



Administration Guide

Version 4.4, December 09, 2017

Copyright for ThoughtSpot publications. © 2017 ThoughtSpot, Inc. All rights reserved.

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

All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. ThoughtSpot is a trademark of ThoughtSpot, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Table of Contents

Introduction to administration.....	5
Login credentials for administration.....	6
Understand the architecture	
Architectural components.....	8
Data caching	9
Authentication frameworks	10
Data and object security.....	11
Performance considerations.....	13
Installation and setup	
About installation and upgrades.....	15
Set your locale.....	16
Test connectivity between nodes.....	17
Set the relay host for SMTP	18
Set up a fiscal calendar year	19
Configure SSL.....	20
Configure SAML.....	22
Integrate LDAP	
About LDAP integration.....	23
Configure LDAP for Active Directory	24
Add the SSL certificate for LDAP	26
Test the LDAP configuration	27
Sync users and groups from LDAP	28
Configure NAS file system.....	30
Set up monitoring.....	31
Configure support services.....	32
Network ports	36
About load balancing and proxies	42
Load and manage data	
Introduction to data management.....	44
Configure casing	46
Load CSV files with the UI.....	47
How to view a data schema	51
Plan the schema	
About schema planning	54
Data types	56

Constraints	58
Sharding	60
Chasm traps	64
Build the schema	
Schema building overview	67
Connect with TQL and create a schema.....	69
How to write a SQL script	71
Schema creation examples	72
Upload an SQL script.....	75
Change the schema	
How to change a schema	77
Convert column data type	80
Load bulk data	
Import CSV files with tsload.....	83
Use a script to load data.....	84
Delete a data source	
Delete a data source (table).....	87
Delete or change a table in TQL.....	90
Improve search with modeling	
About data modeling.....	91
Change a table's data model	92
Edit the system-wide data model.....	94
Data model settings	
Overview of the settings.....	97
Set column name, description, and type	99
Set additive and aggregate values.....	101
Hide a column or define a synonym.....	104
Change the index type.....	106
Add a geographical data setting	108
Set the search suggestions priority.....	111
Set number, date, currency formats	113
Change the Attribution Dimension setting	118
Link tables using relationships	
Link tables using relationships	120
Delete a relationship.....	124
Use stickers.....	126
Simplify search with worksheets	
Create and use worksheets.....	129
How the inclusion rule works	134
How the worksheet join rule works.....	135

Create a formula in a worksheet	137
Edit a worksheet.....	140
Change inclusion, join, or RLS for a worksheet.....	142
Delete a worksheet or table	144
Manage users and groups	
Understand groups and privileges.....	146
Create, edit, or delete a group.....	150
Add, edit, or delete a user	155
Manage jobs	
Job management (scheduled pinboards)	159
Scheduled pinboards management.....	161
Security	
Overview of security features.....	163
System Security.....	164
Data security	
Data security.....	166
Share tables and columns.....	170
Share worksheets.....	172
Share a pinboard.....	174
Security for SpotIQ functions.....	176
Revoke access (unshare).....	177
Row level security (RLS)	
About RLS.....	179
How Rule-Based RLS works.....	180
Set Rule-Based RLS.....	182
System administration	
Overview of System administration.....	186
System monitoring.....	187
Using the Space Utilization chart.....	193
Send logs when reporting problems.....	194
Set up recording for Replay Search.....	196
Backup and restore	
Understand the backup strategies.....	200
Understand backup/snapshot schedules	202
Work with snapshots	205
Work with backups	
Understand backup modes	207
Create a manual backup.....	209
Configure periodic backups	210
About restore operations.....	212

Troubleshooting

About troubleshooting.....	213
Get logs.....	214
Upload logs to ThoughtSpot Support.....	216
Network connectivity issues.....	217
Change the timezone.....	218
Browser untrusted connection error	219
Characters not displaying correctly.....	220
Clear the browser cache.....	221
Cannot open a saved answer that contains a formula.....	223
Data loading too slowly.....	225
Search results contain too many blanks.....	226
About this reference.....	227
Keyword reference	228
TQL reference	239
ThoughtSpot Loader flag reference.....	246
tscli command reference.....	249
Date and time formats reference.....	264
Row level security rules reference.....	267
Formula reference	275
Error code reference	285

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

There are three separate default administrator users:

Login Type	User	Access Method	Password
OS user	admin	SSH. Used for work that requires sudo or root privileges. Does not exist for application login.	Contact ThoughtSpot to obtain the default password.
OS user	thoughtspot	SSH. Used for command line work that does not require sudo or root privileges. Does not exist for application login.	Contact ThoughtSpot to obtain the default password.
Application user	tsadmin	Browser. Access through a Web browser and for non-sudo or root privilege work	Contact ThoughtSpot to obtain the default password.

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.

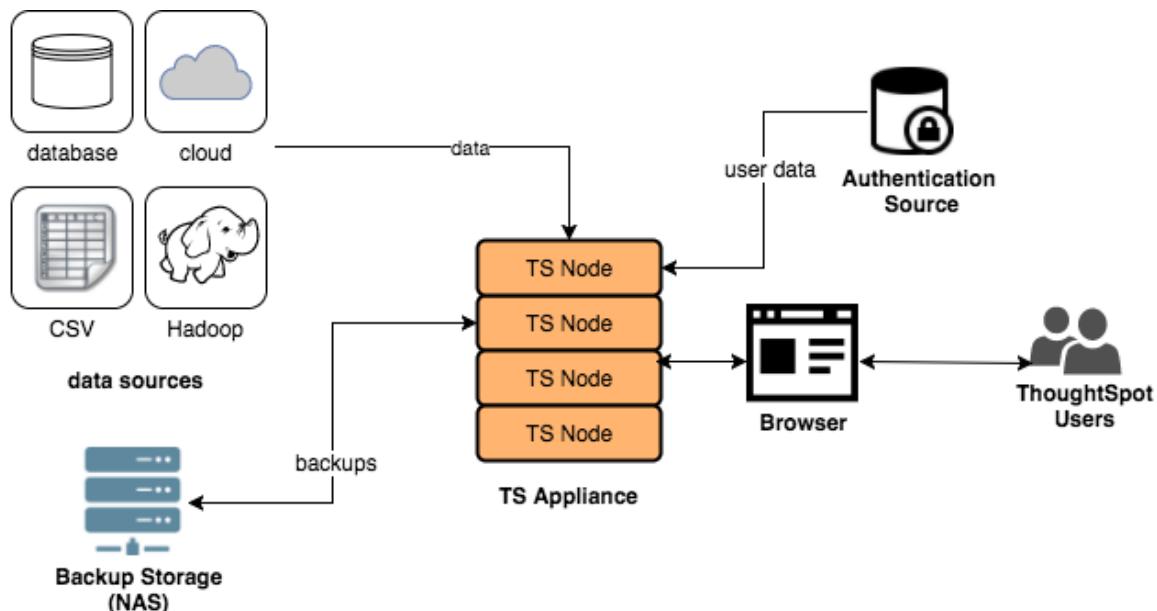
To log in to ThoughtSpot from a browser:

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.

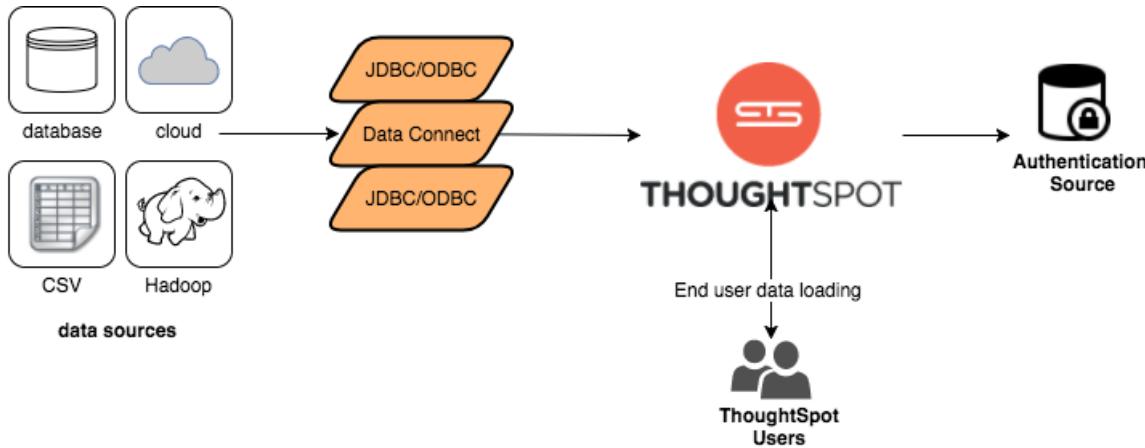
The ThoughtSpot appliance can be a physical appliance that ThoughtSpot ships, one or more AWS instances that are clustered together, or one or more VMware instances that are clustered together. From an external interface, regardless of the appliance type, the appliance appears to be a single instance.

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 search-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
<ul style="list-style-type: none"> Have an ETL load, for example, Informatica, SSIS, and so forth. Have available resources to create and manage ETL. Have smaller daily loads. 	<ul style="list-style-type: none"> 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. 	<ul style="list-style-type: none"> 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 does all analysis against data in memory to help achieve fast results across millions and billions of records of data.

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
<ul style="list-style-type: none">• 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.	<ul style="list-style-type: none">• Configuration.• Users authenticate against LDAP or AD.• Option to sync users and groups with ThoughtSpot to manage group membership.	<ul style="list-style-type: none">• 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.

There are updates planned in future versions that will provide stricter access to columns and stronger governance of data.

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 .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.

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	2B 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

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	10M
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.

To perform the set up and configuration, you must first learn how to [gain administrative access](#). If you haven't already done this, do this now. Then, complete the following tasks as needed:

- [Set your ThoughtSpot locale](#)
In addition to American English, ThoughtSpot also supports German and Japanese.
- [Test network connectivity between nodes](#)
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.
- [Set the relay host for SMTP \(email\)](#)
To enable alert emails, you'll need to set up a relay host for SMTP traffic from ThoughtSpot. This routes the alert and notification emails coming from ThoughtSpot through an SMTP email server.
- [Set up a fiscal calendar year](#)
Many companies start their fiscal calendar in a month other than January. If this is the case in your company, setting a fiscal calendar quarter makes the ThoughtSpot date searches reflect your fiscal year.
- [Configure SSL \(secure socket layers\)](#)
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.
- [Configure SAML](#)
ThoughtSpot can use Security Assertion Markup Language (SAML) to authenticate users. You can set up SAML through the shell on ThoughtSpot using a tscli based configurator.
- [About LDAP integration](#)
Some companies use LDAP (Lightweight Directory Access Protocol) to manage user authentication. Using LDAP provides security and makes user management more centralized.
- [Configure a NAS file system](#)
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.
- [Configure support services](#)
Set up and configure ThoughtSpot support services for your installation. You can designate a support contact (email and phone number), set up a reverse tunnel to allow ThoughtSpot Support to get access to your ThoughtSpot instance. Enable an optional "call home" capability sends usage statistics to ThoughtSpot Support once a day via a secure file server.
- [Set up monitoring](#)
To configure monitoring of your cluster, set up the frequency of heartbeat and monitoring reports and an email address to receive them.
- [Network ports](#)
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.
- [About load balancing and proxies](#)
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.

