBC, or JDBC, it is only visible to administrators. Data imported from a Web browser is visible to administrators and the user who uploaded it. Administrators and owners can share **Can View** or **Can Edit** privileges on tables with other users, who can further share them with others.

Permissive or strict sharing

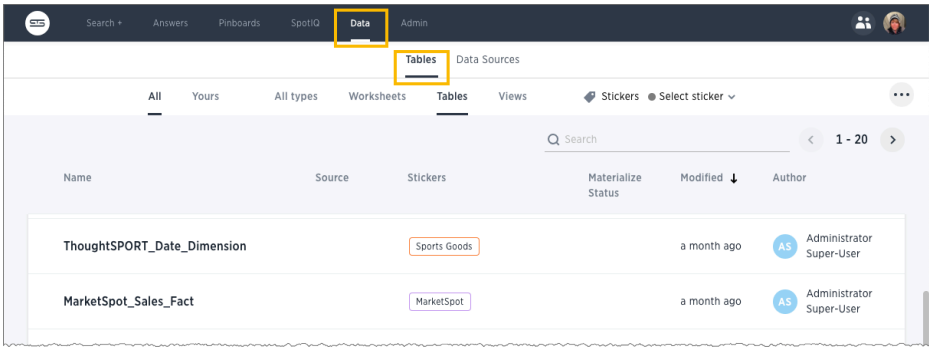
Use caution when sharing tables, because any objects created from them will have dependencies on the tables and their underlying structure. Objects created from tables can include worksheets, answers, and pinboards. This means that if a user wants to drop or modify a table, any object that depends upon it must be edited or removed first, to remove the dependency.

For this reason, it is a best practice to only grant the **Edit** permission on tables to a small number of users. If you want to prevent shares from also revealing the columns regardless of where it appears (worksheets, answers, and pinboards), you can ask ThoughtSpot Customer Support to enable a stricter behavior.

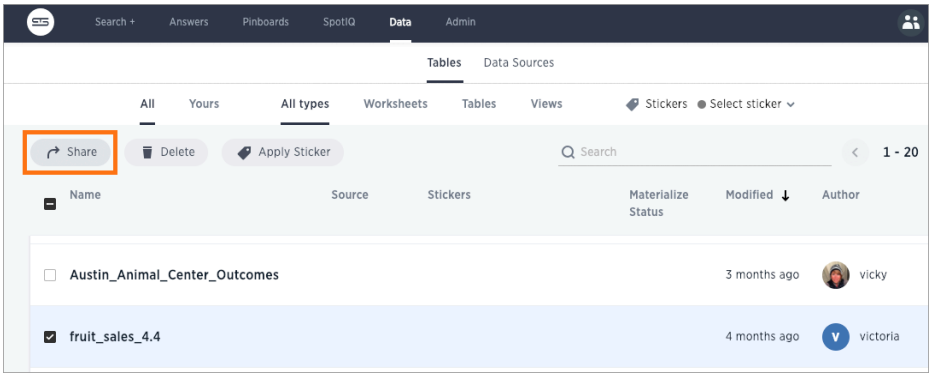
How to share

Share a table or imported data by following these steps:

1. Click **Data** in the top navigation bar.
2. Click on **Tables**.



3. Select one or more tables to share, and click the **Share** icon.
4. Select **Entire Table** or **Specific Columns**.



5. If you selected **Specific Columns**, select the column to share.
6. Click + and select the users and groups with whom you want to share.
7. Configure the level of access by selecting from the dropdown list. You can select:
   - **Can View** to provide read-only access. This enables viewing the table data and defining worksheets on the table.
   - **Can Edit** to allow modification. This enables renaming, modifying, or deleting the entire table and adding or removing its columns.
8. Click **Add and Save**.
9. Click **Done**.

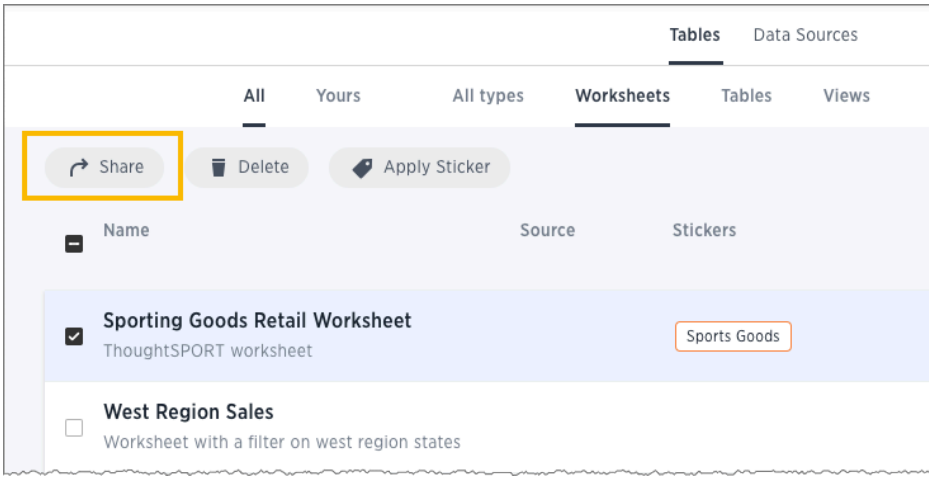
Share worksheets

Summary: You can share worksheets with users or with groups. Sharing a worksheet allows users to select it as a data source and search it.

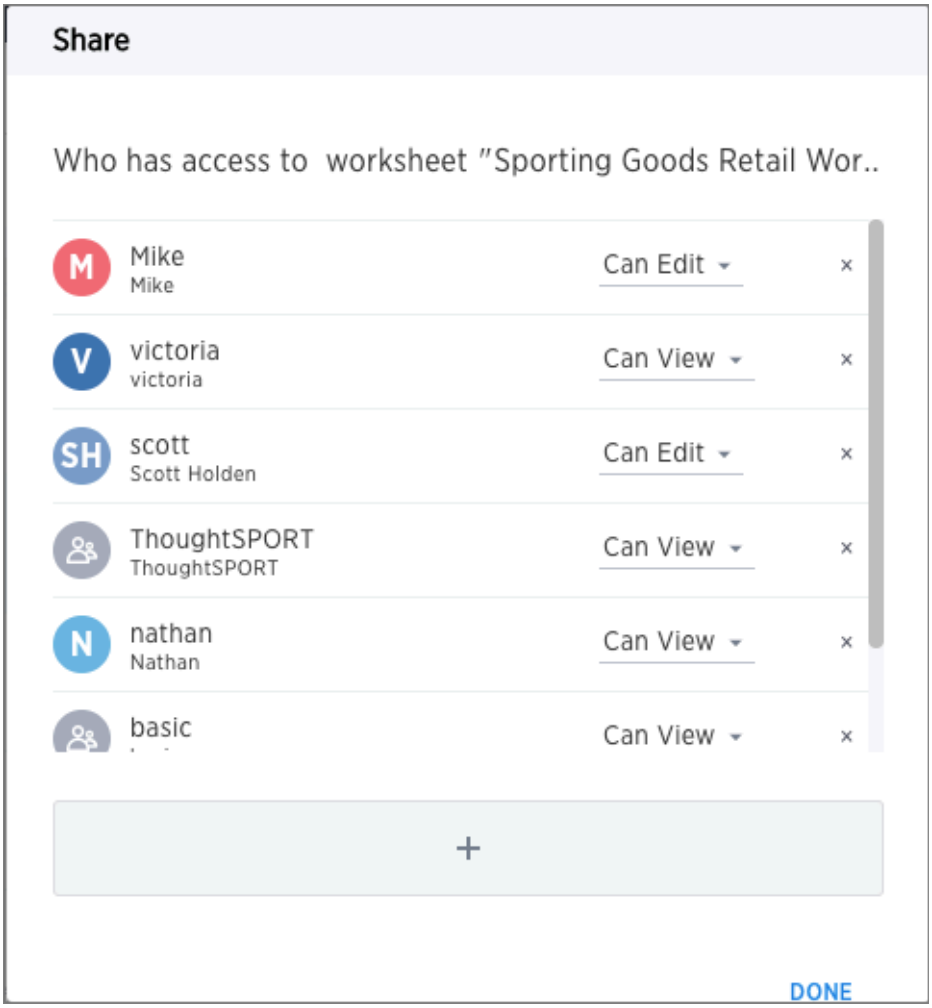
When you share a worksheet, all of its columns are shared. Sharing a worksheet does not share the underlying tables. If you want to share the underlying tables, see Share tables and columns. A worksheet can be shared by the owner of the worksheet, or by an administrator. Users can start searching a worksheet as soon as the worksheet is shared with them.

To share a pinboard:

1. Click **Data** on the top navigation bar and choose **Worksheets**.
2. Select one or more worksheets to share, and click the **Share** icon.



3. Click + **Add users or groups** and select users or groups that you want to share with.



4. Configure the level of access by selecting from the dropdown list. You can select:
    - **Can View** to provide read-only access. Enables viewing the worksheet and searching on it.
    - **Can Edit** to allow modification. Enables renaming, modifying filters, or deleting the worksheet and adding or removing its columns. To add columns to a worksheet a user needs access to the underlying table.

5. Click **Add and Save**.
6. Click **Done**.

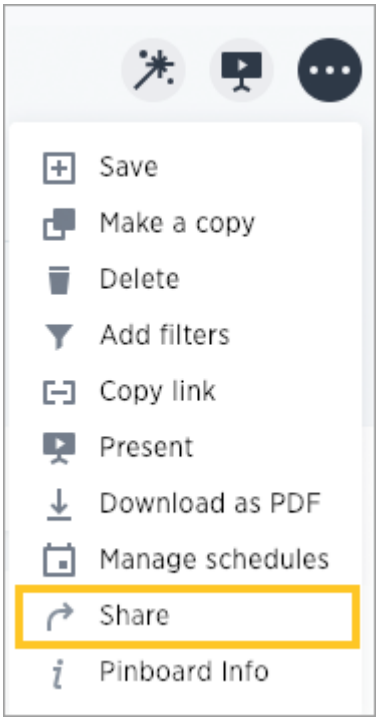
Share a pinboard

Summary: Whenever you view a pinboard you have the option of sharing it with others.

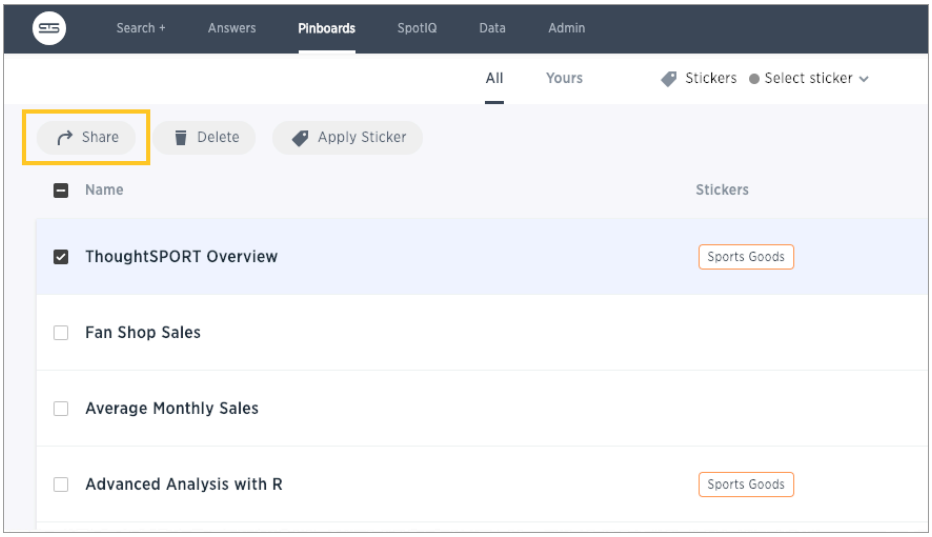
When you share a pinboard what you are really sharing is a live link to the pinboard, when you click **Share with…**. So whenever someone else views it, they will see the most recently saved version with the most recent data. You do not have to be an administrator or the owner to share saved pinboards. Any user can share them, based on the access levels the user has.

To share a pinboard:

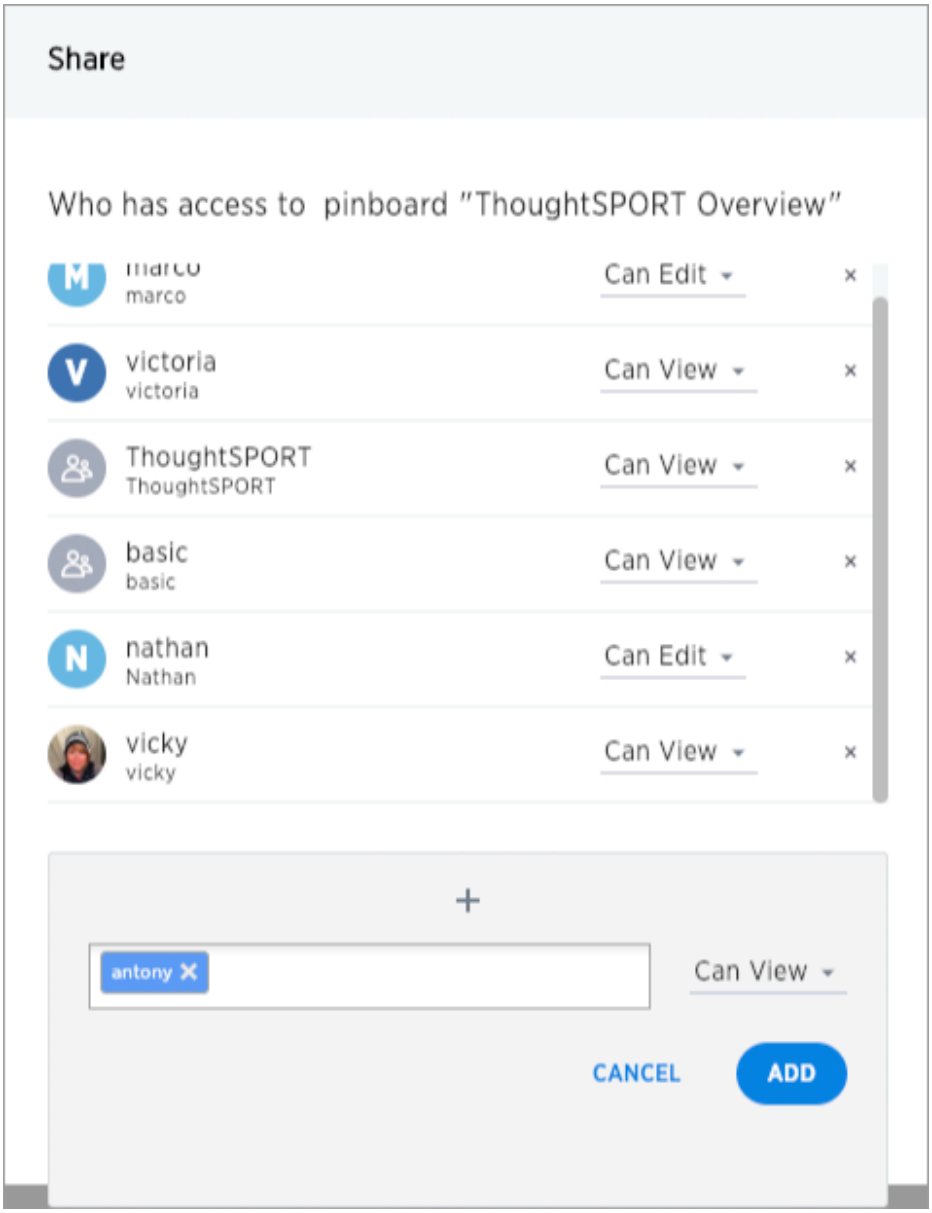
1. Configure it to look as you'll want it to appear when shared.
2. From within a pinboard, click the ellipses icon (3 dots) , and select **Share**.



Alternatively, select the pinboard you want to share from the list of pinboards and click **Share**. (The profile picture or avatar for the owner of each pinboard is shown in the list.)



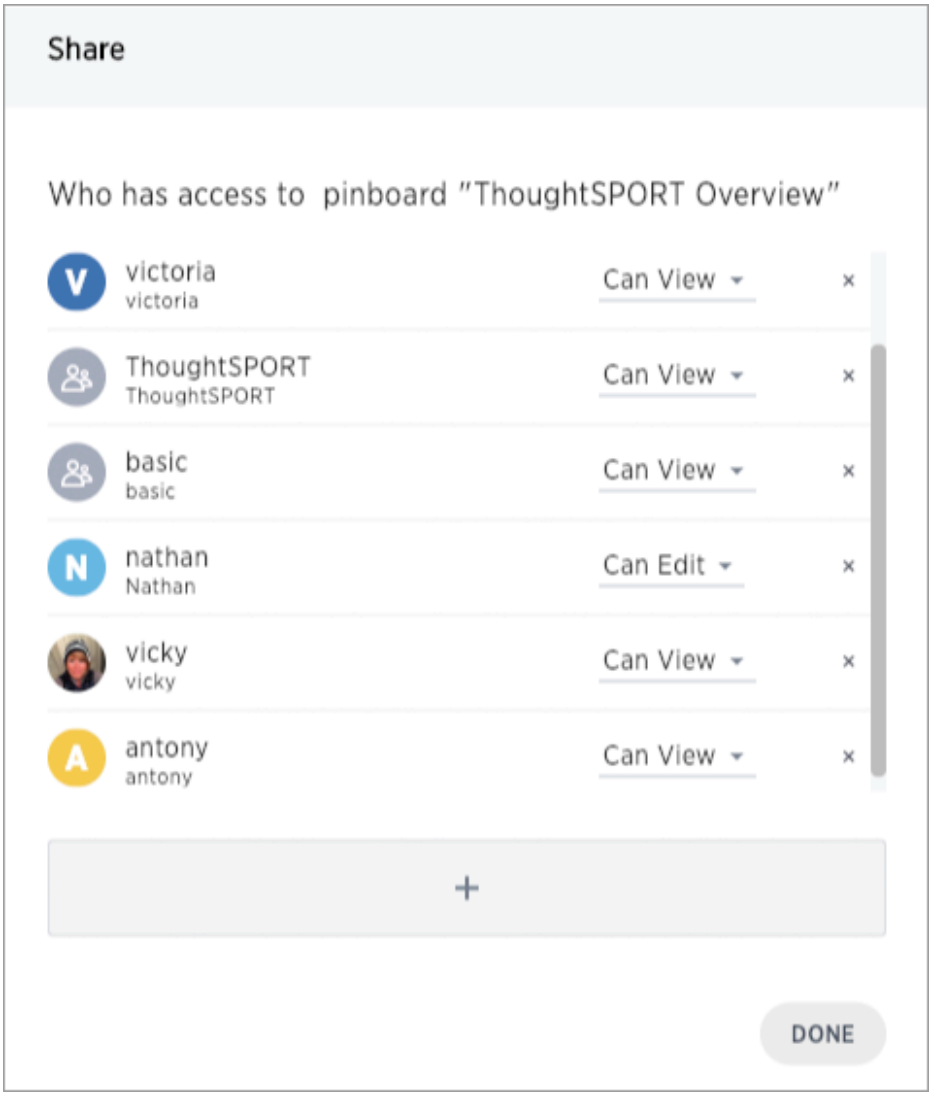
3. Click the plus (+) at the bottom of the Share dialog, and select users or groups with whom you want to share.



4. Configure the level of access by selecting from the dropdown next to each user or group. Available options are based on your own access level. For example, if you have only **View** access, you will not have an option to share as **Edit**. You can select:

    - **Can View** to provide read-only access. If the person doesn't have access to the underlying data, they can only view a shared pinboard. If they change anything on the pinboard, their changes are not saved. In order to persist the changes, the user would need to make a copy of the modified pinboard.
    - **Can Edit** to allow modification. Enables renaming or deleting the shared pinboard. If a person with edit privileges modifies a shared pinboard, their changes will be saved to it.

5. Click **Add** to save your changes, then click **Done**.


Share
Who has access to pinboard "ThoughtSPORT Overview"
victoria
victoria
Can View
ThoughtSPORT
ThoughtSPORT
Can View
basic
basic
Can View
nathan
Nathan
Can Edit
vicky
vicky
Can View
antony
antony
Can View
DONE

Security for SpotIQ functions

SpotIQ is a feature in ThoughtSpot that automatically generates insights into system data. The feature works on all the data in your system, search queries, saved answers, and pinboards. Because SpotIQ uses the same data security model as other data in the system, there is no need to build a new schema or security model to support it.

SpotIQ automatically generates insights into data when a user requests them. These requests can be run immediately or users can schedule regular analysis. SpotIQ requests for insights can put additional load on your system depending on the amount of data being analyzed and how frequently users make use of it.

Users are required to have the **Has Spot IQ privilege** to use this feature. Users without this privilege cannot view insights unless the users that generated the insights add them to a pinboard and share the pinboard with others.

You may want to restrict access to a subset of your users or even to a subset of your managers. To restrict access to this feature:

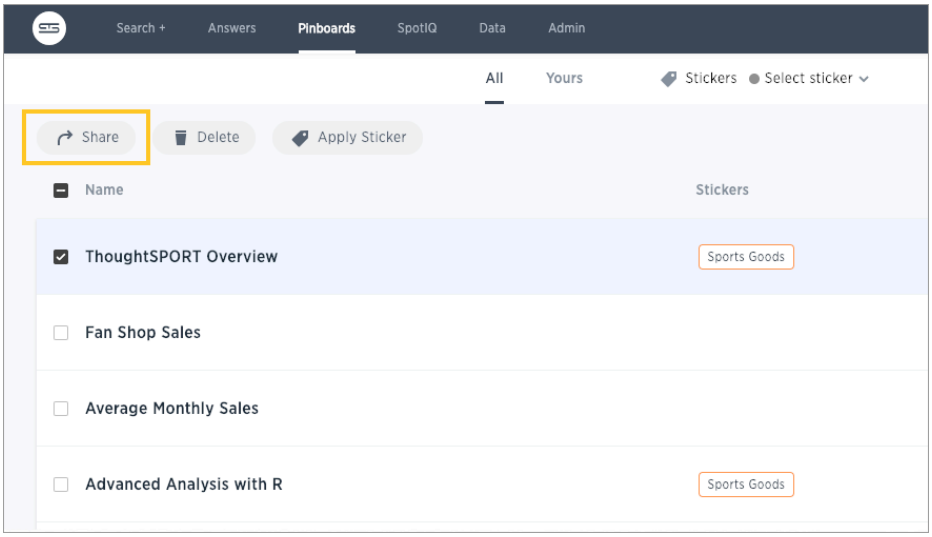
1. Create a group called **SpotIQUsers**.
2. Grant this group **Has Spot IQ privilege**.
3. Enable users to this group.

Revoke access (unshare)

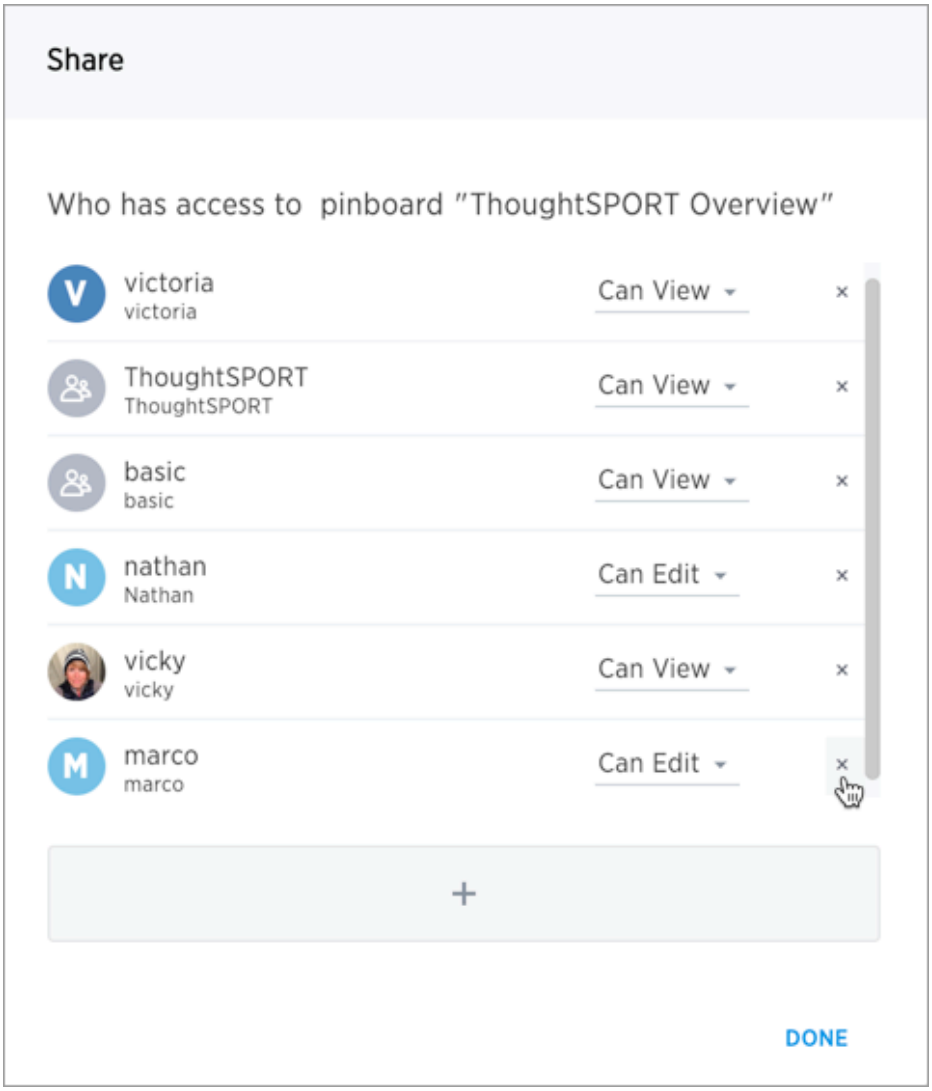
You may need to revoke access to an object (table, worksheet, or pinboard) that you have previously shared. Unsharing an object is very similar to sharing it.

To unshare one or more objects:

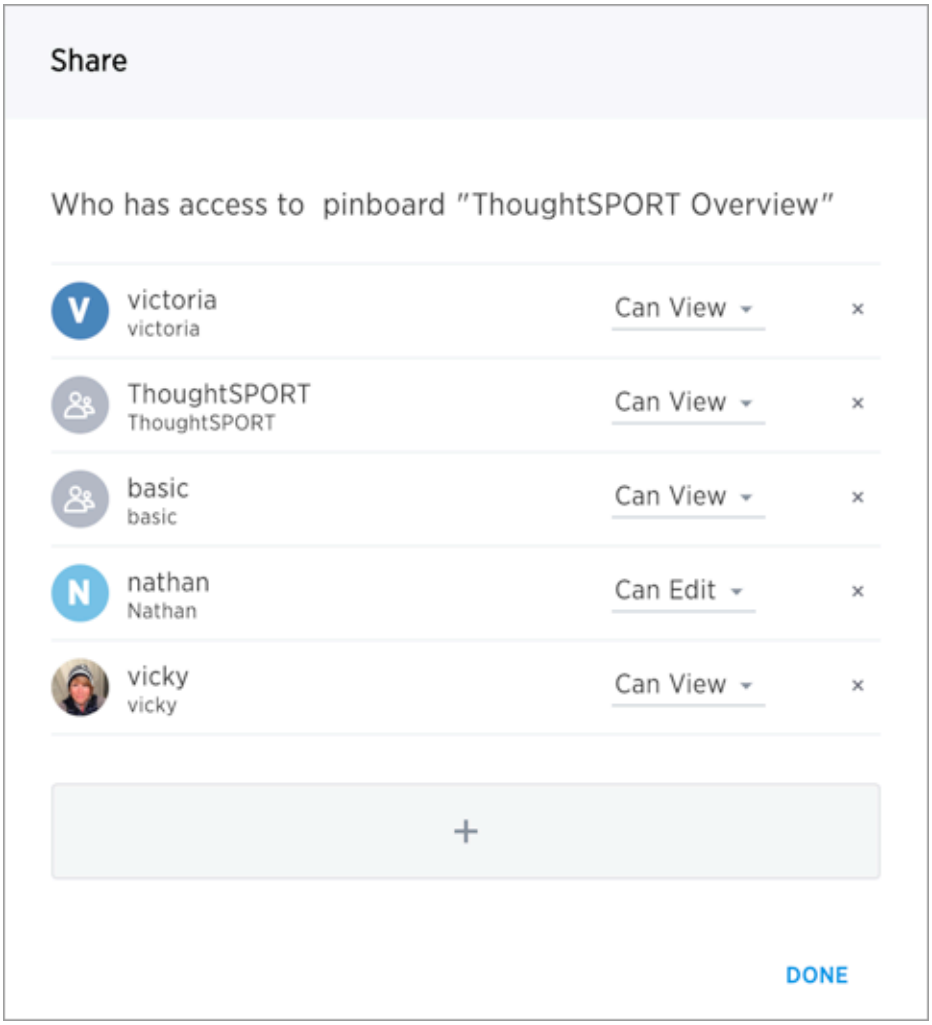
1. Go to the area where the object(s) you want to unshare is located. From the top menu bar:
    - If the object is a table or worksheet, click **Data**.
    - If the object is a pinboard, click **Pinboards**.
    - If the object is an answer, click **Answers**.
2. Find the object(s) in the list, and check the corresponding box(es).
3. Click the **Share** icon.



4. Click the **X** next to the users and groups that you want to remove from sharing.



5. Click **Save**, and then click **Done**.


Share
Who has access to pinboard "ThoughtSPORT Overview"
victoria
victoria
Can View
ThoughtSPORT
ThoughtSPORT
Can View
basic
basic
Can View
nathan
Nathan
Can Edit
vicky
vicky
Can View
DONE

About row level security (RLS)

Summary: Using row level security, you can restrict data that appears in search results and pinboards by group.

Row level security (RLS) allows you to restrict a group's access to table row data. You do this by creating a *rule* that associates a filter with a group. When a group member searches, views an answer, or otherwise works with data, ThoughtSpot evaluates the rules and prevents the display of the restricted data. Users see only the data they are permitted to see.

How does RLS impact user interactions?

The security rules apply to objects shared with users individually or via groups they are a member of. The rules restrict the visible data when users:

- view a table
- view a worksheet derived from the table
- search for data in the worksheet or table
- view answers from restricted data &endash; either that they've created or that were shared with them
- interact with pinboards from restricted data &endash; either that they've created or that were shared with them

Search suggestions also fall under row-level security. If a user would not have access to the row data, then values from the row do not appear in **Search** suggestions.

Why use RLS?

RLS allows you to set up flexible rules that are self-maintaining. An RLS configuration can handle thousands of groups. There are several reasons you might want to use row level security:

Reason	Example
Hide sensitive data from groups who should not see it.	In a report with customer details, hide potential customers (those who have not yet completed their purchase) from everyone except the sales group.
Filter tables to reduce their size, so that only the relevant data is visible.	Reduce the number of rows that appear in a very large table of baseball players, so that players who are no longer active are not shown except to historians.
Enable creation of a single pinboard or visualization, which can display different data depending on the group who is accessing it.	Create one sales pinboard that shows only the sales in the region of the person who views it. This effectively creates a personalized pinboard, depending on the viewer's region.

Related information

- To continue learning about RLS, see How rule-based RLS works.
- **Search** suggestions relies on compile indices to present suggestions to users from your data. See Manage suggestion indexing to learn how to configure suggestions.

How rule-based RLS works

Summary: Use rule-based RLS to restrict a group's access to data. Users see only accessible row data.

Row level security works at the group level and is configured on tables. A table's RLS rules also apply to any objects with data from that table. So, searches, answers, worksheets, and pinboards that rely on a table's data fall under RLS rules.

Worksheet queries and RLS

You cannot set RLS rules on worksheets, only on tables. However, administrators can disable RLS on worksheets that are derived from tables with RLS rules. Once RLS rules are disabled, users with access to the worksheet can see all its data.

By default, worksheet queries only take into account RLS rules on tables whose columns appear in the query. Other related tables that may underly the worksheet are ignored. This means that not all RLS rules on underlying tables are applied when a user queries a worksheet.

You can configure a stricter application of RLS rules to take into account RLS rules from all the tables underlying the worksheet. This is recommended if you have key dimension tables that worksheets rely on but that are not necessarily regularly accessed through query. To do this, contact ThoughtSpot Customer Support.

Privileges that allow users to set, or be exempt from, RLS

Users in the **Administrators** group or with the **Has administration privilege** have full access to everything in the system. As a result:

- Row level security does not apply to them.
- They can create, edit, and delete RLS rules.
- They can also disable RLS rules on individual worksheets.

If your installation has enabled the **Can Administer and Bypass RLS** privilege, administrators can also grant **Can Administer and Bypass RLS** to groups. Members of groups with **Can Administer and Bypass RLS**:

- Are exempt from row-level security (RLS) rules.
- Can add/edit/delete existing RLS rules.
- Can check or uncheck Bypass RLS on a worksheet.

This behavior is true regardless of whether the privilege is from a direct group membership or indirect (through a group hierarchy).

Examples of RLS rules

An RLS rule evaluates against two system variables:

Function	Description	Examples
`ts_groups`	Returns a list of all the groups the current logged in user belongs to. For any row, if the expression evaluates to true for any of the groups, the user can see that row.	`ts_groups = 'east'`
`ts_username`	Returns the user with the matching neame.	`ts_username != 'mark'`

ThoughtSpot filters a table's rows by evaluating a rule against the authenticated user.

A rule is an expression that returns a boolean, `TRUE` or `FALSE`. If the rule evaluates to `TRUE`, a user can see that row. If the rule evaluates to `FALSE` for the user, then the user cannot view the data and instead they see the message `No data to display`.

Rule expression can be implicit or explicit. And rules may or may not contain logic. A simple implicit RLS rule has the format:

`COLUMN_FILTER` = **ts_groups**

An example of an explicit rule that contains logic would be:

```
if ( COLUMN_FILTER ) then true else false
```

Rules can also reference tables other than the table you are securing.

Consider a simple RLS rule example. Your company has `vendor-purchase` table such as:

DATE	VENDOR	AMOUNT
12/11/39..	zendesk	116.00
12/11/39..	getquik com ca	289.70
12/11/39..	ikea	113.91
12/11/39..	costco	274.43
12/11/39..	waiters wheels pa	66.52
12/11/39..	waiters whee	76.49
12/11/39..	chipotle	175.33

You want to give your vendors the ability to see trends in company purchases. You give vendor personnel access to ThoughtSpot *and* add them to self-titled vendor groups. So, all users from the Starbucks vendor are in the `Starbucks` group and all users from `round table` are in the `Round Table` group. Then, you set a **Row security** on the `vendor-purchase` table as follows:

```
VENDOR = ts_groups
```

Only users in `Starbucks` group see `starbucks` data and so forth. Rules ignore case inconsistencies and spaces are evaluated so `round table` in the data matches the `Round table` group but not a group named `RoundTable`.

Rules can be simple or they can incorporate logic such as `if/then` rules. For example, vendors should see their own data but your accounts payable group needs to see all the vendor data:

```
VENDOR = ts_groups or 'Accounts Payable' = ts_groups
```

This rule continues to work as you add data from new vendor or team members to `Accounts Payable`. In this way, a well-written rule is *self maintaining*, meaning you don't have to revisit the rule as your system changes.

You can also create rules that reference tables other than the table you are securing. For example, if you have a `sales` table and `store` dimension table, you can use attributes from the `store` table to secure the `sales` table.

Multiple rules and multiple group membership

You can define multiple rules on table. In this case, ThoughtSpot treats the rules as additive. That is, they are applied using an `OR` operator. If any of the rules evaluate to `true` for a user on a row, that row's data is visible.

If a user is a member of multiple groups, the user can see all the rows that are visible to all of their groups. The most permissive policy is used.

Members of groups with **Can Administer and Bypass RLS** are exempt from row-level security (RLS) rules. This is true regardless of whether the group membership is direct or indirect (through a group hierarchy).

Best practices for using Rule-Based Row Level Security

Use these best practices for Rule-Based Row Level Security:

- Use **Share** as the first level of data access.

  Non-administrative users and groups have no way to access any data without first having it shared with them. So, only share what you need.

  When you share, share worksheets. This is a general best practice. Worksheets simplify the data environment for end users; they only need to choose among a few sources, rather than many tables. Also, one worksheet can also combine data from several tables.

- Set row level security wherever you want to keep data secure.

  It is always a possible that a particular search only includes data from a single table, and a user will see something they shouldn't. So, protect your data by setting row level security wherever you want to keep data secure.