Set your ThoughtSpot locale

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

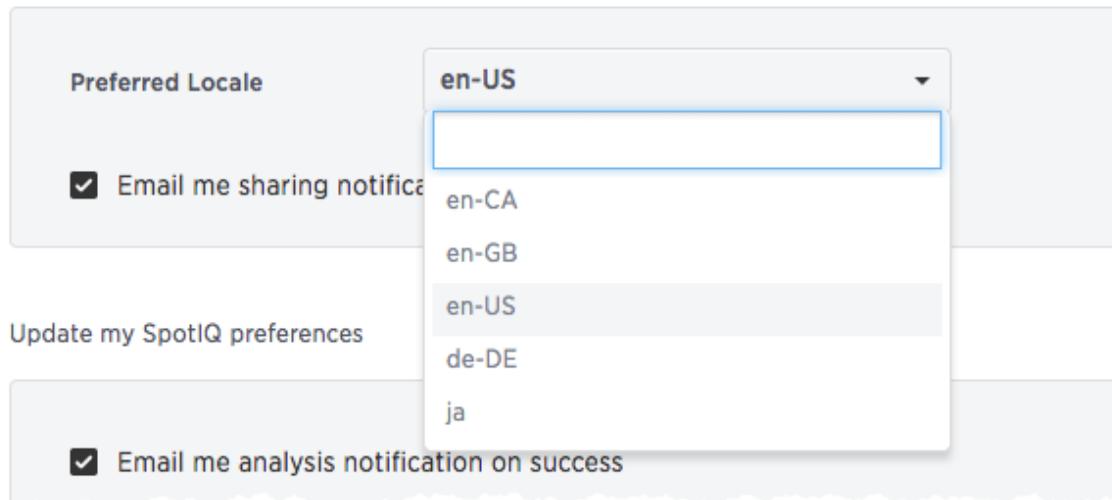
The language the ThoughtSpot UX displays is based off of the locale in a user's profile. The locale preferences control the language and data formats (date and number formats) by geographic locations. In addition to American English (*en-US*), ThoughtSpot supports:

- German (*de-DE*)
- Japanese (*ja*)
- Canadian English (*en-CA*)
- United Kingdom English (*en-GB*)

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

Update my preferences



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**.

⚠ Warning: It is possible that you set your locale but find strings in the UI still appear in English, this indicates an untranslated string. Please notify ThoughtSpot support.

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 is 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.

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 security 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.

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://192.168.2.100:389):
ldap://192.168.2.100:389

Default domain (Example: ldap.thoughtspot.com): ldap.thoughtspot.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@ldap.thoughtspot.com" -W  
-b "dc=ldap,dc=thoughtspot,dc=com" 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`

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)://<host>:<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.thoughtspot.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 currently in LDAP tree  
being synced (y/n): n  
Distinguished name for the base to start searching groups 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 and including the  
specified base DN: 2  
  
Filter string to apply the search to: (|(CN=TestGroupAlpha) (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 members 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=TestGroupBeta))' \  
--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 mountpoint 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>
```

- 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>
```

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](#) to 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. When the ticket has been completed, continue with the remaining steps in this procedure to make the settings on your side.
3. Log in to 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 has been applied:

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

6. Enable the remote tunnel:

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

7. [Contact ThoughtSpot](#) again, so you can test the setup with your ThoughtSpot Support contact.
8. After your remote session with ThoughtSpot Support, you should 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. Test that the remote tunnel has been 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> [--password <password>]
```

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

Enable call home capability

The optional “call home” capability sends usage statistics to ThoughtSpot Support once a day via a secure file server.

Before you can enable the call home feature:

1. [Configure the connection to the file server](#).
2. Obtain the customer name as recognized by the file server.

The customer name is formatted like this example: `Shared/<customer_name>/stats`. If you do not know the customer name, [contact ThoughtSpot Support](#).

This can be helpful when troubleshooting problems with ThoughtSpot Support, because they will be able to see basic usage information over time for your ThoughtSpot instance.

To set up the call home feature:

1. Log in to the Linux shell using SSH.
2. Enable the call home feature by issuing:

```
$ tscli callhome enable --customer_name <customer_name>
```

3. If you want to disable call home in the future, you can do so by issuing:

```
$ 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:

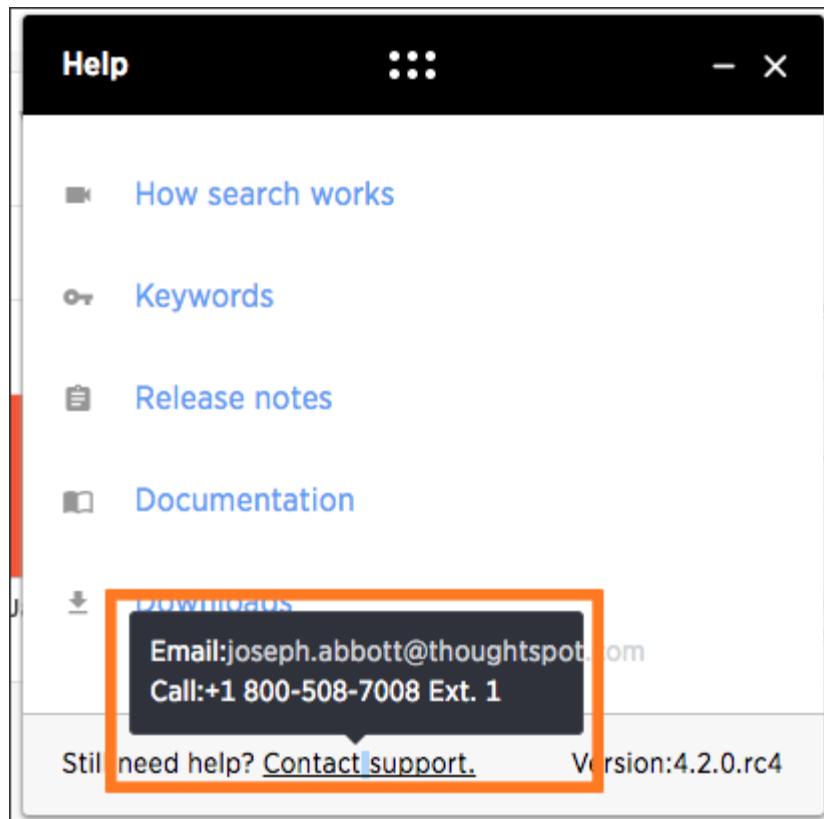
```
$ tscli support rm-admin-email
```

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

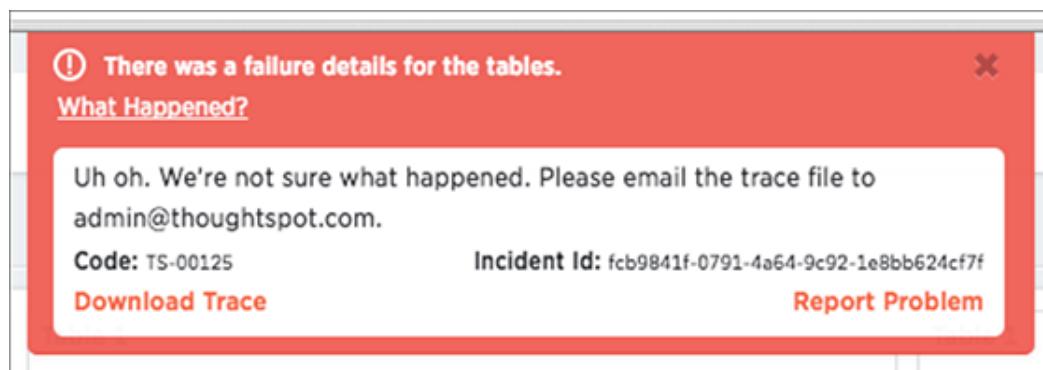
How users find your company's support contact

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

- In the Help Center, when they click **Contact Support**.



- In error messages, when they click 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	All nodes	Port used to debug the stats collector

Port	Protocol	Service Name	Direction	Source	Dest.	Description
				nodes		
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](#)

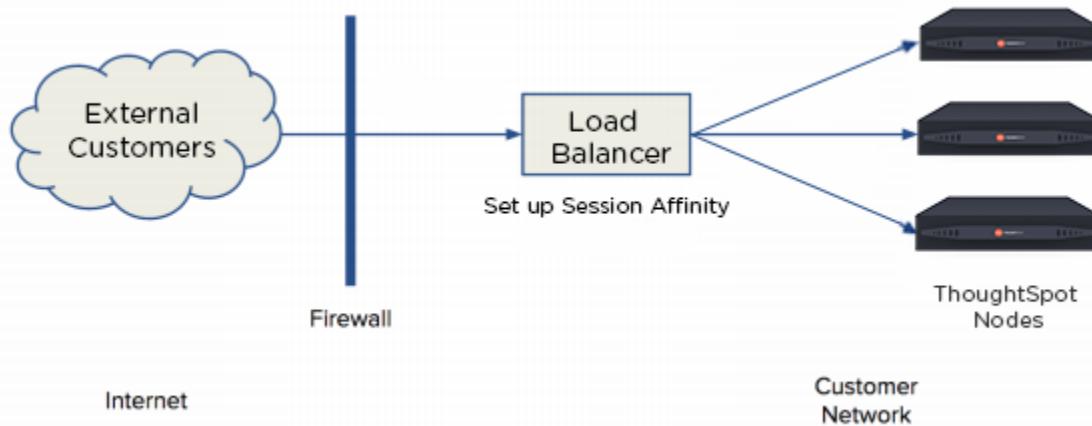
About load balancing and proxies

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.

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.

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.
Use the Informatica Connector	Use the Informatica Connector if you already use Informatica to connect to your other data sources. For information, see the ThoughtSpot Data Integration Guide.	Works with your established data migration processes in Informatica.

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.

Tip: 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.

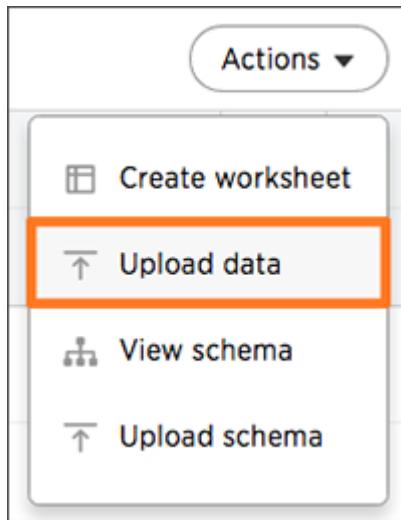
Load the CSV File

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 on DATA, on the top navigation bar.



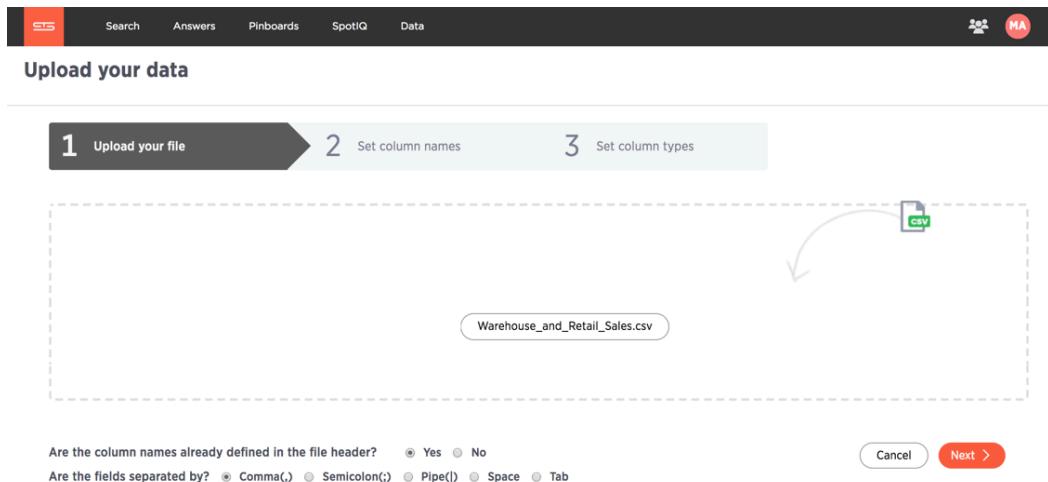
3. Click the Actions button in the upper right corner, and select Upload Data.



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

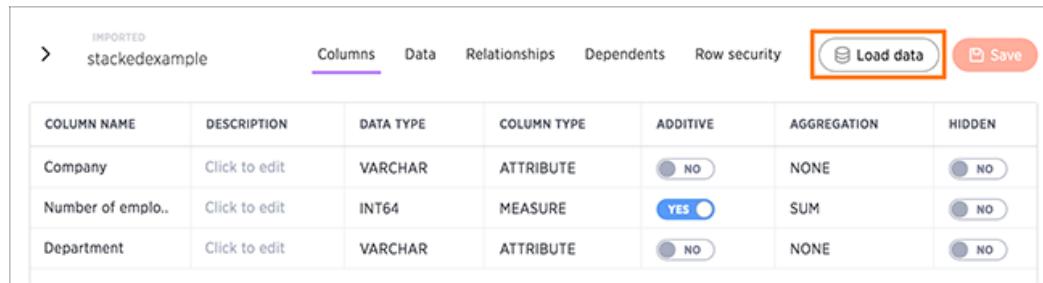
You can append data to your existing system tables through the ThoughtSpot application, even if the tables were initially loaded using Data Connect or `tsload`. The CSV file must have the same structure as the table it is being loaded into, including number and type of columns, in the same order as the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

To append data into ThoughtSpot:

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



3. Click the on the table you would like to append data to.
4. Click the Load data button.



stackedexample		Columns	Data	Relationships	Dependents	Row security	Load data	Save
Column Name	Description	Data Type	Column Type	Additive	Aggregation	Hidden		
Company	Click to edit	VARCHAR	ATTRIBUTE	<input type="radio"/> NO	NONE	<input type="radio"/> NO		
Number of emplo..	Click to edit	INT64	MEASURE	<input checked="" type="radio"/> YES	SUM	<input type="radio"/> NO		
Department	Click to edit	VARCHAR	ATTRIBUTE	<input type="radio"/> NO	NONE	<input type="radio"/> NO		

5. Upload the CSV or Excel file by doing one of these options:
 - Click on **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. Answer the question **Do you want to append to the existing data or overwrite it?**
8. Answer the question **Are the fields separated by?** Click **Next**.
9. Click on the column header names to change them to more useful names, if you'd like. Click **Next**.
10. 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**.
11. Click **Import**.
12. Click **Link to Existing Data** if you want to link the data you uploaded to the data in another table or worksheet. Or click **Search** if you want to begin a new search.

How to view a data schema

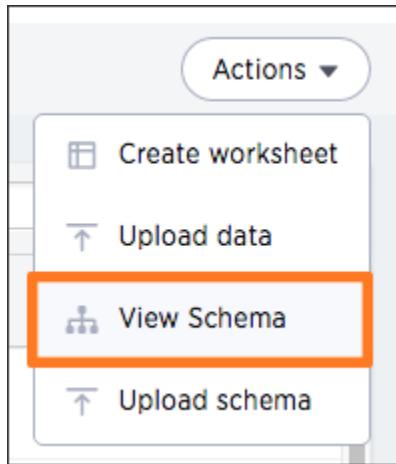
Summary: Use the schema viewer to see tables and worksheets and their relationships.

There is a schema viewer in ThoughtSpot which 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.

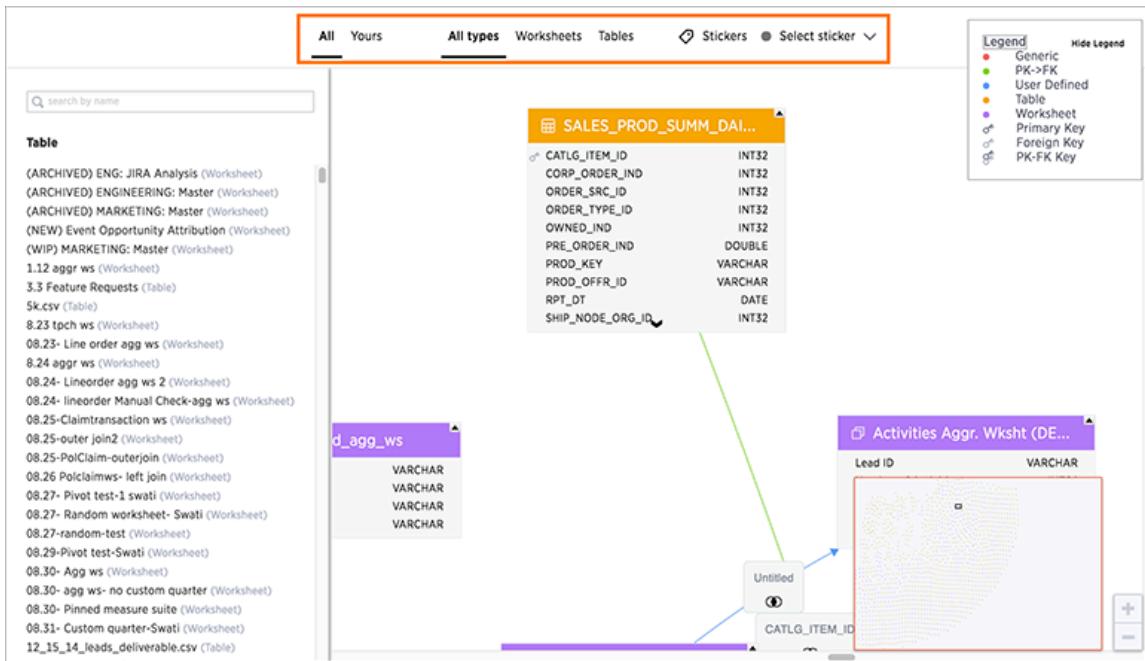
You need Admin or Data Management privilege to use the Schema Viewer.

Bringing up the Schema Viewer

You can access the Schema Viewer from the DATA screen by clicking Actions, and selecting 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:

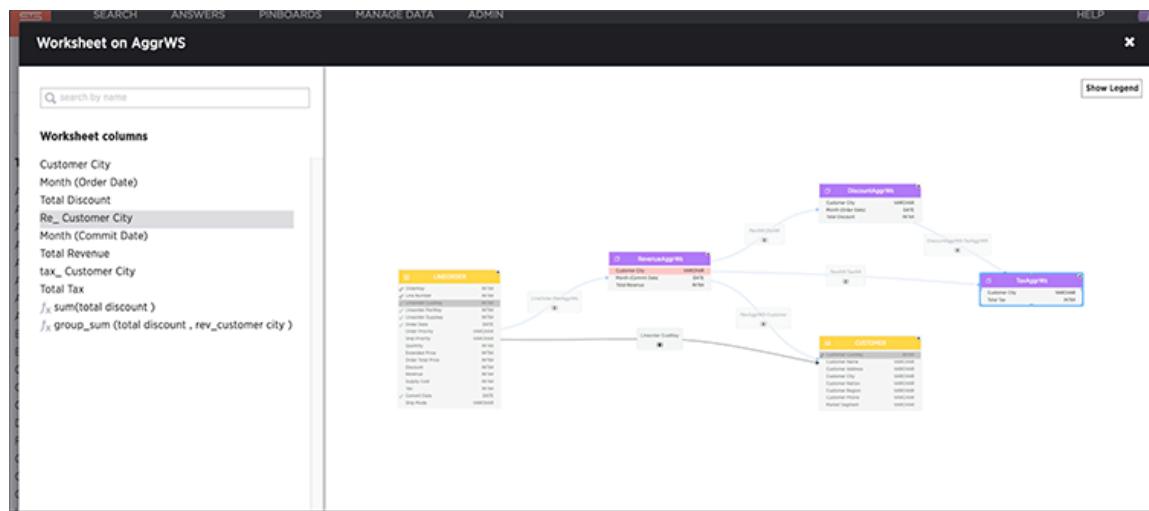
- What is the relationship between two tables?
- What tables make up this worksheet, and how are they joined?

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.

Worksheet view

For worksheets, you can also click on one to view the worksheet. 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 will not show the correct join paths.



The worksheet view does not work for aggregated worksheets, but does works for worksheets built on top of aggregated worksheets.

Schema planning concepts

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

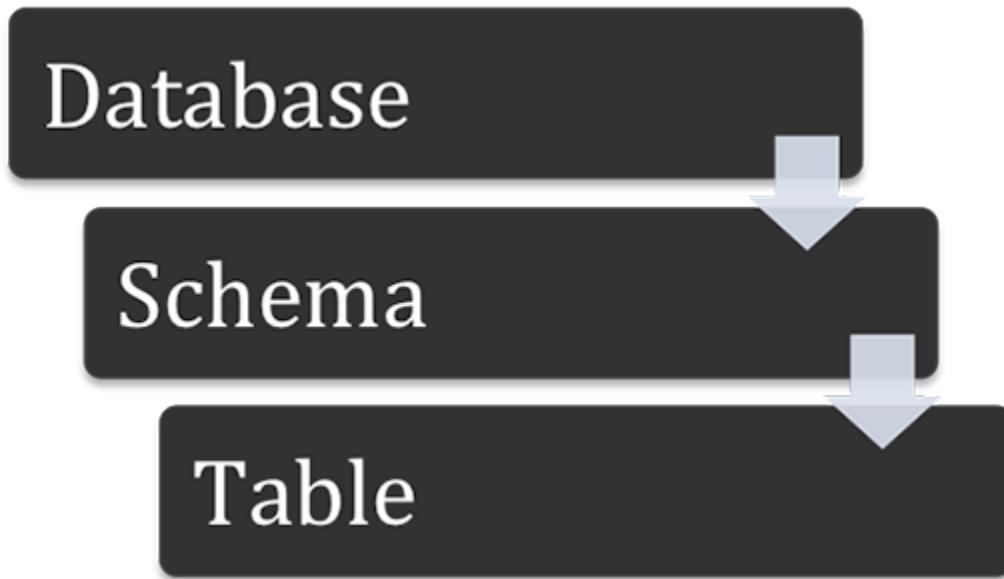
Before you can load data with ThoughtSpot Loader, you must create a schema to receive it, using the SQL command line interface (TQL).