- Explicitly grant access for users that should see all rows.

  As soon as you define a rule on a table for one group, you prevent access by all others outside of that group hierarchy. Subsequent rules should specifically add groups that need access.

- Keep in mind that multiple rules on a table are additive with `or`.

  If you are concerned with security, start with very limited access. Then, expand the access as needed.

- Keep rules simple.

Complex rules can impact the system performance. So, err on the side of simple rules rather than complex rules with a lot of logic.

Related information

- To learn the procedure you follow for setting a rule, Set RLS rules
- For a list of operators and functions you can use to build RLS rules see Row level security rules reference.
- For information on bypassing rules on a worksheet, see Change inclusion, join, or RLS for a worksheet.

Set row level security rules

Summary: Explains the process for setting RLS rules.

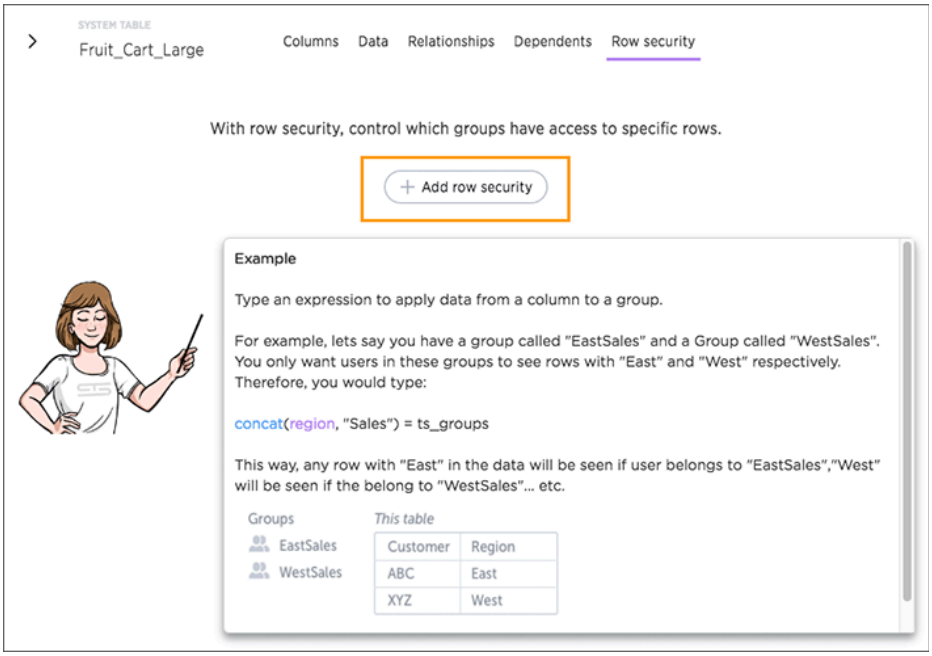
When rule-based row level security (RLS) is set, it prevents users from seeing data they shouldn't in tables and the objects derived from them. You must have administrative rights on ThoughtSpot to set RLS rules.

Before you create a rule, make sure you have read How rule-based RLS works.

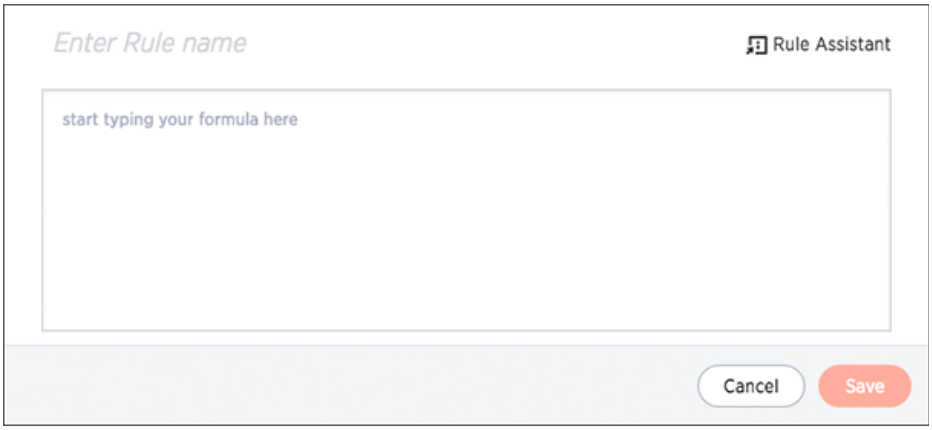
Create a rule on a table

You can set RLS rules *only* on tables. To set up rule-based row level security, do the following:

1. Click **Data**, and double-click on a table.
2. Click **Row security**.
3. Click **+ Add row security**.



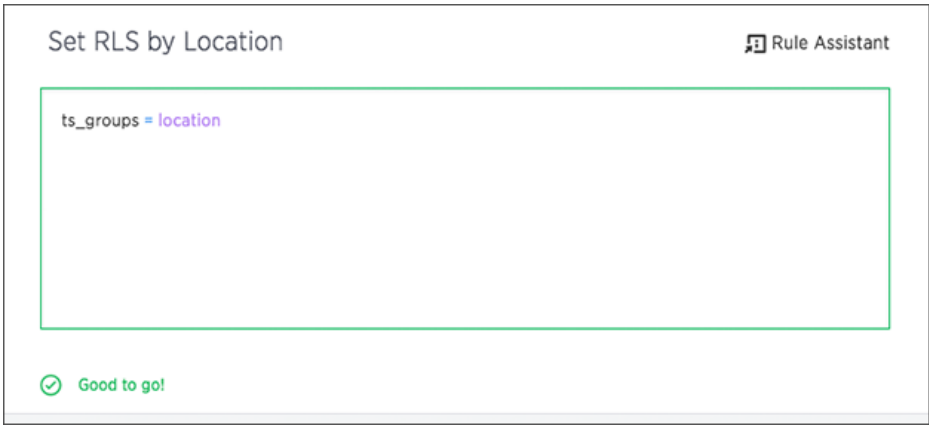
The system displays the Rule Builder.



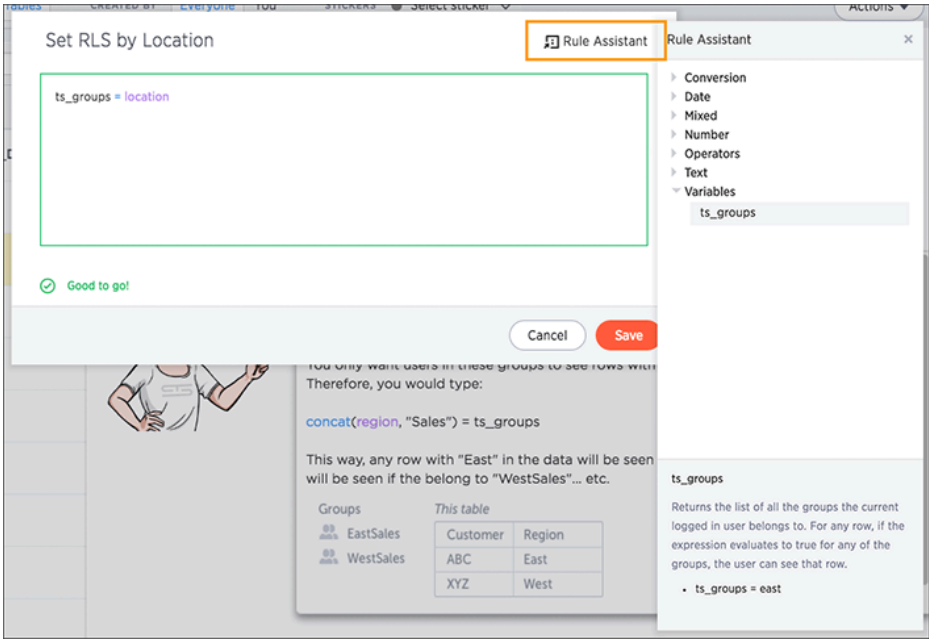
You define row level security by creating an expression that gets evaluated for every row and group combination. This powerful feature can be used with up to thousands of groups.

4. Open the Rule Builder.
5. Give your rule a name.
6. Enter an expression for your rule.

   The rule gets evaluated against an authenticated user for every row and group combination. If the rule evaluates to true, the user can't see that row's data. Use the variable **ts_groups** to refer to the group name.



You can see a list of available operators by clicking on **Rule Assistant**.



As you type, ThoughtSpot suggests formula syntax, variables, and column names. If you can't remember the exact column name or variable you want to use, the suggestions can help.

When your expression is valid, a green indicator appears at the bottom of the Rule Builder.

7. Click **Save**.

The rule you created is listed in the rules. You can edit the rule or add more rules by clicking **+ Add**.

Test your rule with restricted and unrestricted users

To test your rule, log in as users in different groups. Search within the table for data both that you test user can and can't access. Make sure your test users are seeing the appropriate rows.

Related information

- Administrators can bypass the RLS rules set on a the table at the worksheet level. See how to "Change inclusion, join, or RLS for a worksheet in this documentation for more information.
- For a list of operators and functions you can use to build RLS rules see Row level security rules reference.

ThoughtSpot Lifecycle

Summary: This topic covers security processes for the entire lifecycle of a ThoughtSpot deployment from development, release, installation, upgrades, to software patching.

Overview

A ThoughtSpot deployment consists of the following high level software systems:

- Operating System (OS) and software packages installed on the OS
- Third-party software
- ThoughtSpot application services (binaries and configuration)

Operating System

All ThoughtSpot physical appliances, virtual machines (VMs) and public cloud images come pre-installed with CentOS 7. The CentOS distribution of Linux is owned by RedHat and closely tracks versions of RedHat Enterprise Linux (RHEL).

ThoughtSpot uses the minimal install of CentOS 7 with the addition of a few software packages (e.g. Python) needed for ThoughtSpot operations. The most notable change to the installation is to the Linux kernel, which is sourced from the current long term stable kernel version instead of the default included in CentOS 7 (kernel-lt package). To list all the installed packages, see Checking Package Versions below.

Third-Party Software (Middleware)

Third party software used includes Java, Boost C++ libraries, Google protocol buffers, etc. These are software components necessary for operation of the ThoughtSpot application. ThoughtSpot only uses software licensed for distribution.

Development and Release Process

ThoughtSpot releases its software as a tarball containing all the ThoughtSpot application (binaries and configuration), third-party software, and an operating system image. Installation or update using this release tarball on appliances, VMs, or cloud instances updates each of these components.

Operating System

Building the operating system image including software packages is a multi-step process:

1. Begin with the set of packages in the base OS image and our added packages.
2. Configure all installation to only use official public RedHat repositories.
3. For each package, install the current stable version including any security patches.

4. Bring up the image on all supported platforms for stability and performance testing along with the ThoughtSpot application stack. Success criteria: no OS impact on stability or performance.
5. Scan the Operating System and ThoughtSpot application stack using Qualys scans with additional modules enabled: Vulnerability Management, Web App Scanning.
6. Review all vulnerabilities found. Success criteria is zero severity 4+ vulnerabilities.
7. Assuming all above testing and exit criteria are met, the OS image is considered qualified.

Third-Party Software

Third-party software is periodically sourced from the upstream distribution of each software component. Unlike OS and ThoughtSpot application, this changes less frequently and on an as needed basis, when any new security vulnerability or stability issue is discovered in the library. The list of all third-party software as well as licensing details are here.

ThoughtSpot Application

ThoughtSpot follows industry standard best practices for writing robust software. Every code change is reviewed by at least one engineer. Our engineering team consists of senior engineers from Enterprise software and web companies.

ThoughtSpot uses a small number of proven programming languages powering some of the largest enterprises in the world. ThoughtSpot tracks stability, performance, and reliability of our software and services aggressively. The ThoughtSpot platform is trusted by dozens of global F2000 organizations.

Protection of Source Code

Source code is private and not shared publicly, e.g. all distribution to customers is in binary or minified format to discourage reverse engineering.

Automated Tools

We use automated tools and infrastructure like Jenkins, Kubernetes, AWS, partnering with the teams behind these systems so as to adopt best practices. For example, all our automation runs through Jenkins, which is managed by CloudBees (the company behind Jenkins) using an enterprise license with regular security patching, and so on. We upgrade our automation tools regularly.

Independent Testing

Independent testing is done outside of the product team by pre sales and post sales before promoting to production. Some areas are tested by third party testers.

Security Hardening

Starting 4.5.1.5, we have also taken specific steps to incorporate most of CIS standard recommendations towards hardening.

Installation and Upgrade Process

ThoughtSpot is installed or updated from a release tarball which contains the ThoughtSpot application (binaries and configuration), third-party software, and Operating System image.

Operating System Image Installation

Installing ThoughtSpot on any node (VM, cloud instance, appliance) automatically updates the operating system and required packages on the node. No Internet or repository access is required for this, the update is applied directly from the release tarball.

Specifically, all nodes running ThoughtSpot are required to have two root partitions on their boot drive of which one of them is booted from at any given time. During installation or update, the Operating System image contained in the release tarball is copied into the second currently-unused root partition and the system switches to it through a reboot.

Checking OS Package Versions

The following command run from any ThoughtSpot node will indicate versions of all installed packages:

```
rpm -qa
```

Upgrades

ThoughtSpot patches the Operating System at the time of upgrades. The exact same process used during installation is also applied during upgrades. The previous OS image on a node gets replaced by the new image carried in the release tarball.

Only some releases may patch the Operating System, not all. Typically, all major and minor releases (e.g. 4.4, 4.5, 4.5.1, 5.0) upgrade OS patches, whereas only some patch releases (e.g. 4.4.1.4) contain OS patches.

Distributed Clusters and Failure Handling

On distributed clusters, individual nodes receive the OS image from the release tarball individually. Initially, the new image is deployed on a single node only. When that node is deemed healthy following the update and a rich set of tests, the image is made available to remaining nodes in the cluster.

If a node fails to patch, then ThoughtSpot support will modify the upgrade workflow to either retry the patching or skip and exclude the node.

Third-Party Software

Installation or upgrade of ThoughtSpot deployments automatically upgrades all third-party software to the version included in the release tarball.

Security Scanning and Patching Process

The ThoughtSpot Security team continuously scans security bulletins for new vulnerabilities discovered in included OS packages (e.g., Linux Kernel, libc) and third party software (e.g., Java). Additionally, weekly scans are done for all release branches using Qualys with the following additional modules enabled: Vulnerability Management, Web App Scanning. The security scans discover vulnerabilities at all layers: OS, third-party software, as well as ThoughtSpot application binaries and configuration. Additionally, ThoughtSpot periodically scans all source code for third-party software as well as ThoughtSpot's proprietary code base for vulnerabilities or unsafe usage using SourceClear.

Once a critical new vulnerability is found (severity 4 or 5), ThoughtSpot includes the corresponding patch in the next patch release for all supported release branches. Consult ThoughtSpot documentation or support to find out if you are on an active or supported release branch.

Once a new patch release with a critical security vulnerability is available, customers are encouraged to upgrade their deployment quickly.

Latency

We recommend customers to wait for the next regular release for receiving security patches. However, should a critical vulnerability be discovered in the interim, ThoughtSpot can push out a new patch release containing the required patches, if available upstream.

ThoughtSpot targets a three week or less cadence for generating patch releases for all supported release branches. Timeline for the new release and patching depends on availability of the patch upstream (e.g., not all vulnerabilities in Linux are immediately fixed) and qualification (ThoughtSpot qualifies each build on each supported cloud and on-prem platform). If a fix is unavailable upstream at the moment, customers and ThoughtSpot support can work together to identify potential workarounds.

Storage Security

Encryption at Rest

- On-prem: Not supported yet
- Cloud: Supported on AWS, GCP, Azure

Secure Erase

Current erase guide

System administration

System administration includes applying upgrades, backing up and restoring the cluster, snapshotting, and adding or removing nodes.

Administration tools

Use these tools to perform administrative actions:

- tscli: an administrative command line interface.
- tsload: a command for loading data directly into the database.
- TQL: a command line SQL interface to interact with databases.

Send logs when reporting problems

You can generate a log bundle which you can then send to ThoughtSpot Support or you can send logs direct to your administrator.

Generate log bundle

Before you can send a log bundle to ThoughtSpot Support, you must Connect to the ThoughtSpot Support file server. This is a one-time setup operation.

To generate a log bundle:

1. Log in to the Linux shell using SSH.
2. Issue the command to generate the log bundle:

```
tscli callhome generate-bundle
   --d <directory> --since <num_of_daysd>
```

Note: Don't forget to include `d` after your specified number of days. For example, `30d`.

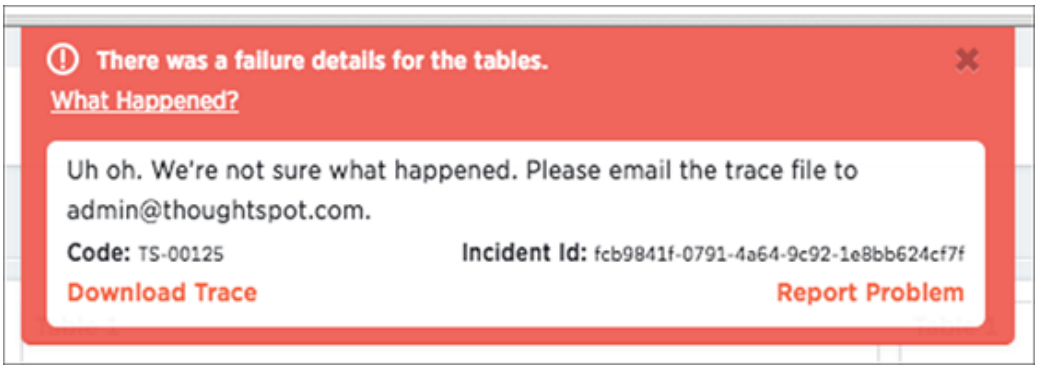
3. Change directories to the directory where you wrote the log bundle.
4. Issue the command to send the log bundle to ThoughtSpot Support:

```
tscli fileserver upload
   --file_name <file>
   --server_dir_path <path>
```

Send a log to the administrator

Alternately, you can easily send log files directly to your administrator with a single click. When ThoughtSpot encounters a problem, a red bar displays in the browser with an error message. You can use the **Report Problem** option to complete this task.

Click **Report Problem** in the bottom right corner of the error message.



The logs will be sent to your administrator as an email attachment from your email account. Your administrator then has the option to followup with ThoughtSpot, if necessary.

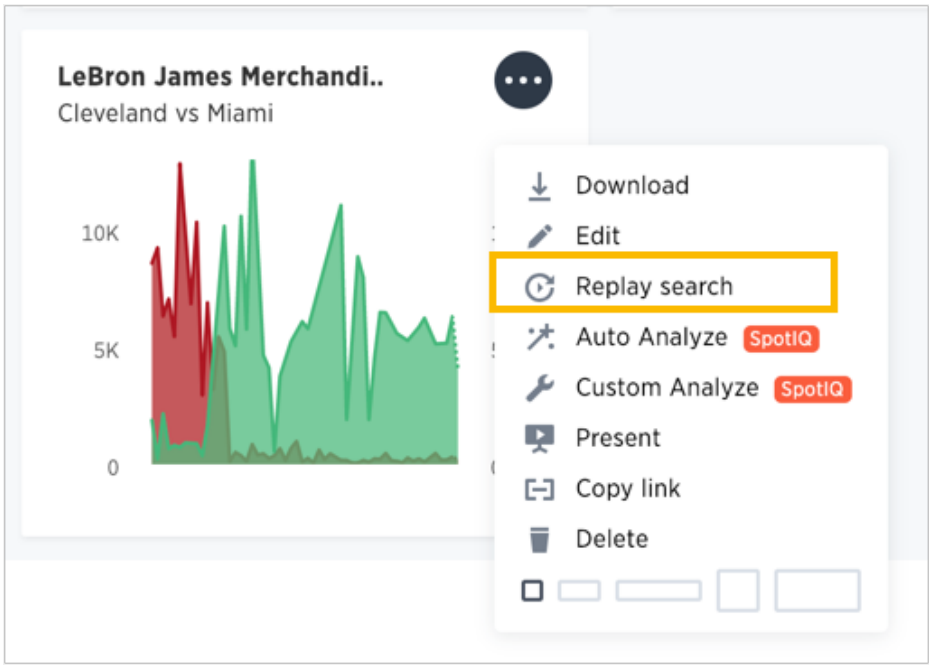
Set up recording for Replay Search

Summary: You can use the recording to create training for your users on how to search your own data.

Recording a search replay requires administrator privileges and a Firefox browser. You must override some of your browser security settings in order to use the ThoughtSpot application to make the recording. This is a one time setup operation. If you do not wish to do this, you can replay the search and record it using QuickTime, Camtasia, or another screen cam recording tool.

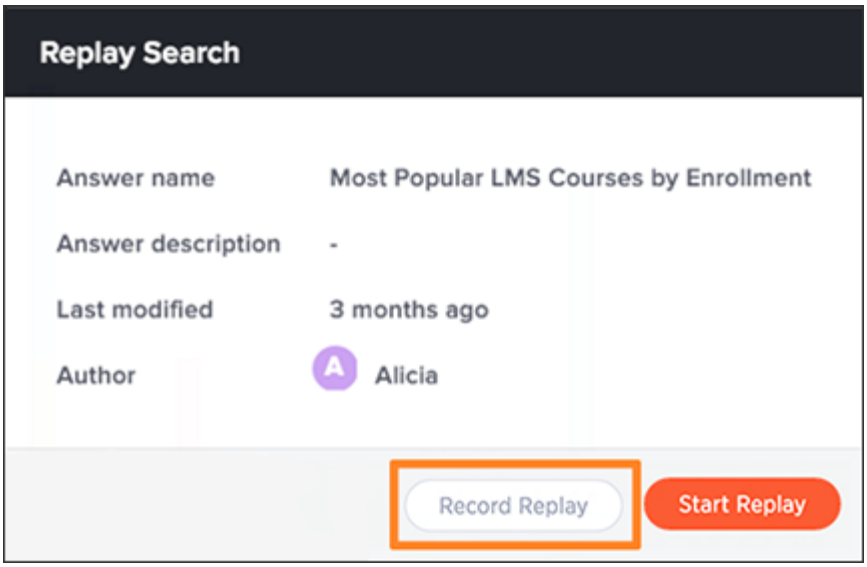
To record a search replay using ThoughtSpot:

1. While viewing a chart or table in ThoughtSpot, click the **Replay Search** icon.



2. Click the **Record Replay** button.

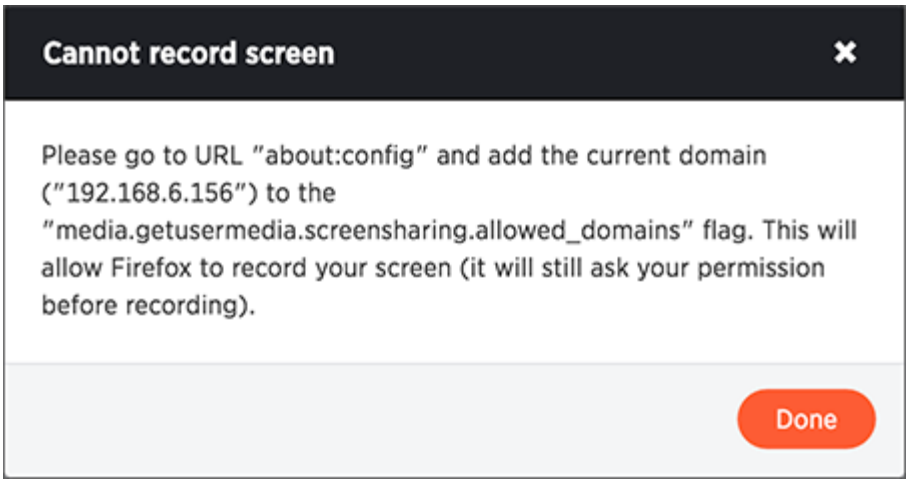
   If you do not see the button, you must log in as a user with administrator privileges.



A message will display, showing a URL and a domain or an IP address.

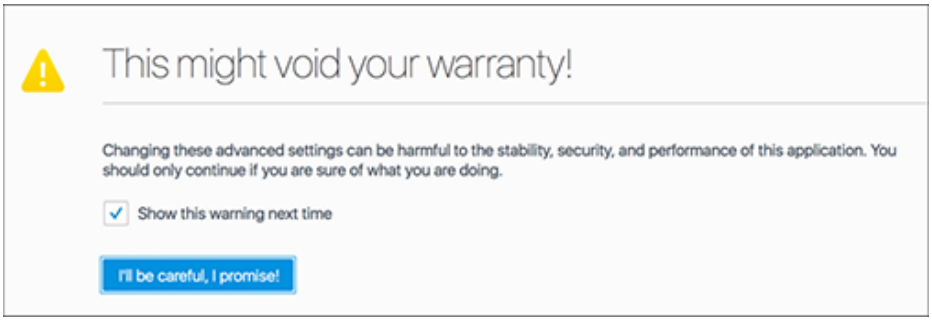
3. Make note of both of these items.
4. Open a new browser tab and go to the URL shown in the message (for example, “about:config”).

   Depending on which browser and version you are using, you may need to access the browser configurations through a menu or by typing in a different URL. Check your own browser help section for information on how to access the browser configuration settings, if necessary.



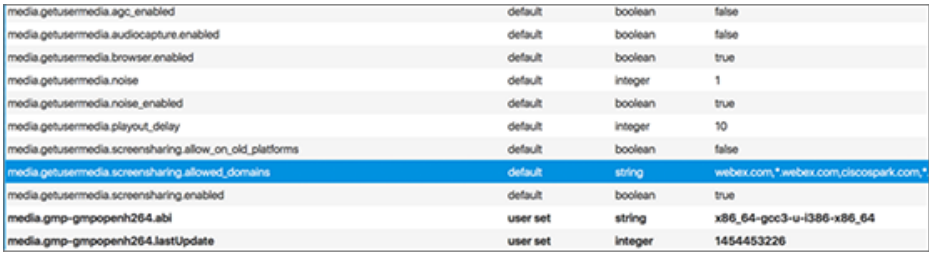
   You may see a message warning that you are about to override the browser settings.

5. If you trust yourself, click “I’ll be careful, I promise!”.



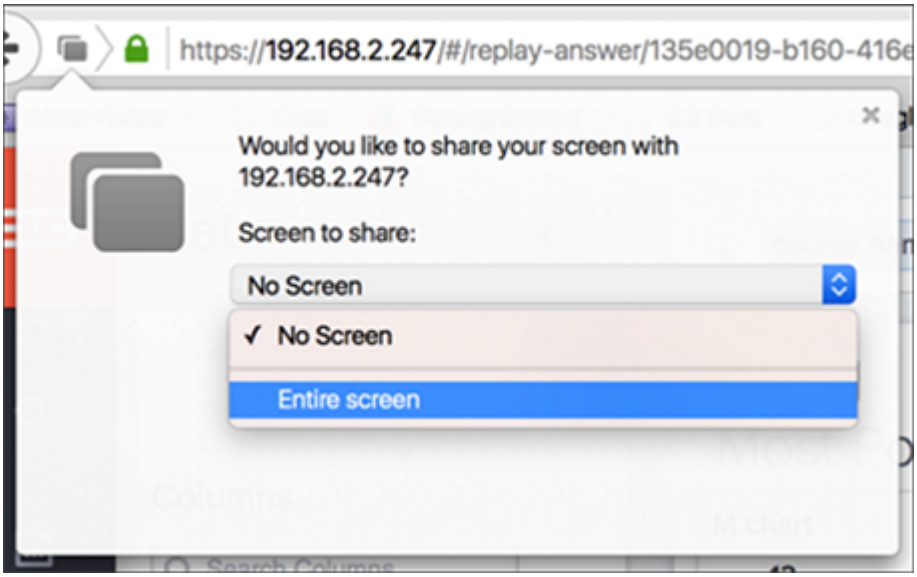
6. Find the setting for **media.getusermedia.screensharing.allowed_domains**, and add the domain used by ThoughtSpot.

   This domain will be the same one you made note of from the **Cannot record screen** message.

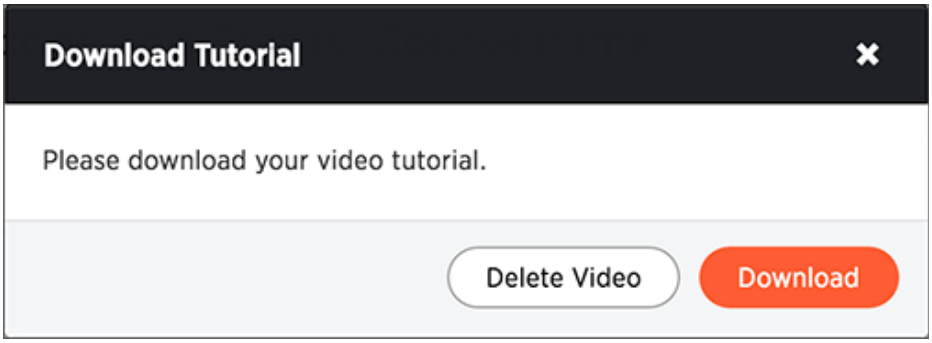


media.getusermedia.agc_enabled	default	boolean	false
media.getusermedia.audiocapture.enabled	default	boolean	false
media.getusermedia.browser.enabled	default	boolean	true
media.getusermedia.noise	default	integer	1
media.getusermedia.noise_enabled	default	boolean	true
media.getusermedia.playout_delay	default	integer	10
media.getusermedia.screensharing.allow_on_old_platforms	default	boolean	false
media.getusermedia.screensharing.allowed_domains	default	string	webex.com,*.webex.com,ciscospark.com,*
media.getusermedia.screensharing.enabled	default	boolean	true
media.gmp-gmpopenh264.abi	**user set**	**string**	**x86_64-gcc3-u-i386-x86_64**
media.gmp-gmpopenh264.lastUpdate	**user set**	**integer**	**1454453226**

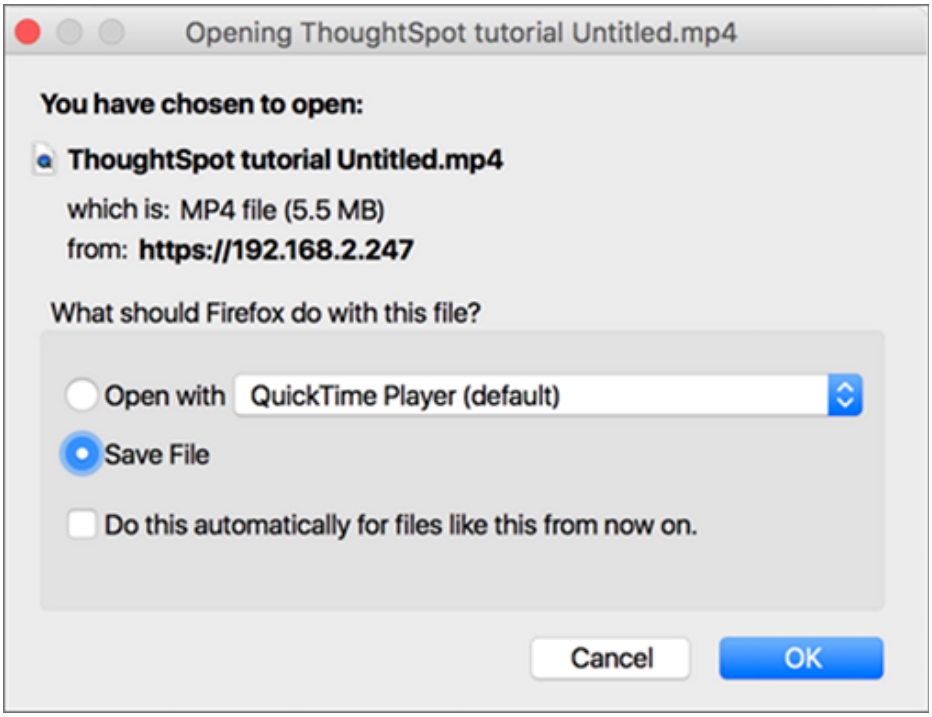
7. If you see a message asking if you'd like to share your screen with the IP address or domain name of ThoughtSpot, select **Entire screen**.



8. When the search replay has been recorded, you'll see a confirmation. Select **Download**.



9. Save the recording on your computer by selecting **Save File** and clicking **OK**.

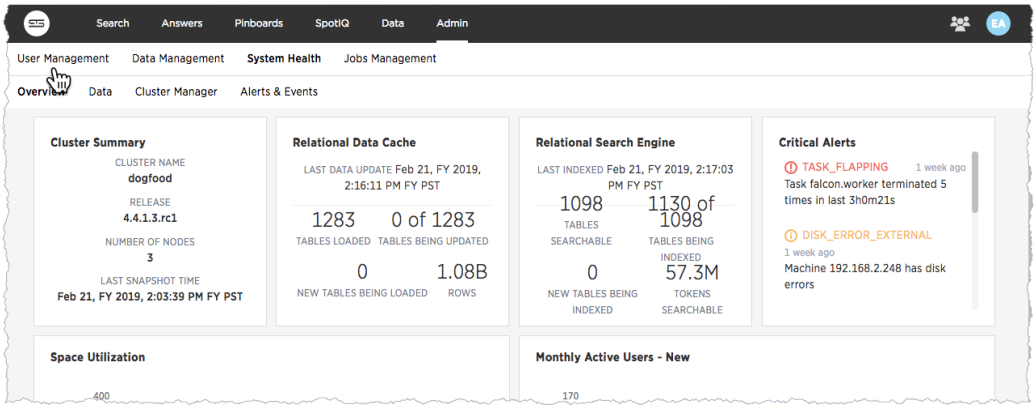


Introduction to monitoring

System monitoring tools in ThoughtSpot include an **Admin** > **System Health** page and system logs. Additionally, ThoughtSpot provides several worksheets and out-of-the-box system monitoring pinboards. From these worksheets, you can create your own custom visualizations and pinboards. This page introduces these features and directs you towards more detailed information.

System Health center

The ThoughtSpot application includes a **System Health** center, for easy monitoring of usage, alerts, events and general cluster health. You view the System Health Center by choosing the **Admin** icon and then selecting **System Health**.



Only users with administrative privileges can view the **System Health** center. However, administrative users can present to others the information that displays in the **System Health** center.

Administrators can also create their own, custom boards that reflect system data in ways that are meaningful to specific departments or groups. For more information, see the following documentation:

- Health Overview board
- Data board
- Cluster Manager board
- Alerts and Events board
- System worksheets
- System pinboards

Much of the data presented by these boards is also available through `tscli` commands.

Log files

Many of the administration commands output logging information to log files. The logs get written into the fixed directory `/export/logs`, with a sub-directory for each subsystem. The individual log directories are the following:

- `/export/logs/orion`
- `/export/logs/oreo`
- `/export/logs/hadoop`
- `/export/logs/zookeeper`

You can also view additional topics that also touch on log files throughout the documentation.

System monitoring notifications

You can configure ThoughtSpot to send emails to addresses you specify with monitoring reports and a cluster heartbeat. Follow these steps to Set up monitoring.

Overview board

The **Overview** pinboard summarizes essential information about your cluster and its users. Choose **Admin** > **System Health** > **Overview** to see this pinboard.

Understand system boards and pinboards

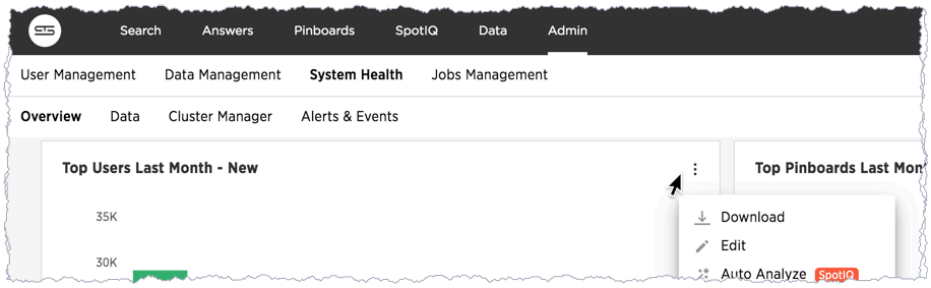
The **Overview** page includes system panels and standard ThoughtSpot *answers*. The system charts are generated in real time and rely on internal system data. The answers rely on underlying system worksheets which are available to ThoughtSpot administrators. The information in these worksheets is updated hourly from internal tables that collect monitoring statistics.

Each answer has a menu. You can present or copy the links to the system charts. The answers have a subset of the ThoughtSpot answer menu. You can use the menu to do additional actions such as download the answer or present information about your ThoughtSpot cluster. While you can interact with and change the search, you cannot save changes to the underlying query.