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.
Boolean	BOOL	Can be true or false.
Integer	INT or 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.

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 help ThoughtSpot with default schema modeling by indicating a connection between two tables. These relationships are used for joining the tables, and not for referential integrity constraint checking. The foreign key relationship is defined on the fact table and references the primary key(s) in 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.

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 "retail_sales" AS
"wholesale_buys"."fruit" = "retail_sales"."fruit" and ("wholesale_buys"."date_ordered" < "retail_sales"."date_sold" and "retail_sales"."date_sold" < "wholesale_buys"."expiration_date");
```

Tip 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.

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.

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*.

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

Number of Nodes	Number of Shards
61-72	768

If you omit the `PARTITION 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 `PARTITION 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.

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.

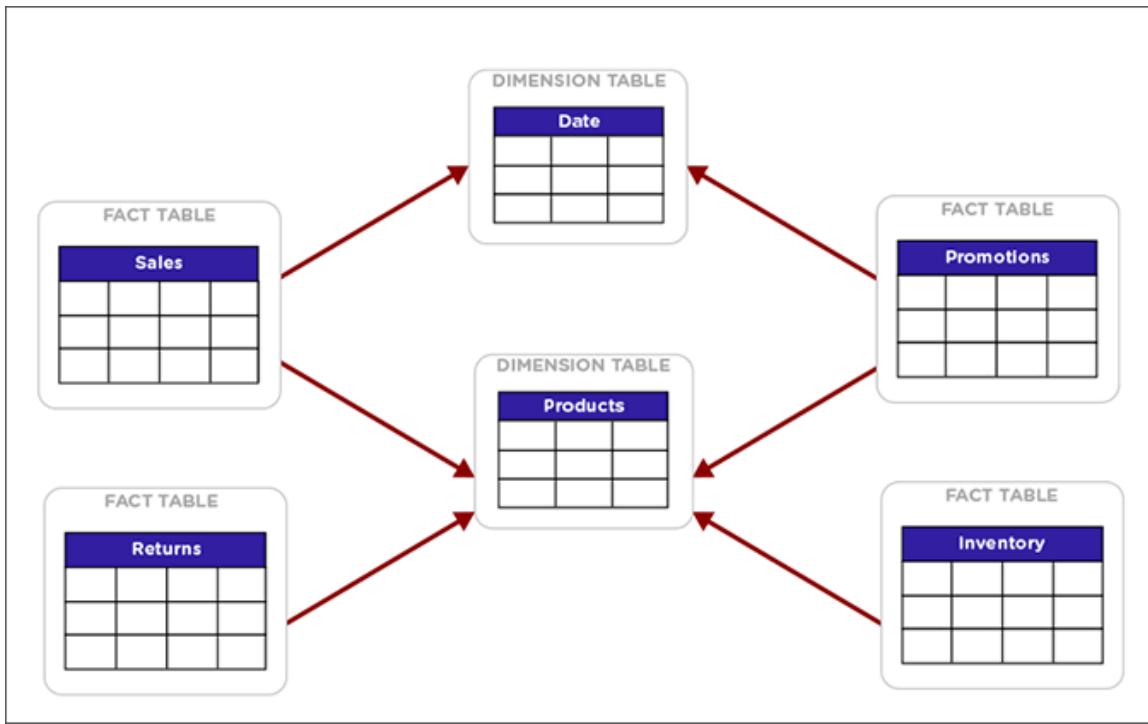
AppleTypeID	Color	Name	Use
55	Red	Red Delicious	Snack
34	Green	Granny Smith	Cooking
09	Yellow	Golden	Snack

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

If your database schema contains any chasm traps, you may encounter these limitations.

Operations

The following limitations on chasm trap schemas will produce a red bar error in the ThoughtSpot application:

- **Show underlying data** does not work for chasm trap searches, whether the search is on a worksheet containing a chasm trap or on base tables that are related over a chasm trap.
- When using the ThoughtSpot APIs, you cannot pass filter values via the URL if the relevant

- searches occur on a worksheet containing a chasm trap or on base tables that are related over a chasm trap.
- Use [Rule-Based Row Level Security](#).

Behavior

The following behavior is different for chasm traps than for schemas that do not contain a chasm trap:

- There are no headlines (single facts based on the data) shown when a search contains a worksheet containing a chasm trap or base tables that are related over a chasm trap.
- 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.
- There are cases when attempting to configure certain charts on chasm trap worksheets or tables will not work. If this happens, you will see the error **Your search needs to have unique y-axis values for each series of data shown on the x-axis**. The workaround is to remove all columns from the search, except for those used in your chart.

Workarounds

In some cases, there is a workaround of saving an answer as a worksheet (Aggregated Worksheet). See the ThoughtSpot User Guide for details on how to do this. If you save a chasm trap search as a worksheet, it becomes a materialized view of the answer. Effectively, it is then just a regular table (no chasm trap). As such, most of the issues above will not affect searches on the saved worksheet.

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 TQL 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:

```
$ tq1
```

```
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_shipppriority" 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_schema"."customer"  
("c_custkey"),  
CONSTRAINT FOREIGN KEY ("lo_suppkey") REFERENCES "sample_schema"."supplier"  
("s_suppkey")  
) PARTITION BY HASH (96) KEY (lo_orderkey);
```

Upload and run a SQL script

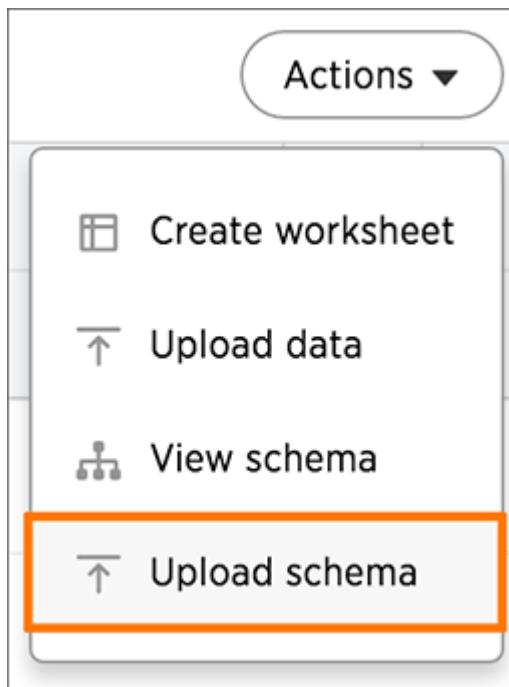
Summary: Importing a schema through the Web browser makes it possible to run your SQL script without needing to have a Linux login.

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:

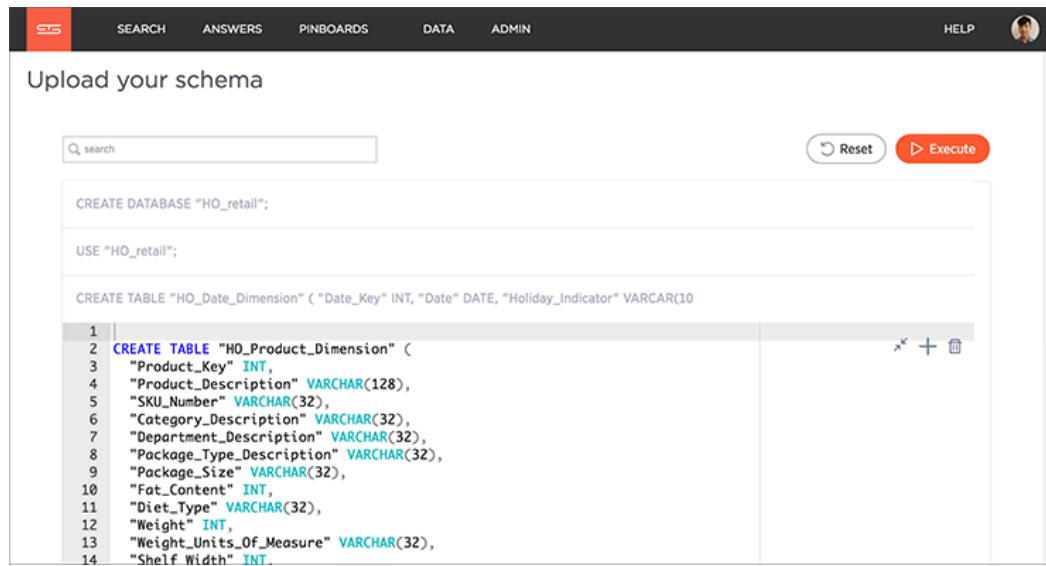
- [Create the SQL script ahead of time](#), and use the browser to run it.
 - Use the editor to type your SQL directly into the browser.
 - Use the browser SQL interface as an interactive SQL editor, for example to test an existing script or make changes to an existing schema.
1. Log into ThoughtSpot from a browser.
 2. Click on DATA, on the top navigation bar.



3. Click Actions 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.



The screenshot shows the ThoughtSpot interface with a dark header bar. The header includes the ThoughtSpot logo, navigation links for SEARCH, ANSWERS, PINBOARDS, DATA, ADMIN, and HELP, and a user profile icon.

The main area is titled "Upload your schema". It features a search bar and two buttons: "Reset" and "Execute".

The code editor contains the following SQL script:

```
CREATE DATABASE "HO_retail";
USE "HO_retail";
CREATE TABLE "HO_Date_Dimension" ( "Date_Key" INT, "Date" DATE, "Holiday_Indicator" VARCAR(10)
1 | 2 CREATE TABLE "HO_Product_Dimension" (
3   "Product_Key" INT,
4   "Product_Description" VARCHAR(128),
5   "SKU_Number" VARCHAR(32),
6   "Category_Description" VARCHAR(32),
7   "Department_Description" VARCHAR(32),
8   "Package_Type_Description" VARCHAR(32),
9   "Package_Size" VARCHAR(32),
10  "Fat_Content" INT,
11  "Diet_Type" VARCHAR(32),
12  "Weight" INT,
13  "Weight_Units_Of_Measure" VARCHAR(32),
14  "Shelf_Width" INT.
```

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 reshuffle 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 [strftime 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);
```

Boolean to string conversions

Boolean to string 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. 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"
```

4. Repeat the data load for each of your CSV files.

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 "|"
```

Use a script to load data

Summary: Learn how to load one or more 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_records.txt
  --has_header_row --alsologtostderr --null_value ""

cat AllstarFull.csv | tsload
  --target_database baseball --target_table "allstarfull"
  --empty_target --field_separator ","
  --max_ignored_rows 10 --bad_records_file bad_records.txt
  --has_header_row --alsologtostderr --null_value ""

cat Appearances.csv | tsload
  --target_database baseball --target_table "appearances"
  --empty_target --field_separator ","
  --max_ignored_rows 10 --bad_records_file bad_records.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 | tsload
--target_database sales --target_table SALES_FACT --max_ignored_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 delimited
--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_ignored_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 delimited
--field_separator "|" --null_value "" --enclosing_character "\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &

cat day13.csv day14.csv day15.csv day16.csv day17.csv day18.csv | tsload
--target_database sales --target_table SALES_FACT --max_ignored_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 delimited
--field_separator "|" --null_value "" --enclosing_character "\""
--boolean_representation 1_0 &

pidlist="$pidlist $!" &

cat day19.csv day20.csv day21.csv day22.csv day23.csv day24.csv | tsload
--target_database sales --target_table SALES_FACT --max_ignored_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 delimited
--field_separator "|" --null_value "" --enclosing_character "\""
--boolean_representation 1_0 &

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

```
cat day25.csv day26.csv day27.csv day28.csv day29.csv day30.csv | tsload  
--target_database sales --target_table SALES_FACT --max_ignored_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 delimited  
--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](#). 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 on 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.

The screenshot shows the 'Tables' tab selected in the top navigation bar. Below it, the 'Data Sources' section is visible. A search bar contains the text 'udemy'. Under the 'All' tab, there is a list of data sources: 'Udemy_user_pr...', 'Udemy_user_co...', 'Udemy_user_act...', and 'Udemy Workshe...'. On the right, the 'Dependents' tab is selected in a row for 'Udemy Worksheet'. This row includes links for 'Columns', 'Data', 'Relationships', 'Dependents' (which is underlined in purple), and 'Row security'. The 'Dependents' table has columns for 'COLUMN NAME', 'DEPENDENT NAME', and 'TYPE'. It lists several entries, all of which are 'CONTENT - Udemy..':

COLUMN NAME	DEPENDENT NAME	TYPE
date enrolled	CONTENT - Udemy..	Pinboard
email	CONTENT - Udemy..	Pinboard
number of course..	CONTENT - Udemy..	Pinboard
number of modul..	CONTENT - Udemy..	Pinboard
last name	CONTENT - Udemy..	Pinboard
first name	CONTENT - Udemy..	Pinboard

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 on 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.

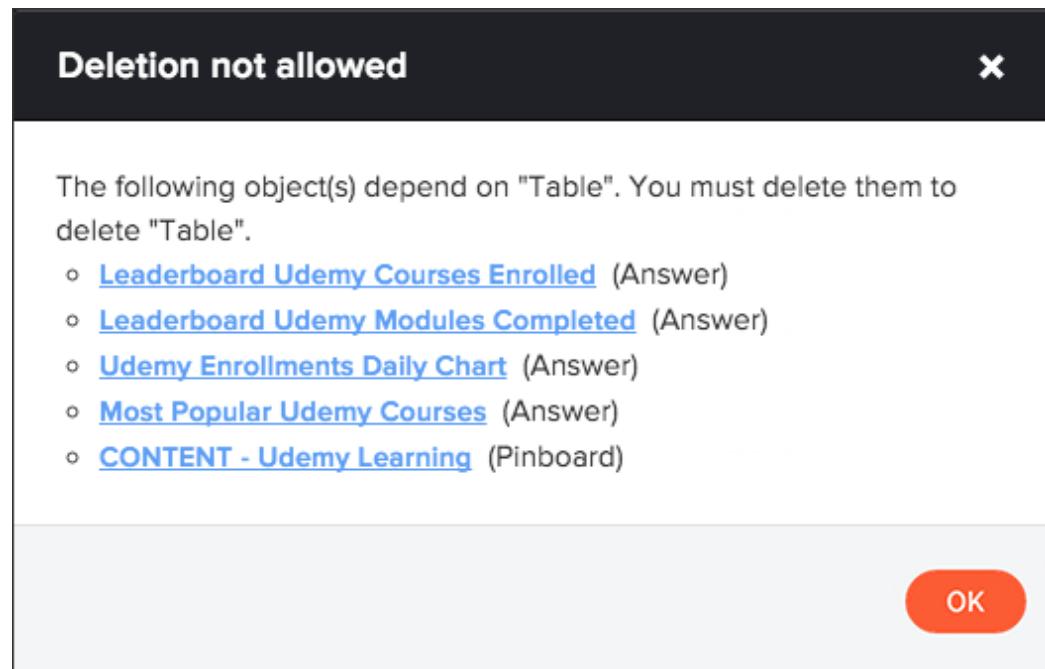
A screenshot of the ThoughtSpot DATA page. The top navigation bar shows the 'DATA' tab is selected. Below the navigation bar, there are two tabs: 'Tables' and 'Data Sources'. The 'Data Sources' tab is selected and highlighted with an orange underline. There are several filters at the top: 'All' and 'Yours' dropdowns, 'All types' dropdown set to 'Worksheets', and buttons for 'Tables' and 'Stickers'. Below the filters are three buttons: 'Share', 'Delete' (which is highlighted with an orange box), and 'Apply Sticker'. The main area displays a table with columns: NAME, DESCRIPTION, and SOURCE. Three data sources are listed:

- Wistia Stats - Vide.. (unchecked)
- Users_Courses_Do.. (unchecked)
- Udemy Worksheet (checked)

The row for 'Udemy Worksheet' is highlighted with a yellow background.

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 relationship "products"
with table "sales", which will break 34 user-visible visualizations 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 table 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.

1. Click on the DATA icon, to get to the data management listing.
2. Click on a data source you own or can edit.

NAME	DESCRIPTION	SOURCE
Olympic medals dat..		

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.

COLUMN NAME	DESCRIPTION	DATA TYPE	COLUMN TYPE	ADDITIVE	AGGREGATION
Athlete Name	Click to edit	VARCHAR	ATTRIBUTE	NO	NONE
Age	Click to edit	INT64	MEASURE	YES	SUM
Country	Click to edit	VARCHAR	ATTRIBUTE	NO	NONE
Year	Click to edit	INT64	MEASURE	YES	SUM
Gold Medals	Click to edit	INT64	ATTRIBUTE	YES	SUM
Silver Medals	Click to edit	INT64	ATTRIBUTE	YES	SUM
Bronze Medals	Click to edit	INT64	MEASURE	YES	SUM

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](#)

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.

The screenshot shows the ThoughtSpot Data Management interface. At the top, there are tabs for SEARCH, ANSWERS, PINBOARDS, DATA, and ADMIN. Under the DATA tab, sub-options include User Management, Data Management, System Health, Business Data Model, and Data Security. The Business Data Model option is highlighted.

The main content area displays three numbered steps:

- 1 Download the modeling file**: Click on the button to download the modeling file. A small icon with a downward arrow and the word "Download" is shown.
- 2 Update the file**: Set the column names and copy cells. An icon of a grid with an "XLS" extension is shown.
- 3 Upload the modeling file (.xls, size < 50MB)**: After your file is ready, just drag and drop it in the zone below or click on the button to get it.

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.

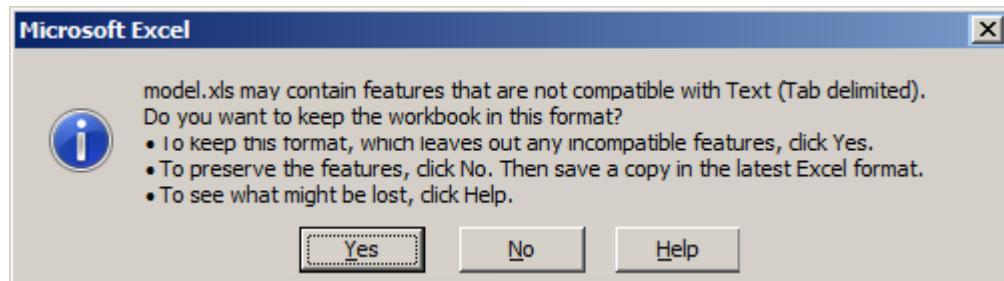
2. Find the column you want to modify.

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.

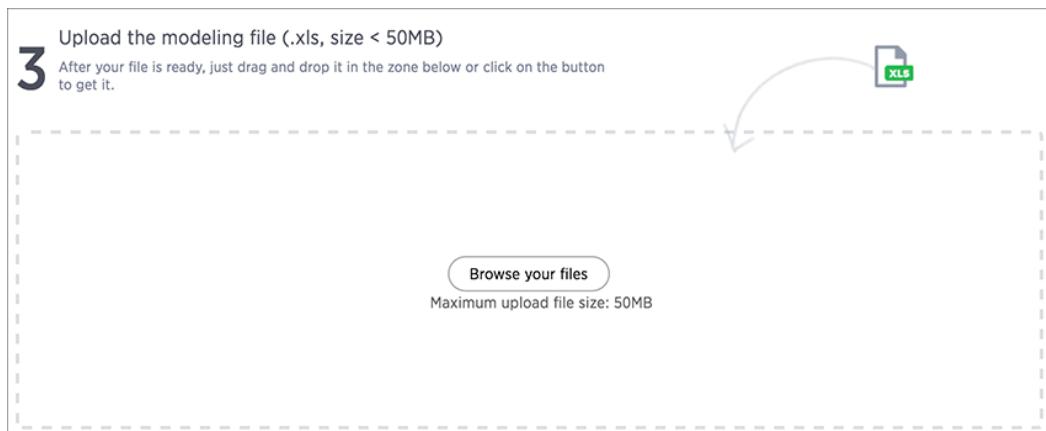
Upload the edited file

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 have changed 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 modelling 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	Can be modeled in a Worksheet
Column Name	Sets the name of the column to be used in searches.	Yes
Description	Adds a text description of what the column contains.	Yes
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.	Yes
Index Type	Sets the type of index that will be created for a column.	
Geo Type	Enables a column to be used in GeoMap visualizations.	
Priority	Changes the priority of a column in search suggestions.	
Number Format	Specifies the format to use when showing a numeric value in the column.	