You can also interact with the answers, drilling into them to explore the detail as with any other pinboard answer.

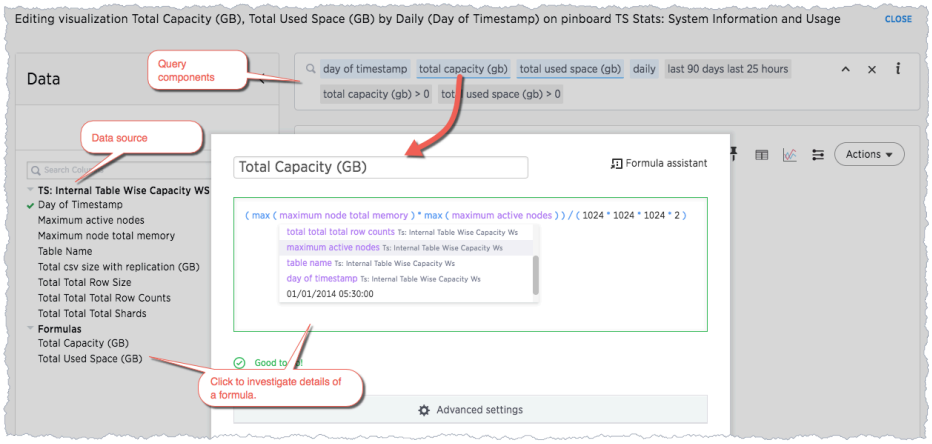
To find out how a particular answer was created, do the following:

1. Select **Edit** from the panel menu.



   This displays a **Search** bar.

2. Investigate the components of the search as you would normally.



Cluster Summary

This system panel contains basic information about your cluster. The **NUMBER OF NODES** is the number of installed nodes. This doesn't reflect the active nodes which may be more or less.

This summary includes the **LAST SNAPSHOT TIME** it reflects whether regular snapshots of your cluster are collected. This value should update regularly in real time. If you do not see it change or empty, you should check your cluster snapshot policy using the `tscli` command:

```
$ tscli snapshot-policy show
schedule {
  period {
    number: 1
    unit: HOUR
  }
  retention_policy {
    bucket {
      time {
        number: 1
        unit: HOUR
      }
      capacity: 3
    }
    bucket {
      time {
        number: 4
        unit: HOUR
      }
      capacity: 2
    }
  }
  offset_minutes_from_sunday_midnight: 0
}
enabled: false
```

You can see this policy is disabled, which is a problem. Production clusters should enable the default snapshot policy. When you show or enable the snapshot policy, you'll see your `tscli` command reflected in the **Configuration Events** panel on this same page.

Relational Data Cache

This section reports real-time information about tables in your cluster. Worksheet data is not included.

Value	Description
TABLES LOADED	Number of currently loaded tables.
TABLES BEING UPDATED	Number of table loads in-progress.
NEW TABLES BEING LOADED	Number of tables being loaded for the first time.
ROWS	Number of rows combined across all tables in ThoughtSpot.

Relational Search Engine

Value	Description
TABLES SEARCHABLE	Tables that are indexed and can be searched.
TABLES BEING INDEXED	Total of in-progress table indexing.
NEW TABLES BEING INDEXED	Total of first-time, in-progress table indexing.
TOKENS SEARCHABLE	Number of *tokens* of all table (combined) indexed in ThoughtSpot.

Critical Alerts

Displays critical and warning alerts. This includes when an alert was generated and from which service and machine. Administrators can get a custom report by issuing a `tscli alert list` on the appliance:

```
tscli alert list --since 4w
```

The critical alerts you can encounter in this display are the following:

- `TASK_FLAPPING`

  Msg: `Task {{.Service}}.{{.Task}} terminated {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

  This alert is raised when a task is crashing repeatedly. The service is evaluted across the whole cluster. So, if a service crashes 5 times in a day across all nodes in the cluster, this alert is generated.

- `OREO_TERMINATED`

  Msg: `Oreo terminated on machine {{.Machine}}`

  This alert is raised when the Oreo daemon on a machine terminates due to an error. This typically happens due to an error accessing Zookeeper, HDFS, or a hardware issue.

- `HDFS_DISK_SPACE`

Msg: `HDFS has less than {{.Perc}}% space free`

Raised when a HDFS cluster is low on total available disk space.

- `ZK_INACCESSIBLE`

  Msg: `Zookeeper is not accessible`

  Raised when Zookeeper is inaccessible.

- `PERIODIC_BACKUP_FLAPPING`

  Msg: `Periodic backup failed {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

  This alert is raised when a periodic backup failed repeatedly.

- `PERIODIC_SNAPSHOT_FLAPPING`

  Msg: `Periodic snapshot failed {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

  This alert is raised when periodic snapshot failed repeatedly.

- `APPLICATION_INVALID_STATE_EXTERNAL`

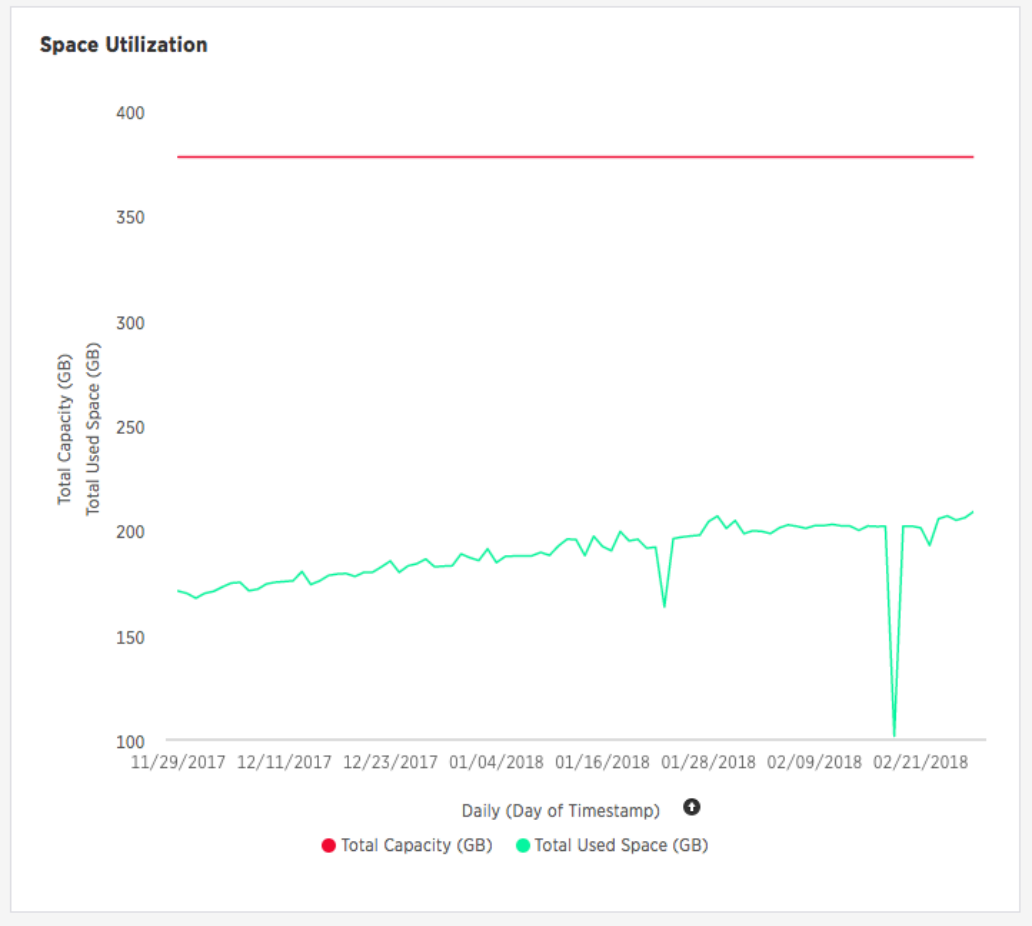
  Msg: `{{.Service}}.{{.Task}} on {{.Machine}} at location {{.Location}}`

  Raised when Application raises invalid state alert.

The possible alert types are `CRITICAL`, `WARNING`, `ERROR`, and `INFO`. For a full reference, see the Alert code reference.

Space Utilization

The **Space Utilization** chart is one of the available charts for you to use when checking the cluster overview. This line chart displays the total capacity and estimated used capacity over time.



The x-axis is by time and the y-axis measures the size in GB. You can zoom in and see daily or hourly utilization data. So, in the **Space Utilization** chart above, the green line shows the amount of capacity in use in the system, while the red line shows the total capacity. An increase in the red line at the end of a time period indicates the addition of extra hardware, resulting in increased capacity.

The query for this answer is the following:

```
day of timestamp
total capacity (gb)
total used space (gb)
daily
last 90 days last 25 hours
total capacity (gb) > 0
total used space (gb) > 0
```

The chart relies on the `TS: Internal Table Wise Capacity WS` worksheet. It tracks total used space, which consists of raw uncompressed data, including replication.

Monthly Active Users

This chart shows the number of active users in the system over the last four months and current month. An active user is defined as a user who has logged in at least once in the defined time interval, in this case months.

The query for this answer is the following:

```
monthly
last 4 months this month
active users
user != {null}
```

This query relies on the `TS: BI server` worksheet.

Monthly Ad-hoc Searches

Number of ad-hoc searches (queries) issued per month. An ad-hoc query is defined as any search or change to a search that builds a new answer (result). An ad-hoc search can also be generated through SpotIQ or another UI/API interaction.

ThoughtSpot considers all of the following as ad-hoc searches (queries):

- User edits tokens (boxed terms) in the search bar.
- User opens an existing saved answer and makes changes to tokens in the search bar.
- User opens an existing saved pinboard, edits a visualization, and makes change to the search tokens.
- Searches initiated by an API call for data with runtime filters

It is not considered a search (query) in this context if a user opens an existing saved aggregated worksheet and makes changes to its underlying query.

The query for this answer is the following:

```
ad-hoc search
user action = 'answer_pinboard_context' 'answer_saved' 'answe
r_unsaved'
monthly
last 4 months this month
```

This answer relies on data from the `TS: BI Server` worksheet.

Monthly Pinboard Views

Number of times a saved pinboard is viewed by a user. These scenarios are considered pinboard views:

- User opens an existing saved pinboard.
- User opens an embedded pinboard from a URL.
- Pinboard data is accessed using the an API.

These scenarios are not considered pinboard views:

- A user opens SpotIQ tab pinboards.
- A user opens admin tab pinboards.
- The system loads a pinboard on the homepage.
- The system loads the 'learn how to use ThoughtSpot' pinboard.

The query underlying this answer is:

```
pinboard views
user action = 'pinboard_embed_view' 'pinboard_tspublic_no_runti
me_filter' 'pinboard_tspublic_runtime_filter' 'pinboard_view'
monthly
last 4 months this month
```

The query uses the `TS: BI Server` data source.

Top Users Last Month

This answer shows the top ThoughtSpot users ranked by number of actions the users performed in the last 30 days. The possible user actions include:

`answer_unsaved`	User makes a change to tokens in the search bar.
`answer_saved`	User opens an existing saved answer and makes changes to tokens in the search bar.
`answer_pinboard_context`	User opens an existing saved pinboard, edits a context viz and makes a change to tokens in the search bar.
`answer_aggregated_worksheet`	User opens an existing saved aggregated worksheet and makes changes to tokens in the search bar.
`answer_upgrade`	Requests made for the sole purpose of upgrade.
`pinboard_view`	User opens an existing saved pinboard.

`pinboard_filter`	User adds, removes or applies values to a pinboard filter.
`pinboard_ad_hoc`	User drills down in a pinboard viz.
`data_chart_config`	Request for new data being generated following a chart config change.
`data_show_underlying_row`	Request to show underlying data for a data row(s).
`data_export`	Request to export data.
`pinboard_tspublic_runtime_filter`	Request to TSPublic/pinboarddata with runtime filters.
`answer_aggregated_worksheet_save`	User updates aggregated worksheet.
`answer_add_new_filter`	User adds a filter using the UI.
`data_show_underlying_viz`	Request to show underlying data for a data row(s).
`answer_view`	User opens an existing, saved answer.
`answer_viz_context_view`	User opens an existing saved pinboard, edits a context viz.
`pinboard_insight_view`	User opens SpotIQ tab pinboards.
`pinboard_admin_view`	User opens admin tab pinboards.
`pinboard_embed_view`	User opens embed pinboard from a URL.
`pinboard_homepage_view`	On loading of homepage pinboard.
`pinboard_learn_view`	On loading learn pinboard.
`pinboard_tspublic_no_runtime_filter`	Request to TSPublic/pinboard data without runtime filters.

The query underlying this answer is:

```
top 10
ranked by user actions
user action != 'invalid'
user != {null}
user
last 30 days today
```

The query uses the `TS: BI Server` data source.

Top Pinboards Last Month

This answer shows the top ThoughtSpot users ranked by number of pinboard actions the user performed in the last 30 days. The possible user actions include:

- User opens an existing saved pinboard (‘pinboard_view’).
- User opens an embedded pinboard from a URL (‘pinboard_embed_view’).
- Pinboard data is accessed using an API (‘pinboard_tspublic_no_runtime_filter’ or ‘pinboard_tspublic_runtime_filter’).

The query underlying this answer is:

```
top 10
ranked by pinboard views
user action = 'pinboard_embed_view' 'pinboard_tspublic_no_runti
me_filter' 'pinboard_tspublic_runtime_filter' 'pinboard_view'
pinboard != {null}
pinboard
last 30 days today
```

The query uses the `TS: BI Server` data source.

Configuration Events

This system answer displays recent events that changed the configuration of the system. This panel displays configuration events related to:

Cluster Configuration	Reports configuration actions from the `tscli` and `tql` commands.
Metadata Management	Events related to metadata such as column names, column visibility, column and data definition, column rank and so forth.
User Management	Events related to creating, updating, or adding new users and groups.

For a more detailed list, including the user that issued a command, you can use the `tscli event list` command. Administrators can `ssh` into the cluster and specify a time period or even a type of command to include.

```
[admin@testsystem ~]$ tscli event list --since 3d
+------------------------------+----------------------
+---------------------------+
|             DATE             |       USER
|           SUMMARY          |
+------------------------------+----------------------
+---------------------------+
| 2018-03-06 11:57:10 -0800 PST |eadmin@thoughtspot.int | User
Management: User      |
|                               |                       | "gues
t_1" updated          |
| 2018-03-06 11:48:10 -0800 PST |admin                  | tscl
i node ls              |
| 2018-03-06 11:17:04 -0800 PST |eadmin@thoughtspot.int | Metad
ata Management:       |
|                               |                       | Metad
ata object "Number of |
|                               |                       | Oppor
tunity                |
|                               |                       | AE an
d Stage" of type      |
...
```

About deprecated boards

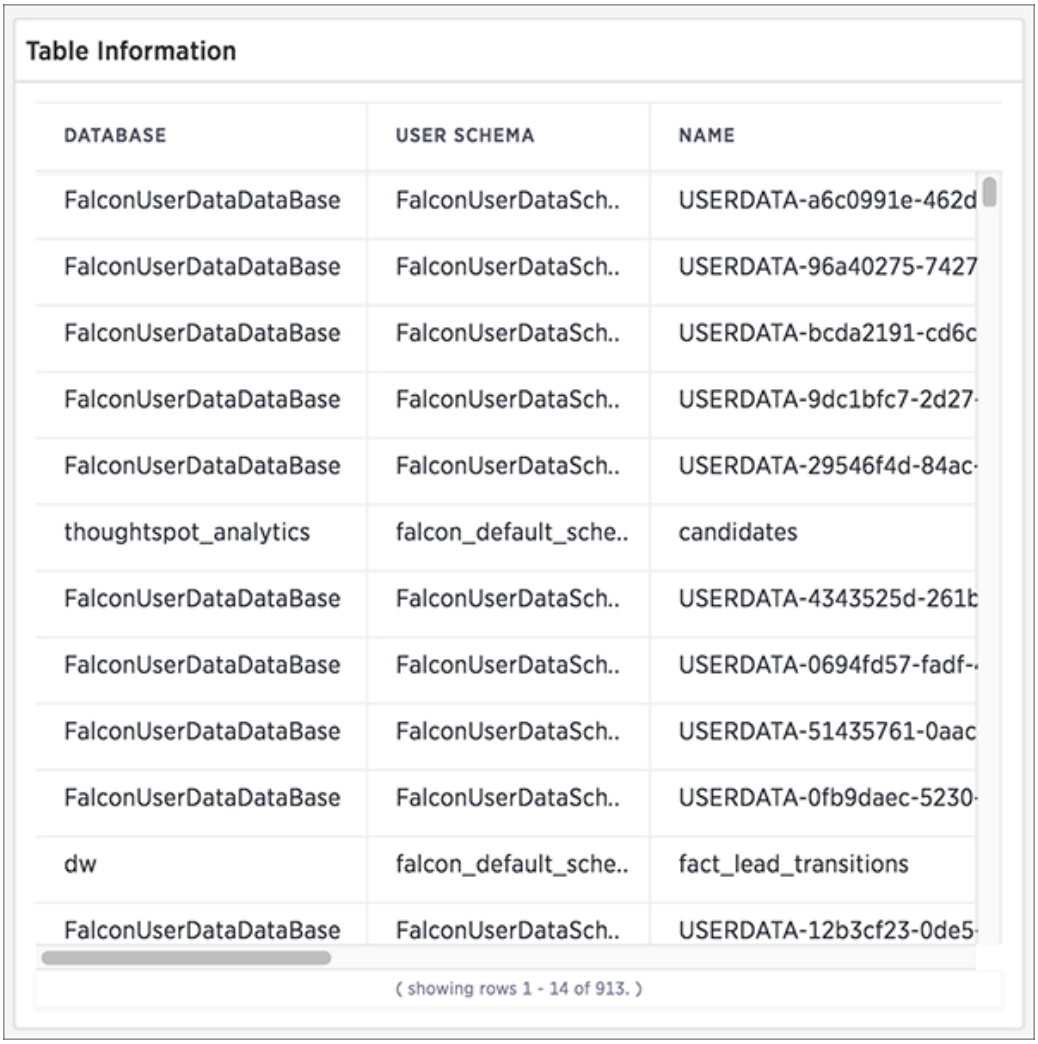
There are a number of deprecated boards on this page. They are there to support older installations that relied on them. New installations, should not use or rely on deprecated boards. Older installations that have used these boards in some way, should use the new boards and remove any dependencies.

Related information

`tscli logs` command

Data board

The **Data** page shows all the stored tables with details on the last update time, time taken for auto-indexing, number of rows, and so forth.



You can click on a column name to sort by table name. This table information is based on an underlying system data, you can present it or copy the link, but you cannot access the underlying query.

Database Status

The **Database Status** column can have the following possible values:

Status	Meaning
`READY`	The data has been loaded.

Status	Meaning
IN PROGRESS	The data is still being loaded.
STALE	The data is not up to date.
ERROR	The table is invalid. Call Customer Support.

Replication Status

The **Replicated** column indicates if the table has been replicated or sharded. This can be used in conjunction with the **Total Shards** column to see how your data is distributed.

If the table is replicated on a multi node system, the **Used Capacity (MB)** column will indicate the total space used on all nodes. a 10MB table replicated on a 4 node system will show 40MB used capacity for example.

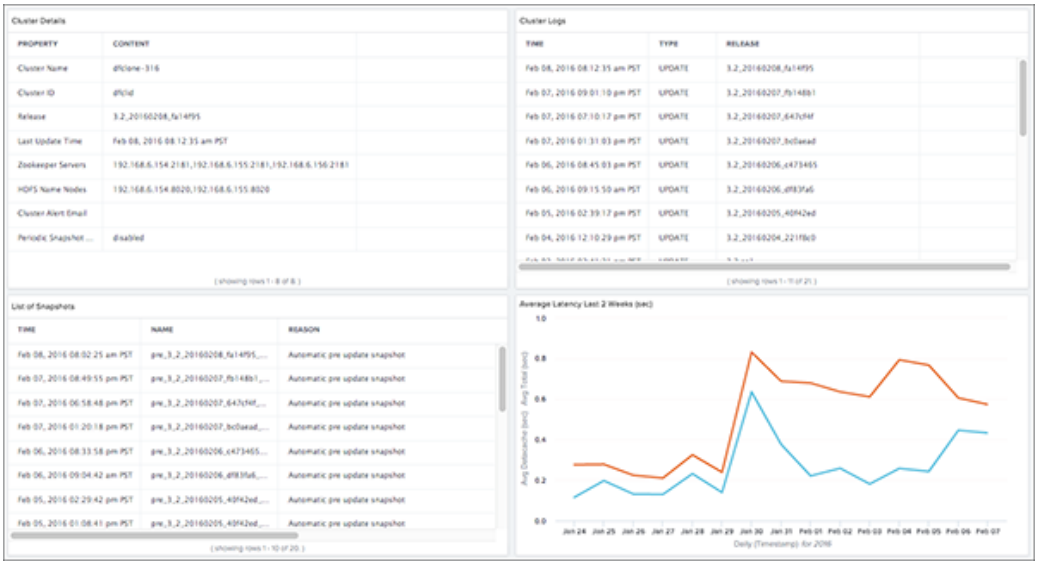
Search Status

The **Search Status** column can have the following possible values:

Status	Meaning
READY	The data is up to date and searchable.
NOT READY	The data is not ready to be searched.
DELETING INDEX	The table has already been deleted, but the index still exists due to the latency between the database and search engine.
INDEXING DISABLED	Either too many tokens exist in a column for it to be indexed, or indexing has been disabled manually.
CREATING INDEX	The index is being created.
UPDATING INDEX	A change has been made to indexing or the data, and the index is being updated to reflect it.

Cluster Manager board

The **Cluster Manager** section show detailed information about a cluster including latency over time, snapshot status, installed release, node functions, and logs.



Only the **Average Latency Last 2 Weeks (sec)** panel contains a standard ThoughtSpot visualization. The other visualizations rely on internal queries that are not accessible. You can, however, **Present** or **Copy link** on them.

Cluster Details

The fields on this visualization has the following meaning:

Field	Description
Cluster Name	The name of the cluster defined at installation time.
Cluster ID	The ID of the cluster set at installation time.
Last Update Time	Last time the cluster was updated.
Release	Version of the current release.
Zookeeper Servers	IP addresses of the Zookeeper servers.
HDFS Name Nodes	Control nodes for Hadoop Distributed File System (HDFS).

Cluster Logs

The fields on this visualization has the following meaning:

Field	Description
Time	A timestamp indicating when an action occurred.
Type	Type of action.
Release	Identifies the full release number.

You can also use the `tscli logs` command to review log data from your cluster.

List of Snapshots

This visualization shows the snapshots *and the backups* taken on the cluster. The fields on this visualization has the following meaning:

Field	Description
Time	A timestamp indicating when a snapshot or backup happened occurred.
Name	Name of the snapshot file. These files are stored in the `/usr/local/scaligent/backup` directory on your cluster.
Reason	Identifies the reason the snapshot/backup was created. You should see several period snapshots if your cluster is configured properly. You may also see evidence here of manual backups. For example, you should be sure your cluster is backed up before major events such as upgrades. Email support@thoughtspot.com if you don't see evidence your cluster is periodically creating snapshots.
Size	Size of the backup in gigabytes.

Average Latency Last 2 Weeks (sec)

This visualization relies on the `TS: BI Server` worksheet to display the average database latency over the last 15 days. The database latency measures how long it takes for a search to return data from ThoughtSpot - this does not include the time taken to send the answer back to the client, it measures internal processing time. You can use the visualization menu to drill down to its underlying query:

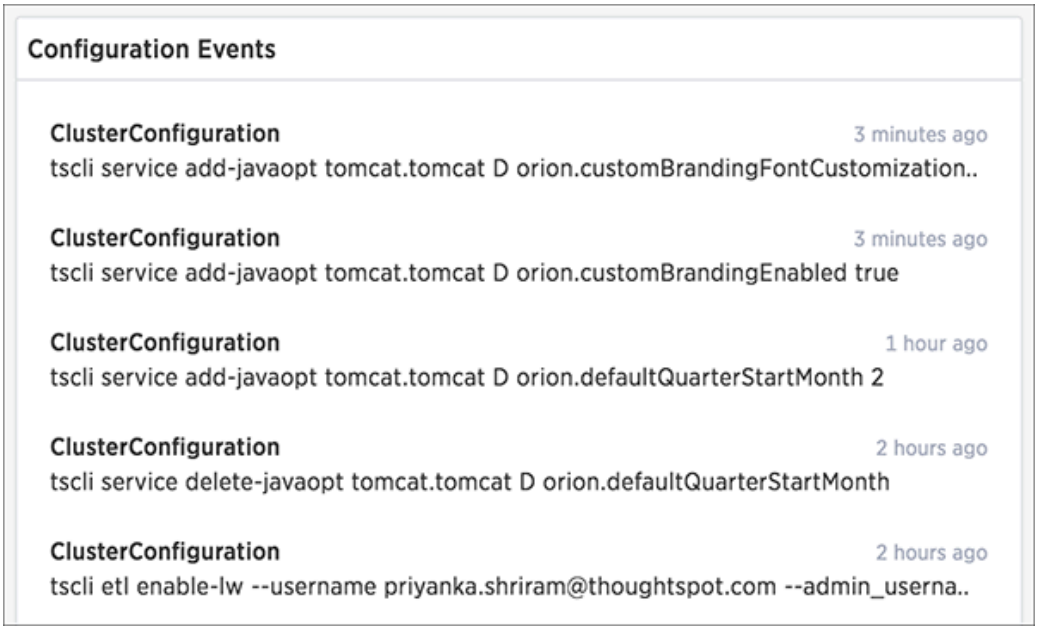
```
average datacache (sec)
average total (sec)
daily
last 15 days
for database latency (us) > 0
```

Related information

`tscli logs` command

Alerts and Events board

The **Alerts and Events** section shows notifications, alerts, and an audit trail of cluster configuration changes.



Alerts

The fields on this answer have the following meaning:

Field	Description
Time	When the alert was sent.
Type	The ID of the event.
Message	The text of the alert message.

For a full reference of possible alerts, see the Alert code reference.

Configuration Events

This system answer displays recent events that changed the configuration of the system. This list can contain the same types of information available on the **Admin System Health** > **Overview** page. This answer displays the **Time**, the **User** that performed the action, and a **Summary** of the action.

Notification events

This answer displays notifications of data loads. The display the **Time**, the **User** that performed the action, and a **Summary** of the action. Notifications are kept for 90 days before being discarded.

Related information

Alert code reference

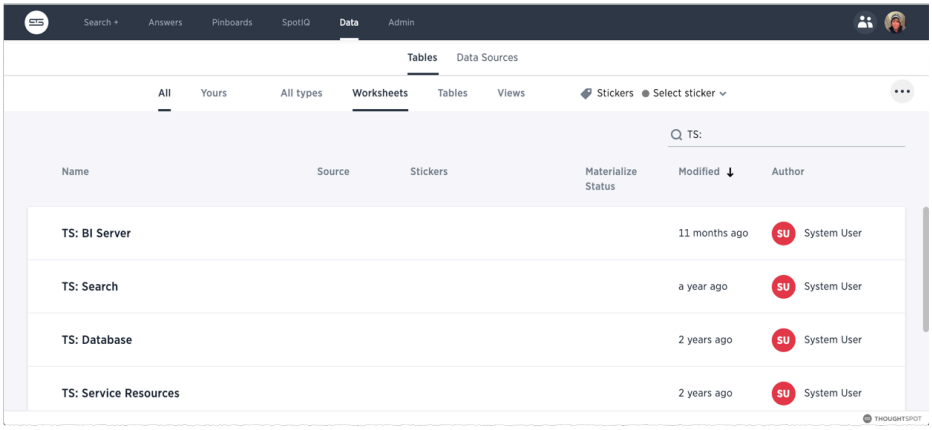
System worksheets

Most of the monitoring information in **System Health** are sourced from system worksheet which administrators can view, but not modify. The underlying tables are protected system tables that cannot be accessed directly. However, administrators can create new, custom monitoring reports from the worksheets.

List the system worksheets

To list the system worksheets:

1. Go to the **Data** tab.
2. Choose **All** and **Worksheets**.
3. Enter `TS:` in the search field.



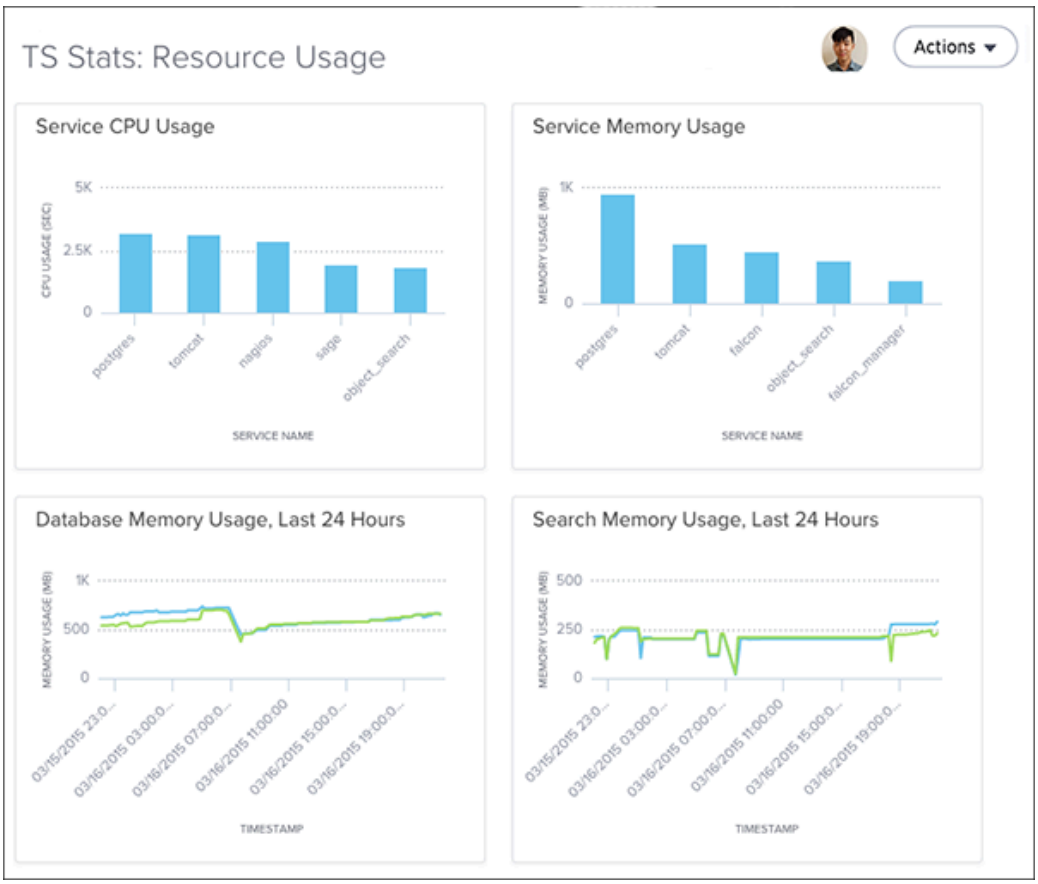
Summary of the worksheets

Worksheet	Description
`TS: BI Server`	Contains data related to the systems associated with underlying the ThoughtSpot BI server. This includes database latency, browser clients, size of responses, and more.
`TS: Database`	Contains information related to the database cache and queries run on the database. For example, you could use this worksheet to see data on the query errors returned by the database.
`TS: Internal Capacity WS`	Describes cluster memory capacity by node over time.
`TS: Internal Table Wise Capacity WS`	Describes memory capacity by node, table name, shard count, and CSV replication over time.
`TS: Metrics`	Contains metrics by cluster and host over time.

`TS: Search`	Contains data related to the number of searches (queries) run in the system. This contains information such as uptime, host, and timestamps.
`TS: Service Resources`	Contains data related to cluster nodes including page faults, memory usage, memory failures, and more.
`TS: Table Info`	Describes the named tables by timestamp, row count, row shards, and row size.
`TS: Table Row Counts`	Describes the named tables by timestamp and row count.
`TS: Table Row Counts and Shards`	Describes the named tables by timestamp, row count, row shards, and row size.
`TS: Table Row Size`	Describes the named tables by timestamp and row size.
`TS: Table Shards`	Describes the named tables by timestamp and shard count.

System pinboards

There are several system monitoring pinboards in ThoughtSpot that provide answers for system status and resource usage questions. The information in these pinboards are updated hourly from internal data sources that collect monitoring statistics.

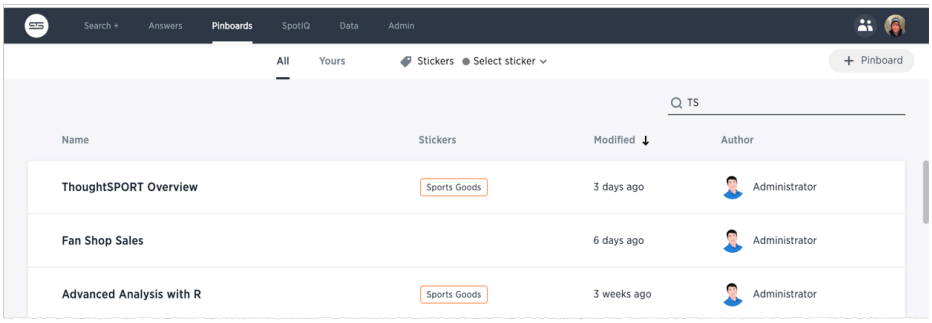


Only users with administrative privilege can view the monitoring pinboards. They are based on worksheets, which administrators can view, but not modify. However, you can create new monitoring pinboards from the worksheets.

List system pinboards

To list these system pinboards:

1. Go to the **Pinboards** tab.
2. Choose **All**.
3. Enter `TS:` in the search field.



Summary of system pinboards

Pinboard	Description
`Learn how to use ThoughtSpot`	Contains search replays created in the system.
`TS Stats: Alert Detail`	Combines alerts, notification events, and configuration events boards.
`TS Stats: Cluster Detail`	Contains cluster details, logs, snapshots, and latency data.
`TS Stats: System Information and Usage`	Replicates the **Admin > System Health > Overview** page.
`TS Stats: Latency Visualizations`	Latency on servers and *impression* counts. This data comes from the `TS: BI Server` worksheet.
`TS Stats: Table Status`	Replicates the **Data** board.

About deprecated boards

The deprecated boards (`TS status: Usage - Deprecated` and `TS StaTS: Queries --Deprecated`) are there to support older installations. New installations, should not use or rely on deprecated boards. Older installations that have used these boards in some way, should use the new boards and remove any dependencies.

Understand the backup strategies

This section discusses three strategies for backing up your ThoughtSpot cluster.

Snapshots

A snapshot is a point-in-time image of your running cluster. Snapshots are both taken on and restored to a cluster while it is running. Each cluster has a periodic snapshot configuration enabled by default. This configuration instructs the system to periodically take snapshots. Creation of a snapshot takes about 20 seconds. Once taken, a snapshot is persisted on disk in the cluster's HDFS.

You can also create a snapshot manually. You should create a snapshot before making any changes to the environment, loading a large amount of new data, or changing the structure of a table. A snapshot may only be restored to the same cluster on which it was taken. The cluster software release version must match the snapshot release version.

If you need to move data between clusters or restore to a cluster that was updated to a new release, contact ThoughtSpot Support.

Backups

A backup is a procedure that stores a snapshot outside of a ThoughtSpot cluster. Backups are stored in a directory on a local or network file system. You can store all of the data associated with a snapshot or a portion of that data or only metadata. There is no default configuration enabled for backing up a cluster. You can configure on yourself or you can take backups manually. Backing up periodically protects your company from losing data and/or user work.

You can use a backup to restore a cluster to a prior state, a differently configured appliance, or move the backup from an appliance to a virtual cluster or vice versa.

Offline backup cluster

The most robust strategy for backup and recovery requires having a backup cluster offline that is kept in sync with the production cluster. Then, if the production cluster fails, the backup cluster can be drafted to take its place with minimal loss of work and disruption to operations.

Details on this architecture and instructions on setting it up are available in the ThoughtSpot Disaster Recovery Guide, which you can request from ThoughtSpot.