Date Format	Specifies the format to use when showing the dates in a column.	
Currency Format	Specifies the format to use when showing the currencies in a column.	

Setting name	Description	Can be modeled in a Worksheet
Attribution Dimension	Only applies to tables that join over a Chasm Trap . Designates whether the tables depend on this column for attribution.	

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 the search suggestions priority 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

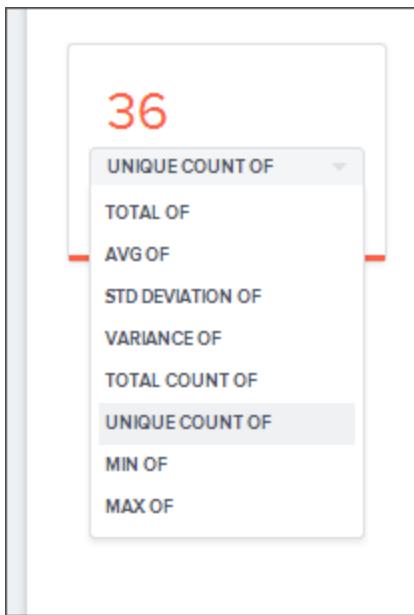
Your data may contain a column with a numeric data type that you have defined as an ATTRIBUTE rather than a MEASURE. For example, you may have ATTRIBUTE column with an INTEGER data type that represents age. Typically, these columns have an ADDITIVE setting of NO. Within a search result that contains 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.
STD_DEVIATION	Calculates the standard deviation of all the values.

Aggregate type	Description
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 are 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](#)

Change the index type

Summary: ThoughtSpot indexes column names and unique column values. The indexes are used to dynamically generate suggestions in the search bar when typing a search.

A column's index influences the suggestions that appear for that column in search. You can modify a specific column's **INDEX TYPE** in the **DATA > Tables > Columns** page or to set a system-wide **Index** value in the modeling file.

Warning: If a column has a very large free text values, ThoughtSpot does not recommended to changing the column indexing. These should not to be indexed, because indexing on these values is not useful and may generate confusing suggestions.

Default indexing

The default behavior of indexing is as follows:

- All column names are indexed using their **ColumnName** value.
- Values for columns with the column type of **MEASURE** are not indexed.
- Values for columns with the data type of **DATE** are not indexed.
- Columns that contain a large amount of free-form text (i.e. the number of characters in more than a few of the fields is more than 50) are indexed as **PREFIX_ONLY** by default.
- Short strings (like a **firstname** column) are indexed using **PREFIX_AND_SUBSTRING** by default, which indexes both prefix and substrings.

You can override the default behavior by editing the modeling file to change the **Index** value for any columns that should be indexed differently. There are several different supported index types:

Index type	Description
DEFAULT	This is the default value. The default indexing behavior will apply to the column values, depending on their type. PREFIX_AND_SUBSTRING for short values and PREFIX_ONLY for long values and free-form text.
DONT_INDEX	Prevents indexing on the column values.
PREFIX_AND_SUBSTRING	Allows full indexing such that prefix and sub-string search both work for the column values.
PREFIX_ONLY	Allows indexing such that only prefix search works 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.

Make a change

1. Find the column whose index type you want to modify
2. Set its **Index Type**.

If you are using the model file, double click in the **Index** cell, and type in the index type you want to use. 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'

3. Save your changes.

Related information

[Model the data for searching](#)

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.

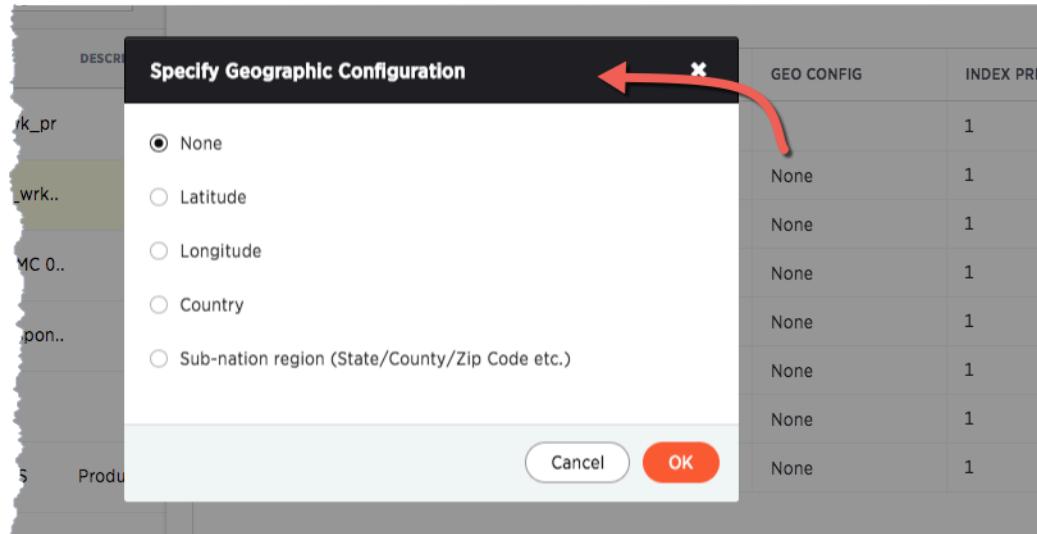
If you are using a column with the data type DOUBLE for latitude and longitude, you will also need to change the following settings for those columns:

- 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

</tr> </tbody> </table> ## Related information [Model the data for searching](semantic-modeling.html#)

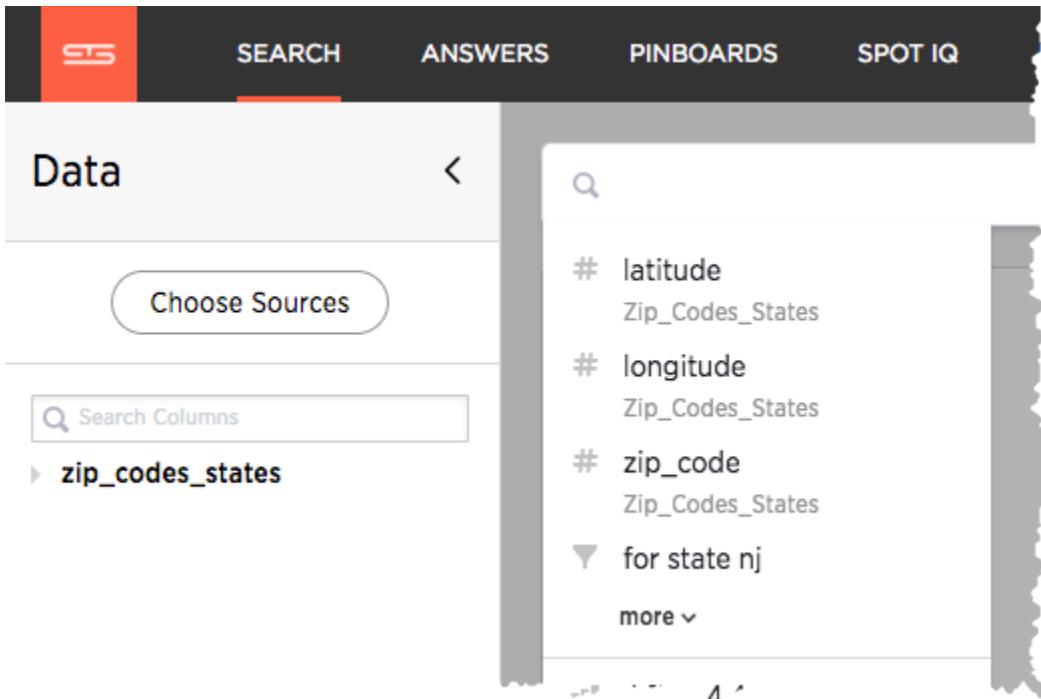
GeoType	Description	Type: Example
COUNTRY_REGION	Countries	<ul style="list-style-type: none"> • name: United States • long name: United States • name_sort: United States of America • abbreviation: U.S.A. • adm0_a3: USA • adm0_a3_is: USA • adm0_a3_us: USA • admin: United States of America • brk_a3: USA • 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	<ul style="list-style-type: none"> • santa clara county • pike county, ohio • pike county, OH
STATE_PROVINCE	States in the United States	<ul style="list-style-type: none"> • name: California • US Postal Service abbreviation: CA
LATITUDE</td>	Must be used with LONGITUDE	<ul style="list-style-type: none"> • 37.421023 • 1.282911
LONGITUDE	Must be used with LATITUDE	<ul style="list-style-type: none"> • -122.142103 • 103.848865
ZIP_CODE	Zip codes and zip codes +4 in the United States	<ul style="list-style-type: none"> • po_name: MT MEADOWS AREA

GeoType	Description	Type: Example
		<ul style="list-style-type: none">• ZIP: "00012"• zip2: 12
Other Sub-nation Regions	Administrative regions found in countries other than the United States	<ul style="list-style-type: none">• bremen• normandy• west midlands

Set the search suggestions priority

Summary: You can change the priority that determines which columns are shown in search suggestions and the order in which they appear.

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).

The screenshot shows the ThoughtSpot interface for managing a dataset named 'zip_codes_states'. At the top, there's a header with 'IMPORTED' and a back arrow. Below it, the dataset name 'zip_codes_states' is displayed next to a 'Load data' button and a red 'Save' button. The main area contains two tables. The first table lists columns with their current index priority values: zip_code (1), latitude (100), longitude (1), city (1), state (1), and county (1). The second table, titled 'INDEX PRIORITY', shows the same six rows with the value '1' in the 'INDEX PRIORITY' column.

COLUMN NAME	DESCR	INDEX PRIORITY
zip_code	Click ↗	1
latitude	Click ↗	100
longitude	Click ↗	1
city	Click ↗	1
state	Click ↗	1
county	Click ↗	1

INDEX PRIORITY
1
100
1
1
1
1

You should only use numbers between 1-10 in the INDEX PRIORITY field. These values impact the [usage based rankings \(UBR\)](#).

1. Change the value to a number between 1 and 10. Use a value between 8-10 for important columns to improve their search ranking. Use 1-3 for low priority columns.
2. Save your changes.

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, BIGINT).
- #,###.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

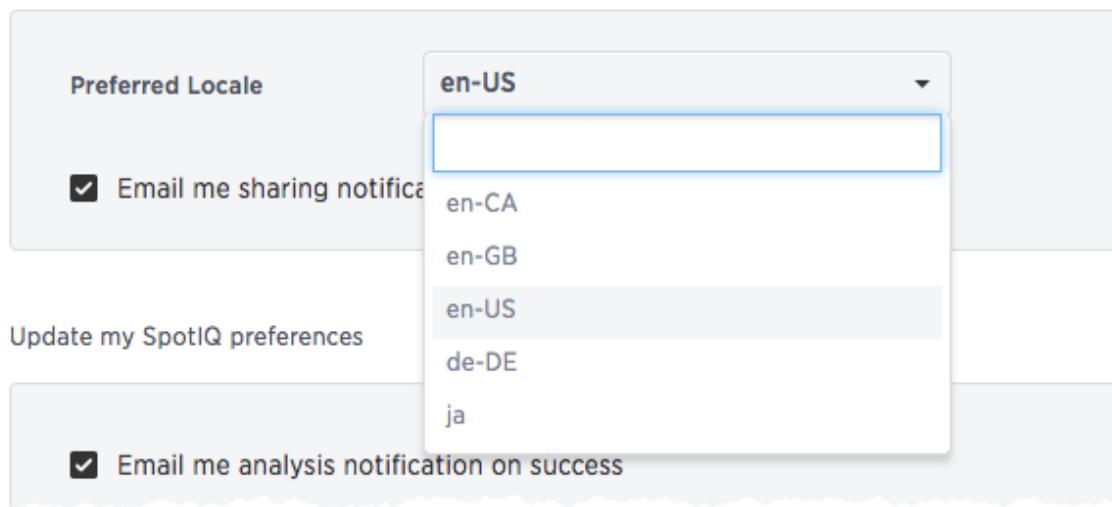
You can change the date format used to display a column's values [for a single table](#) or, by editing the data model, for [the entire ThoughtSpot instance](#). Editing the data model file requires that you have administrative privileges.

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.

Update my preferences



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
D	Day of year without a leading zero 0-365
DD	Day of year with up to one leading zero 01-365

Format mask	Description
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

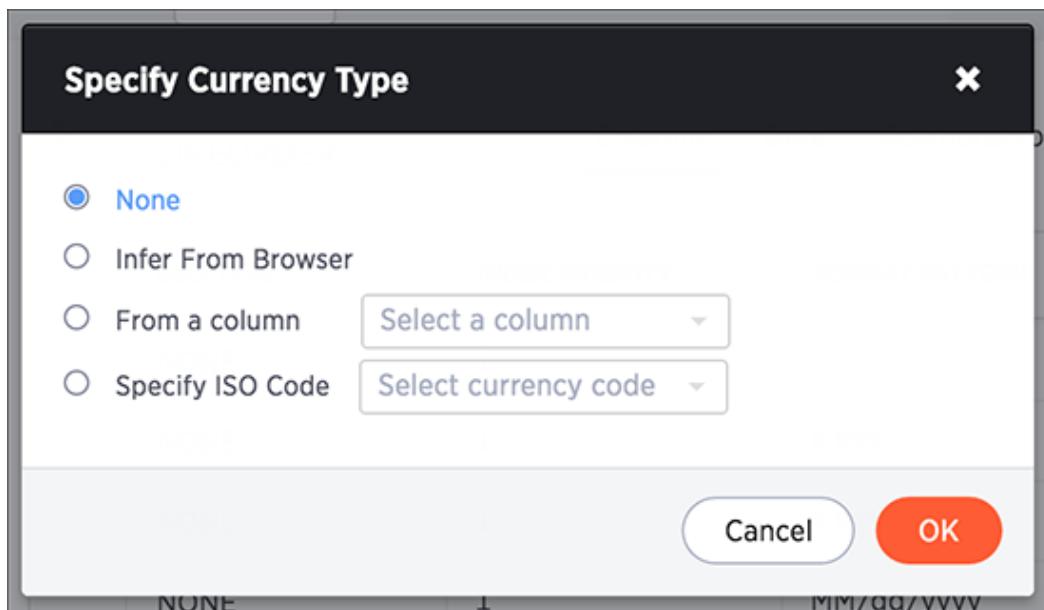
You can set a format for how currencies are displayed in tables and charts when using the ThoughtSpot Data API or embedding. For example, you can display currencies in a standard European Euro or US Dollar format based on the data modeling setting **Currency Type**.

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 on the DATA page, 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 złoty 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.
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.

Option	Description
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.

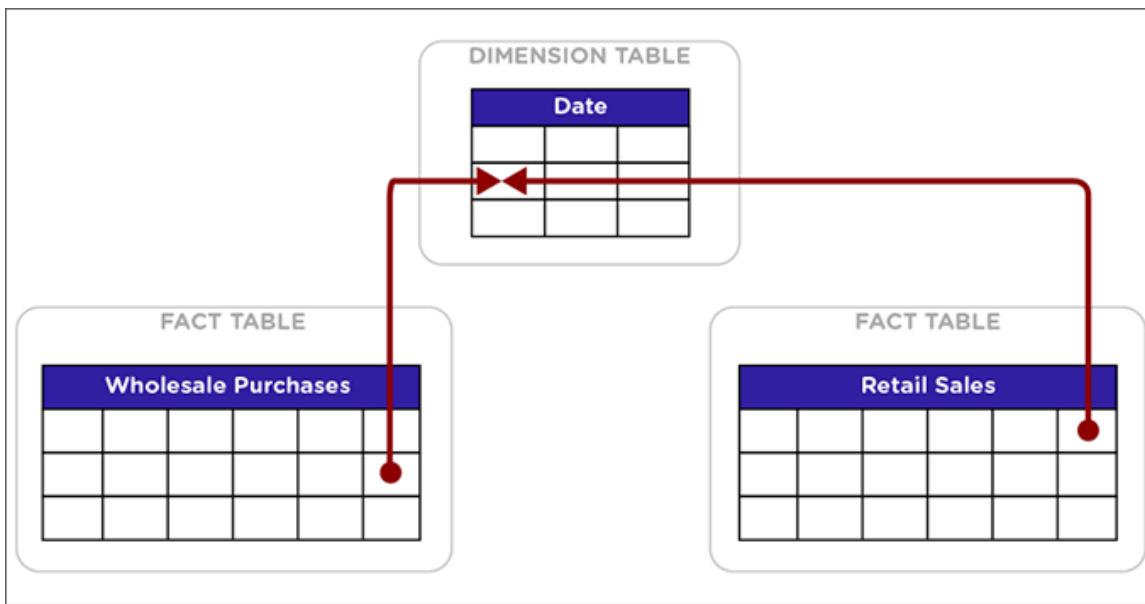
The **Attribution Dimension** setting only applies to tables that join over a [Chasm Trap](#). By default, the attribution dimension setting will be set to **YES**, but you can override that by setting the column's attribution dimension property to **NO**, as described here.

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.

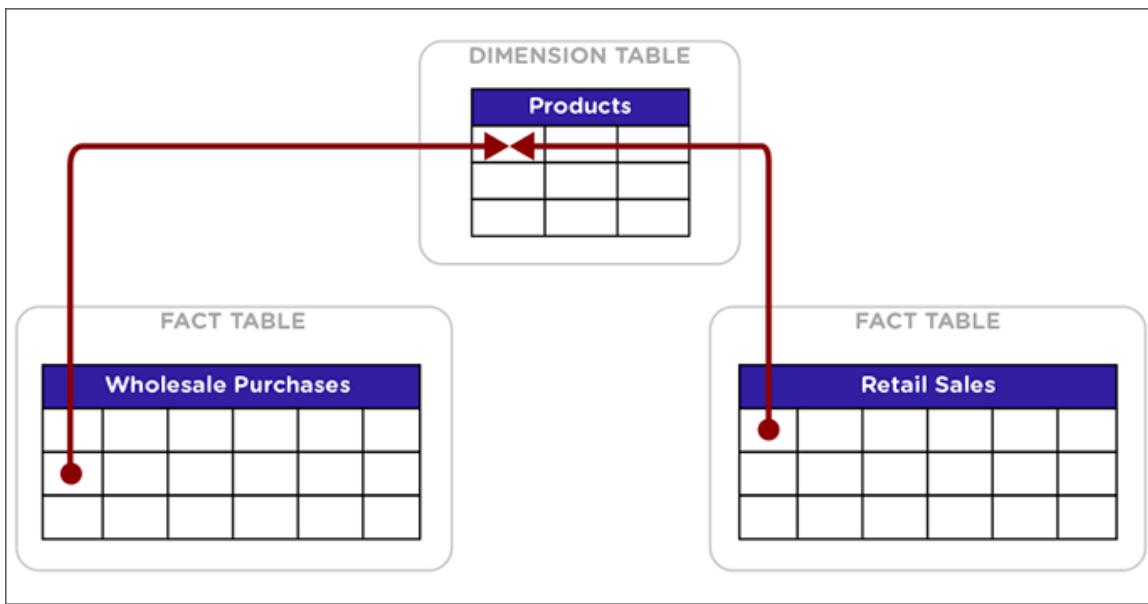
Usually, all goes well using this method. But sometimes an incorrect or illogical attribution can result. This can happen because the column chosen is not meaningful for performing this attribution. If you are seeing unexpected results in searches that include tables across a chasm trap, this setting 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](#)

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.

Create a relationship

Summary: Explains how to create a link between two data sources through a relationship.

You can quickly create a relationship (or link) between tables that allows you to combine them in a single search. Choose a column to join on that both tables contain (e.g. employee ID or product key).

You must have either administration privilege or modify access permission to the columns to create a relationship.

When creating a link 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. Click on DATA, on the top navigation bar.



2. Click on the name of the data source you want to link from.
3. Select Relationships.

A screenshot of the ThoughtSpot Relationships editor. The top navigation bar shows 'Tables' and 'Data Sources'. Below it, there are tabs for 'All', 'Yours', 'All types', 'Worksheets', 'Tables', 'Stickers', and 'Select sticker'. The 'Relationships' tab is highlighted with a red box. On the right, there is a table titled 'Relationships' with columns: COLUMN NAME, DESCRIPTION, DATA TYPE, COLUMN TYPE, ADDITIVE, AGGREGATION, and HIDE. The table contains five rows for columns: id, Queue, Subject, Status, and TimeEstimated. The 'ADDITIVE' column for id has a radio button set to 'YES'. The 'AGGREGATION' column for id has a radio button set to 'SUM'. The 'ADDITIVE' column for TimeEstimated has a radio button set to 'YES'. The 'AGGREGATION' column for TimeEstimated has a radio button set to 'SUM'. There are also 'Load data' and 'Save' buttons at the bottom right of the table area.