Choosing when to use a strategy

Depending on your situation and your goals, you can choose to use a snapshot or a backup. This table should help you decide:

Snapshot	Backup

Used to	To restore to a cluster to particular point in time.	• Restore a cluster to a prior state. • Move a cluster to a different HW appliance. • Move a cluster to VM appliance. • Removal of a node. • Restoring to a cluster running a different release from the one where the backup was taken.
Stored	In the cluster's HDFS	Outside the cluster on either local or NAS disk.
Advantages	• Can be taken on or restored to a running cluster • Creation and restore is fast	• Very stable medium. • Can be used to recover from data loss or corruption, even if your cluster was destroyed. • Can be typed as full, lightweight, or dataless.
Limitations	• Includes all data, state, and metadata etc. created between snapshot creation and restore. • Are lost if the HDFS name node fails, you lose multiple disks at once, or the entire cluster is destroyed • Can only restored to the cluster they were taken from	• Backups require deleting the existing cluster first. • You are responsible for validating your backup configuration as viable for restoring a cluster. • Best practice recommends you to maintain multiple backups. • Are typically large in size. </td> </tr> </table> You should never restore from a snapshot or backup yourself, instead contact ThoughtSpot Support for help.

Understand backup/snapshot schedules

Summary: Explains how schedules and gives examples.

You can schedule periodic snapshots and backups. For snapshots, ThoughtSpot comes configured with a strongly recommended periodic snapshot policy. For backups, there is no such policy but you may want to create one or several of your own configurations. This section helps to understand existing schedules and how to configure new schedules.

Configuration format

ThoughtSpot uses a protocol buffer configuration file to hold snapshot and backup policies. There are slight differences between the configuration of snapshots and backups. You'll read more about these later. However, the file format defines a `schedule` structure which is the same for both snapshots and backups. The following example shows the `schedule` format:

```
schedule {
    period {
        number: integer
        unit: MINUTE | HOUR | DAY
        }
    retention_policy {
        bucket {
            time {
                number: integer
                unit: MINUTE | HOUR | DAY
             }
             capacity: integer
        }
    }
    offset_minutes_from_sunday_midnight: integer
}
```

The `schedule` has the following components:

`period`	Specifies the frequency in the chosen `unit`. You can specify the `unit` as `MINUTE`, `HOUR`, or `DAY`.
`retention_policy`	Specifies retention intervals. Retention is on a first-in-first-out (FIFO) basis. So, the oldest result is always discarded. You can specify the `unit` as `MINUTE`, `HOUR`, or `DAY`. You can specify multiple retention buckets and they can have different retention policies.

`offset_minutes_from_sunday_midnight`	Determines the minute within the hour you'd like execution to start. Setting this to zero is equivalent to midnight.

Work through an example schedule

In this section, you work through an example schedule. This is a working example that is the actual default snapshot schedule set on every ThoughtSpot instance.

```
schedule {
    period {
        number: 1
        unit: HOUR
    }
    retention_policy {
     bucket {
         time {
            number: 1
            unit: HOUR
         }
         capacity: 3
     }
     bucket {
        time {
            number: 4
            unit: HOUR
        }
        capacity: 2
        }
    }
    offset_minutes_from_sunday_midnight: 0
}
```

Under this policy, a snapshot is taken every hour starting at midnight. You can see that by combining the `period` of 1 hour with the midnight offset of 0.

```
    schedule {
        period {
            number: 1
            unit: HOUR
        }
        ...
        }
        offset_minutes_from_sunday_midnight: 0
    }
```

Using this frequency, a total of 24 snapshots are taken in a day.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	…	24

If you were to specify a `number` of 2, the frequency changes. The first execution would start at midnight but subsequent executions would happen every 2 hours as shown here:

2	4	6	8	10	12	14	…	24

You use the `retention_policy` to control how many snapshots are kept. In this example, the first bucket retains a snapshot every three hours.

```
retention_policy {
    bucket {
        time {
            number: 1
            unit: HOUR
        }
        capacity: 3
    }
    ...
 }
```

When the fourth hour comes along, the snapshot from first hour is discarded as per FIFO behavior. So in the 4th hour, you'll have the snapshots from hours 2, 3, and 4 in this retention bucket.

1D	2R	3R	4R	5	6	7	8	9	10	11	12	13	14	…	24

The second bucket retains the snapshot taken at four hour intervals.

```
 retention_policy {
    ...
    bucket {
        time {
            number: 4
            unit: HOUR
        }
        capacity: 2
    }
}
```

It retains two of these four-hour-interval snapshots at any one time. By hour 9 during the day, you have the snapshots from hour 4 and hour 8 in this second bucket.

```
1  2  3  4R  5  6  7  8R  9  10  11  12  13  14  ...  24
```

Consider what you will have in the first bucket in hour 9? The first bucket will have the snapshots from hour 9, 8, and 7.

At the end of the day, in the first bucket, you will have 22, 23, and 24th snapshot. While in the second bucket, you will have the 20th hour and the 24th hour snapshots.

```
1  ...  12  13  14  15  16  17  18  19  20R  21  22R  23R  24R
```

What if you changed the `period` frequency to every 2 hours? What would you have retained in your buckets at hour 24?

```
1  ...  12  14  16  18R  20R  22R  24R
```

As you can see, when defining a policy it can be helpful to graphically represent the frequency you configure. Then, determine which time blocks are important to retain before determining your retention bucket.

Work with snapshots

> **Summary:** A snapshot is a point-in-time image of your running cluster you can use to restore the cluster back to a specific point in time.

In this section, you learn how to work with the default snapshot configuration that is enabled on ever cluster and how to take manual snapshots of your own.

Create a manual snapshot

You should create a snapshot before making any changes to the environment, loading a large amount of new data, or changing the structure of a table. You can have up to 20 manual snapshots at a time, after which, you have to clear one before you are able to create another. If you need to delete a snapshot, contact ThoughtSpot Support.

> **Note:** When you upgrade, all existing snapshots from the previous version of ThoughtSpot will become manual snapshots.

Taking a snapshot is fast, about 20 seconds. It happens invisibly in the background of a running cluster. If you would like to restore from a snapshot instead, contact ThoughtSpot Support.

To create a snapshot:

1. Log in to the Linux shell using SSH.
2. Initiate a snapshot, providing a name and reason for creating it: Snapshot names must be 44 characters or less.

```
$ tscli snapshot create <name> <reason> <ttl>
```

3. Check that the snapshot was created:

```
$ tscli snapshot ls
```

Configure periodic snapshots

By default, each ThoughtSpot cluster is configured to take automatic, periodic snapshots of your cluster. This section explains how to learn more about the periodic snapshots in your cluster.

The default snapshot policy is enabled for every cluster. You can use the `tscli snapshot-policy show` command to display the current policy for periodic snapshots.

```
[admin@dogfood1 ~]$ tscli snapshot-policy show
schedule {
  period {
    number: 1
    unit: HOUR
  }
  retention_policy {
    bucket {
      time {
        number: 1
        unit: HOUR
      }
      capacity: 3
    }
    bucket {
      time {
        number: 4
        unit: HOUR
      }
      capacity: 4
    }
    bucket {
      time {
        number: 1
        unit: DAY
      }
      capacity: 4
    }
    bucket {
      time {
        number: 1
        unit: WEEK
      }
      capacity: 2
    }
  }
  offset_minutes_from_sunday_midnight: 0
}
enabled: true
```

This policy starts at midnight on Sunday. It retains the snapshots from the last three 4-hour intervals and two snapshots from two of the previous 4 hour intervals. That means, there are 7 periodic snapshots retained overall. For detailed information about understanding the schedule, see Understand backup/snapshot schedules.

You shouldn't change this default policy unless instructed to by support. If you have to adjust it for some reason, you can use, `tscli snapshot-policy update` command. This opens the current policy in an editor. Your policy should never retain more than 20 snapshots at any point in time. Exceeding this number can impact cluster performance.

You cannot delete the snapshot policy. However, you can disable the policy by executing the `tscli snapshot-policy disable` command. And you can re-enable it by running, `tscli snapshot enable-policy`.

> ❗ **Warning:** Backups rely on the snapshot system. For this reason, you should never disable the periodic snapshot system. For example, if you have disabled the periodic snapshots and periodic backups are enabled, then the periodic backup may use a very outdated snapshot or it may fail all together.

To check your current periodic snapshot policy:

1. Log in to the Linux shell using SSH.
2. Enter `tscli snapshot-policy show` to view the policy.

Understand backup modes

A backup is a procedure that stores a snapshot outside of a ThoughtSpot cluster. You can use a backup to restore a cluster to a prior state, a differently configured appliance, or move it to from an appliance to a virtual cluster or vice versa. Other advanced administrative operations also use backups.

You can create a manual backup or configure an automated, periodic backup A backup stores snapshot outside of a ThoughtSpot cluster. For manual backups, the system creates a backup using the named snapshot you specify. For periodic backups, the system uses the most recent snapshot to create the backup.

> **Warning:** You should never disable the periodic snapshot system as backups rely on it. For example, if you have disabled the periodic snapshots system and periodic backups are enabled, then the periodic backup may use a very outdated snapshot or it may fail all together.

Backups are usually stored on a NAS (network attached storage) file system but you can store them on a local disk as well. When creating a backup, ThoughtSpot copies a release tarball and several supporting files to a disk you specify. Storing these supporting files takes about 10 GB of extra space beyond the backup itself. The final backup image is smaller because these extra files are removed after the backup completes successfully. So, make sure you have enough disk space both to *take* a backup and store the result. Use the `tscli storage df` command to identify the amount of space available.

You can create a backup using one of three modes, full, lightweight or dataless.

Full backups

Full backups are entire backups of the cluster with all data, whether loaded from the web interface or from `tsload`. This is the best mode for restoring a cluster and all your data. Once a `FULL` backup is created, you can move them between clusters, even if the cluster configuration is different. Full backups can be as large as 20 GB in addition to the 5 GB of additional files. Some installations can exceed these limits, this is why it is important to test your backup configuration.

Before creating a manual backup or configuring automated backups, make sure there is enough disk space on the target disk. Consider an example, where you want to store three backups. If the backup itself takes 18GB, you need about 18 + 5 = 23 GB of free disk space. Don't forget that the backup size can grow over time, so you should occasionally check to ensure you are not in danger of running out of disk space to store backups.

Lightweight backups

Lightweight backups contain everything that makes up a cluster so they contain the following:

- Cluster configuration (SSH, LDAP, etc.)
- In-memory data cache
- All data that is stored unencrypted in HDFS
- Data uploaded by users
- Metadata for the data store
- Users, groups and permissions

- Objects created by users (pinboards, worksheets, and formulas) with their shares and permissions.
- Data model and row-level security rules.

Data loaded through ThoughtSpot Loader (`tsload`), ODBC/JDBC drivers, and Data Connect is excluded. The expectation is that data loaded via `tsload` is from external sources and so can be re-loaded after the cluster is restored. An exception is if these mechanisms were used to load data into tables that were first created through CSV import (that is, a user first loaded the tables via the GUI). In this case, the data, like the tables they were loaded into, are saved.

Dataless backups

A dataless backup saves a backup of the schema (metadata), with no data. Dataless backups allow you to send a copy of your cluster metadata to ThoughtSpot Support for troubleshooting without compromising data security and privacy. The size of a dataless backup is usually within 10's of megabytes provided you do not have customized binaries.

When restoring from a dataless backup, you must supply the correct release tarball, since this type of backup does not include the software release.

Create a manual backup

Use this procedure when you want to manually create a backup. If you would like to restore from a backup, contact ThoughtSpot Support.

You create a manual backup from an existing snapshot. So, you must identify an existing snapshot to use or take a new snapshot first. The time required to take a backup depends on the data size. Taking a backup does not take long, and happens in the background while the cluster is running.

1. Log in to the Linux shell using SSH.
2. Create a manual snapshot or find a snapshot you want to use. To find a snapshot you want to back up use the following command:

```
$ tscli snapshot ls


------------------------------------------------------------
-------------------

 Name          : pre330
 Reason        : pre3.3.0
 Hdfs snapshot : pre330
 Start         : Wed May 4 18:07:32 2016
 End           : Wed May 4 18:08:23 2016
 Size(Full)    : 13.24 GB
 Size(LW)      : 4.96 GB
 Size(Dataless): 39.76 MB

------------------------------------------------------------
-------------------

 ...
```

3. Make sure you have enough room on the target disk.

   In addition to the size of the snapshot, you will need 10 to 12 GB of disk space. This is because the process requires space for temporary files. You can use the `df` command to check disk size.

```
$ df –h
```

4. Create the backup, designating the type of backup, the snapshot name, and a directory:

   Choose the mode of backup you want to create, either full, lightweight, or dataless. The destination directory is created for you; do not specify an existing directory. The BASE value is the name

```
$ tscli backup create [-h]
    [--mode {full|light|dataless}]
    [--type {full}]
    [--base snapshot_name>]
    [--storage_type {local|nas}][--remote]
     <name> <directory>
```

5. Check that the backup was created:

```
$ tscli backup ls
```

Configure periodic backups

You can configure ThoughtSpot to backup automatically at specified times. The policy allows you to control the type, frequency, retention periods (first-in-first-out), and output location for a periodic backup.

A periodic backup uses the same steps as creating a backup manually. However, you do not need to specify a snapshot name, the system uses the most recent backup. You can backup to a local file system or mount a NAS (network attached storage) file system to hold the backup. A NAS is recommended. Make sure you have adequate space to store the number of backups you want to archive.

The format for a policy includes the following:

```
name: "name_for_backup"
param {
    mode: FULL | DATALESS | LIGHTWEIGHT
    type: STANDALONE
}
schedule {
    period {
      number: integer
      unit: MINUTE | HOUR | DAY
    }
    retention_policy {
          time {
          number: integer
          unit:  MINUTE | HOUR | DAY
        }
        capacity: integer
      }
  }
  offset_minutes_from_sunday_midnight: integer
}
directory: "NAME"
storage_type: NAS | LOCAL
```

Before creating a policy, make sure you have read Understand backup/snapshot schedules for information on configuring a `schedule` element. In addition, you must specify:

Element	Description
`mode`	The backup mode. `FULL` backups are necessary for restoring a cluster. See Work with backups for details on each backup mode.
`type`	Currently, only `STANDALONE` is supported.

Element	Description
`directory`	The location on the disk to place the backup.
`storage_type`	The type of storage you are using. `NAS` storage is recommended for `FULL` backups.

Backups cannot start when another backup is still running. So, choose a reasonable frequency for the mode in you policy. For example, a `FULL` backup takes longer than a `DATALESS` backup. Consider the load on the system when configuring. Do not backup up when the system would experience a heavy load. For example, you may want to take `FULL` backups late in the evening or on weekends.

The retention system deletes the oldest stored backup and the corresponding snapshot on a first-in first-out basis (FIFO). This means that if you set a bucket retention of 1 the system stores a single backup at any one time. The system deletes the older backup after the new full backup is successful.

To configure periodic backups:

1. Log in to the Linux shell using SSH.
2. Find a directory with enough disk space to support the `retention_policy number` you configure.

   You can use `df -h` to see free disk space and `tscli snapshot ls` to view existing snapshots and their size on disk.
3. Use the `tscli backup-policy create` command.

   The command opens a `vi` editor for you to configure the backup policy.
4. Write and save the file to store your configuration.

   By default, newly created policies are automatically enabled. To disable a policy, use the `tscli backup-policy disable` command.
5. Verify the policy using the `tscli backup periodic-config <name>` command.

Doing more with backup

The following table lists some additional backup commands you can use.

To	Command
List present backup policies.	`tscli backup-policy ls`
Show a backup policy.	`tscli backup-policy show <name>`
Check the status of a policy.	`tscli backup-policy status <name>`
Change an existing policy.	`tscli backup-policy update <name>`
Disable or enable an existing policy.	`tscli backup-policy disable` or `enable`
Delete a policy	`tscli backup-policy delete <name>`

Finally, you can time a `crontab` job with your periodic backup configuration to move a backup to longer term storage. Simply create a `crontab` job that moves the backup to a location outside of the `directory` defined in the periodic schedule.

About restore operations

When restoring to a running cluster that where the ThoughtSpot software was not updated, you'll usually use a snapshot. But in the case where you've updated the cluster to a new release, the configuration has changed significantly, or you're restoring to a different cluster, you'll need to restore from a backup.

Restoring from backup require that you first delete the old cluster. Changes to a cluster that require restoring from a backup instead of a snapshot include:

- Removal of a node.
- Restoring to a different cluster from the one where the snapshot/backup was taken.
- Restoring to a cluster running a different release from the one where the snapshot/backup was taken.

You should never restore from a snapshot or backup yourself. To perform a restore from a snapshot or backup, contact ThoughtSpot Support.

About troubleshooting

The information here provides very basic troubleshooting. For more detailed troubleshooting, Contact ThoughtSpot.

- Get your configuration and logs
  For troubleshooting on specific incidents or cluster problems, getting a log bundle can help.
- Network connectivity issues
  If network connectivity to and from ThoughtSpot is not working, try using these steps to find and correct the issue.
- Check the timezone
  ThoughtSpot comes configured with the timezone where it is to be installed.
- Browser untrusted connection error
  If you are not using a SSL certificate for authentication, users will see an untrusted connection error in their browser when accessing ThoughtSpot. The error looks slightly different depending upon the Web browser being used.
- Characters not displaying correctly
  Your CSV files are more likely to load smoothly if they are encoded with UTF-8. If you're having problems with some characters rendering incorrectly, you can convert the files to UTF-8 encoding before loading the data.
- Clear the browser cache
  You might occasionally see unexpected behavior that is due to the Web browser caching information from ThoughtSpot. In this case, clearing the browser cache and reloading the page should resolve the problem.
- Cannot open a saved answer that contains a formula
- Data loading too slowly
  Some tables may take an unusually long time to load due to a high data version issue. This issue normally arises when ThoughtSpot completes an upgrade or the system is recovering after a crash.
- Search results contain too many blanks
  If you find that your search results contain too many blanks when your data source is a worksheet, there is a simple adjustment you can make to fix this.

Get your configuration and logs

For troubleshooting on specific incidents or cluster problems, two things are important. Understanding your current configuration and getting a log bundle.

Check your configuration

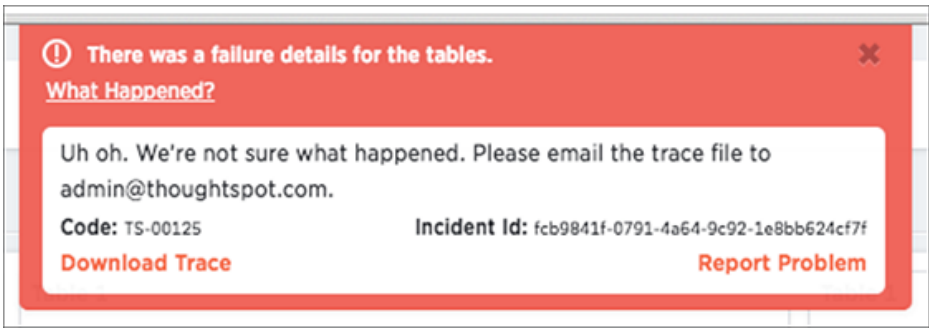
1. Log into the ThoughtSpot cluster as the `admin` user.
2. Use the `tscli feature` subcommand to display your current configuration.

```
 $ tscli feature get-all-config
 +---------------------------------+----------
+---------------+
 |              NAME               |  STATUS  | CONFIGUR
ATION |
 +---------------------------------+----------
+---------------+
 | Firewall                        | Disabled
|               |
 | Saml                            | Disabled
|               |
 | Ldap                            | Disabled
|               |
 | CustomBranding                  | Disabled
|               |
 | CustomBrandingFontCustomization | Disabled
|               |
 | DataConnect                     | Disabled
|               |
 | RLS                             | Enabled
|               |
 | Callhome                        | Enabled
|               |
 | SSHTunnel                       | Enabled
|               |
 | Fileserver                      | Disabled
|               |
 +---------------------------------+----------
+---------------+
```

How to get logs

There are two ways to get logs:

- When ThoughtSpot encounters a problem, a red bar displays in the browser with an error message. You can click on **What Happened?** in the error message for more details. To download related logs, click **Download Trace**. Send the logs as an email attachment to the support contact that is provided. Clicking **Report Problem** will also send the logs as an email attachment to your administrator.



- You can generate a log bundle using the `tscli` command `tscli logs collect` if you are comfortable with Linux. The command lets you specify which logs to collect and from what time periods.

  Usage for this command is:

```
tscli logs collect
  --include <selector | glob>
  [--exclude <selector | glob>]
  [--since <hours,minutes,days>
  | --from <yyyymmdd-HH:MM>
  --to <yyyymmdd-HH:MM>]
  [--out <path>]
  [--maxsize <size_in_MB_or_GB>]
  [--sizeonly]
```

  The full list of all selectors is:

  - `all` collects all of the logs listed from the system and the ThoughtSpot application.
  - `system` collects all system logs, e.g. syslog, upstart, mail logs, etc.
  - `ts` collects all logs from the ThoughtSpot application. This includes falcon, sage, orion core (cluster management), etc.
  - `orion` collects all orion logs including cluster management, hdfs, zookeeper, etc.

  Detailed syntax and options are listed in the tscli command reference.

Examples

Here are some examples of usage for the command `tscli logs collect`:

To collect all logs from the past day to the default location (`/tmp/logs.tar.gz`):

```
$ tscli logs collect --include all --since 1d
```

In this example, `all` is a selector for all the available logs.

In most cases, you'll probably use the selector `ts` to only capture logs for the ThoughtSpot application:

```
$ tscli logs collect --include ts --since 2d
```

For debugging cluster management issues, use a command like this one, which collects logs for system and orion from the past 2 hours. The output is written to /tmp/debug.tar.gz as specified using `--out`:

```
$ tscli logs collect --include system,orion --since 2h --out /t
mp/debug.tar.gz
```

This command collects logs from a specific time window:

```
$ tscli logs collect --include system,orion --from 20150520-1
2:00:00 --to 20150522-12:30:00
```

Advanced usage alert! You can also use `--include` and `--exclude` to specify filesystem paths as a glob pattern. This works like the Linux find(1) command. Pass all the entries in `--include` starting with `/` to find(1), and all entries in `--exclude` which are not selectors to find(1) using the `-not -path` flag.

```
$ tscli logs collect --include system,orion --exclude *hadoo
p*,*zookeeper* --since 2h
```

The above command collects all system and all orion logs, but excludes hadoop (hdfs) and zookeeper logs. See Upload logs to ThoughtSpot Support about using a secure file sever to collect log files or other files needed for troubleshooting. You can easily send log files to this file serve directly from the ThoughtSpot instance.

Upload logs to ThoughtSpot Support

ThoughtSpot Support uses a secure file sever to collect log files or other files needed for troubleshooting. You can easily send log files to this file serve directly from the ThoughtSpot instance.

Metrics collection

ThoughtSpot collects the diagnostic information from your system on an ongoing basis: there is no time needed to collect diagnostic information after a problem is reported. These metrics allow for:

- Our support team can begin working to remediate any issue with you at once.
- Metrics provides direct visibility to the ThoughtSpot team on your system's limits. Therefore, our Support team can proactively identify critical threshold issues and work to prevent failures. Metrics also help reduce SLA times as the team can debug much faster.
- ThoughtSpot can tune search algorithms by studying search history and schema.
- ThoughtSpot analyzes expensive and complex query patterns to look for performance optimizations.

Finally, the metrics pipeline allows ThoughtSpot to identify application-use patterns that contribute to performance bottlenecks with specific browsers and help your team prevent or alleviate them.

Other log uploads

Before you can upload a file to the secure file server:

1. Configure the connection to the file server.
2. Obtain the directory path on the file server.

The server directory path for uploading a file is formatted like this example: `/Shared/support/<customer_name>`. If you do not know the customer name, contact ThoughtSpot Support.

You can upload files directly to the file server using this procedure:

1. Log in to the Linux shell using SSH.
2. Navigate to the directory where the file to be uploaded is located.
3. Issue the command to upload the file, specifying the file name and directory path:

```
 $ tscli fileserver upload --file_name <file> --server_d
ir_path <path>
```

When your upload succeeds, you will see a confirmation message.

Network connectivity issues

If network connectivity to and from ThoughtSpot is not working, try using these steps to find and correct the issue.

To troubleshoot network connectivity for ThoughtSpot:

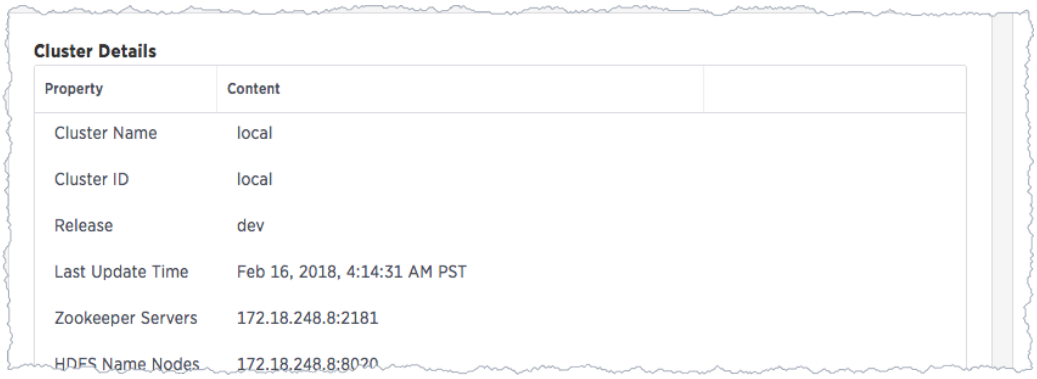
1. Make sure that the network cables are connected correctly.
2. Check that the network cable is connecting the nodes to the network switch.
3. Try replacing the cable with a cable from a known working system to rule out a bad cable or switch connectivity issues.
4. Make sure the eth0 interface is connected to the network by issuing: `ethtool eth0` The port that's currently connected will have "link detected" in the last line of the output.
5. If the networking settings have been reconfigured, reboot each of the nodes.

Check the timezone

ThoughtSpot comes configured with the timezone where it is to be installed. Data is imported based on the timezone of the node from which `tsload` or `tql` is run. To see the timezone your ThoughtSpot node is running under, log into the server as the `thoughtspot` user and run the `date` command:

```
[thoughtspot@ts.server etc]$ date
Tue Feb 20 09:07:04 PST 2018
```

To see the current timezone used by the ThoughtSpot application, choose **Admin** > **Cluster Manager** and review the **Cluster Details**:



Cluster Details

Property	Content
Cluster Name	local
Cluster ID	local
Release	dev
Last Update Time	Feb 16, 2018, 4:14:31 AM PST
Zookeeper Servers	172.18.248.8:2181
HDFS Name Nodes	172.18.248.8:8020

The timezones should match.

Sometimes the timezone that is listed by `date` is not the active timezone for the ThoughtSpot application and the application needs to be reset. If you need to change the timezone, contact ThoughtSpot Support and they will change the timezone for you. You may need to change the timezone if you move the server between data centers.

Browser untrusted connection error

If you are not using a SSL certificate for authentication, users will see an untrusted connection error in their browser when accessing ThoughtSpot. The error looks slightly different depending upon the Web browser being used.

ThoughtSpot uses secure HTTP (the HTTPS protocol) for communication between the browser and ThoughtSpot. By default there is no SSL certificate for authentication. This must be added by the site administrator. If the site administrator has not added the certificate, the browser warns the user.

Browser	Warning
Google Chrome	The site's security certificate is not trusted!
Mozilla Firefox	This Connection is Untrusted

If you see the warning message, choose one of the following options:

- Ask the site administrator to install the certificate.
- Ask the site administrator to turn off SSL using this command in the shell on the ThoughtSpot instance:

```
$ tscli ssl off
```

- You can choose to ignore the message, and access ThoughtSpot without SSL.

Characters not displaying correctly

Your CSV files are more likely to load smoothly if they are encoded with UTF-8. If you're having problems with some characters rendering incorrectly, you can convert the files to UTF-8 encoding before loading the data.

You might see unexpected characters in your data, especially characters whose ASCII values are at the high and low end of possible values. Some examples of characters that can appear incorrectly are: æ, ñ, ä, í, ö.

If this happens, your data will look like this:

Instead of displaying correctly like this:

To encode your data as UTF-8:

1. On Windows, open your CSV file in Notepad. Save the file as CSV with the Unicode option.
2. On Linux or MacOS, issue a command like:

```
$ iconv -f -t UTF-8 <in_file>.csv > <out_file>.csv
```

3. Reload the data.
4. Attempt to import it again.

Clear the browser cache

You might occasionally see unexpected behavior that is due to the Web browser caching information from ThoughtSpot. In this case, clearing the browser cache and reloading the page should resolve the problem. You can usually resolve these situations by clearing the browser cache:

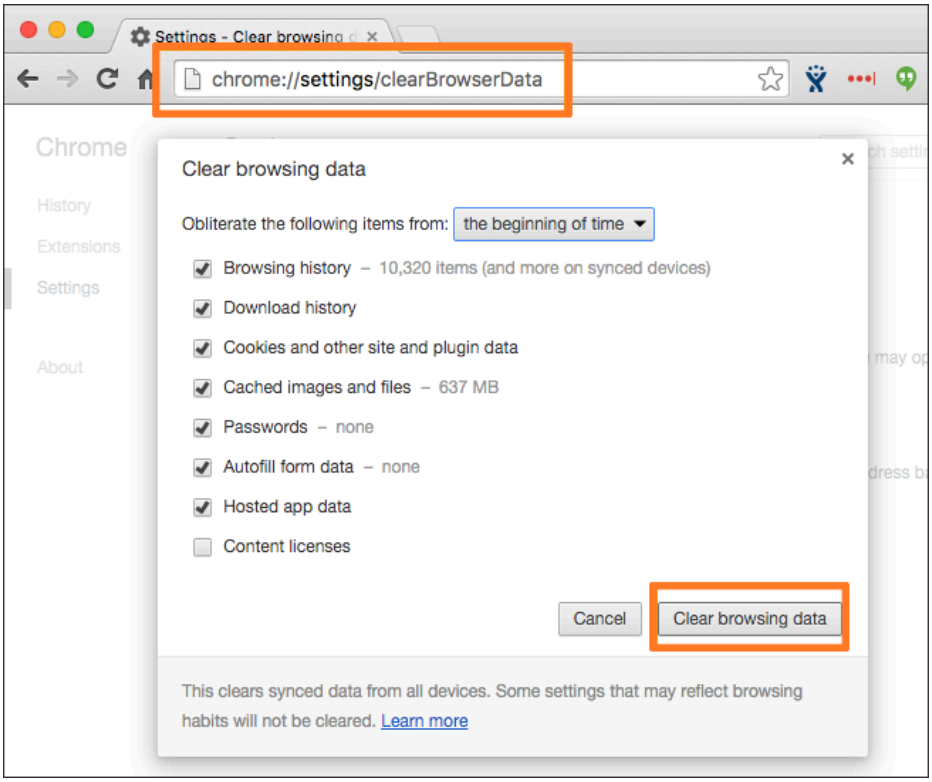
- During a ThoughtSpot session, the browser suddenly displays a white screen and reloading does not fix the problem. This is due to a self-signed SSL certificate that has timed out during the session.
- When accessing the Help Center, you see a login screen. This is due to a problem during automatic authentication in the Help Center, after which the bad login gets cached by the browser.

To resolve any of these situations, clear the browser cache:

1. Clear the browser cache.

   This works a little differently on individual browser versions and operating systems. For example, when using Chrome, to get to the browser cache settings, navigate to:

   ```
   chrome://settings/clearBrowserData
   ```

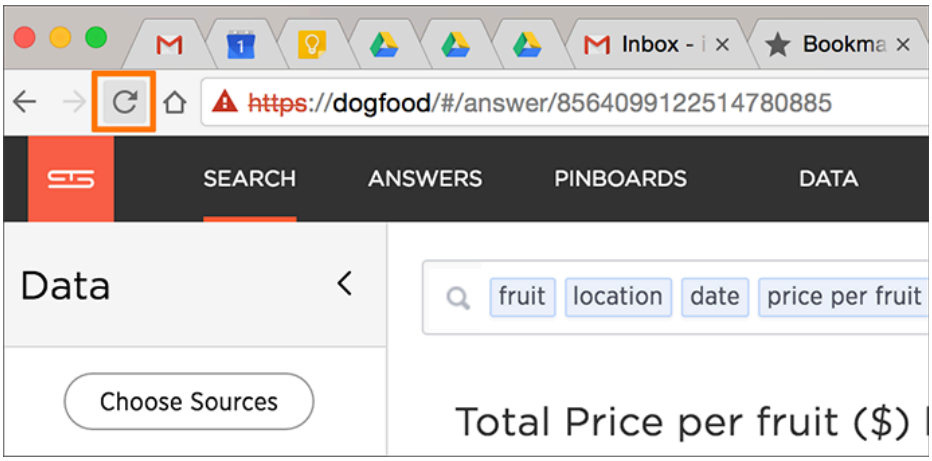


2. Click **Clear browsing data**.

   This is the name of the button on Chrome. The name may vary slightly on other browsers.

3. Reload the page.

For example, on Chrome you would click the **Reload** icon:



Now the problem should be fixed, and the page will appear as expected.

Cannot open a saved answer that contains a formula

When working with formulas, keep in mind the data types they return. You may occasionally see unexpected results, or even be unable to open a saved answer, due to problems with data types and formulas.

In this scenario, “data type” refers the data type as defined in the column definition when creating the schema (INT, TIMESTAMP, VARCHAR, etc.).

When you define a formula, both the data type it returns is set automatically. This can lead to problems, if you build another formula that uses the output of the first formula as input. This can be hard to understand, so an example will be helpful.

Suppose you have created a worksheet that contains a formula called “weekday” defined as:

```
day_of_week(date)
```

The output of that formula is the day of the week (Monday, Tuesday, etc.) returned as a text string (VARCHAR, ATTRIBUTE).

Then suppose you create an answer using the worksheet as a source. And in the answer, you create another formula on top of the formula column in the worksheet. This formula is supposed to return the day of the week that is two days after the given day of the week:

```
weekday + 2
```

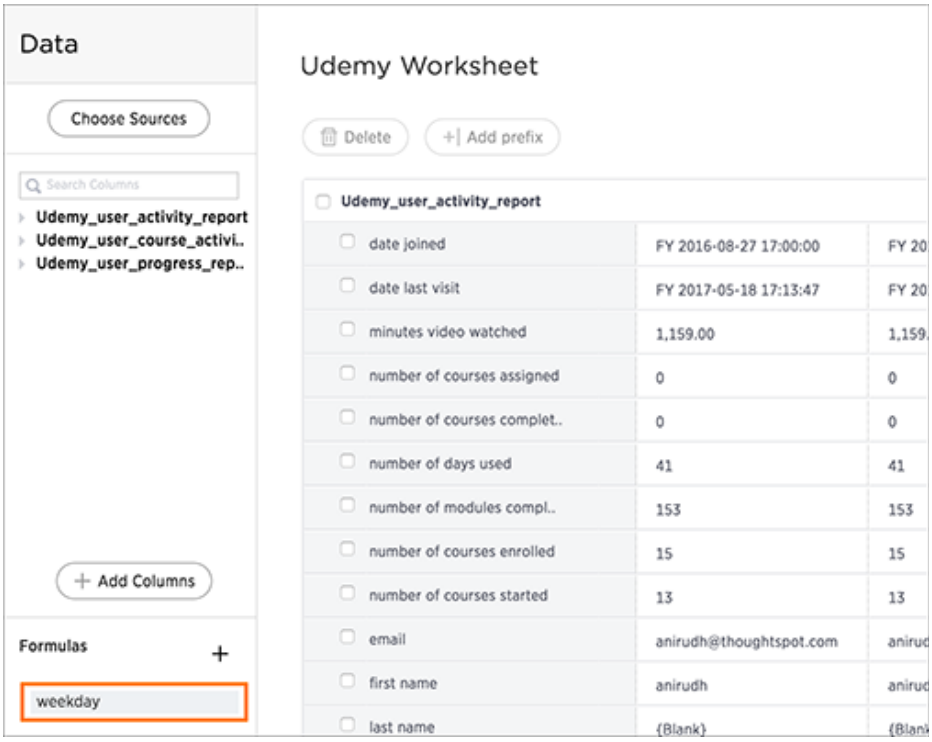
In this case, you have effectively created a formula on top of another formula. This works fine, so long as the data types in the worksheet formula can work in the answer formula. If not, you may not be able to save the answer, or open it once it has been saved. Here, the second formula you created does not work, because it is invalid. It is trying to subtract a number from a text string.

If you encounter this issue, you will need to open the worksheet and edit its formula so that it returns the type expected by the formula that was built on top of it. In this case, a numeric data type.

You must change the underlying worksheet column to use day_number_of_week instead of day_of_week. This is because day_number_of_week returns a numeric data type.

Here are the steps to resolve an issue like this:

1. Open the underlying worksheet that contains the formula whose output data type you need to change.
2. Click on the formula name to edit the formula.



3. In the Formula Builder, modify the formula, so that it returns the expected data type.

   There are data type conversion formulas available to make this easier. To view them and their syntax, open the **Formula Assistant**, and expand the section called **Conversion**.

4. Make your changes, and saving the formula by clicking **Save**.
5. Save the worksheet by clicking **Save**.
6. Now you will be able to open the answer that was created on top of the worksheet.

Data loading too slowly

Some tables may take an unusually long time to load due to a high data version issue. This issue normally arises when ThoughtSpot completes an upgrade or the system is recovering after a crash.

The data version is the number of loads that have been historically applied to a table. Every completed load increments the version number of the table by one. ThoughtSpot would need to process each version of the table during restoration, which could increase the time it takes to build the table.

There are a few steps you can take to check for a high data version issue and fix it. To improve data loading speed, run the following command to find the number of tables that are building and their names

```
tscli cluster status –-mode table
```

You may notice that a few small tables are taking up a lot of time to be built. This could simply be due to the deceivingly large size of the table. There is also the chance it could be due to a high data version issue. To determine if this is a high data version issue, check the size of the table by running the following command:

```
echo 'show statistics for server;' | tql
```

If there is a large number of rows in the table, go on to shard the table. If the table has a small number of rows, then the slow loading speed is caused by a high data version issue, and you do not have to shard the table. Use the compact table functionality to trim the table down to its actual size:

```
tql> compact table <table name>;
```

Search results contain too many blanks

If you find that your search results contain too many blanks when your data source is a worksheet, there is a simple adjustment you can make to fix this.

If you find that the charts and tables built on a worksheet contain a large number of null values (which display as {blank} in the web browser), you can fix this by changing the inclusion rule for the worksheet.

An inclusion rule that specifies **Exclude Empty Rows (Inner Join)** will reduce the number of null values in the result. Imagine a worksheet that includes data about a retail grocery store. There are rows in the worksheet from these source tables:

Table Name	Description
sales	Fact table with sales made per product per store.
products	Dimension table with information about every product.
stores	Dimension table with information about every store.

When creating the worksheet, suppose you choose **Include Empty Rows (Left Outer Join)** for the inclusion rule and **Progressive Joins** for the join rule. In this case, if you type “product name” in your search, you’ll see a list of all the products that exist. Suppose you then add “store name” to your search. You will see a lot of null ({blank}) values in the result. This happens because the columns “store name” and “product name” are joined through the fact table, “sales”. So for every product that has never been sold in a particular store, you’ll see {blank} in the “store name” column. This may be what you want to see, in which case, you can leave the worksheet as is, and choose **Exclude** for the {blank} values in your table or chart, whenever you don’t want to see them.

However, in may cases, including all the {blank} values could confuse end users. So if you encounter this problem, you can edit the worksheet, and change the inclusion rule to **Exclude Empty Rows (Inner Join)**. Now when searching for “store name” and “product name” on the worksheet, users will not be overwhelmed by null values. They’ll only see the rows where a particular product has been sold in a particular store.

Keyword reference

You can use keywords when asking a question to help define your search. This reference lists the various keywords. You can also see this list of keywords and examples from within the help center.

Keywords in Other Languages

Currently, we offer the following keyword translations.

日本語	中文 (簡体)	Deutsche	Español (latín)	Español (España)
Français (Canada)	Français (France)	Português (Brasil)	Português (Portugal)	Italiano
Dansk	Suomi	Svenska	Norsk	Nederland

Also, see the topic on how to set locale preferences in your user profile to control language, date, and number formats on the ThoughtSpot UI.

General

Keyword	Examples
top	• `top sales rep by count sales for average revenue > 10000` • `sales rep average revenue for each region top`
bottom	• `bottom revenue average` • `bottom revenue by state` • `customer by revenue for each sales rep bottom`
n	`top 10 sales rep revenue`
n	`bottom 25 customer by revenue for each sales rep`
sort by	• `revenue by state sort by average revenue` • `revenue by customer sort by region`

Date

Keyword	Examples
after	`order date after 10/31/2014`

Keyword	Examples
before	`order date before 03/01/2014`
between … and ...	`order date between 01/30/2012 and 01/30/2014`
daily year-over-year	`growth of revenue by order date daily year-over-year`
daily	`shipments by region daily`
day	`count monday restaurant`
day of week	`revenue by day of week last 6 months`
day of week	`count shipments Monday`
n days for each month	`sales last 2 days for each month`
n days for each quarter	`revenue last 15 days for each quarter`
n days for each week	`total sold last 2 days for each week`
n days for each year	`revenue last 300 days for each year`
growth of … by … daily	`growth of sales by order date daily`
growth of … by … monthly	`growth of sales by date shipped monthly sales > 24000`
growth of … by … quarterly	`growth of sales by date shipped quarterly`
growth of … by … weekly	`growth of sales by receipt date weekly for pro-ski2000`
growth of … by … yearly	`growth of sales by date closed yearly`
growth of … by ...	`growth of sales by order date`
n hours for each day	`sales last 2 hours for each day`
last day by	`customers last day by referrer`
last month by	`customers last month by day`
last *n* days	`visitors last 7 days`
last *n* quarters	`visitors last 2 quarters by month by campaign`
last *n* weeks	`visitors last 10 weeks by day`

Keyword	Examples
last quarter	`customers last quarter sale > 300`
last week	`customers last week by store`
last year	`top 10 customers last year by sale by store for region west`
last *n* years	`visitors last 5 years by revenue for sum revenue > 5000`
month to date	`sales by product month to date sales > 2400`
month year	`commission by sales rep February 2014`
month	`commission January`
month	`revenue by month last year`
monthly year-over-year	`growth of revenue by receipt date monthly year-over-year`
monthly	`commission > 10000 monthly`
n months for each quarter	`cost last 2 months for each quarter`
n months for each year	`last 8 months for each year`
n days ago	`sales 2 days ago`
n months ago	`sales 2 months ago by region`
n months	`visitors last 6 months for homepage visits > 30 by month`
n quarters ago	`sales 4 quarters ago by product name contains deluxe`
n weeks ago	`sales 4 weeks ago by store`
n years ago	`sales 5 years ago by store for region west`
n years	`opportunities next 5 years by revenue`
next day	`shipments next day by order`
next month	`appointments next month by day`
next *n* days	`shipments next 7 days`
next *n* months	`openings next 6 months location`
next *n* quarters	`opportunities next 2 quarters by campaign`
next *n* weeks	`shipments next 10 weeks by day`

Keyword	Examples
next quarter	`opportunities next quarter amount > 30000`
next week	`shipments next week by store`
next year	`opportunities next year by sales rep`
quarter to date	`sales by product quarter to date for top 10 products by sales`
quarterly year-over-year	`growth of revenue by date shipped quarterly year-over-year`
quarterly	`sales quarterly for each product`
n quarters for each year	`last 2 quarters for each year`
today	`sales today by store`
week to date	`sales by order date week to date for pro-ski200`
week	`revenue by week last quarter`
weekly year-over-year	`growth of revenue by date shipped weekly year-over-year`
weekly	`revenue weekly`
n weeks for each month	`sales last 3 weeks for each month`
n weeks for each quarter	`last 2 weeks for each quarter`
n weeks for each year	`last 3 weeks for each year`
year to date	`sales by product year to date`
year	`revenue by product 2014 product name contains snowboard`
yearly	`shipments by product yearly`
yesterday	`sales yesterday for pro-ski200 by store`

Time

Keyword	Examples
detailed	`ship time detailed`

Keyword	Examples
last minute	`count homepage views last minute`
last hour	`count unique visits last hour`
n minutes	`count visitors last 30 minutes`
n hours	`count visitors last 12 hours`
hourly	`visitors by page name hourly`
n minutes ago	`sum inventory by product 10 minutes ago`
n hours ago	`sum inventory by product by store 2 hours ago`

Text

Keyword	Examples
begins with	`product name begins with 'pro'`
contains	`product name contains "alpine" description contains "snow shoe"`
ends with	`product name ends with 'deluxe'`
not begins with	`product name not begins with "tom's"`
not contains	`product color not contains 'tan' product color not contains 'red'`
not ends with	`product name not ends with "trial"`
similar to	`course name similar to 'hand'`
not similar to	`course name not similar to 'hand'`

Number

Function	Examples
sum	`sum revenue`
average	`average revenue by store`
count	`count visitors by site`
max	`max sales by visitor by site`
min	`min revenue by store by campaign for cost > 5000`
standard deviation	`standard deviation revenue by product by month for date after 10/31/2010`
unique count	`unique count visitor by product page last week`
variance	`variance sale amount by visitor by product for last year`

Comparative

Function	Examples
all	`all`
between... and	`revenue between 0 and 1000`
vs, versus	`revenue east vs west`
>	`sum sale amount by visitor by product for last year sale amount > 2000`
<	`unique count visitor by product by store for sale amount < 20`
>=	`count calls by employee lastname >= m`
<=	`count shipments by city latitude <= 0`
=	`unique count visitor by store purchased products = 3 for last 5 days`
!=	`sum sale amount region != canada region != mexico`

Location

Keyword	Examples
near	`revenue store name county near san francisco`
near... within *n* miles|km|meters	`revenue store name county near alameda within 50 miles`
farther than *n* miles|km|meters from	`average hours worked branch farther than 80 km from scarborough`

Location keywords only work for searches where the data source includes latitude/longitude data.

Period

Keyword	Example
quarter (*date*)	`quarter (purchase date)`
quarter of year (*date*)	`quarter of year (purchase date)`
month of quarter (*date*)	`month of quarter (purchase date)`
week of year (*date*)	`week of year (ship date)`
week of quarter (*date*)	`week of quarter (ship date)`
week of month (*date*)	`week of month (ship date)`
day of year (*date*)	`day of year (ship date)`
day of quarter (*date*)	`day of quarter (ship date)`
day (*date*)	`day (ship date)`
day of month (*date*)	`day of month (order date)`
day of week (*date*)	`day of week (order date)`
hour (*datetime*)	`hour (timestamp)`

TQL reference

TQL is the ThoughtSpot language for entering SQL commands. This reference lists TQL commands you can use to do things like creating a schema or verifying a data load.

About using TQL

You can use TQL either through the ThoughtSpot application's web interface or the command line interface in the Linux shell.

Use `--query_results_apply_top_row_count <number>` flag to limit the number of result rows returned by a query. For example:

```
$ tql --query_results_apply_top_row_count 100
```

As a best practice, you should enclose object names (database, schema, table, and column) in double quotes, and column values in single quotes. When referring to objects using fully qualified object names, the syntax is:

```
"database"."schema"."table"
```

To get help on SQL when using TQL, enter `help` on the command line.

You can use TQL to view and modify schemas and data in tables. Remember to add a semicolon after each command. Commands are not case sensitive but are capitalized here for readability.

> **Note:** Worksheets and pinboards in ThoughtSpot are dependent upon the data in the underlying tables. Use caution when modifying tables directly. If you change or remove a schema on which those objects rely, the objects could become invalid.

View schemas and data

Syntax	Description
`SHOW DATABASES`	Lists all available databases. Examples: `SHOW DATABASES;`

Syntax	Description
`USE <database>`	Switches the context to the specified database. This is required if queries do not use fully qualified names (database.schema.table) for specifying tables. Examples: `USE "fruit_database";`
`SHOW SCHEMAS`	Lists all schemas within the current database. For example: `SHOW SCHEMAS;`
`SHOW TABLES`	Lists all tables within the current database by schema. For example: `SHOW TABLES;`
`SHOW TABLE <table>`	Lists all the columns for a table. For example: `SHOW TABLE "locations";`
`SCRIPT SERVER`	Generates the TQL schema for all tables in all databases on the server. For example: `SCRIPT SERVER;`
`SCRIPT DATABASE <database>`	Generates the TQL schema for all tables in a database. For example: `SCRIPT DATABASE "fruit_database";`
`SCRIPT TABLE <table>`	Generates the TQL schema for a table. For example: `SCRIPT TABLE "vendor";`

Syntax	Description

`SELECT <cols_or_expr> FROM <table_list> [WHERE <predicates>] [GROUP BY <expr>] [ORDER BY <expr>]`	Shows specified set of table data.
	
	If you do not specify the TOP number of rows to select, the top 50 rows will be returned by default. The number of rows to return can be set using the TQL command line flag:
	
	`--query_results apply_top_row_count`
	
	You can use the following aggregation functions:
	
	• sum
	• count
	• count distinct
	• stddev
	• avg
	• variance
	• min
	• max
	
	You can use the following date functions:
	• absyear
	• absmonth
	• absday
	• absquarter
	• date
	• time
	
	For example:
	
	`SELECT TOP 10 "quantity" FROM "sales_fact";`
	
	`SELECT COUNT(*) FROM "vendor";`
	
	`SELECT "vendor", SUM("quantity") FROM "sales_fact" GROUP BY "vendor";`
	
	`SELECT "vendor", SUM("amount") FROM "vendor", "sales_fact" WHERE "sales_fact"."vendorid" = "vendor"."vendorid" AND "amount" > 100 GROUP BY "vendor" ORDER BY "amount" DESC;`
	
	`SELECT "vendor", SUM("quantity") FROM "sales_fact" GROUP BY "vendor" LIMIT 10;`

Schema creation

Syntax	Description
`CREATE DATABASE <database>`	Creates a database. For example: `CREATE DATABASE "fruit_database";`
`CREATE SCHEMA <schema>`	Creates a schema within the current database. For example: `CREATE SCHEMA "fruit_schema";`

`CREATE TABLE <table> (<column_definitions> [<constraints>]) [PARTITION BY HASH (<number>) [KEY ("<column>")]])`	Creates a table with the specified column definitions and constraints.
	Use `PARTITION BY HASH` to shard a table across all nodes. If no `KEY` is specified, the table will be randomly sharded.
	Do not specify relationship constraints (`FOREIGN KEY` or `RELATIONSHIP`) in the `CREATE TABLE` statement. Instead, define these using `ALTER TABLE` statements at the end of your TQL script, after creating your tables. This method guarantees that tables are created before they are referenced in the constraint definitions. For example:
	`CREATE TABLE "vendor" ("vendorid" int, "name" varchar(255));`
	`CREATE TABLE "sales_fact" ("saleid" int, "locationid" int, "vendorid" int, "quantity" int, "sale_amount" double, "fruitid" int, CONSTRAINT PRIMARY KEY("saleid")) PARTITION BY HASH(96) KEY ("saleid");`

Schema modification

Syntax	Description
`DROP DATABASE <database>`	Drops a database and all of its schemas and tables. For example: `DROP DATABASE "fruit_database";`

Syntax	Description
`DROP SCHEMA <schema>`	Drops a schema within the current database, and drops all of the tables in the schema. For example: `DROP SCHEMA "fruit_schema";`
`DROP TABLE <table>`	Drops a table. For example: `DROP TABLE "location";`
`TRUNCATE TABLE <table>`	Removes all data from a table, but preserves its metadata, including all GUIDs, relationships, etc. This can be used to force a new schema for a table without losing the metadata. However, this operation removes all existing data from the table and must be used with caution. You must reload the data following a `TRUNCATE`, or all dependent objects (worksheets and pinboards) in ThoughtSpot will become invalid. For example: `TRUNCATE TABLE "location";`

`ALTER TABLE <table> ADD|DROP|RENAME COLUMN <column>`	Alters a table to add, drop, or rename a column.
	When you add a column to an existing table, you must provide a default value to use for existing rows. For example:
	`ALTER TABLE "cart" ADD COLUMN "nickname" varchar(255) DEFAULT 'no nickname';`
	`ALTER TABLE "cart" DROP COLUMN "nickname";`
	`ALTER TABLE "cart" RENAME COLUMN "nickname" TO "shortname";`

`ALTER TABLE <table> DROP CONSTRAINT PRIMARY KEY;`	Drops the primary key from a table.
	Note that if you then add a new primary key, the same upsert behavior will be applied as with adding any primary key. This can result in data deletion, so make sure you understand how the upsert will affect your data ahead of time. For example:
	`ALTER TABLE "sales" DROP CONSTRAINT PRIMARY KEY;`
	`ALTER TABLE "sales" ADD CONSTRAINT PRIMARY KEY ("PO_number");`

Syntax	Description

`ALTER TABLE <table> DROP [FOREIGN KEY| RELATIONSHIP] <name>;`	Drops the named foreign key or relationship between two tables. For example:
	`ALTER TABLE "sales_fact" DROP CONSTRAINT FOREIGN KEY "FK_PO_number";`
	`ALTER TABLE "fruit_dim" DROP RELATIONSHIP "REL_dates";`

`ALTER TABLE <table> DROP [CONSTRAINT FOREIGN KEY [<table_name>] | RELATIONSHIP [WITH <table_name>];`	You must use this syntax when dropping relationships between tables created before ThoughtSpot version 3.2. This is because relationships could not be named in older versions.
	Drops the foreign key or relationship between two tables where you cannot reference it by relationship name. If the relationship was created without a name, use:
	• the name of the referenced table, for a foreign key.
	• the name of the related table, for a relationship.
	If you drop a foreign key without specifying the referenced table, all foreign keys from the table you are altering will be dropped.
	Examples:
	`ALTER TABLE "shipments" DROP CONSTRAINT FOREIGN KEY "orders";`
	`ALTER TABLE "wholesale_buys" DROP RELATIONSHIP WITH "retail_sales";`
	Drops all relationships that have wholesale_buys as a source.
	`ALTER TABLE "wholesale_buys" DROP RELATIONSHIP;`
	Drops all foreign keys from wholesale_buys.
	`ALTER TABLE "wholesale_buys" DROP CONSTRAINT FOREIGN KEY;`

Syntax	Description

`ALTER TABLE <table> [SET DIMENSION | SET FACT [PARTITION BY HASH [(<shards>)] [KEY(<column>)]]]`	Changes the partitioning on a table by doing one of:
	• re-sharding a sharded table
	• changing a replicated table to a sharded table
	• changing a sharded table to a replicated (unsharded) table
	By default, ThoughtSpot does not shard dimension tables.
	To change the partitioning on a table, or to change a dimension table to a sharded table, use `ALTER TABLE...SET FACT PARTITION BY HASH...;`
	To make a sharded table into a dimension table (replicated on every node), use `ALTER TABLE...SET DIMENSION;` command.
	Examples of this statement:
	`ALTER TABLE "sales_fact" SET FACT PARTITION BY HASH (96) KEY ("PO_number");`
	`ALTER TABLE "fruit_dim" SET DIMENSION;`

`ALTER TABLE <table> MODIFY COLUMN <column> <new_data_type>;`	Changes the data type of a column. This can have implications on sharding and primary key behavior. See About data type conversion. For example: `ALTER TABLE fact100 MODIFY COLUMN product_id int;`

Modify data

Syntax	Description
`INSERT INTO <table> VALUES ...`	Inserts values into a table. Only use this for testing. Do not use `INSERT` on a production system. For example: `INSERT INTO "vendor" VALUES ('helen rose', 'jacob norse', 'eileen ruff', 'manny gates');`
`ALTER TABLE <table> SET LOAD PRIORITY <value> <new_data_type>;`	Sets the load priority for a table. Load priority determines the order in which a table is loaded on a cluster restart. You can set any value from `1-100` . The system default for all tables is `50` . For example: `ALTER TABLE 'sales_facts' SET LOAD PRIORITY 1;`

Syntax	Description
`UPDATE <table> ... SET ... [WHERE ...]`	Updates rows in a table that match optionally provided predicates. Predicates have the form `column = value` connected by the `AND` keyword. Sets the column values to the specified values. For example: `UPDATE "location" SET "borough" = 'staten island', "city" = 'new york' WHERE "borough" = 'staten isl' AND city = 'NY';`
`DELETE FROM <table> [WHERE...]`	Deletes rows from a table that match optionally provided predicates. Predicates have the form `column = value` connected by the `AND` keyword. For example. `DELETE FROM "vendor" WHERE "name" = 'Joey Smith' AND "vendorid" = '19463';`

Constraints and relationships

Constraints and relationships in ThoughtSpot are used to define the relationships between tables (i.e. how they can be joined). However, constraints are not enforced, as they would be in a transactional database. You can define the following constraints when creating a table with `CREATE TABLE`, or add them to an existing table using the `ADD CONSTRAINT` syntax:

Syntax	Description

PRIMARY KEY	Designates a unique, non-null value as the primary key for a table. This can be one column or a combination of columns. If values are not unique, an upsert will be performed if a row includes a primary key that is already present in the data. Some examples: </p>
	`CREATE TABLE "schools" ( "schoolID" varchar(15), "schoolName" varchar(255), "schoolCity" varchar(55), "schoolState" varchar(55), "schoolNick" varchar(55), CONSTRAINT PRIMARY KEY ("schoolID") ) ;`
	`ALTER TABLE "cart" ADD CONSTRAINT PRIMARY KEY ("cart_id");`
	`ALTER TABLE "cart" DROP CONSTRAINT PRIMARY KEY "cart_id";`

Syntax	Description

FOREIGN KEY	Defines a relationship where the value(s) in the table are used to join to a second table. Uses an equality operator. The foreign key must match the primary key of the table that is referenced in number, column type, and order of columns.
	
	When creating a foreign key, give it a name. You can reference the foreign key name later, if you want to remove it.
	
	Examples of this statement:
	
	`ALTER TABLE "batting" ADD CONSTRAINT "FK_player" FOREIGN KEY ("playerID") REFERENCES "players" ("playerID");`
	
	`ALTER TABLE "batting" ADD CONSTRAINT "FK_lg_team" FOREIGN KEY ("lgID" ,"teamID") REFERENCES "teams" ("lgID" ,"teamID");`
	
	`ALTER TABLE "shipment" ADD CONSTRAINT "FK_PO_vendor" FOREIGN KEY ("po_number", "vendor") REFERENCES "orders" ("po_number", "vendor");`
	
	`ALTER TABLE "shipment" DROP CONSTRAINT "FK_PO_vendor";`

RELATIONSHIP	Defines a relationship where the value(s) in the table can be used to join to a second table, using an equality condition (required) and one or more range conditions (optional). These conditions act like a WHERE clause when the two tables are joined. They are applied using AND logic, such that all conditions must be met for a row to be included.
	
	You may add multiple relationships between tables. When creating a relationship, give it a name. You can reference the relationship name later, if you want to remove it.
	
	Examples of this statement:
	
	`ALTER TABLE "wholesale_buys" ADD RELATIONSHIP "REL_fruit" WITH "retail_sales" AS "wholesale_buys"."fruit" = "retail_sales"."fruit" AND ("wholesale_buys"."date_order" < "retail_sales"."date_sold" AND "retail_sales"."date_sold" < "wholesale_buys"."expire_date");`
	
	`ALTER TABLE "wholesale_buys" DROP RELATIONSHIP "REL_fruit";`

Data types

ThoughtSpot supports a simplified list of data types:

Syntax	Description	Examples
Character	• VARCHAR(*n*)	Specify the maximum number of characters, as in VARCHAR(255). The size limit is 1GB for VARCHAR values.
Floating point	• DOUBLE • FLOAT	DOUBLE is recommended.
Boolean	• BOOL	Can be `true` or `false`.
Integer	• INT • BIGINT	INT holds 32 bits. BIGINT holds 64 bits.
Date or time	• DATE • DATETIME • TIMESTAMP • TIME	DATETIME, TIMESTAMP, and TIME are stored at the granularity of seconds . TIMESTAMP is identical to DATETIME, but is included for syntax compatibility.

tsload flag reference

For recurring data loads and for scripting loads, use `tsload` (the ThoughtSpot Loader). This reference section lists all the flags that can be used to modify the behavior of `tsload`.

General tsload flags

Flag	Description	Notes
`--target_database <database>`	Specifies the pre-existing target database into which tsload should load the data.	
`--target_schema <schema>`	Specifies the target schema.	Default is “falcon_default_schema”.
`--target_table <table>`	Specifies the tables that you want to load into the database.	The tables must exist in the database specified by `--target_database`.
`--empty_target`	Specifies that any data in the target table is to be removed before the new data is loaded.	If supplied, any rows that exist in the table specified by `--target_database` and `--target table` will be deleted before this data load. To perform an “upsert” on the existing data, omit this flag or specify `--noempty_target`.
`--max_ignored_rows <number>`	Specifies the maximum number of rows that can be ignored if they fail to load.	If the number of ignored rows exceeds this limit, the load will be aborted.
`--bad_records_file <path_to_file>/<file_name>`	Specifies the file to use for storing rows that failed to load.	Input rows that do not conform to the defined schema in ThoughtSpot will be ignored and inserted into this file.
`--date_format <date_formatmask>`	Specifies the format string for date values.	The default format is `yearmonthday` e.g. “Dec 30th, 2001” and is represented as `20011230`. Use the date format specifications supported in the strptime library function.
`--date_time_format <date_formatmask> <time_formatmask>`	Specifies the format string for datetime values.	The default is yearmonthday hour:minute:second e.g. Dec 30th, 2001 1:15:12 and is represented as 20011230 01:15:12. Use the datetime format specifications supported in the strptime library function.
`--time_format <time_formatmask>`	Specifies the format string for time values.	The default is hour:minute:second. Use the time format specifications supported in the strptime library function.

Flag	Description	Notes
`--v=[0|1|2|3`	Specifies the verbosity of log messages.	Provide a value for verbosity level. By default, verbosity is set to the minimum, which is 0. This value is similar to a volume control. At higher levels your log receives more messages and that log more frequently. This is used for debugging. You should not change this value unless instructed by ThoughtSpot Support.
`--skip_second_fraction`	Skips fractional seconds when loading data.	If supplied, the upserts logic may be affected, especially if the date time being loaded is a primary key, and the data has millisecond granularity. Load the data twice, once as a string with a primary key, and again with second granularity date time. There is no support to store fractional seconds in the ThoughtSpot system.

File loading tsload flags

The following flags are used when loading data from an input file:

Flag	Description	Notes
`--source_file <path_to_file>/<file_name>`	Specifies the location of the file to be loaded.	
`--source_data_format [csv|delimited]`	Specifies the data file format.	Optional. The default is csv.
`--field_separator "<delimiter>"`	Specifies the field delimiter used in the input file.	
`-- trailing_field_separator`	Specifies that the field separator appears after every field, including the last field per row.	Example row with trailing field separator: a,b,c,The default is false.
`--null_value "<null_representation>"`	Specifies how null values are represented in the input file.	These values will be converted to NULL upon loading.
`--date_converted_to_epoch [true|false]`	Specifies whether the “date” or “datetime” values in the input file are represented as epoch values.	
`--boolean_representation [true_false | 1_0 | T_F | Y_N]`	Specifies the format in which boolean values are represented in the input file.	The default is T_F. You can also use this flag to specify other values. For example, if your data used Y for true and NULL for false, you could specify: `--boolean_representation Y_NULL`

Flag	Description	Notes
`--has_header_row`	Indicates that the input file contains a header row.	If supplied, column names in the header row are used to match column names in the target table in ThoughtSpot. If not supplied, the first row of the file is loaded as data, the same as all subsequent rows.
`--escape_character "<character>"`	Specifies the escape character used in the input file.	If no value is specified, the default is “(double quotes).
`--enclosing_character "<character>"`	Specifies the enclosing character used in the input file.	If the enclosing character is double quotes, you need to escape it, as in this example: `--enclosing_character "\""`
`--use_bit_boolean_values = [true | false]`	Specifies how boolean values are represented in the input file.	If supplied, the input CSV file uses a bit for boolean values, i.e. the false value is represented as 0x0 and true as 0x1. If omitted or set to false, boolean values are assumed to be T_F, unless you specify something else using the flag `--boolean_representation [true_false | 1_0 | T_F | Y_N]` .

tscli command reference

The `tscli` command line interface is an administration interface for the ThoughtSpot instance. Use `tscli` to take snapshots (backups) of data, apply updates, stop and start the services, and view information about the system. This reference defines each subcommand and what you can accomplish with it.

The command returns 0 upon success and a non-zero exit code upon failure. Because the `tscli` command is typically running a command on multiple codes, an error may be called at different points. As much as possible, the command attempts to save errors to the `stderr` directory as configured on a node.

How to use the tscli command

The `tscli` command has the following syntax:

```
tscli [-h] [--helpfull] [--verbose] [--noautoconfig]
          [--autoconfig] [--yes] [--cluster <cluster>]
          [--zoo <zookeeper>] [--username username] [--identit
y_file identity_file]
          {access,alert,backup,backup-policy,callhome,cassandr
a,cluster,command,dr-mirror,etl,event,feature,fileserver,firewa
ll,hdfs,ipsec,ldap,logs,map-tiles,monitoring,nas,node,patch,rpa
ckage,saml,scheduled-pinboards,smtp,snapshot,snapshot-policy,sp
ot,sssd,ssl,storage,support,tokenauthentication}
```

The `tscli` command has several subcommands such as `alert`, `backup`, and so forth. You issue a subcommand using the following format:

```
tscli [subcommand ]
```

Subcommands have their own additional options and actions such as `tscli backup create` or `tscli backup delete` for example. To view help for a subcommand:

```
tscli [subcommand] -h
```

A subcommand itself may have several options.

tscli subcommands

This section lists each subcommand and its syntax.

access

```
tscli access [-h] {list} ...
```

Use this subcommand to do the following:

- `tscli access list` Lists objects by last access time.

alert

```
tscli alert [-h] {count,info,list,off,on,refresh,silence,statu
s,unsilence} ...
```

Use this subcommand to do the following:

- `tscli alert info` Lists all alerts.
- `tscli alert list` Lists the generated alerts.
- `tscli alert off` Disables all alerts from the cluster in the cluster's timezone.
- `tscli alert on` Enables alerts from the cluster.
- `tscli alert silence --name <alert_name>`

  Silences the alert with *`alert_name`*. For example, DISK_ERROR. Silenced alerts are still recorded in postgres, however emails are not sent out.

- `tscli alert status` Shows the status of cluster alerts.
- `tscli alert unsilence-name` *`alert_name`*

  Unsilences the alert with *`alert_name`*. For example, `DISK_ERROR`.

backup

```
tscli backup [-h] {create,delete,ls,restore} ...
```

Use this subcommand to do the following:

- `tscli backup create [-h] [--mode {full,light,dataless}] [--type {full,incremental}] [--base BASE] [--storage_type {local,nas}] [--remote] name out`

  Pulls a snapshot and saves it as a backup where:

  - `--mode {full,light,dataless}`

Mode of backups. To understand these diffrent modes see Understand backup modes.

- `--type {full,incremental}` Type of backup.(Incremental `incremental` is not implemented yet) (default: full)
- `--base BASE`

  Based snapshot name for incremental backup. (Not Implemented yet) (default: None)
- `--storage_type {local,nas}`

  Storage type of output directory. (default: local)
- `--remote`
  Take backup through orion master. (default: True)

- `tscli backup delete *` name `*` Deletes the named backup.
- `tscli backup ls` List all backups taken by the system.
- `tscli backup restore` Restore cluster using backup.

backup-policy

```
tscli backup-policy [-h] {create,delete,disable,enable,ls,sho
w,status,update} ...
```

Use this subcommand to do the following:

- `tscli backup-policy create` Prompts an editor for you to edit the parameters of the backup policy.
- `tscli backup-policy delete` *`name`* Deletes the backup policy with *`name`*.
- `tscli backup-policy disable` *`name`* Disables the policy *`name`*.
- `tscli backup-policy enable` *`name`* Enables the policy *`name`*.
- `tscli backup-policy ls` List backup policies.
- `tscli backup-policy show` *`name`* Show the policy *`name`*.
- `tscli backup-policy status` *`name`* Enables the policy *`name`*.
- `tscli backup-policy update *` name `*` Prompts an editor for you to edit the policy *`name`*.

callhome

```
tscli callhome [-h] {disable,enable,generate-bundle} ...
```

Use this subcommand to do the following:

- `tscli callhome disable` Turns off the periodic call home feature.
- `tscli callhome enable --customer_name` *`customer_name`*`

  Enables the “call home” feature, which sends usage statistics to ThoughtSpot This feature is enabled by default.

The parameter `customer_name` takes the form `Shared/*`customer_name`*/stats`.

- `tscli callhome generate-bundle` –d `directory` `--since` `DAYS`
  - `--d D` Dest folder where tar file will be created. (default: None)
  - `--since` `DAYS`

    Grab callhome data from this time window in the past. Should be a human readable duration string, e.g. `4h` (4 hours), `30m` (30 minutes), `1d` (1 day). (default: None) Generates a tar file of the cluster metrics and writes it to the specified directory where `DAYS` is how far back you'd like to generate the tar file from in days. For example, `30`. If this parameter is not specified, the command will collect the stats from the last `7` days by default.

cassandra

```
tscli cassandra [-h] {backup,restore} ...
```

Use this subcommand to do the following:

- `tscli cassandra backup` Take a backup of cassandra
- `tscli cassandra restore` Restore cassandra from a backup

cluster

```
tscli cluster [-h] {abort-reinstall-os,check,create,get-confi
g,load,reinstall-os,report,restore,resume-reinstall-os,resume-u
pdate,set-config,set-min-resource-spec,show-resource-spec,star
t,status,stop,update,update-hadoop} ...
```

Use this subcommand to do the following:

- `tscli cluster abort-reinstall-os` Abort in-progress reinstall.
- `tscli cluster check --includes {all,disk,zookeeper,hdfs,orion-cgroups,orion-oreo}` Check the status nodes in the cluster.

  You must specify a component to check.

- `tscli cluster create` `release`

  Creates a new cluster from the release file specified by `release`. This command is used by ThoughtSpot Support when installing a new cluster, for example, `tscli cluster create 2.0.4.tar.gz`

- `tscli cluster get-config` Get current cluster network and time configuration. Prints JSON configuration to stdout. If for some reason the system cannot be connected to all interfaces, the command returns an error but continues to function.
- `tscli cluster load` Load state from given backup onto existing cluster
- `tscli cluster reinstall-os` Reinstall OS on all nodes of the cluster.
- `tscli cluster report` Generate cluster report.
- `tscli cluster restore --release` `release_tarball` `backupdir`

Restores a cluster using the backup in the specified directory `backupdir`. If you're restoring from a dataless backup, you must supply the release tarball for the corresponding software release.

- `tscli cluster resume-reinstall-os` Resume in-progress reinstall.
- `tscli cluster resume-update` Resume in-progress updates.
- `tscli cluster set-config` Set cluster network and time configuration. Takes JSON configuration from stdin.
- `tscli cluster set-min-resource-spec` Sets min resource configuration of the cluster
- `tscli cluster show-resource-spec` Prints default or min.
- `tscli cluster start` Start cluster.
- `tscli cluster status` Gives the status of the cluster, including release number, date last updated, number of nodes, pending tables time, and services status.
- `tscli cluster stop` Pauses the cluster (but does not stop storage services).
- `tscli cluster update` Update existing cluster.
- `tscli cluster update-hadoop` Updates Hadoop/Zookeeper on the cluster.

command

```
tscli command [-h] {run} ...
```

Command to run a command on all nodes.

`tscli command run [-h] [--nodes NODES ] --dest_dir DEST_DIR [--copyfirst COPYFIRST ] [--timeout TIMEOUT ] command`

- `--nodes NODES` Space separated IPs of nodes where you want to run the command. (default: `all`)
- `--dest_dir DEST_DIR` Directory to save the files containing output from each nodes. (default: None)
- `--copyfirst COPYFIRST` Copy the executable to required nodes first. (default: `False`)
- `--timeout TIMEOUT` Timeout waiting for the command to finish. (default: `60`)

dr-mirror

```
tscli dr-mirror [-h] {start,status,stop} ...
```

- `tscli dr-mirror start` Starts a mirror cluster which will continuously recover from a primary cluster.
- `tscli dr-mirror status` Checks whether the current cluster is running in mirror mode.
- `tscli dr-mirror stop` Stops mirroring on the local cluster.

etl

```
tscli etl [-h] {change-password,disable-lw,download-agent,enable-lw,show-lw} ...
```

- `tscli etl change-password --admin_username admin_user --username Informatica_user`

  Changes the Informatica Cloud account password used by ThoughtSpot Data Connect. Required parameters are:

  - `--admin_username admin_user` specifies the Administrator username for ThoughtSpot.
  - `--username Informatica_user` specifies the username for the Informatica Cloud.

- `tscli etl disable-lw` Disables ThoughtSpot Data Connect.
- `tscli etl download-agent` Downloads the ThoughtSpot Data Connect agent to the cluster.
- `tscli etl enable-lw [-h] --username USERNAME --thoughtspot_url THOUGHTSPOT_URL --admin_username ADMIN_USERNAME [--groupname GROUPNAME ] --org_id ORG_ID [--pin_to PIN_TO ] [--proxy_host PROXY_HOST ] [--proxy_port PROXY_PORT ] [--proxy_username PROXY_USERNAME ] [--max_wait MAX_WAIT ]`

  You should contact ThoughtSpot Support for assistance in setting this up. Required parameters are:

  - `--username USERNAME` Username for Informatica Cloud (default: None)
  - `--thoughtspot_url THOUGHTSPOT_URL` URL to reach thoughtspot. (default: None)
  - `--admin_username ADMIN_USERNAME` Admin username for ThoughtSpot (default: None)
  - `--groupname GROUPNAME`
  - `--org_id ORG_ID` specifies the Informatica `id` of the organization (company). For ThoughtSpot, this is `001ZFA`. `org_id` shouldn't include the prefix `Org`. For example, if on Informatica cloud, the `orgid` is `Org003XYZ`, then use only
  - `--pin_to PIN_TO` specifies the IP address to pin to. If you specify an IP to pin to, that node becomes sticky to the Informatica agent, and will always be used. Defaults to the public IP address of the localhost where this command was run.
  - `--proxy_host PROXY_HOST` Proxy server host for network access (default:)
  - `--proxy_port PROXY_PORT` Proxy server port (default:)
  - `--proxy_username PROXY_USERNAME` Proxy server username (default:)
  - `--max_wait MAX_WAIT` Maximum time in seconds to wait for Data Connect agentto start (default: None)

- `tscli etl show-lw` Shows the status of ThoughtSpot Data Connect. It also returns the Informatica username and OrgId.

event

```
tscli event [-h] {list} ...
```

This subcommand has the following actions:

`tscli event list [-h] [--include INCLUDE ] [--since SINCE ] [--from FROM ] [--to TO ] [--limit LIMIT ] [--detail] [--summary_contains SUMMARY_CONTAINS ] [--detail_contains DETAIL_CONTAINS ] [--attributes ATTRIBUTES ]`

- `--include` *INCLUDE* Options are all, config, notification. Default config. (default: config)
- `--since` *SINCE* Grab events from this time window in the past. Should be a human readable duration string, e.g. `4h` (4 hours), `30m` (30 minutes), `1d` (1 day). (default: None)
- `--from` *FROM* Begin timestamp, must be of the form: `yyyymmdd-HH:MM` (default: None)
- `--to` *TO* End timestamp, must be of the form: `yyyymmdd-HH:MM` (default: None)
- `--limit` *LIMIT* Max number of events to fetch. (default: 0)
- `--detail` Print events in detail format. This is not tabular. Default is a tabular summary. (default: False)
- `--summary_contains` *SUMMARY_CONTAINS* Summary of the event will be checked for this string. Multiple strings to check for can be specified by separating them with `|` (event returned if it matchesALL). Put single quotes around the param value to prevent undesired glob expansion (default: None)
- `--detail_contains` *DETAIL_CONTAINS* Details of the event will be checked for this string. Multiple strings to check for can be specified by separating them with `|` (event returned if it matches ALL). Put single quotes around the param value to prevent undesired glob expansion (default: None)
- `--attributes` *ATTRIBUTES* Specify attributes to match as key=value. Multiple attributes to check for can be specified by separating them with `|` (event returned if it matches ALL). Put single quotes around the param value to prevent undesired glob expansion (default: None)

feature

```
tscli feature [-h] {get-all-config} ...
```

This subcommand has the following actions:

`tscli feature get-all-config` Gets the configured features in a cluster. The command will return a list of features, such as custom branding, Data Connect, and call home, and tell you whether they are enabled or disabled.

fileserver

```
tscli fileserver [-h] {configure,download-release,purge-confi
g,show-config,upload} ...
```

This subcommand has the following actions:

- `tscli fileserver configure [-h] --user` *USER* `[--password` *PASSWORD* `]` Configures the secure file server username and password for file upload/download and the call home feature. You only need to issue this command once, to set up the connection to the secure file server. You only need to reissue this command if the password changes. The parameter *PASSWORD* is optional. If a password is not specified, you will be prompted to enter it.
- `tscli fileserver download-release [-h] [--user` *USER* `] [--password` *PASSWORD* `]` *release* Downloads the specified release file and its checksum. Specify the release by number, to the second decimal point (e.g. 3.1.0, 3.0.5, etc.). You may optionally specify the `--user` and `--password` to bypass the credentials that were specified when configuring the file server connection with `tscli fileserver configure` . Before using this command for the first time, you need to set up the file server connection using `tscli`

`fileserver configure`.

- `tscli fileserver purge-config` Removes the file server configuration.
- `tscli fileserver show-config` Shows the file server configuration.
- `tscli fileserver upload [-h] [--user` *`USER`* `] [--password` *`PASSWORD`* `] --file_name` *`FILE_NAME*` *–server_dir_path* `*` *SERVER_DIR_PATH`*

  Uploads the file specified to the directory specified on the secure file server. You may optionally specify the `--user` and `--password` to bypass the credentials that were specified when configuring the file server connection with `tscli fileserver configure`. Before using this command for the first time, you need to set up the file server connection using `tscli fileserver configure`.

  Accepts these flags

  - `--user` *`USER`* Username of fileserver (default: None)
  - `--password` *`PASSWORD`* Password of fileserver (default: None). This is required and the command prompts you for it if you do not supply it.
  - `--file_name` *`FILE_NAME`* Local file that needs to be uploaded (default: None)
  - `--server_dir_path` *`SERVER_DIR_PATH`* Directory path on fileserver. (default: None) The *`SERVER_DIR_PATH`* parameter specifies the directory to which you want to upload the file. It is based on your customer name, and takes the form `/Shared/support/*` customer_name `*`.

firewall

```
tscli firewall [-h] {close-ports,disable,enable,open-ports,stat
us} ...
```

- `tscli firewall close-ports`

  Closes given ports through firewall on all nodes. Takes a list of ports to close, comma separated. Only closes ports which were previously opened using “open-ports”. Ignores ports which were not previously opened with “open-ports” or were already closed.

- `tscli firewall disable` Disable firewall.
- `tscli firewall enable` Enable firewall.
- `tscli firewall open-ports` `--ports` *`ports`*

  Opens given ports through firewall on all nodes. Takes a list of ports to open, comma separated. Ignores ports which are already open. Some essential ports are always kept open (e.g. `ssh`), they are not affected by this command or by `close-ports`.

- `tscli firewall status` Shows whether firewall is currently enabled or disabled.

hdfs

```
tscli hdfs [-h] {leave-safemode} ...
```

This subcommand has the following actions:

`tscli hdfs leave-safemode` Command to get HDFS namenodes out of safemode.

ipsec

```
tscli ipsec [-h] {disable,enable,status} ...
```

This subcommand has the following actions:

`tscli ipsec disable` Disable IPSec `tscli ipsec enable` Enable IPSec `tscli ipsec status` Show IPSec status on all nodes

ldap

```
tscli ldap [-h] {add-cert,configure,purge-configuration} ...
```

This subcommand has the following actions:

- `tscli ldap add-cert` *`name`* *`certificate`*

  Adds an SSL certificate for LDAP. Use only if LDAP has been configured without SSL and you wish to add it. Use `*` name `*` to supply an alias for the certificate you are installing.

- `tscli ldap configure`

  Configures LDAP using an interactive script. You can see detailed instructions for setting up LDAP in About LDAP integration.

- `tscli ldap purge-configuration` Purges (removes) any existing LDAP configuration.

logs

```
tscli logs [-h] {collect,runcmd} ...
```

This subcommand has the following actions:

- `tscli logs collect [-h] [--include` *`INCLUDE`* `] [--exclude` *`EXCLUDE`* `] [--since` *`SINCE`* `] [--from` *`FROM`* `] [--to` *`TO`* `] [--out` *`OUT`* `] [--maxsize` *`MAXSIZE`* `] [--sizeonly] [--nodes` *`NODES`* `]`

  Extracts logs from the cluster. Does not include any logs that have been deleted due to log rotation.

  These parameters have the following values:

  - `--include` *`INCLUDE`*

Specifies a comma separated list of logs to include. Each entry is either a "selector" or a glob for matching files.Selectors must be among: `all`, `orion`, `system`, `ts`. Anything starting with `/` is assumed to be a glob pattern and interpreted via `find(1)`. Other entries are ignored. Put single quotes around the param value to prevent undesired glob expansion (default: `all`)

- `--exclude` *EXCLUDE*

  Comma separated list of logs to exclude. Applies to the list selected by –include. Params are interpreted just like in –include (default: None)

- `--since` *SINCE*

  Grab logs from this time window in the past. Should be a human readable duration string, e.g. 4h (4 hours), 30m (30 minutes), 1d (1 day). (default: None)

- `--from` *FROM* Timestamp where collection begins, must be of the form: `yyyymmdd-HH:MM` (default: None)
- `--to` *TO* Timestamp where collection ends, must be of the form: `yyyymmdd-HH:MM` (default: None)
- `--out` *OUT* Tarball path for dumping logs from each node (default: `/tmp/logs.tar.gz`)
- `--maxsize` *MAXSIZE* Only fetch logs if size is smaller that this value. Can be specified in megabytes/gigabytes, e.g. 100MB, 10GB. (default: None)
- `--sizeonly` Do not collect logs. Just report the size. (default: False)
- `--nodes` *NODES* Comma separated list of nodes from where to collect logs. Skip this to use all nodes. (default: None)

- `tscli logs runcmd [-h] --cmd` *CMD* `[--include` *INCLUDE* `] [--exclude` *EXCLUDE* `] [--since` SINCE] `[--from` *FROM* `] [--to` *TO* `] [--outfile` *OUTFILE* `] [--outdir` *OUTDIR* `] [--cmd_infmt` *CMD_INFMT* `] [--cmd_outfmt` *CMD_OUTFMT* `] [--nodes` *NODES* `]`

  Runs a Unix command on logs in the cluster matching the given constraints. Results are reported as text dumped to standard out, the specified output file, or as tarballs dumped into the specified directory.

  - `--cmd` *CMD*

    Unix-Command to be run on the selected logs. Use single quotes to escape spaces etc. Language used to specify CMDSTR has following rules.

    - A logfile and its corresponding result file can be referred by keywords `SRCFILE` & `DSTFILE`. eg. `cp SRCFILE DSTFILE`
    - Without any reference to DSTFILE in CMDSTR, `> DSTFILE` will be appended to CMDSTR for output redirection. eg `du -sch SRCFILE` gets auto- transtalted to `du -sch SRCFILE > DSTFILE`
    - Without any reference to SRCFILE, content of log is streamed to CMDSTR via pipe. eg. `tail -n100 | grep ERROR` gets auto-transtalted to `cat SRCFILE | tail -n100 | grep ERROR > DSTFILE` (default: None)

  - `--include` *INCLUDE*

Comma separated list of logs to include,each entry is either a “selector” or a glob for matching files.Selectors must be among: `all`, `orion`, `system`, `ts`. Anything starting with / is assumed to be a glob pattern and interpreted via `find(1)`. Other entries are ignored. TIP: put single quotes around the param value to prevent undesired glob expansion (default: all)

- `--exclude` *EXCLUDE*

  Comma separated list of logs to exclude. Applies to the list selected by `--include`. Params are interpreted just like in `--include` (default: None)

- `--since` *SINCE*

  Grab logs from this time window in the past. Should be a human readable duration string, e.g. `4h` (4 hours), `30m` (30 minutes), `1d` (1 day). (default: None)

- `--from` *FROM* Timestamp where collection begins, must be of the form: `yyyymmdd-HH:MM` (default: None)
- `--to` *TO* Timestamp where collection ends, must be of the form: `yyyymmdd-HH:MM` (default: None)
- `--outfile` *OUTFILE* File path for printing all the results. By default printed to stdout (default: None)
- `--outdir` *OUTDIR* Directory path for dumping results with original dir structure from each node. Used as an alternative to printing output to outfile/stdout (default: None)
- `--cmd_infmt` *CMD_INFMT* Specify if the inputfile should be compressed/uncompressed before running `CMD`. `C` =compressed, `U` =uncompressed. Don’t use this flag if `CMD` works on both (default: None)
- `--cmd_outfmt` *CMD_OUTFMT* Specify if *OUTFILE* generated by `CMD` will be compressed/uncompressed. `C` =compressed, `U` =uncompressed. Don’t use this flag if output file will be of same format as input file (default: None)
- `--nodes` *NODES* Comma separated list of nodes where to run command. Skip this to use all nodes. (default: None)

map-tiles

```
tscli map-tiles [-h] {disable,enable,status} ...
```

This subcommand supports the following actions:

- `tscli map-tiles enable [-h] [--online] [--offline] [--tar TAR] [--md5 MD5]`

  Enables ThoughtSpot’s map tiles, which are used when constructing geomap charts. If you don’t have interest access, you must download the map tiles tar and md5 files. Then you must append the following to the `tscli` command.

  - `--online` Download `maptiles` tar from internet. (default: True)
  - `--offline` Using `maptiles` tar from local disk. (default: False)
  - `--tar` *TAR* Specified tar file for map-tiles. (default:)
  - `--md5` *MD5* Specified md5 file for map-tiles. (default:)

- `tscli map-tiles disable` Disable map-tiles functionality.
- `tscli map-tiles status` Check whether map-tiles is enabled.

monitoring

```
tscli monitoring [-h] {set-config,show-config} ...
```

This subcommand has the following actions:

- `tscli monitoring set-config [-h] [--email EMAIL] [--clear_email] [--heartbeat_interval HEARTBEAT_INTERVAL] [--heartbeat_disable] [--report_interval REPORT_INTERVAL] [--report_disable]` Sets the monitoring configuration.
  - `--email` *`EMAIL`* Comma separated list (no spaces) of email addresses where the cluster will send monitoring information.
  - `--clear_email` Disable emails by clearing email configuration. (default: False)
  - `--heartbeat_interval` *`HEARTBEAT_INTERVAL`* Heartbeat email generation interval in seconds. Should be greater than 0.
  - `--heartbeat_disable` Disable heartbeat email generation. (default: False)
  - `--report_interval` *`REPORT_INTERVAL`* Cluster report email generation interval in seconds. Should be greater than 0.
  - `--report_disable` Disable cluster report email generation. (default: False)
- `tscli monitoring show-config` Shows the monitoring configuration.

nas

```
tscli nas [-h] {ls,mount-cifs,mount-nfs,unmount} ...
```

This subcommand has the following actions:

- `tscli nas ls [-h]` List mounts managed by NAS mounter service.
- `tscli nas mount-cifs [-h] --server` *`SERVER`* `[--path_on_server` *`PATH_ON_SERVER`* `] --mount_point` *`MOUNT_POINT`* `--username` *`USERNAME`* `--password` *`PASSWORD`* `[--uid` *`UID`* `] [--gid` *`GID`* `] [--options` *`OPTIONS`* `]`

  Mounts a CIFS device on all nodes.
  - `--server` *`SERVER`* IP address or DNS name of CIFS service. For example, `10.20.30.40` (default: None)
  - `--path_on_server` *`PATH_ON_SERVER`* Filesystem path on the CIFS server to mount (source). For example: `/a` (default: `/`)
  - `--mount_point` *`MOUNT_POINT`*

Directory on all cluster nodes where the NFS filesystem should be mounted (target). This directory does not need to already exist. If this directory already exists, a new directory is not created and the existing directory is used for mounting. For example: `/mnt/external` (default: None)

- `--username` *`USERNAME`* Username to connect to the CIFS filesystem as (default: None)
- `--password` *`PASSWORD`* CIFS password for `--username` (default: None)
- `--uid` *`UID`*

  `UID` that will own all files or directories on the mounted filesystem when the server does not provide ownership information. See `man mount.cifs` for more details. (default: `1001`)

- `--gid` *`GID`*

  Gid that will own all files or directories on the mounted filesystem when the server does not provide ownership information. See `man mount.cifs` for more details. (default: `1001`)

- `--options` *`OPTIONS`* Other command-line options to forward to `mount.cifs` command (default: `noexec`)

- `tscli nas mount-nfs [-h] --server SERVER [--protocol PROTO --path_on_server PATH_ON_SERVER] --mount_point MOUNT_POINT [--options OPTIONS]`

  Mounts a NFS device on all nodes. Parameters are:

  - `--server` `SERVER` IP address or DNS name of NFS service. For example, `10.20.30.40` (default: None)
  - `--path_on_server` `PATH_ON_SERVER` Filesystem path on the NFS server to mount (source). For example: `/a/b/c/d` (default: `/`)
  - `--mount_point` `MOUNT_POINT`

    Directory on all cluster nodes where the NFS filesystem should be mounted (target). This directory does not need to already exist. If this directory already exists, a new directory is not created and the existing directory is used for mounting. For example: `/mnt/external` (default: None)

  - `--options` `OPTIONS` Command-line options to forward to mount command (default: `noexec`).
  - `--protocol PROTO` One of `nfs` or `nfs4` . The default is `nfs` .

- `tscli nas unmount [-h] --dir` *`DIR`*

  Unmounts all devices from the specified *`DIR`* (directory) location. This command returns an error if nothing is currently mounted on this directory via `tscli nas mount` (default: None)

node

```
tscli node [-h] {check,ls,reinstall-os,resume-reinstall-os,stat
us} ...
```

This subcommand has the following actions:

- `tscli node check [-h] [--select {reinstall-preflight}] [--secondary SECONDARY ]`

  Run checks per node. Takes the following parameters:

  - `--select {reinstall-preflight}` Select the type of node check (default: `reinstall-preflight`)
  - `--secondary` *`SECONDARY`* Secondary drive for `reinstall-preflight` (default: `sdd`)

- `tscli node ls [-h] [--type {all,healthy,not-healthy}]` Filter by node state (default: `all`)
- `tscli node reinstall-os [-h] [--secondary` *`SECONDARY`* `] [--cluster]` Reinstall OS on a node. This takes the following parameters:

  - `--secondary` *`SECONDARY`* Secondary drive to be used to carry to reinstall (default: `sdd`)
  - `--cluster` Is the node part of a cluster (default: `False`)

- `tscli node resume-reinstall-os` Resume in-progress reinstall

patch

```
tscli patch [-h] {apply,ls,resume-apply,resume-rollback,rollback} ...
```

This subcommand has the following actions:

- `tscli patch apply [-h] [` *`release`* `]`

  Apply the patch on an existing cluster. Takes the following parameters:

  - *`release`* The relative path to the patch tar ball

- `tscli patch ls [-h] [--applied] [--rolled_back] [--service SERVICE] [--md5 MD5] [--history]` Lists the patches currently applied. This takes the following parameters:

  - `--applied` Show only the patches applied since last full release (default: ` False`)
  - `--rolled_back` Show only the patches rolled back since last full release (default: ` False`)
  - `--service SERVICE` Show patches filtered by service (default: ` None`)
  - `--md5 MD5` Shows the details of the patch specified (default: ` None`)
  - `--history` Shows the history of all patch apply/rollback release (default: ` False`)

- `tscli patch resume-apply [-h]`

  Resume patch apply

- `tscli patch resume-rollback [-h]`

  Resume patch roll-backup

- `tscli patch rollback [-h]`

  Rollback the patch from an existing cluster

rpackage

```
tscli rpackage [-h] {add,delete,list} ...
```

Manages R packages available to SpotIQ.

- `tscli rpackage add [-h] [--repo REPO ] [--timeout TIMEOUT ] [--dest_dir DEST_DIR ] [--nodes NODES ] package_name` Command to add an R `package_name` to the cluster. This command has the following options:
  - `--repo REPO` Specify the url of a specific repo to download packages
  - `--timeout REPO` Timeout waiting for the R Package to be installed (default: 60)
  - `--dest_dir REPO` Directory where output of this command will be placed (default: None)
  - `--nodes NODES` Space separated IPs of nodes where you want to run the command. (default: all).
- `tscli rpackage add [-h] [--timeout TIMEOUT ] [--dest_dir DEST_DIR ] [--nodes NODES ] package_name` Command to delete an installed R package from the cluster. This command has the following options:
  - `--timeout REPO` Timeout waiting for the R Package to be removed (default: 60)
  - `--dest_dir REPO` Directory where output of this command will be placed (default: None)
  - `--nodes NODES` Space separated IPs of nodes where you want to run the command. (default: all).
- `tscli rpackage list [-h] [--detailed]` List all R packages installed on the cluster.

saml

```
tscli saml [-h] {configure,purge-configuration}
```

This subcommand has the following actions:

- `tscli saml configure [-h]` Configures SAML. To see a list of prerequisites refer to Configure SAML.
- `tscli saml purge-configuration` Purges any existing SAML configuration.

scheduled-pinboards

```
tscli scheduled-pinboards [-h] {disable,enable}
```

This subcommand has the following actions:

- `tscli scheduled-pinboards disable [-h]` Disable scheduled pinboards for this cluster.
- `tscli scheduled-pinboards enable [-h]` Enables scheduled pinboards, which is disabled in prod clusters by default.

> **Note:** When you enable scheduled pinboards, you should also configure a whitelist of intended email domains. Contact ThoughtSpot Support for help configuring a whitelist.

smtp

```
tscli smtp [-h] {remove-mailfromname,remove-mailname,remove-rel
ayhost,remove-saslcredentials,reset-canonical-mapping,set-canon
ical-mapping,set-mailfromname,set-mailname,set-relayhost,set-sa
slcredentials,show-canonical-mapping,show-mailfromname,show-mai
lname,show-relayhost}
```

This subcommand takes supports the following actions:

- `tscli smtp remove-mailfromname` Removes current cluster mailfromname
- `tscli smtp remove-mailname` Removes current cluster mailname
- `tscli smtp remove-relayhost` Removes current cluster relayhost
- `tscli smtp remove-saslcredentials` Clears SASL credentials and disables SMTP AUTH
- `tscli smtp reset-canonical-mapping` Deletes the current postmap mapping.
- `tscli smtp set-canonical-mapping [-h]` *`new_key`* *`new_value`* Sets a new Postmap mapping.
- `tscli smtp set-mailfromname` *`mailfromname`* Sets the name, an email address, from which email alerts are sent, for the cluster.
- `tscli smtp set-mailname` *`mailname`* Sets the mailname, a domain, where email alerts are sent, for the cluster.
- `tscli smtp set-relayhost [-h] [--force FORCE] relayhost` Sets the Relay Host for SMTP (email) sent from the cluster.
    - `--force` *`FORCE`* Set even if relay host is not accessible. (default: `False`)
- `tscli smtp set-saslcredentials` Sets SASL credentials and enables SMTP AUTH
- `tscli smtp show-canonical-mapping` Shows the current postmap mapping.
- `tscli smtp show-mailfromname` Shows the mailname, from which email alerts are sent, for the cluster.
- `tscli smtp show-mailname` Shows the mailname, where email alerts are sent, for the cluster.
- `tscli smtp show-relayhost` Shows the for SMTP (email) sent from the cluster. If there is no Relay Host configured, the command returns `NOT FOUND`.

snapshot

```
tscli snapshot [-h] {backup,create,delete,ls,pin,restore,unpi
n,update-ttl}
```

Learn more about snapshots and backups see the Understand the backup strategies documentation. This subcommand supports the following actions:

- `tscli snapshot backup [-h] [--mode {full,light,dataless}] [--type {full,incremental}] [--base` *`BASE`* `] [--storage_type {local,nas}] [--remote]` *`name`* *`out`*

  Pull snapshot out as a backup. This takes the following parameters:

  - `--mode {full,light,dataless}` Mode of backups. (default: `full`)
  - *`name`* Name of snapshot to pull out as a backup. To list all snapshots, run `tscli snapshot ls`.
  - *`out`* Directory where backup will be written, must not already exist.
  - `--type {full,incremental}` Type of backup.(Incremental backup is not implemented yet) (default: `full`)
  - `--base` *`BASE`* Based snapshot name for incremental backup. (Not Implemented yet) (default: None)
  - `--storage_type {local,nas}` Storage type of output directory. (default: `local`)
  - `--remote` Take backup through Orion master. (default: `True`)

- `tscli snapshot create [-h]` *`name`* *`reason`* *`ttl`*

  Creates a new snapshot with the *`name`* and *`reason`* provided. This command does not accept `.` (periods), but does accept `-` (dashes). The `ttl` parameters is the number of days after which this snapshot will be automatically deleted. A value of `-1` disables automatic deletion.

- `tscli snapshot pin [-h]` *`name`* Pins a snapshot so it cannot be deleted or garbage collected.
- `tscli snapshot delete [-h]` *`name`* Deletes the named snapshot.
- `tscli snapshot ls [-h]` List available snapshots.
- `tscli snapshot restore [-h] [--allow_release_change] [--only_service_state]` *`name`* Restore cluster to an existing snapshot. This takes the following parameters:

  - `--allow_release_change` Allow restoration to a snapshot at a different release. (default: `False`)
  - `--only_service_state` Restore only service state. (default: `False`)

- `tscli snapshot unpin [-h]` *`name`* Unpin a snapshot so it can be deleted or garbage collected.

- `tscli snapshot update-ttl [-h] [--disable` *`DISABLE`* `]` *`name`* *`ttl`*

  Updates manual snapshot garbage collection policy.

- `name` Specifies which snapshot to update.
- `ttl` Extends the manual snapshot `ttl` (time-to-live) value. Use a positive value to increase `ttl`. Use negative value to decrease it.
- `--disable` *`DISABLE`* Disable manual snapshot garbage collection. Setting this value to `True` will override any `ttl` value. (default: False)

snapshot-policy

```
tscli snapshot-policy [-h] {disable,enable,show,update}
```

This subcommand supports the following actions:

- `tscli snapshot-policy disable [-h]` Disable snapshot policy.
- `tscli snapshot-policy enable -h` Enable specified snapshot policy.
- `tscli snapshot-policy show [-h]` Show snapshot policy.
- `tscli snapshot-policy update [-h] [--config` *`CONFIG`* `]` Update periodic snapshot config. This takes the following parameter:
  - `--config` *`CONFIG`* Text format of periodic backup policy config (default: None)

spot

```
tscli spot [-h] {enable} ...
```

Enables Spot integration. This subcommand supports the following actions:

`tscli spot enable [-h] --token` *`TOKEN`* `--thoughtspot_url` *`THOUGHTSPOT_URL`* `[--cache_timeout` *`CACHE_TIMEOUT`* `]`

- `--token` *`TOKEN`* Slack authroization token for Spot bot. This is required. You receive this token when your Slack administrator adds the Spot application.
- `--thoughtspot_url` *`THOUGHTSPOT_URL`* URL for the ThoughtSpot application. This is required.
- `--cache_timeout` *`CACHE_TIMEOUT`* Internal cache timeout (default: `60000`)

ssl

```
tscli ssl [-h] {add-cert,clear-min-tls-version,off,on,rm-cert,set-min-tls-version,status,tls-status} ...
```

This subcommand supports the following actions:

- `tscli ssl add-cert [-h]` *`key certificate`* Adds an SSL certificate, key pair.
- `tscli ssl clear-min-tls-version [-h]` Clears any customizations for the minimum TLS version to support.

- `tscli ssl off`

  Disables SSL. Disabling SSL will stop users from seeing a security warning when accessing ThoughtSpot from a browser if there is no SSL certificate installed.

- `tscli ssl on [-h]` If SSL is enabled and there is no certificate, users will see a security warning when accessing ThoughtSpot from a browser.
- `tscli ssl rm-cert` Removes the existing SSL certificate, if any.
- `tscli ssl set-min-tls-version [-h] {1.0,1.1,1.2}` Sets the minimum supported TLS version. Sets the minimum SSL version to be supported by the ThoughtSpot application. Please ensure that client browsers are enabled for this version or newer.
- `tscli ssl status` Shows whether SSL authentication is enabled or disabled.
- `tscli ssl tls-status [-h]` Prints the status of TLS support.

sssd

```
tscli sssd {enable, disable, set-sudo-group, clear-sudo-group}
...
```

This subcommand uses system security services daemon (SSSD), and has the following actions:

- `tscli sssd enable --user` *`USER`* `--domain` *`DOMAIN`*

  Enables system Active Directory (AD) user access on a single node. You will be prompted for password credentials. The user must have permission to join a computer or VM to the domain.

- `tscli sssd disable`

  Disables system AD based access on a local node. Running this command will also remove the AD group from sudoers list.

- `tscli sssd set-sudo-group` *`ACTIVE_DIRECTORY_GROUP_NAME`*

  Allows `sudo` permissions for AD group.

- `tscli sssd clear-sudo-group` *`ACTIVE_DIRECTORY_GROUP_NAME`*

  Clears any set AD sudo group.

For more about setting up Active Directory access, see Enable Active Directory based access.

storage

```
tscli storage [-h] {gc,df} ...
```

This subcommand supports the following actions:

- `tscli storage gc [-h] [--log_age` *`LOG_AGE`* `] [--force] [--localhost_only]`

  Garbage collect unused storage. Before issuing this command, you must stop the cluster using `tscli cluster stop`. After garbage collection has completed, you can restart the cluster with `tscli cluster start`. The command frees space in these directories:

- `/tmp`
- `/usr/local/scaligent/logs/`
- `/export/logs/orion`
- `/export/logs/oreo`
- `/export/logs/hadoop`
- `/export/logs/zookeeper`
- `cores`

Accepts these optional flags:

- `--log_age` *`LOG_AGE`*

  Delete logs older than these many hours. Use a non-zero value ideally. A zero value will cause all temporary files to be deleted, including say those which are just temporarily closed while they are being passed from one component to the next. (default: `4`)

- `--force` Forces deletion of all logs and temporary files regardless of age. This must only be run on a stopped cluster. (default: False)

- `--localhost_only` If used, only the logs on the localhost will be removed. If not specified, the command acts on the entire cluster.

- `tscli storage df [--mode disk|hdfs]`

  Checks the disk usage on the relevant mounts. Returns output similar to the Linux system command `df -h <directory>`.

support

```
tscli support [-h]
{bundle,restart-remote,rm-admin-email,rm-admin-phone,rm-feedback-email,set-admin-email,set-admin-phone,set-debug-ui-password,set-feedback-email,set-remote,show-admin-email,show-admin-phone,show-feedback-email,show-remote,start-remote,stop-remote} ...
```

This subcommand supports the following actions:

- `tscli support bundle [-h] [--include INCLUDE] [--exclude EXCLUDE] [--list_selectors] [--since SINCE] [--from FROM] [--to TO] [--out OUT] [--nodes NODES]`

  - `--include` *`INCLUDE`* Comma separated list of selectors to include, each entry is either a “selector” or a glob for matching files. To see the list of valid selectors, run this command with `--list_selectors`. You may also specify: “`all` to get all selectors and logs, and “basic” to get only the basic selectors. Selectors may also be selectors meant for logs collect: all, orion, system, ts, or the name of a service. Anything starting with / is assumed to be a glob pattern and interpreted via find(1). Other entries are ignored. TIP: put single quotes around the param value to prevent undesired glob expansion. Use “all” to collect all selectors and all logs (default: all_but_logs)

  - `--exclude` *EXCLUDE* Comma separated list of selectors to exclude. Applies to the list selected by `--include`. Params are interpreted just like in `--include`. Use the special keyword “logs” to exclude logs collection all together. (default: None)
  - `--list_selectors` List the selectors available for `--include` and `--exclude`, and then exit. (default: `False`)
  - `--since` *SINCE* Grab logs from this time window in the past. Should be a human readable duration string, e.g. `4h` (4 hours), `30m` (30 minutes), `1d` (1 day). (default: None)
  - `--from` *FROM* Timestamp where collection begins, must be of the form: `yyyymmdd-HH:MM` (default: None)
  - `--to` *TO* Timestamp where collection ends, must be of the form: `yyyymmdd-HH:MM` (default: None)
  - `--out` *OUT* Tarball path for dumping the support bundle (default: `/tmp/support_bundle.tar.gz`)
  - `--nodes` *NODES* Comma separated list of nodes from where to collect logs. Skip this to use all nodes. (default: None)

- `tscli support restart-remote` Restarts remote support.
- `tscli support rm-admin-email` Removes the email address for contacting the customer administrator. Replaces it with the default ThoughtSpot Support email address.
- `tscli support rm-feedback-email` Removes the email address for product feedback. Replaces it with the default ThoughtSpot Support email address.
- `tscli support rm-admin-phone` Removes the phone number for contacting the customer administrator. Replaces it with the default ThoughtSpot Support phone number.
- `tscli support rm-feedback-email` Removes the email for sending feedback out of the system. If you would like to set a blank email address, issue the command `tscli support set-feedback-email ' '`.
- `tscli support set-admin-email` *email* Sets the email address for contacting the customer administrator. If you would like to display a blank email address, issue the command `tscli support set-admin-email ' '`.
- `tscli support set-feedback-email` *email* Sets the email address for sending feedback. If you would like to display a blank email address, issue the command `tscli support set-feedback-email ' '`.
- `tscli support set-admin-phone` *phone_number* Sets the phone number for contacting the customer administrator. Specify a phone number using any value (e.g. `+1 800-508-7008 Ext. 1`). If you would like to display a blank phone number, issue the command `tscli support set-admin-phone`.
- `tscli support set-remote [-h] [--addr` *ADDR* `] [--user` *USER* `]` Configures the cluster for remote support through SSH tunneling, where *ADDR* is the address of support, e.g. t`unnel.thoughtspot.com`, and *USER* is the support username.
- `tscli support show-admin-email` Shows the email address for customer administrator, if set.
- `tscli support show-feedback-email` Shows the email address for product feedback, if set.
- `tscli support show-admin-phone` Shows the phone number for customer administrator, if set.
- `tscli support show-remote` Shows the status and configuration of remote support.
- `tscli support start-remote` Starts remote support.
- `tscli support stop-remote` Stops remote support.

tokenauthentication

```
tscli cli tokenauthentication [-h] {disable,enable}
```

- `tscli cli tokenauthentication enable` Generates a token.
- `tscli cli tokenauthentication disable` Purges token login configuration.

Date and time formats reference

This is a references for the date and time contexts and formats you can use with ThoughtSpot. You define data formats in specific contexts and, depending on the context, your choices in data formatting differ. You must understand date and time when you load data in these contexts:

- using data upload from the browser
- through `tsload` command
- through data connect or another extract, transform, load (ETL) tool

Data loading formats do not change how data is displayed in tables and charts.

The context where you *can control* date and time formats is data modeling. Data modeling controls how data is displayed in search and their resulting answers.

Data loading formats via tsload

When loading via the `tsload` command you must specify `date` and `timestamp` formats using the format specifications defined in the `strptime` library function. Data is imported based on the timezone of the node from which `tsload` is run.

For `date` data types, the default format is `%Y%m%d` which translates to `yearmonthday` For example, `Dec 30th, 2001` is represented as `20011230`. For `time` and `datetime` data types, the default is `%Y%m%d %H:%M:%S` which translates to `yearmonthday hour:minute:second`, for example, `Dec 30th, 2001 1:15:12` is represented as `20011230 01:15:12`.

Data modeling formats for browser data upload

These date and time formats are supported in a CSV file when uploading via the browser. You cannot specify the date format; ThoughtSpot will pick the format that fits your data best:

- 1/30/2014
- 2014-01-30
- 2014-1-30
- 30-Jan-2014
- 2014-Jan-30
- 2014-01-30 10:32 AM
- 2014-01-30 14:52
- 2014-01-30 10:32:22
- 2014-01-30 10:32:22 AM
- 2014-01-30 10:32:22.0
- 2014-01-30 10:32:22.0 AM
- 2014-01-30 10:32:22.000
- 2014-01-30 10:32:22.000 AM
- 1/30/2014
- 30-Jan-14
- 01-Mar-02 (assumes 2002)
- 30/1/2014 10:32 AM
- 30/1/2014 14:52
- 30/1/2014 10:32:22
- 30/1/2014 10:32:22 AM
- 30/1/2014 10:32:22.0
- 30/1/2014 10:32:22.0 AM

- 30/1/2014 10:32:22.000
- 30/1/2014 10:32:22.000 AM
- 30-Jan-14 10:32 AM
- 30-Jan-14 14:52
- 30-Jan-14 10:32:22
- 30-Jan-14 10:32:22 AM
- 30-Jan-14 10:32:22.0
- 30-Jan-14 10:32:22.0 AM
- 30-Jan-14 10:32:22.000
- 30-Jan-14 10:32:22.000 AM
- Fri Jan 30 2014 3:26 PM
- Fri Jan 30 2014 13:46
- Fri Jan 30 2014 10:32:22
- Fri Jan 30 2014 10:32:22 AM
- Fri Jan 30 2014 10:32:22.0
- Fri Jan 30 2014 10:32:22.0 AM
- Fri Jan 30 2014 10:32:22.000
- Fri Jan 30 2014 10:32:22.000 AM
- 14:52
- 10:32 AM
- 10:32:22
- 10:32:22 AM
- 10:32:22.0
- 10:32:22.000
- 10:32:22.0 AM
- 10:32:22.000 AM

Data loading formats via data connect or another ETL tool

Data that is loaded via ETL arrives through ODBC or JDBC connection. After you extract the data from the source and before you load it into ThoughtSpot, you must transform any date or timestamp into a valid format for ThoughtSpot. Once transformed, no explicit data masking is required. See the data integration guide for more details of loading data via ODBC and JDBC.

Data modeling formats

A user with administrative rights can configure data modeling for data on one or all files. You can set number, date, and currency display formats. These formats define how these value types display in tables and charts. See the Admin Guide for more information about data modeling settings. The following format strings are available for use:

Format mask	Description
`YYYY` or `yyyy`	four digit year such as `2017`
`YY` or `yy`	last two digits of year such as `17`
`M`	month with no leading zero `1` - `12`
`MM`	Two digit month `01` - `12`

Format mask	Description
`MMM`	Three letter month such as `Jan`
`D`	Day of year without a leading zero `0` - `365`
`DD`	Day of year with up to one leading zero `01` - `365`
`DDD`	Day of year with up to two leading zeroes `001` - `365`
`d`	Day of month with no leading zero `1` - `31`
`dd`	Two digit day of month `01` - `31`
`HH`	Two digit `24` hour representation of hour `00` - `23`
`hh`	Two digit `12` hour representation of hour `01` - `12`
`H`	24 hour representation of hour with no leading zero `0` - `23`
`h`	12 hour representation of hour with no leading zero `1` - `12`
`mm`	Minutes `00` - `59`
`m`	Minutes with no leading zero `0` - `59`
`ss`	Seconds `00` - `59`
`s`	Seconds with no leading zero `0` - `59`
`a`	AM/PM indicator

Valid delimiters include most non-alphabet characters. This includes but is not limited to:

- `\` (forward slash)
- `/` (backward slash)
- `|` (pipe symbol)
- `:` (colon)
- `-` (dash)
- `_` (underscore)
- `=` (equal sign)

Examples of valid format masks you can produce for display are as follows:

- `MM/dd/yyyy`
- `MMM`
- `DD/MM/yyyy`
- `MM/dd/yyyy HH:mm`
- `DD/MM/yyyy HH:mm`

Row level security rules reference

ThoughtSpot allows you to create row level security rules using expressions. If an expression evaluates to “true” for a particular row and group combination, that group will be able to see that row. This reference lists the various operators and functions you can use to create rules.

For information on how to use the row level security functions and operators, see About Rule-Based Row Level Security. There is a special variable called ts_groups, which you can use when creating row level security rules. It fetches a list of the groups that the currently logged in user belongs to. For each row, f the expression in the rule evaluates to ‘true’ for any one of these groups, that row will be shown to the user.

You can also see this list of operators and examples from within the Rule Builder by selecting **Rule Assistant**.

Conversion functions

These functions can be used to convert data from one data type to another. Conversion to or from date data types is not supported.

Function	Description	Examples
`to_bool`	Returns the input as a boolean (true or false).	`to_bool (0) = false` `to_bool (married)`
`to_date`	Accepts a date represented as an integer or text string, and a second string parameter that can include strptime date formatting elements. Replaces all the valid strptime date formatting elements with their string counterparts and returns the result. Does not accept epoch formatted dates as input.	`to_date (date_sold, '%Y-%m-%d')`
`to_double`	Returns the input as a double.	`to_double ('3.14') = 3.14` `to_double (revenue * .01)`
`to_integer`	Returns the input as an integer.	`to_integer ('45') + 1 = 46` `to_integer (price + tax - cost)`

`to_string`	Returns the input as a text string. To convert a date to a string, specify the date format you want to use.	`to_string (45 + 1) = '46'`
		`to_string (revenue - cost)`
		`to_string (date,('%m/%d/%y'))`

Date functions

Function	Description	Examples
`add_days`	Returns the result of adding the specified number of days to the given date.	`add_days (01/30/2015, 5) = 02/04/2015` `add_days (invoiced, 30)`
`date`	Returns the date portion of a given date.	`date (home visit)`
`day`	Returns the number (1-31) of the day for the given date.	`day (01/15/2014) = 15` `day (date ordered)`
`day_number_of_quarter`	Returns the number of the day in a quarter for a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`day_number_of_quarter (01/30/2015) = 30` `day_number_of_quarter (01/30/2015, 'fiscal') = 91`
`day_number_of_week`	Returns the number (1-7) of the day in a week for a given date with 1 being Monday and 7 being Sunday.	`day_number_of_week(01/15/2014) = 3` `day_number_of_week (shipped)`

Function	Description	Examples

`day_number_of_year`	Returns the number (1-366) of the day in a year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`day_number_of_year (01/30/2015) = 30`
		`day_number_of_year ( 01/30/2015, 'fiscal' ) = 275`
		`day_number_of_year (invoiced)`

`day_of_week`	Returns the day of the week for the given date.	`day_of week (01/30/2015) = Friday` `day_of_week (serviced)`
`diff_days`	Subtracts the second date from the first date and returns the result in number of days, rounded down if not exact.	`diff_days (01/15/2014, 01/17/2014) = -2` `diff_days (purchased, shipped)`
`diff_time`	Subtracts the second date from the first date and returns the result in number of seconds.	`diff_time (01/01/2014, 01/01/2014) = -86,400` `diff_time (clicked, submitted)`
`hour_of_day`	Returns the hour of the day for the given date.	`hour_of_day (received)`
`is_weekend`	Returns true if the given date falls on a Saturday or Sunday.	`is_weekend (01/31/2015) = true` `is_weekend (emailed)`
`month`	Returns the month from the given date.	`month (01/15/2014) = January` `month (date ordered)`

Function	Description	Examples

`month_number`	Returns the number (1-12) of the month from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`month_number (09/20/2014) = 9`
		`month_number ( 09/20/2014, 'fiscal' ) = 5`
		`month_number (purchased)`

`month_number_of_quarter`	Returns the month (1-3) number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`month_number_of_quarter (02/20/2018) = 2` `month_number_of_quarter (02/20/2018,'fiscal' ) = 1`
`now`	Returns the current timestamp.	`now ()`

`quarter_number`	Returns the number (1-4) of the quarter associated with the given date. Add an optional second parameter to specify 'fiscal' or 'calendar' dates. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`quarter_number ( 04/14/2014) = 2`
		`quarter_number ( 04/14/2014, 'fiscal' ) = 4`
		`quarter_number ( shipped )`

Function	Description	Examples
`start_of_month`	Returns `MMM yyyy` for the first day of the month. Your installation configuration can override this setting so that it returns a different format such as `MM/dd/yyyy` . Speak with your ThoughtSpot administrator for information on doing this.	`start_of_month ( 01/31/2015 ) = Jan FY 2015` `start_of_month (shipped)`

`start_of_quarter`	Returns the date for the first day of the quarter for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`start_of_quarter ( 04/01/2014) = Apr 2014`
		`start_of_quarter ( 04/01/2014, 'fiscal') = Feb 2014`
		`start_of_quarter (sold)`

`start_of_week`	Returns the date for the first day of the week for the given date.	`start_of_week ( 06/01/2015 ) = 05/30/2015 Week` `start_of_week (emailed)`

`start_of_year`	Returns the date for the first day of the year for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`start_of_year (04/01/2014) returns Jan 2014`
		`start_of_year (04/01/2014, 'fiscal') returns May 2013`
		`start_of_year (joined)`

`time`	Returns the time portion of a given date.	`time (3/1/2002 10:32) = 10:32` `time (call began)`

Function	Description	Examples
`week_number_of_month`	Returns the week number for the given date in a month.	`week_number_of_month(03/23/2017) = 3`
`week_number_of_quarter`	Returns the week number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`week_number_of_quarter (04/03/2017) = 1` `week_number_of_quarter (04/03/2017, 'fiscal') = 10`
`week_number_of_year`	Returns the week number for the given date in a year. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`week_number_of_year (01/17/2014) = 3` `week_number_of_year ( 01/17/2014, 'fiscal') = 38`

`year`	Returns the year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01. Per standard convention, the fiscal year is defined by the year-end date.)	`year (01/15/2014) = 2014`
		`year (12/15/2013, 'fiscal' ) = 2014`
		`year (date ordered)`

Mixed functions

These functions can be used with text and numeric data types.

Function	Description	Examples
`!=`	Returns true if the first value is not equal to the second value.	`3 != 2 = true` `revenue != 1000000`
`<`	Returns true if the first value is less than the second value.	`3 < 2 = false` `revenue < 1000000`
`<=`	Returns true if the first value is less than or equal to the second value.	`1 <= 2 = true` `revenue <= 1000000`
`=`	Returns true if the first value is equal to the second value.	`2 = 2 = true` `revenue = 1000000`
`>`	Returns true if the first value is greater than the second value.	`3 > 2 = true` `revenue > 1000000`
`>=`	Returns true if the first value is greater than or equal to the second value.	`3 >= 2 = true` `revenue >= 1000000`
`greatest`	Returns the larger of the values.	`greatest (20, 10) = 20` `greatest (q1 revenue, q2 revenue)`
`least`	Returns the smaller of the values.	`least (20, 10) = 10` `least (q1 revenue, q2 revenue)`

Number functions

Function	Description	Examples
`*`	Returns the result of multiplying both numbers.	`3 * 2 = 6` `price * taxrate`
`+`	Returns the result of adding both numbers.	`1 + 2 = 3` `price + shipping`
`-`	Returns the result of subtracting the second number from the first.	`3 - 2 = 1` `revenue - tax`
`/`	Returns the result of dividing the first number by the second.	`6 / 3 = 2` `markup / retail price`

Function	Description	Examples
`^`	Returns the first number raised to the power of the second.	`3 ^ 2 = 9` `width ^ 2`
`abs`	Returns the absolute value.	`abs (-10) = 10` `abs (profit)`
`acos`	Returns the inverse cosine in degrees.	`acos (0.5) = 60` `acos (cos-satellite-angle)`
`asin`	Returns the inverse sine (specified in degrees).	`asin (0.5) = 30` `asin (sin-satellite-angle)`
`atan`	Returns the inverse tangent in degrees.	`atan (1) = 45` `atan (tan-satellite-angle)`
`atan2`	Returns the inverse tangent in degrees.	`atan2 (10, 10) = 45` `atan2 (longitude, latitude)`
`cbrt`	Returns the cube root of a number.	`cbrt (27) = 3` `cbrt (volume)`
`ceil`	Returns the smallest following integer.	`ceil (5.9) = 6` `ceil (growth rate)`
`cos`	Returns the cosine of an angle (specified in degrees).	`cos (63) = 0.45` `cos (beam angle)`
`cube`	Returns the cube of a number.	`cube (3) = 27` `cube (length)`
`exp`	Returns Euler's number (~2.718) raised to a power.	`exp (2) = 7.38905609893` `exp (growth)`
`exp2`	Returns 2 raised to a power.	`exp2 (3) = 8` `exp2 (growth)`
`floor`	Returns the largest previous integer.	`floor (5.1) = 5` `floor (growth rate)`
`ln`	Returns the natural logarithm.	`ln (7.38905609893) = 2` `ln (distance)`
`log10`	Returns the logarithm with base 10.	`log10 (100) = 2` `log10 (volume)`
`log2`	Returns the logarithm with base 2 (binary logarithm).	`log2 (32) = 5` `log2 (volume)`
`mod`	Returns the remainder of first number divided by the second number.	`mod (8, 3) = 2` `mod ( revenue , quantity )`

Function	Description	Examples
`pow`	Returns the first number raised to the power of the second number.	`pow (5, 2) = 25` `pow (width, 2)`
`random`	Returns a random number between 0 and 1.	`random ( ) = .457718` `random ( )`
`round`	Returns the first number rounded to the second number (the default is 1).	`round (35.65, 10) = 40` `round (battingavg, 100)`
`safe_divide`	Returns the result of dividing the first number by the second. If the second number is 0, returns 0 instead of NaN (not a number).	`safe_divide (12, 0) = 0` `safe_divide (total_cost, units)`
`sign`	Returns +1 if the number is greater than zero, -1 if less than zero, 0 if zero.	`sign (-250) = -1` `sign (growth rate)`
`sin`	Returns the sine of an angle (specified in degrees).	`sin (35) = 0.57` `sin (beam angle)`
`spherical_distance`	Returns the distance in km between two points on Earth.	`spherical_distance (37.465191, -122.153617, 37.421962, -122.142174) = 4,961.96` `spherical_distance (start_latitude, start_longitude, start_latitude, start_longitude)`
`sq`	Returns the square of a numeric value.	`sq (9) = 81` `sq (width)`
`sqrt`	Returns the square root.	`sqrt (9) = 3` `sqrt (area)`
`tan`	Returns the tangent of an angle (specified in degrees).	`tan (35) = 0.7` `tan (beam angle)`

Operators

Operator	Description	Examples

`and`	Returns true when both conditions are true, otherwise returns false.	`(1 = 1) and (3 > 2) = true`
		`lastname = 'smith' and state ='texas'`
		Note: Not available for row level security (RLS) formulas.

`if…then…else`	Conditional operator.	`if (3 > 2) then 'bigger' else 'not bigger'` `if (cost > 500) then 'flag' else 'approve'`
`ifnull`	Returns the first value if it is not null, otherwise returns the second.	`ifnull (cost, 'unknown')`
`isnull`	Returns true if the value is null.	`isnull (phone)`
`not`	Returns true if the condition is false, otherwise returns false.	`not (3 > 2) = false` `not (state = 'texas')`
`or`	Returns true when either condition is true, otherwise returns false.	`(1 = 5) or (3 > 2) = true` `state = 'california' or state ='oregon'`

Text functions

Function	Description	Examples
`concat`	Returns the one or more values as a concatenated text string. Be sure to use single quotes instead of double quotes around each of the strings.	`concat ( 'hay' , 'stack' ) = 'haystack'` `concat (last_name , first_name )`
`contains`	Returns true if the first string contains the second string, otherwise returns false.	`contains ('broomstick', 'room') = true` `contains (product, 'trial version')`

Function	Description	Examples
`edit_distance`	Accepts two text strings. Returns the edit distance (minimum number of operations required to transform one string into the other) as an integer. Works with strings under 1023 characters.	`edit_distance ('attorney', 'atty') = 4` `edit_distance (color, 'red')`
`edit_distance_with_cap`	Accepts two text strings and an integer to specify the upper limit cap for the edit distance (minimum number of operations required to transform one string into the other). If the edit distance is less than or equal to the specified cap, returns the edit distance. If it is higher than the cap, returns the cap plus 1. Works with strings under 1023 characters.	`edit_distance_with_cap ('pokemon go', 'minecraft pixelmon', 3) = 4` `edit_distance_with_cap (event, 'burning man', 3)`
`similar_to`	Accepts a document text string and a search text string. Returns true if relevance score (0-100) of the search string with respect to the document is greater than or equal to 20. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document.	`similar_to ('hello world', 'hello swirl') = true` `similar_to (current team, drafted by)`
`similarity`	Accepts a document text string and a search text string. Returns the relevance score (0-100) of the search string with respect to the document. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document. If the two strings are an exact match, returns 100.	`similarity ('where is the burning man concert', 'burning man') = 46` `similarity (tweet1, tweet2)`
`spells_like`	Accepts two text strings. Returns true if they are spelled similarly and false if they are not. Works with strings under 1023 characters.	`spells_like ('thouhgtspot', 'thoughtspot') = true` `spells_like (studio, distributor)`

Function	Description	Examples
`strlen`	Returns the length of the text.	`strlen ('smith') = 5` `strlen (lastname)`
`strpos`	Returns the numeric position (starting from 0) of the first occurrence of the second string in the first string, or -1 if not found.	`strpos ('haystack_with_needles', 'needle') = 14` `strpos (complaint, 'lawyer')`
`substr`	Returns the portion of the given string, beginning at the location specified (starting from 0), and of the given length.	`substr ('persnickety', 3, 7) = snicket` `substr (lastname, 0, 5)`

Variables

These variables can be used in your expressions.

Function	Description	Examples
`ts_groups`	Returns a list of all the groups the current logged in user belongs to. For any row, if the expression evaluates to true for any of the groups, the user can see that row.	`ts_groups = 'east'`
`ts_username`	Returns the user with the matching neame.	`ts_username != 'mark'`

Formula function reference

ThoughtSpot allows you to create derived columns in worksheets using formulas. You create these columns by building formulas using the **Formula Assistant**. An individual formula is constructed from *n* combination of operators and functions.

This reference lists the various operators and functions you can use to create formulas.

Operators

Operator	Description	Examples

`and`	Returns true when both conditions are true, otherwise returns false.	`(1 = 1) and (3 > 2) = true`
		`lastname = 'smith' and state ='texas'`
		Note: Not available for row level security (RLS) formulas.

`if…then…else`	Conditional operator.	`if (3 > 2) then 'bigger' else 'not bigger'` `if (cost > 500) then 'flag' else 'approve'`
`ifnull`	Returns the first value if it is not null, otherwise returns the second.	`ifnull (cost, 'unknown')`
`isnull`	Returns true if the value is null.	`isnull (phone)`
`not`	Returns true if the condition is false, otherwise returns false.	`not (3 > 2) = false` `not (state = 'texas')`
`or`	Returns true when either condition is true, otherwise returns false.	`(1 = 5) or (3 > 2) = true` `state = 'california' or state ='oregon'`

Aggregate functions (group aggregate)

These functions can be used to aggregate data.

Function	Description	Examples
`average`	Returns the average of all the values of a column.	`average (revenue)`

Function	Description	Examples
`average_if`	Returns the average of all the columns that meet a given criteria.	`average_if(city = "San Francisco", revenue)`
`count`	Returns the number of rows in the table containing the column.	`count (product)`
`count_if`	Returns the number of rows in the table containing the column.	`count_if(region ='west', region)`
`cumulative_average`	Takes a measure and one or more attributes. Returns the average of the measure, accumulated by the attribute(s) in the order specified.	`cumulative_average (revenue, order date, state)`
`cumulative_max`	Takes a measure and one or more attributes. Returns the maximum of the measure, accumulated by the attribute(s) in the order specified.	`cumulative_max (revenue, state)`
`cumulative_min`	Takes a measure and one or more attributes. Returns the minimum of the measure, accumulated by the attribute(s) in the order specified.	`cumulative_min (revenue, campaign)`
`cumulative_sum`	Takes a measure and one or more attributes. Returns the sum of the measure, accumulated by the attribute(s) in the order specified.	`cumulative_sum (revenue, order date)`

`group_aggregate`	Takes a measure and, optionally, attributes and filters. These can be used to aggregate measures with granularities and filters different from the terms/columns used in the search. Especially useful for comparison analysis.
	
	This formula takes the form: group_aggregate (< aggregation (measure) >, < groupings >, < filters >)
	
	Lists can be defined with {} and optional list functions `query_groups` or `query_filters` , which by default specify the lists or filters used in the original search. Plus (+) or (-) can be used to add or exclude specific columns for query groups.
		`group_aggregate (sum (revenue) , {ship mode, date} , {} )`
		
		`group_aggregate (sum (revenue) , {ship mode , date}, {day_of_week (date) = 'friday'} )`
		
		`group_aggregate (sum (revenue) , query_groups() , query_filters() )`
		
		`group_aggregate (sum (revenue) , query_groups() + {date} , query_filters() )`

Function	Description	Examples
`group_average`	Takes a measure and one or more attributes. Returns the average of the measure grouped by the attribute(s).	`group_average (revenue, customer region, state)`
`group_count`	Takes a measure and one or more attributes. Returns the count of the measure grouped by the attribute(s).	`group_count (revenue, customer region)`
`group_max`	Takes a measure and one or more attributes. Returns the maximum of the measure grouped by the attribute(s).	`group_max (revenue, customer region)`
`group_min`	Takes a measure and one or more attributes. Returns the minimum of the measure grouped by the attribute(s).	`group_min (revenue, customer region)`
`group_stddev`	Takes a measure and one or more attributes. Returns the standard deviation of the measure grouped by the attribute(s).	`group_stddev (revenue, customer region)`
`group_sum`	Takes a measure and one or more attributes. Returns the sum of the measure grouped by the attribute(s).	`group_sum (revenue, customer region)`
`group_unique_count`	Takes a measure and one or more attributes. Returns the unique count of the measure grouped by the attribute(s).	`group_unique_count (product , supplier)`
`group_variance`	Takes a measure and one or more attributes. Returns the variance of the measure grouped by the attribute(s).	`group_variance (revenue, customer region)`
`max`	Returns the maximum value of a column.	`max (sales)`
`max_if`	Returns the maximum value among columns that meet a criteria.	`max_if( (revenue > 10) , customer region )`
`min`	Returns the minimum value of a column.	`min (revenue)`
`min_if`	Returns the minimum value among columns that meet a criteria.	`min_if( (revenue < 10) , customer region )`

Function	Description	Examples
`moving_average`	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1…Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the average of the measure over the given window. The attributes are the ordering columns used to compute the moving average.	`moving_average (revenue, 2, 1, customer region)`
`moving_max`	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1…Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the maximum of the measure over the given window. The attributes are the ordering columns used to compute the moving maximum.	`moving_max (complaints, 1, 2, store name)`
`moving_min`	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1…Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the minimum of the measure over the given window. The attributes are the ordering columns used to compute the moving minimum.	`moving_min (defects, 3, 1, product)`
`moving_sum`	Takes a measure, two integers to define the window to aggregate over, and one or more attributes. The window is (current - Num1…Current + Num2) with both end points being included in the window. For example, “1,1” will have a window size of 3. To define a window that begins before Current, specify a negative number for Num2. Returns the sum of the measure over the given window. The attributes are the ordering columns used to compute the moving sum.	`moving_sum (revenue, 1, 1, order date)`

Function	Description	Examples
stddev	Returns the standard deviation of all values of a column.	stddev (revenue)
stddev_if	Returns a standard deviation values filtered to meet a specific criteria.	stddev_if((revenue > 10) , (revenue/10.0))
sum	Returns the sum of all the values of a column.	sum (revenue)
sum_if	Returns sum values filtered by a specific criteria.	sum_if(region='west', revenue)
unique count	Returns the number of unique values of a column.	unique count (customer)
unique_count_if	Returns the number of unique values of a column provided it meets a criteria.	unique_count_if((revenue > 10) , order date)
variance	Returns the variance of all the values of a column.	variance (revenue)
variance_if	Returns the variance of all the values of a column provided it meets a criteria..	variance_if((revenue > 10) , (revenue/10.0))

Conversion functions

These functions can be used to convert data from one data type to another. Conversion to or from date data types is not supported.

Function	Description	Examples
to_bool	Returns the input as a boolean (true or false).	to_bool (0) = false to_bool (married)
to_date	Accepts a date represented as an integer or text string, and a second string parameter that can include strptime date formatting elements. Replaces all the valid strptime date formatting elements with their string counterparts and returns the result. Does not accept epoch formatted dates as input.	to_date (date_sold, '%Y-%m-%d')
to_double	Returns the input as a double.	to_double ('3.14') = 3.14 to_double (revenue * .01)

Function	Description	Examples
to_integer	Returns the input as an integer.	to_integer ('45') + 1 = 46 to_integer (price + tax - cost)
to_string	Returns the input as a text string. To convert a date to a string, specify the date format you want to use.	to_string (45 + 1) = '46' to_string (revenue - cost) to_string (date,('%m/%d/%y'))

Date functions

Function	Description	Examples
add_days	Returns the result of adding the specified number of days to the given date.	add_days (01/30/2015, 5) = 02/04/2015 add_days (invoiced, 30)
date	Returns the date portion of a given date.	date (home visit)
day	Returns the number (1-31) of the day for the given date.	day (01/15/2014) = 15 day (date ordered)
day_number_of_quarter	Returns the number of the day in a quarter for a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_quarter (01/30/2015) = 30 day_number_of_quarter (01/30/2015, 'fiscal') = 91
day_number_of_week	Returns the number (1-7) of the day in a week for a given date with 1 being Monday and 7 being Sunday.	day_number_of_week(01/15/2014) = 3 day_number_of_week (shipped)

Function	Description	Examples

`day_number_of_year`	Returns the number (1-366) of the day in a year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`day_number_of_year (01/30/2015) = 30`
		`day_number_of_year ( 01/30/2015, 'fiscal' ) = 275`
		`day_number_of_year (invoiced)`

`day_of_week`	Returns the day of the week for the given date.	`day_of week (01/30/2015) = Friday` `day_of_week (serviced)`
`diff_days`	Subtracts the second date from the first date and returns the result in number of days, rounded down if not exact.	`diff_days (01/15/2014, 01/17/2014) = -2` `diff_days (purchased, shipped)`
`diff_time`	Subtracts the second date from the first date and returns the result in number of seconds.	`diff_time (01/01/2014, 01/01/2014) = -86,400` `diff_time (clicked, submitted)`
`hour_of_day`	Returns the hour of the day for the given date.	`hour_of_day (received)`
`is_weekend`	Returns true if the given date falls on a Saturday or Sunday.	`is_weekend (01/31/2015) = true` `is_weekend (emailed)`
`month`	Returns the month from the given date.	`month (01/15/2014) = January` `month (date ordered)`

Function	Description	Examples

`month_number`	Returns the number (1-12) of the month from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`month_number (09/20/2014) = 9`
		`month_number ( 09/20/2014, 'fiscal' ) = 5`
		`month_number (purchased)`

`month_number_of_quarter`	Returns the month (1-3) number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`month_number_of_quarter (02/20/2018) = 2` `month_number_of_quarter (02/20/2018,'fiscal' ) = 1`
`now`	Returns the current timestamp.	`now ()`

`quarter_number`	Returns the number (1-4) of the quarter associated with the given date. Add an optional second parameter to specify 'fiscal' or 'calendar' dates. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`quarter_number ( 04/14/2014) = 2`
		`quarter_number ( 04/14/2014, 'fiscal' ) = 4`
		`quarter_number ( shipped )`

Function	Description	Examples
`start_of_month`	Returns `MMM yyyy` for the first day of the month. Your installation configuration can override this setting so that it returns a different format such as `MM/dd/yyyy` . Speak with your ThoughtSpot administrator for information on doing this.	`start_of_month ( 01/31/2015 ) = Jan FY 2015` `start_of_month (shipped)`

`start_of_quarter`	Returns the date for the first day of the quarter for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`start_of_quarter ( 04/01/2014) = Apr 2014`
		`start_of_quarter ( 04/01/2014, 'fiscal') = Feb 2014`
		`start_of_quarter (sold)`

`start_of_week`	Returns the date for the first day of the week for the given date.	`start_of_week ( 06/01/2015 ) = 05/30/2015 Week` `start_of_week (emailed)`

`start_of_year`	Returns the date for the first day of the year for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`start_of_year (04/01/2014) returns Jan 2014`
		`start_of_year (04/01/2014, 'fiscal') returns May 2013`
		`start_of_year (joined)`

`time`	Returns the time portion of a given date.	`time (3/1/2002 10:32) = 10:32` `time (call began)`

Function	Description	Examples
`week_number_of_month`	Returns the week number for the given date in a month.	`week_number_of_month(03/23/2017) = 3`
`week_number_of_quarter`	Returns the week number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`week_number_of_quarter (04/03/2017) = 1` `week_number_of_quarter (04/03/2017, 'fiscal') = 10`
`week_number_of_year`	Returns the week number for the given date in a year. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	`week_number_of_year (01/17/2014) = 3` `week_number_of_year ( 01/17/2014, 'fiscal') = 38`

`year`	Returns the year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01. Per standard convention, the fiscal year is defined by the year-end date.)	`year (01/15/2014) = 2014`
		`year (12/15/2013, 'fiscal' ) = 2014`
		`year (date ordered)`

Mixed functions

These functions can be used with text and numeric data types.

Function	Description	Examples
`!=`	Returns true if the first value is not equal to the second value.	`3 != 2 = true` `revenue != 1000000`
`<`	Returns true if the first value is less than the second value.	`3 < 2 = false` `revenue < 1000000`
`<=`	Returns true if the first value is less than or equal to the second value.	`1 <= 2 = true` `revenue <= 1000000`
`=`	Returns true if the first value is equal to the second value.	`2 = 2 = true` `revenue = 1000000`
`>`	Returns true if the first value is greater than the second value.	`3 > 2 = true` `revenue > 1000000`
`>=`	Returns true if the first value is greater than or equal to the second value.	`3 >= 2 = true` `revenue >= 1000000`
`greatest`	Returns the larger of the values.	`greatest (20, 10) = 20` `greatest (q1 revenue, q2 revenue)`
`least`	Returns the smaller of the values.	`least (20, 10) = 10` `least (q1 revenue, q2 revenue)`

Number functions

Function	Description	Examples
`*`	Returns the result of multiplying both numbers.	`3 * 2 = 6` `price * taxrate`
`+`	Returns the result of adding both numbers.	`1 + 2 = 3` `price + shipping`
`-`	Returns the result of subtracting the second number from the first.	`3 - 2 = 1` `revenue - tax`
`/`	Returns the result of dividing the first number by the second.	`6 / 3 = 2` `markup / retail price`

Function	Description	Examples
`^`	Returns the first number raised to the power of the second.	`3 ^ 2 = 9` `width ^ 2`
`abs`	Returns the absolute value.	`abs (-10) = 10` `abs (profit)`
`acos`	Returns the inverse cosine in degrees.	`acos (0.5) = 60` `acos (cos-satellite-angle)`
`asin`	Returns the inverse sine (specified in degrees).	`asin (0.5) = 30` `asin (sin-satellite-angle)`
`atan`	Returns the inverse tangent in degrees.	`atan (1) = 45` `atan (tan-satellite-angle)`
`atan2`	Returns the inverse tangent in degrees.	`atan2 (10, 10) = 45` `atan2 (longitude, latitude)`
`cbrt`	Returns the cube root of a number.	`cbrt (27) = 3` `cbrt (volume)`
`ceil`	Returns the smallest following integer.	`ceil (5.9) = 6` `ceil (growth rate)`
`cos`	Returns the cosine of an angle (specified in degrees).	`cos (63) = 0.45` `cos (beam angle)`
`cube`	Returns the cube of a number.	`cube (3) = 27` `cube (length)`
`exp`	Returns Euler's number (~2.718) raised to a power.	`exp (2) = 7.38905609893` `exp (growth)`
`exp2`	Returns 2 raised to a power.	`exp2 (3) = 8` `exp2 (growth)`
`floor`	Returns the largest previous integer.	`floor (5.1) = 5` `floor (growth rate)`
`ln`	Returns the natural logarithm.	`ln (7.38905609893) = 2` `ln (distance)`
`log10`	Returns the logarithm with base 10.	`log10 (100) = 2` `log10 (volume)`
`log2`	Returns the logarithm with base 2 (binary logarithm).	`log2 (32) = 5` `log2 (volume)`
`mod`	Returns the remainder of first number divided by the second number.	`mod (8, 3) = 2` `mod ( revenue , quantity )`

Function	Description	Examples
pow	Returns the first number raised to the power of the second number.	pow (5, 2) = 25 pow (width, 2)
random	Returns a random number between 0 and 1.	random ( ) = .457718 random ( )
round	Returns the first number rounded to the second number (the default is 1).	round (35.65, 10) = 40 round (battingavg, 100)
safe_divide	Returns the result of dividing the first number by the second. If the second number is 0, returns 0 instead of NaN (not a number).	safe_divide (12, 0) = 0 safe_divide (total_cost, units)
sign	Returns +1 if the number is greater than zero, -1 if less than zero, 0 if zero.	sign (-250) = -1 sign (growth rate)
sin	Returns the sine of an angle (specified in degrees).	sin (35) = 0.57 sin (beam angle)
spherical_distance	Returns the distance in km between two points on Earth.	spherical_distance (37.465191, -122.153617, 37.421962, -122.142174) = 4,961.96 spherical_distance (start_latitude, start_longitude, start_latitude, start_longitude)
sq	Returns the square of a numeric value.	sq (9) = 81 sq (width)
sqrt	Returns the square root.	sqrt (9) = 3 sqrt (area)
tan	Returns the tangent of an angle (specified in degrees).	tan (35) = 0.7 tan (beam angle)

Text functions

Function	Description	Examples
concat	Returns the one or more values as a concatenated text string. Be sure to use single quotes instead of double quotes around each of the strings.	concat ('hay' , 'stack') = 'haystack' concat (last_name , first_name)

Function	Description	Examples
`contains`	Returns true if the first string contains the second string, otherwise returns false.	`contains ('broomstick', 'room') = true` `contains (product, 'trial version')`
`edit_distance`	Accepts two text strings. Returns the edit distance (minimum number of operations required to transform one string into the other) as an integer. Works with strings under 1023 characters.	`edit_distance ('attorney', 'atty') = 4` `edit_distance (color, 'red')`
`edit_distance_with_cap`	Accepts two text strings and an integer to specify the upper limit cap for the edit distance (minimum number of operations required to transform one string into the other). If the edit distance is less than or equal to the specified cap, returns the edit distance. If it is higher than the cap, returns the cap plus 1. Works with strings under 1023 characters.	`edit_distance_with_cap ('pokemon go', 'minecraft pixelmon', 3) = 4` `edit_distance_with_cap (event, 'burning man', 3)`
`similar_to`	Accepts a document text string and a search text string. Returns true if relevance score (0-100) of the search string with respect to the document is greater than or equal to 20. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document.	`similar_to ('hello world', 'hello swirl') = true` `similar_to (current team, drafted by)`
`similarity`	Accepts a document text string and a search text string. Returns the relevance score (0-100) of the search string with respect to the document. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document. If the two strings are an exact match, returns 100.	`similarity ('where is the burning man concert', 'burning man') = 46` `similarity (tweet1, tweet2)`

Function	Description	Examples
`spells_like`	Accepts two text strings. Returns true if they are spelled similarly and false if they are not. Works with strings under 1023 characters.	`spells_like ('thouhgtspot', 'thoughtspot') = true` `spells_like (studio, distributor)`
`strlen`	Returns the length of the text.	`strlen ('smith') = 5` `strlen (lastname)`
`strpos`	Returns the numeric position (starting from 0) of the first occurrence of the second string in the first string, or -1 if not found.	`strpos ('haystack_with_needles', 'needle') = 14` `strpos (complaint, 'lawyer')`
`substr`	Returns the portion of the given string, beginning at the location specified (starting from 0), and of the given length.	`substr ('persnickety', 3, 7) = snicket` `substr (lastname, 0, 5)`

Alerts code reference

This reference identifies the messages that can appear in the **System Health** > **Overview** > **Critical Alerts** and in the **Alerts** dashboard.

Informational alerts

TASK_TERMINATED

Msg: `Task {{.Service}}.{{.Task}} terminated on machine {{.Machine}}`

Type: INFO

This alert is raised when a task terminates.

DISK_ERROR

Msg: `Machine {{.Machine}} has disk errors`

Type: INFO

Raised when a machine has disk errors.

ZK_AVG_LATENCY

Msg: `Average Zookeeper latency is more than {{.Num}} msec`

Type: INFO

Raised when average Zookeeper latency is above a threshold.

ZK_MAX_LATENCY

Msg: `Max Zookeeper latency is more than {{.Num}} msec`

Type: INFO

Raised when max Zookeeper latency is above a threshold.

ZK_MIN_LATENCY

Msg: `Min Zookeeper latency is more than {{.Num}} msec`

Type: INFO

Raised when min Zookeeper latency is above a threshold.

ZK_OUTSTANDING_REQUESTS

Msg: `Number of outstanding Zookeeper requests exceeds {{.Num}}`

Type: INFO

Raised when there are too many outstanding Zookeeper requests.

ZK_NUM_WATCHERS

Msg: `Number of Zookeeper watchers exceeds {{.Num}}`

Type: INFO

Raised when there are too many Zookeeper watchers.

MASTER_ELECTION

Msg: `{{.Machine}} elected as Orion Master`

Type: INFO

Raised when a new Orion Master is elected.

PERIODIC_BACKUP

Msg: `{{.Process}} periodic backup for policy {{.Name}} failed.`

Type: INFO

Raised when periodic backup fails.

PERIODIC_SNAPSHOT

Msg: `{{.Process}} periodic snapshot {{.Name}} failed.`

Type: INFO

Raised when a periodic snapshot fails.

HDFS_CORRUPTION

Msg: `HDFS root directory is in a corrupted state.`

Type: INFO

Raised when HDFS root directory is corrupted.

APPLICATION_INVALID_STATE

Msg: `{{.Service}}.{{.Task}} on {{.Machine}} at location {{.Location}}`

Type: INFO

Raised when Application raises invalid state alert.

UPDATE_START

Msg: `Starting update of ThoughtSpot cluster {{.Cluster}}`

Type: INFO

Raised when update starts.

UPDATE_END

Msg: `Finished update of ThoughtSpot cluster {{.Cluster}} to release {{.Release}}`

Type: INFO

Raised when update completes.

Errors

TIMELY_JOB_RUN_ERROR

Msg: `Job run {{.Message}}`

Type: ERROR

Raised when a job run fails.

TIMELY_ERROR

Msg: `Job manager {{.Message}}`

Type: ERROR

Raised when a job manager runs into an inconsistent state.

Warnings

DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free`

Type: WARNING

Raised when a disk is low on available disk space. Valid only in the 3.2 version of ThoughtSpot.

ROOT_DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free on root partition`

Type: WARNING

Raised when a machine is low on available disk space on root partition.

BOOT_DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free on boot partition`

Type: WARNING

Raised when a machine is low on available disk space on boot partition.

UPDATE_DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free on update partition`

Type: WARNING

Raised when a machine is low on available disk space on update partition.

EXPORT_DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free on export partition`

Type: WARNING

Raised when a machine is low on available disk space on export partition.

HDFS_NAMENODE_DISK_SPACE

Msg: `Machine {{.Machine}} has less than {{.Perc}}% disk space free on HDFS namenode drive`

Type: WARNING

Raised when a machine is low on available disk space on HDFS namenode drive.

MEMORY

Msg: `Machine {{.Machine}} has less than {{.Perc}}% memory free`

Type: WARNING

Raised when a machine is low on free memory.

OS_USERS

Msg: `Machine {{.Machine}} has more than {{.Num}} logged in users`

Type: WARNING

Raised when a machine has too many users logged in.

OS_PROCS

Msg: `Machine {{.Machine}} has more than {{.Num}} processes`

Type: WARNING

Raised when a machine has more too many processes.

SSH

Msg: `Machine {{.Machine}} doesn't have an active SSH server`

Type: WARNING

Raised when a machine has more than 600 processes.

DISK_ERROR_EXTERNAL

Msg: `Machine {{.Machine}} has disk errors`

Type: WARNING

Raised when more than 2 disk errors happen in a day.

ZK_FD_COUNT

Msg: `Zookeeper has more than {{.Num}} open file descriptors`

Type: WARNING

Raised when there are too many open Zookeeper files.

ZK_EPHEMERAL_COUNT

Msg: `Zookeeper has more than {{.Num}} ephemeral files`

Type: WARNING

Raised when there are too many Zookeeper ephemeral files.

HOST_DOWN

Msg: `{{.Machine}} is down`

Type: WARNING

Raised when a host is down.

TASK_UNREACHABLE

Msg: `{{.ServiceDesc}} on {{.Machine}} is unreachable over HTTP`

Type: WARNING

Raised when a task is unreachable over HTTP.

TASK_NOT_RUNNING

Msg: `{{.ServiceDesc}} is not running`

Type: WARNING

Raised when a service task is not running on any machine in the cluster.

Critical alerts

TASK_FLAPPING

Msg: `Task {{.Service}}.{{.Task}} terminated {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

Type: CRITICAL

This alert is raised when a task is crashing repeatedly. The service is evaluted across the whole cluster. So, if a service crashes 5 times in a day across all nodes in the cluster, this alert is generated.

OREO_TERMINATED

Msg: `Oreo terminated on machine {{.Machine}}`

Type: CRITICAL

This alert is raised when the Oreo daemon on a machine terminates due to an error. This typically happens due to an error accessing Zookeeper, HDFS, or a hardware issue.

HDFS_DISK_SPACE

Msg: `HDFS has less than {{.Perc}}% space free`

Type: CRITICAL

Raised when a HDFS cluster is low on total available disk space.

ZK_INACCESSIBLE

Msg: `Zookeeper is not accessible`

Type: CRITICAL

Raised when Zookeeper is inaccessible.

PERIODIC_BACKUP_FLAPPING

Msg: `Periodic backup failed {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

Type: CRITICAL

This alert is raised when a periodic backup failed repeatedly.

PERIODIC_SNAPSHOT_FLAPPING

Msg: `Periodic snapshot failed {{._actual_num_occurrences}} times in last {{._earliest_duration_str}}`

Type: CRITICAL

This alert is raised when periodic snapshot failed repeatedly.

APPLICATION_INVALID_STATE_EXTERNAL

Msg: `{{.Service}}.{{.Task}} on {{.Machine}} at location {{.Location}}`

Type: CRITICAL

Raised when Application raises invalid state alert.

User action code reference

This reference identifies the user action codes that can appear in the **System Health** pages and in logs or other reports.

`answer_unsaved`	User makes a change to tokens in the search bar.
`answer_saved`	User opens an existing saved answer and makes changes to tokens in the search bar.
`answer_pinboard_context`	User opens an existing saved pinboard, edits a context viz and makes a change to tokens in the search bar.
`answer_aggregated_worksheet`	User opens an existing saved aggregated worksheet and makes changes to tokens in the search bar.
`answer_upgrade`	Requests made for the sole purpose of upgrade.
`pinboard_view`	User opens an existing saved pinboard.
`pinboard_filter`	User adds, removes or applies values to a pinboard filter.
`pinboard_ad_hoc`	User drills down in a pinboard viz.
`data_chart_config`	Request for new data being generated following a chart config change.
`data_show_underlying_row`	Request to show underlying data for a data row(s).
`data_export`	Request to export data.
`pinboard_tspublic_runtime_filter`	Request to TSPublic/pinboarddata with runtime filters.
`answer_aggregated_worksheet_save`	User updates aggregated worksheet.
`answer_add_new_filter`	User adds a filter using the UI.
`data_show_underlying_viz`	Request to show underlying data for a data row(s).
`answer_view`	User opens an existing, saved answer.
`answer_viz_context_view`	User opens an existing saved pinboard, edits a context viz.
`pinboard_insight_view`	User opens SpotIQ tab pinboards.

`pinboard_admin_view`	User opens admin tab pinboards.
`pinboard_embed_view`	User opens embed pinboard from a URL.
`pinboard_homepage_view`	On loading of homepage pinboard.
`pinboard_learn_view`	On loading learn pinboard.
`pinboard_tspublic_no_runtime_filter`	Request to TSPublic/pinboard data without run-time filters.

Error code reference

Summary: List of error codes and messages.

This section lists error codes that can appear in ThoughtSpot, with summary information and actions to take. Error codes and messages are shown in ThoughtSpot when something goes wrong. These messages can appear in the application and in logs.

When you see an error code, you will also see a message with a brief summary of what has happened. If there is a remediation action you can take, it will be listed in this references. If there is no action listed, please contact ThoughtSpot Support.

☑ **Tip:** Only the base code number is listed for each error. So keep this in mind when searching through these codes. For example, error code `TS-00125` is simply listed as `125`.

Metadata Errors (100 - 499)

Code	Severity	Summary	Details	Action
TS-00100	INFO	Success. {1} has been added to {2}. \# {1} – name of visualization \# {2} – {name/link to pinboard}	None	None
TS-00101	ERROR	Failure adding {1} to {2}	Visualization could not be added to {2} \# {1} – name of visualization \# {2} – name/link to pinboard	None
TS-00102	ERROR	Failure adding {1} to {2} due to corruption	{1} could not be added to {2} as the pinboard has one or more invalid visualizations	Please try again after removing the invalid visualization(s) from {2} \# {1} – name of visualization \# {2} – name/link to pinboard
TS-00103	INFO	Success. Visualization has been deleted from {1}. 1 – name/link to pinboard	None	None

TS-00104	ERROR	Failure deleting visual from {1}	Visualization could not be deleted from the pinboard. 1 – name/link to pinboard	None
TS-00105	ERROR	Failure deleting visual from {1} due to corruption	Visualization could not be deleted from {1} as the pinboard has one or more invalid visualizations. 1 – name/link to pinboard	Please try again after removing the invalid visualization(s) from the pinboard
TS-00106	INFO	Success. {1} created successfully. 1 – name/link to pinboard	None	None
TS-00107	ERROR	Failure creating {1}. 1 – name/link to pinboard	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00108	INFO	Sticker created successfully.	None	None
TS-00109	ERROR	Failure creating the sticker.	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00110	INFO	Sticker deleted successfully.	None	None
TS-00111	ERROR	Failure deleting sticker.	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00112	INFO	Pinboards deleted successfully.	None	None
TS-00113	ERROR	Failure deleting pinboards	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00114	INFO	Answers deleted successfully.	None	None

TS-00115	ERROR	Failure deleting answers	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00116	INFO	Tables deleted successfully.	None	None
TS-00117	ERROR	Failure deleting tables	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00118	INFO	Relationship created successfully.	None	None
TS-00119	ERROR	Failure creating relationship	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00120	INFO	Relationship updated successfully.	None	None
TS-00121	ERROR	Failure updating the relationship	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00122	INFO	Relationship deleted successfully.	None	None
TS-00123	ERROR	Failure deleting the relationship	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00124	ERROR	Failure fetching details for table	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00125	ERROR	Failure fetching details for the tables	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None

TS-00126	ERROR	Failure fetching details for datasource	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00127	ERROR	Failure fetching details for datasources	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00128	ERROR	Failure fetching details for metadata items	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00129	ERROR	Failure opening the answer	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00130	ERROR	Failure opening the pinboard	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00131	ERROR	Failure opening the worksheet	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00132	INFO	Table saved successfully.	None	None
TS-00133	ERROR	There was a problem saving the table	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00134	INFO	Visualization update successful	None	None
TS-00135	ERROR	Visualization failed to update	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None

TS-00136	INFO	{1} saved 1 – name of answer	None	None
TS-00137	ERROR	{1} could not be saved 1 – name of answer	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00138	INFO	{1} saved 1 - name of pinboard / link	None	None
TS-00139	ERROR	{1} could not be saved 1 - name of pinboard / link	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00140	INFO	{1} saved 1 – name of worksheet	None	None
TS-00141	ERROR	{1} could not be saved 1 – name of worksheet	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00142	INFO	{1} saved 1 – name of answer	None	None
TS-00143	ERROR	{1} could not be saved	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}. 1 – name of answer	None
TS-00144	INFO	{1} saved 1 – name/ link to pinboard	None	None
TS-00145	ERROR	{1} could not be saved	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}. 1 – name of pinboard	None
TS-00146	INFO	Worksheet saved	None	None
TS-00147	ERROR	Worksheet could not be saved	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None

TS-00148	INFO	Sticker updated	None	None
TS-00149	ERROR	The sticker could not be updated	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00150	INFO	Successfully assigned sticker	None	None
TS-00151	ERROR	The sticker could not be assigned	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00152	INFO	Successfully unassigned sticker	None	None
TS-00153	ERROR	The sticker could not be unassigned	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00154	ERROR	Failed to fetch metadata list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00155	ERROR	Failed to fetch table list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00156	ERROR	Failed to fetch relationship list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00157	ERROR	Failed to fetch answer list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None

TS-00158	ERROR	Failed to fetch pinboard list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00159	ERROR	Failed to fetch worksheet list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00160	ERROR	Failed to fetch aggregated worksheet list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00161	ERROR	Failed to fetch imported data list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00162	ERROR	Failed to fetch system table list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00163	ERROR	Failed to DB view list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00164	ERROR	Failed to fetch data source list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00165	ERROR	Failed to fetch column list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00166	ERROR	Failed to label list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None

TS-00167	ERROR	Failed to fetch answer	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00168	ERROR	Failed to fetch worksheet	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00169	INFO	Aggregated worksheet {1} created 1 – name of aggregated worksheet	None	None
TS-00170	ERROR	Failure creating Aggregated Worksheet.	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00171	INFO	{1} updated 1 – name of aggregated worksheet	None	None
TS-00172	ERROR	{1} failed to update 1 – name of aggregated worksheet	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00173	ERROR	{1} failed to update 1 – name of the formula	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00174	ERROR	Comments cannot be fetched	Failed to save client state	None
TS-00175	ERROR	Comment cannot be created	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00176	ERROR	Comment cannot be updated	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None

TS-00177	ERROR	Comment cannot be deleted	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00178	INFO	Rule saved successfully	None	None
TS-00179	ERROR	Rule could not be saved	We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00180	INFO	Rule deleted successfully	None	None
TS-00181	ERROR	Rule could not be deleted	We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00182	INFO	Item deleted successfully.	None	None
TS-00183	ERROR	Item could not be deleted.	We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-00184	INFO	Related link created successfully.	None	None
TS-00185	ERROR	Related link could not be created.	Uh oh. We're not sure what happened. Please click 'Report Problem' to email a report to your administrator, {adminEmail}.	None
TS-00186	INFO	Related link updated successfully.	None	None
TS-00187	ERROR	Related link could not be updated.	Uh oh. We're not sure what happened. Please click 'Report Problem' to email a report to your administrator, {adminEmail}.	None
TS-00188	INFO	Related link deleted successfully.	None	None

TS-00189	ERROR	Related link could not be deleted.	Uh oh. We're not sure what happened. Please click 'Report Problem' to email a report to your administrator, {adminEmail}.	None
TS-00190	INFO	Related link detail fetched successfully.	None	None
TS-00191	ERROR	Related link detail could not be fetched.	Uh oh. We're not sure what happened. Please click 'Report Problem' to email a report to your administrator, {adminEmail}.	None

Data Service Errors (500 - 699)

Code	Severity	Summary	Details	Action
TS-00500	ERROR	Failed to fetch leaf level data	Failed to fetch leaf level data.	None
TS-00501	ERROR	Failed to fetch excel data	Failed to fetch excel data.	None
TS-00502	ERROR	Failed to fetch visualization data	Failed to fetch visuzliation data.	None
TS-00503	ERROR	Failed to fetch visualizations data	Failed to fetch data for visualizations.	None
TS-00504	ERROR	Failed to fetch chart data	Failed to fetch table data.	None
TS-00505	ERROR	Failed to fetch table data	Failed to fetch table data.	None
TS-00506	ERROR	Failed to fetch worksheet data	Failed to fetch worksheet data.	None
TS-00507	ERROR	Failed to fetch filter data	Failed to fetch filter data.	None
TS-00508	ERROR	Failed to fetch headline data	Failed to fetch filter data.	None

Code	Severity	Summary	Details	Action
TS-00509	ERROR	Failed to fetch natural query	Failed to fetch natural query.	None
TS-00510	INFO	File upload successful	None	None
TS-00511	ERROR	Failed to upload file	Failed to upload	None
TS-00512	ERROR	The pinboard data could not be exported to pdf.	Uh oh. We're not sure what happened. Please click 'Report Problem' to email a report to your administrator, {adminEmail}.	None

Dependency Errors (700 - 799)

Code	Severity	Summary	Details	Action
TS-00700	ERROR	Failure fetching table dependents	Failed to fetch dependents for the table.	None
TS-00701	ERROR	Failure fetching column dependents	Failed to fetch dependents for the column.	None
TS-00702	ERROR	Failure fetching incomplete items	Failed to fetch incomplete items.	None

Admin Service Errors (800 - 899)

Code	Severity	Summary	Details	Action
TS-00800	ERROR	Failure fetching MemCache stats	Failed to fetch MemCache stats.	None
TS-00801	ERROR	Failure MemCache Clear	Failed to clear MemCache.	None
TS-00802	ERROR	Failure searching from MemCache	Failed to search from MemCache.	None
TS-00803	ERROR	Failure fetching Loggers	Failed to fetch Loggers.	None
TS-00804	ERROR	Failure setting LogLevel	Failed to set Log Level.	None
TS-00805	ERROR	Failure getting debug info	Failed to get debug info.	None
TS-00806	INFO	Memcache cleared successfully	None	None

Code	Severity	Summary	Details	Action
TS-00807	INFO	Log level set successfully	None	None
TS-00808	ERROR	Failed to report problem	None	None
TS-00809	INFO	Problem reported successfully	None	None

Permissions Errors (900 - 999)

Code	Severity	Summary	Details	Action
TS-00900	ERROR	Failure fetching table permissions	Failed to fetch table permissions.	None
TS-00901	ERROR	Failure fetching answer permissions	Failed to fetch answer permissions.	None
TS-00902	ERROR	Failure fetching pinboard permissions	Failed to fetch pinboard permissions.	None
TS-00903	ERROR	Failure getting metadata permissions	Failed to get metadata permissions.	None

Import Data Errors (1000 - 1099)

Code	Severity	Summary	Details	Action
TS-01000	ERROR	Data caching failed	Data caching failed.	None
TS-01001	ERROR	Read Columns failed.	Failed to read columns.	None
TS-01002	ERROR	Failed to read keys.	Failed to read keys.	None
TS-01003	ERROR	Failed to read relationships.	Failed to read relationships.	None
TS-01004	ERROR	Failed to load data.	Failed to load data.	None
TS-01005	ERROR	Failed to create table.	Failed to create table.	None
TS-01006	ERROR	Failed to fetch data rows.	Failed to fetch data rows.	None
TS-01007	ERROR	Failed to delete files.	Failed to fetch data rows.	None
TS-01008	ERROR	Failed to abort create table.	Failed to abort create table.	None
TS-01009	ERROR	Failed to create schema.	Failed to create schema.	None

Code	Severity	Summary	Details	Action
TS-01010	ERROR	Failed to fetch table models.	Failed to fetch table models.	None
TS-01011	ERROR	Failed to fetch sample values.	Failed to fetch sample values.	None

Scheduled Jobs Errors (1100 - 1199)

Code	Severity	Summary	Details	Action
TS-01100	INFO	The list of jobs.	None	Please click 'Report Problem' to email a report to your administrator.
TS-01110	INFO	Successfully created job.	None	None
TS-01111	ERROR	The job could not be created.	None	Please click 'Report Problem' to email a report to your administrator.
TS-01112	INFO	Successfully updated job.	None	None
TS-01113	ERROR	The job could not be updated.	None	Please click 'Report Problem' to email a report to your administrator.
TS-01114	INFO	Successfully deleted jobs.	None	None
TS-01115	ERROR	The job could not be deleted.	None	Please click 'Report Problem' to email a report to your administrator.
TS-01116	INFO	The job was paused.	None	None
TS-01117	ERROR	The job could not be paused.	None	Please click 'Report Problem' to email a report to your administrator.
TS-01118	INFO	The job was resumed	None	None
TS-01119	ERROR	The job could not be resumed.	None	Please click 'Report Problem' to email a report to your administrator.

User Admin Service Errors (1200 - 1399)

Code	Severity	Summary	Details	Action
TS-01200	ERROR	Failed to fetch users list	Failed to fetch users list	None

Code	Severity	Summary	Details	Action
TS-01201	ERROR	Failed to fetch groups list	Failed to fetch groups list	None
TS-01202	ERROR	Failed to fetch users and groups list	Failed to fetch users and groups list	None
TS-01203	ERROR	Successfully created user	Successfully created user	None
TS-01204	ERROR	Failed to create user	Failed to create user	None
TS-01205	ERROR	Successfully created group	Successfully created group	None
TS-01206	ERROR	Failed to create group	Failed to create group	None
TS-01207	ERROR	Successfully updated user	Successfully updated user	None
TS-01208	ERROR	Failed to update user	Failed to update user	None
TS-01209	ERROR	Successfully updated users	Successfully updated users	None
TS-01210	ERROR	Failed to update users	Failed to update users	None
TS-01211	ERROR	Successfully updated group	Successfully updated group	None
TS-01212	ERROR	Failed to update group	Failed to update group	None
TS-01213	ERROR	Successfully updated password	Successfully updated password	None
TS-01214	ERROR	Failed to update password	Failed to update password	None
TS-01215	ERROR	Successfully deleted users	Successfully deleted users	None
TS-01216	ERROR	Failed to delete users	Failed to delete users	None
TS-01217	ERROR	Successfully deleted groups	Successfully deleted groups	None
TS-01218	ERROR	Failed to delete groups	Failed to delete groups	None
TS-01219	ERROR	Successfully assigned users to groups	Successfully assigned users to groups	None
TS-01220	ERROR	Failed to assign users to groups	Failed to assign users to groups	None
TS-01221	ERROR	Failed to fetch profile pic	Failed to fetch profile pic	None
TS-01222	INFO	Successfully uploaded profile pic	None	None
TS-01223	ERROR	Failed to upload profile pic	Failed to upload profile pic	None
TS-01224	ERROR	Successfully assigned groups to group	Failed to assign user to group	None

Code	Severity	Summary	Details	Action
TS-01228	ERROR	Successfully created role	Successfully created role	None
TS-01229	ERROR	Failed to create role	Failed to create role	None
TS-01230	ERROR	Successfully deleted role	Successfully deleted role	None
TS-01231	ERROR	Failed to delete role	Failed to delete role	None
TS-01232	ERROR	Successfully updated role	Successfully updated role	None
TS-01233	ERROR	Failed to update role	Failed to update role	None

Session Service Errors (1400 - 1599)

Code	Severity	Summary	Details	Action
TS-01400	ERROR	Failed to fetch session info	Failed to fetch session info	None
TS-01401	ERROR	Failed to login	Uh oh. We’re not sure what happened. Please email the trace file to {adminE-mail}.	None
TS-01402	ERROR	Failed to logout	Failed to logout	None
TS-01403	ERROR	Failed to save client state	Failed to save client state	None
TS-01404	ERROR	Failed to fetch login con-fig	Failed to fetch login config	None
TS-01405	ERROR	Failed to fetch slack config	Failed to fetch slack config	None
TS-01406	ERROR	Health check failed	Health check failed	None
TS-01407	ERROR	Failed to fetch health portal token	Failed to fetch health portal token	None
TS-01408	ERROR	The health portal release name could not be re-trieved	Uh oh. We’re not sure what happened. Please email the trace file to {adminE-mail}.	None

Data Management Service Errors (1600 - 1799)

Code	Severity	Summary	Details	Action
TS-01600	ERROR	Failed to fetch data source types	Failed to fetch data source types	None
TS-01601	ERROR	Failed to fetch data source sample values	Failed to fetch data source sample values	None
TS-01602	ERROR	Failed to delete data source	Failed to delete data source	None
TS-01603	ERROR	Failed to execute DDL	Failed to execute DDL	None
TS-01604	ERROR	Failed to update schedule	Failed to update schedule	None
TS-01605	ERROR	Failed to reload tasks	Failed to reload tasks	None
TS-01606	ERROR	Failed to stop tasks	Failed to stop tasks	None
TS-01607	ERROR	Failed to get creation DDL	Failed to get creation DDL	None
TS-01608	ERROR	Failed to load from data source	Failed to load from data source	None
TS-01609	ERROR	Failed to create connection to data source	Failed to create connection to data source	None
TS-01610	ERROR	Failed to create data source	Failed to create data source	None
TS-01611	ERROR	Failed to connect to data source	Failed to connect to data source	None
TS-01612	ERROR	Failed to get data source connection field info	Failed to get data source connection field info	None
TS-01613	ERROR	Failed to get connection list for data source	Failed to get connection list for data source	None
TS-01614	ERROR	Failed to get connection attributes for data source	Failed to get connection attributes for data source	None
TS-01615	ERROR	Failed to get connections to data source	Failed to get connections to data source	None

Code	Severity	Summary	Details	Action
TS-01616	ERROR	Failed to fetch data source config	Failed to fetch data source config	None
TS-01617	ERROR	Failed to parse sql.	Failed to parse sql.	None
TS-01618	ERROR	Failed to execute sql.	Failed to execute sql.	None
TS-01619	INFO	Successfully created connection to data source	None	None
TS-01620	INFO	Successfully updated data upload schedule	None	None
TS-01621	ERROR	Failed to execute sql.	Please check the failing command, executed {1} statements successfully.	None
TS-01622	ERROR	Lightweight data-cache disabled	Lightweight data-cache disabled	None
TS-01623	INFO	Selected tables were queued for loading.	Selected tables were queued for loading.	None
TS-01624	ERROR	DataType conversion error.	No mapping found for source datatype to ThoughtSpot datatype.	None
TS-01625	INFO	Successfully reload task started.	None	None
TS-01626	INFO	Successfully connected to data source.	None	None
TS-01627	INFO	Successfully created data source.	None	None
TS-01628	INFO	Successfully stopped the tasks.	None	None
TS-01629	INFO	Successfully deleted the connection.	None	None
TS-01630	ERROR	There was an error deleting this connection.	None	None
TS-01631	INFO	Successfully executed the DDL.	None	None

Cluster Status Service Errors (1800 - 1899)

Code	Severity	Summary	Details	Action
TS-01800	WARNING	Failed to fetch cluster information from search service.	None	None
TS-01801	WARNING	Failed to fetch table detail information from search service.	None	None
TS-01802	WARNING	Failed to fetch cluster information from database service.	None	None
TS-01803	WARNING	Failed to fetch table detail information from databse service.	None	None
TS-01804	WARNING	Failed to fetch cluster information from cluster management service.	None	None
TS-01805	WARNING	Failed to fetch detail information from cluster management service.	None	None
TS-01806	WARNING	Failed to fetch log from cluster management service.	None	None
TS-01807	WARNING	Failed to fetch snapshot list from cluster management service.	None	None
TS-01808	WARNING	Failed to fetch cluster information from alert management service.	None	None
TS-01809	WARNING	Failed to fetch cluster information from event service.	None	None
TS-01810	WARNING	Failed to fetch alerts information from alert management service.	None	None
TS-01811	WARNING	Failed to fetch events information from alert management service.	None	None
TS-01812	INFO	Thanks for your feedback!	None	None
TS-01813	WARNING	Sorry! Unable to submit the feedback at this moment!	None	None
TS-01814	INFO	Successfully exported objects. File can be found at {1}.	None	None
TS-01815	ERROR	Sorry! Unable to export objects at this moment!	What happened? {1}.	None

Code	Severity	Summary	Details	Action
TS-01816	INFO	Successfully imported objects	None	None
TS-01817	ERROR	Sorry! Unable to import objects at this moment!	What happened? {1}.	None
TS-01818	INFO	Successfully deleted data source object(s).	None	None

Callosum API Errors (9000 - 9199)

Code	Severity	Summary	Details	Action
TS-09000	ERROR	The data you are trying to delete has some dependencies	Some objects depend on the data you are trying to delete	delete the dependencies before deleting this data.
TS-09001	ERROR	Uh oh. We're not sure what happened.	Please email the trace file to {adminEmail}.	None
TS-09002	ERROR	Could not authorize user	Try logging in again	None
TS-09003	ERROR	Uh oh. We're not sure what happened.	Please email the trace file to {adminEmail}.	None
TS-09004	WARNING	Still loading data, come back soon	None	None
TS-09005	ERROR	Uh oh. We're having trouble getting data for this request.	Please email the trace file to {adminEmail}.	None
TS-09006	ERROR	Uh oh. We're having trouble getting data for this request.	Please email the trace file to {adminEmail}.	None
TS-09007	ERROR	Uh oh. We're having trouble getting data for this request.	Please email the trace file to {adminEmail}.	None
TS-09008	ERROR	Something went wrong with your search	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None

Code	Severity	Summary	Details	Action
TS-09009	ERROR	The calculation engine has timed out. Please try again.	Please email the trace file to {adminEmail}.	None
TS-09010	ERROR	Cannot open Object	Object cannot be opened due to errors in some of its dependencies	None
TS-0Blink Generated Errors (9500 - 9599)				
TS-09500	WARNING	Cannot connect to the calculation engine. Please try again soon.	None	None
TS-09501	WARNING	The calculation engine has timed out. Please try again.	None	None
TS-09502	WARNING	Cannot connect to the search engine. Please try again soon.	None	None
TS-09503	WARNING	The search engine has timed out. Please try again.	None	None
TS-09504	ERROR	Cannot open {1}	{1} cannot be opened due to errors in the following dependencies 1 - Type of the object Table/Answer/Pinboard etc.	None
TS-09505	WARNING	We’re still indexing this data, try again soon	None	None
TS-09506	ERROR	Object is not present in the system	{1} is not present in the system 1 - Type of the object Table/Answer/Pinboard etc.	None
TS-09507	ERROR	ThoughtSpot is unreachable. Please try again soon	None	None

Common Errors (10000 - 10099)

Code	Severity	Summary	Details	Action
TS-10000	ERROR	A system error has occurred	Uh oh. We're not sure what happened. Please contact your administrator.	None
TS-10001	ERROR	Connection failed	The metadata store is not reachable.	Please contact your administrator
TS-10002	ERROR	The input is invalid	Input from the client to the server is invalid.	Please contact your administrator
TS-10003	ERROR	Unfortunately, you can't do that	You are not authorized to perform {1}. # {1} – action user is not authorized for	Please request access from your administrator
TS-10004	ERROR	The user could not be authorized	User {0} is not authorized to perform {1}. # {0} – name of the user # {1} – action user is not authorized for	Please request access from your administrator
TS-10005	ERROR	The base object is missing	An underlying object referenced by this object is missing in store.	Please contact your administrator
TS-10006	ERROR	The connection to Zookeeper has failed	Zookeeper is not reachable.	Please contact your administrator
TS-10007	ERROR	There's invalid parameter(s)	Invalid parameter values: {0}.	Please contact your administrator
TS-10008	ERROR	The user cannot be found	User {0} not found in store. # {0} – name of the user	Please contact your administrator
TS-10009	ERROR	Cannot add group	This group already belongs to the group you are trying to add it to.	None

Falcon Errors (10600 - 10699)

Code	Severity	Summary	Details	Action
TS-10603	ERROR	Falcon query cancelled	None	None

Data Errors (11000 - 11099)

Code	Severity	Summary	Details	Action
TS-11001	ERROR	Invalid row	None	None
TS-11002	ERROR	Invalid table/query resultset	None	None
TS-11003	ERROR	Invalid column identifier	None	None
TS-11004	ERROR	Invalid visualization identifier	None	None
TS-11005	ERROR	No data	Query execution resulted in no data.	None
TS-11006	ERROR	Query execution failed	Error in query execution to Falcon.	None
TS-11007	ERROR	Answer data generation failed	Error in Answer data generation for Sage input.	None
TS-11008	ERROR	Data export failed	None	None
TS-11009	ERROR	Data generation failed	Error in data generation in Callosum.	None

Report Generation Errors (12000 - 13000)

Code	Severity	Summary	Details	Action
TS-12700	ERROR	Error while exporting data file.	None	None
TS-12701	ERROR	Invalid input.	The definition of the job is invalid.	None
TS-12702	ERROR	No author provided.	None	None
TS-12703	ERROR	No pinboard provided.	None	None

Code	Severity	Summary	Details	Action
TS-12704	ERROR	No recipients provided.	None	None
TS-12705	ERROR	This format is not supported.	None	None
TS-12706	ERROR	No job name provided.	None	None
TS-12707	ERROR	No job description provided.	None	None
TS-12708	ERROR	Pinboard data export error.	None	None
TS-12709	ERROR	Visualization data export error.	None	None
TS-12710	ERROR	User data unavailable.	None	None
TS-12711	ERROR	Configuration information un-available.	None	None
TS-12712	ERROR	There are too many recipients.	The max number of recipients is 1000.	None
TS-12713	ERROR	Attachment size limit exceeded.	None	None
TS-12714	ERROR	Recipient domain is not whitelisted.	None	None

More Metadata Errors (13000 - 13099)

Code	Severity	Summary	Details	Action
TS-13001	ERROR	Schema creation failed	Error creating database schema.	None
TS-13002	ERROR	Views creation failed	Error creating view.	None
TS-13003	ERROR	The object can-not be found in store	Object with Id: {0} of type: {1} not found. # {0} – identity of the ob-ject # {1} – type of ob-ject	None
TS-13004	ERROR	The object is in an invalid state	Oject with Id: {0} of type: {1} in invalid state. # {0} – identity of the object # {1} – type of object	None

Code	Severity	Summary	Details	Action
TS-13005	ERROR	Object already exists	Object with Id: {0} of type: {1} already exists. # {0} – identity of the object # {1} – type of object	None
TS-13006	ERROR	Invalid object type	Invalid type: {0} provided. # {1} – type of object	None
TS-13007	ERROR	Invalid Sage question	Insufficient or invalid input from Sage: {0}. # {0} – the invalid input	None
TS-13008	ERROR	Invalid Sage question	Input from from Sage – missing columns of type: {0}. # {0} – column type	None
TS-13009	ERROR	Invalid Sage question	Invalid input from Sage – invalid expression: {0}. # {0} – the invalid expression	None
TS-13010	ERROR	Sending logical metadata to Sage failed	Sending logical metadata to Sage failed due to: {0}. # {0} – reason for failure	None
TS-13011	ERROR	Answer generation failed	Answer generation failed due to: {0}. # {0} – reason for failure	None
TS-13012	ERROR	Worksheet generation failed	Worksheet generation failed due to: {0}. # {0} – reason for failure	None
TS-13013	ERROR	Service provider unavailable	Service provider unavailable: {0}. # {0} – provider details	None
TS-13015	ERROR	Physical model not loaded	None	None
TS-13016	ERROR	Invalid physical schema proto	Inconsistency in physical schema from Falcon: {0}. # {0} – error details	None
TS-13017	ERROR	Invalid duplicate columns	Duplicate columns: {0}. # {0} – List of duplicate column identities	None

Code	Severity	Summary	Details	Action
TS-13018	ERROR	Cyclic relationship	Detected cycles: {0}. # {0} – cycle details	None
TS-13019	WARNING	Older physical schema version received	Schema update for older version: {0} received and ignored. # {0} – received version number	None
TS-13020	ERROR	Invalid relationship	Attempted to create invalid relationship: {0}. # {0} – relationship details	None
TS-13022	ERROR	Invalid filter values: {values}	None	None
TS-13023	ERROR	Creating relationship failed.	None	None
TS-13024	ERROR	Deleting schema failed.	None	None
TS-13025	ERROR	Expression validation failed.	None	None
TS-13026	INFO	Load schedule successfully disabled.	None	None
TS-13027	ERROR	Load schedule could not be disabled.	None	None
TS-13028	ERROR	Objects fetched from the connection are invalid for editing datasource.	None	To proceed with editing the datasource, please edit the connection below to fetch valid source objects.
TS-13029	INFO	Successfully edited data source connection.	None	None
TS-13030	ERROR	Connection test failed.	None	Please verify connection attributes.

Loading Errors (30000 - 30099)

Code	Severity	Summary	Details	Action
TS-30000	ERROR	Table is not ready (data loading in progress).	None	None

Timely Errors (60000 - 64999)

Code	Severity	Summary	Details	Action
TS-60000	ERROR	Failed to initialize.	None	None

Frequently asked questions

Where can I find the version of ThoughtSpot I am using?

Users with administrative privileges can see this displayed on the **Admin** > **System Health** > **Overview** page.

I'm not seeing certain columns/values in the drop-down, why?

It could be the index has not built with the latest data or something is causing the column to be dropped.

- Verify the the column is available using the **Data** page.
- View the table columns and check the **INDEX TYPE** value. If it is set to `DONT_INDEX`, change it.
- Check the column's **INDEX PRIORITY** &emdash. make sure it is `1`.

To learn more about modeling data see modeling data in this documentation.

How do I track progress of current index build?

If you are an administrator, you can use the **Admin** > **System Health** > **Overview** page to see the number of tables currently being indexed. You can also review the

How do I display the features used in my cluster configuration?

1. Log into the ThoughtSpot cluster as the `admin` user.
2. Use the `tscli feature` subcommand to display your current configuration.

```
 $ tscli feature get-all-config
 +---------------------------------+----------
+---------------+
 |              NAME               |  STATUS  | CONFIGUR
ATION |
 +---------------------------------+----------
+---------------+
 | Firewall                        | Disabled
|               |
 | Saml                            | Disabled
|               |
 | Ldap                            | Disabled
|               |
 | CustomBranding                  | Disabled
|               |
 | CustomBrandingFontCustomization | Disabled
|               |
 | DataConnect                     | Disabled
|               |
 | RLS                             | Enabled
|               |
 | Callhome                        | Enabled
|               |
 | SSHTunnel                       | Enabled
|               |
 | Fileserver                      | Disabled
|               |
 +---------------------------------+----------
+---------------+
```

Is it possible to create a max(date) filed and set it to filter?

If you have a date field in my set of data and want to return the most recent set of data based on specific date. To do this:

1. Create a formula called `Max Date`, for example:

```
date = group_max ( date_to_filter_by )
```

2. In the search bar, filter your dates by this formula for example:

```
max date = true
```

This returns only those fields that pass the filter.