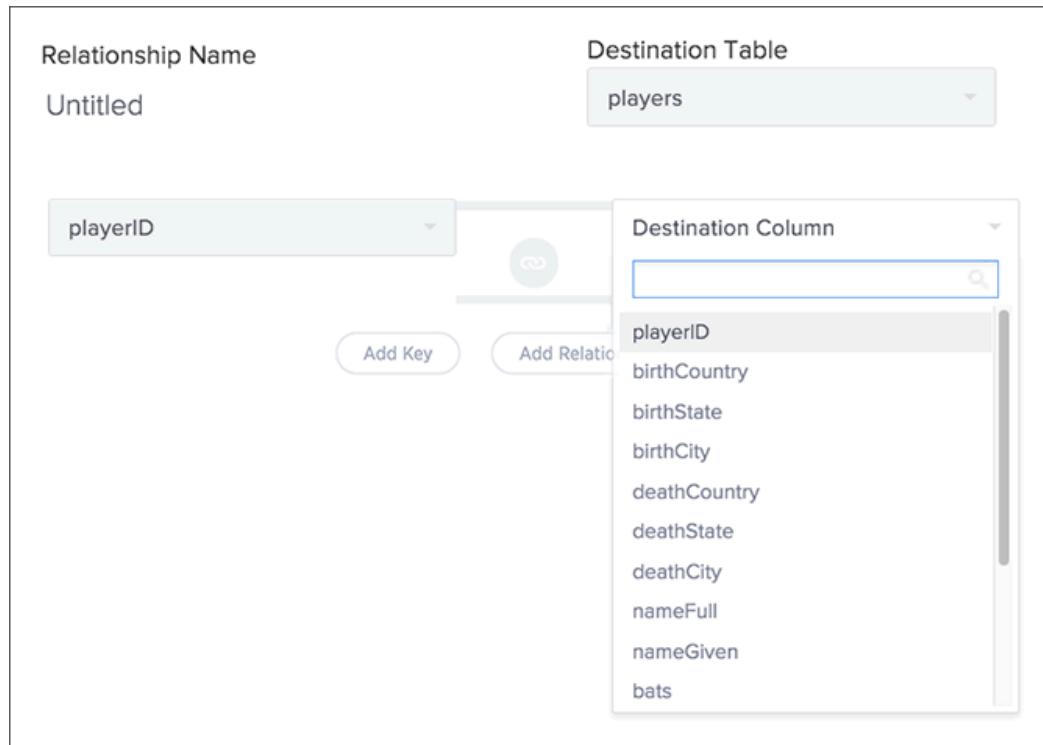
4. If there are already some existing relationships, scroll down and click Add Relationship. Otherwise, continue to the next step.
5. Click on Source Column and select the column you want to link in the source table.

The screenshot shows the ThoughtSpot interface with the 'Relationships' tab selected. The worksheet title is 'Olympic medals data 1896 to 2008'. The 'Source column' dropdown is set to 'Int Olympic Committee code'. The 'Destination Table' dropdown shows a list of tables, with 'cou' selected.

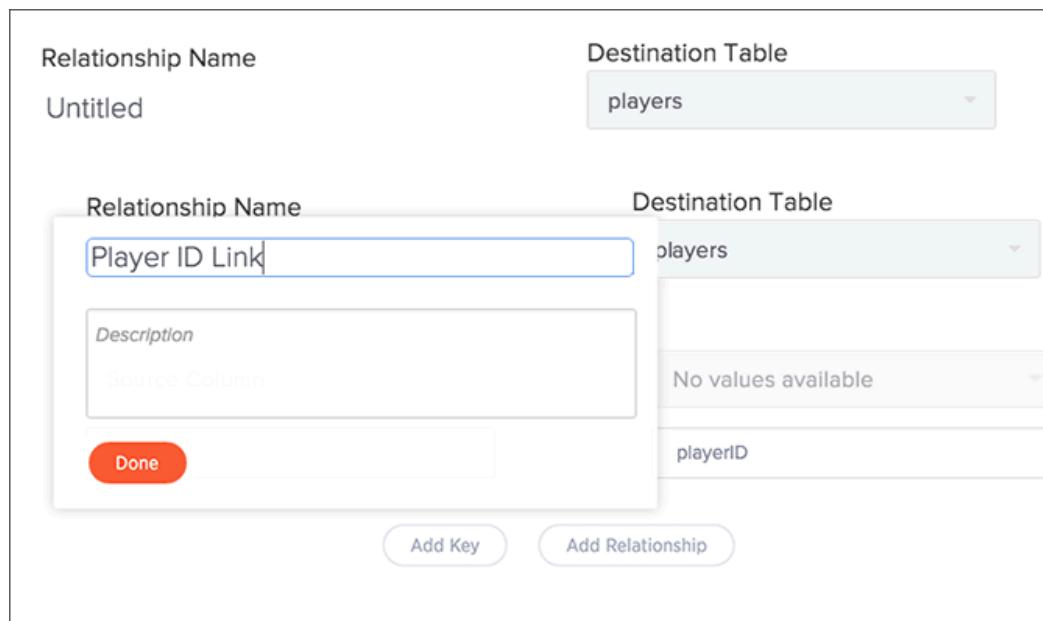
- Under Destination Table find and select the table that you want to link to.

The screenshot shows the ThoughtSpot interface with the 'Relationships' tab selected. The worksheet title is 'Olympic medals data 1896 to 2008'. The 'Source column' dropdown is set to 'Int Olympic Committee code'. The 'Destination Table' dropdown shows a list of tables, with 'cou' selected.

- Click on Destination Column and select the column you want to link to in the destination table.



8. Click Add Key to add the link.
9. Name your relationship and optionally give it a description.



10. Click Add Relationship.
11. Repeat these steps for creating a link until all the links you want to make for your table have been created.

Delete a relationship

Summary: You can delete relationship (link) between tables through the application or TQL.

You must have either administration privilege or modify access permission to the columns to delete a relationship. If you created a relationship (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 FOREIGN KEY...` statement. To delete a relationship from the Web interface:

1. Click on the DATA icon on the top navigation bar and then on Worksheets.

The screenshot shows the ThoughtSpot web interface. At the top, there is a navigation bar with tabs: SEARCH, ANSWERS, PINBOARDS, DATA (which is highlighted with a red box), and ADMIN. Below the navigation bar, there are two main tabs: Tables and Data Sources. Under the Tables tab, there are filters: All, Yours, All types, Worksheets (which is highlighted with a red box), Tables, and a search bar. Below the filters are three buttons: Delete, Share, and Apply Sticker. The main area displays a table with columns: NAME, DESCRIPTION, and SOURCE. A single row is visible, representing a 'Fruit Worksheet'. The entire interface has a clean, modern design with a dark header and light body.

2. Click on the name of the data source you from which you want to remove the relationship.
3. Select Relationships.

The screenshot shows the ThoughtSpot web interface focusing on relationships. At the top, there is a navigation bar with tabs: SEARCH, ANSWERS, PINBOARDS, DATA, and ADMIN. Below the navigation bar, there are two main tabs: Tables and Data Sources. Under the Tables tab, there are filters: All, Yours, All types, Worksheets, Tables, and a search bar. Below the filters are buttons for Stickers and Actions. The main area displays a table with columns: COLUMN NAME, DESCRIPTION, DATA TYPE, COLUMN TYPE, ADDITIVE, AGGREGATION, and HIDE. Several rows are listed, each representing a column with its properties. The 'Relationships' tab is highlighted with a red box. The interface is designed for data management and visualization.

4. Find the relationship you want to delete, and click Delete.

Relationship Name	Untitled
Destination Table	tickets
SOURCE COLUMN	
Owner	
DESTINATION COLUMN	
Owner	
Delete	Update
<hr/>	
Add Relationship	

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:

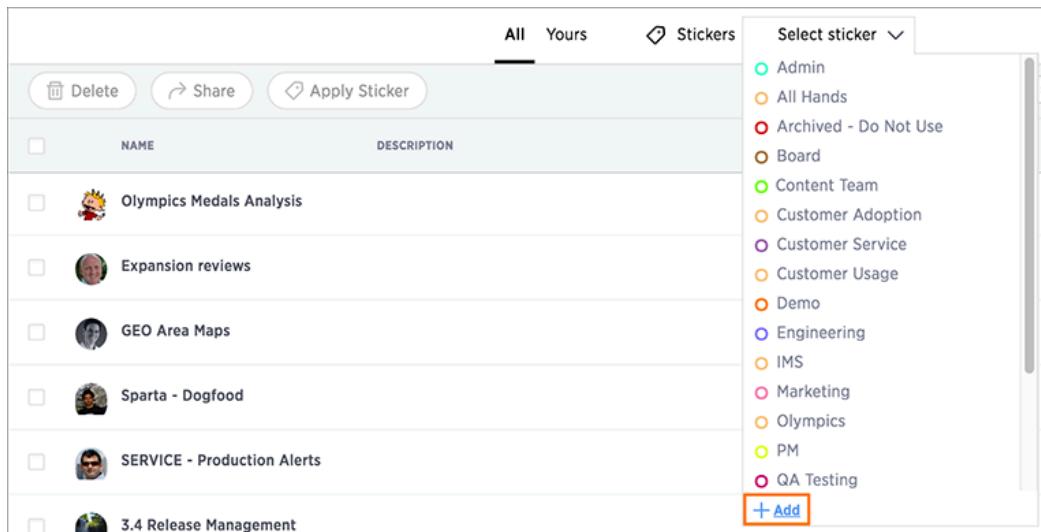
- Only administrators can create stickers.
- Anyone can apply a sticker.
- Anyone can filter by a sticker.

Create a sticker

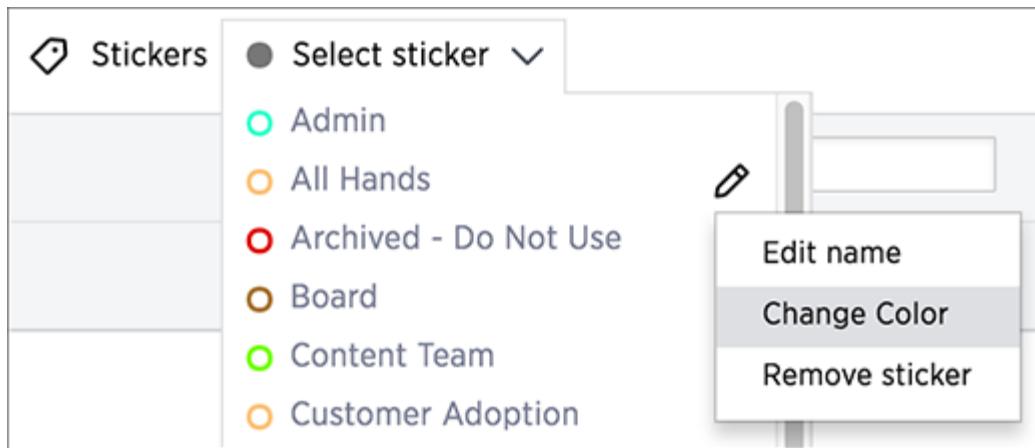
Only administrator users can create stickers. Anyone can apply the stickers you create, or use them as filters when selecting from a list of sources or pinboards.

To create a sticker:

1. Navigate to the **Manage Data** or **Pinboards** screen using the icons in the top navigation bar.
2. Choose **Select 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 or color of a sticker by clicking the edit icon next to its name.



Apply a sticker

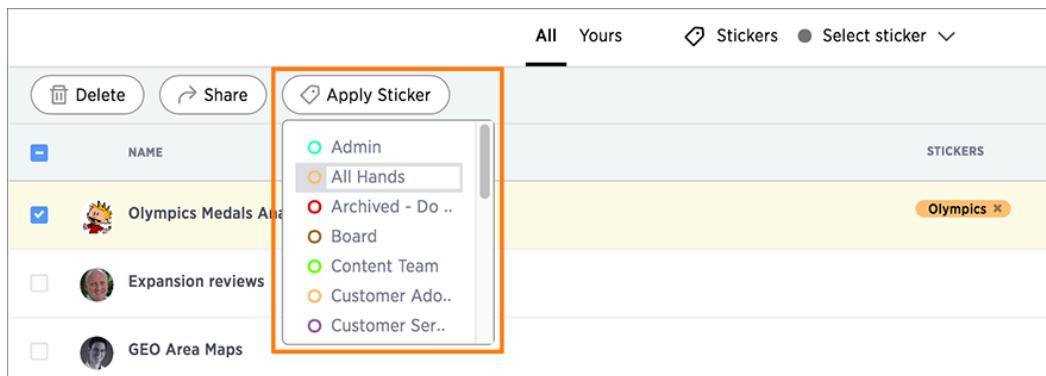
Only administrators create stickers, but anyone with edit privileges can tag an object with a sticker.

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

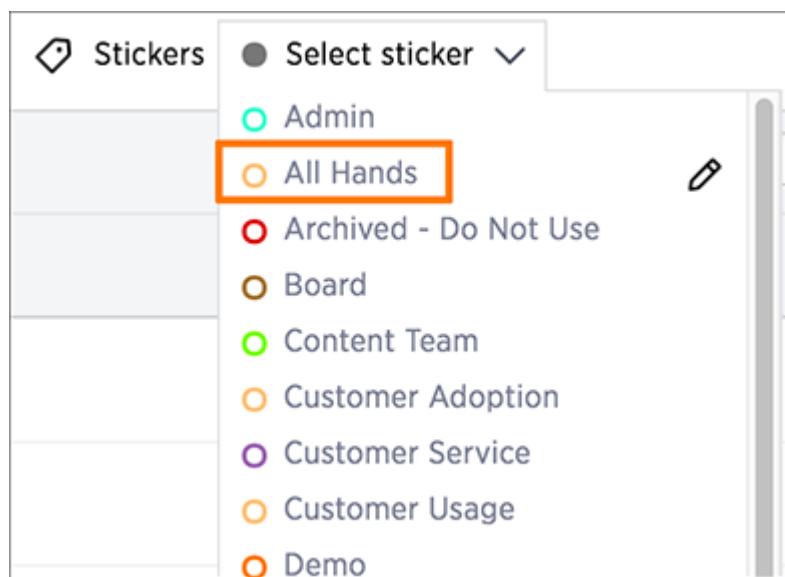
Whenever you are selecting objects from a list, you can filter by sticker to find what you're looking for. Anyone can use stickers to filter lists of pinboards or data sources. You can also filter by sticker when selecting data sources.

To filter by sticker:

1. From the top menu, choose Answers, Pinboards, or Data.



2. Click on Select sticker, and select a sticker to filter by. Click on its name.



Simplify search with worksheets

Summary: Worksheets are flat tables created by joining columns from a set of one or more tables or imported datasets.

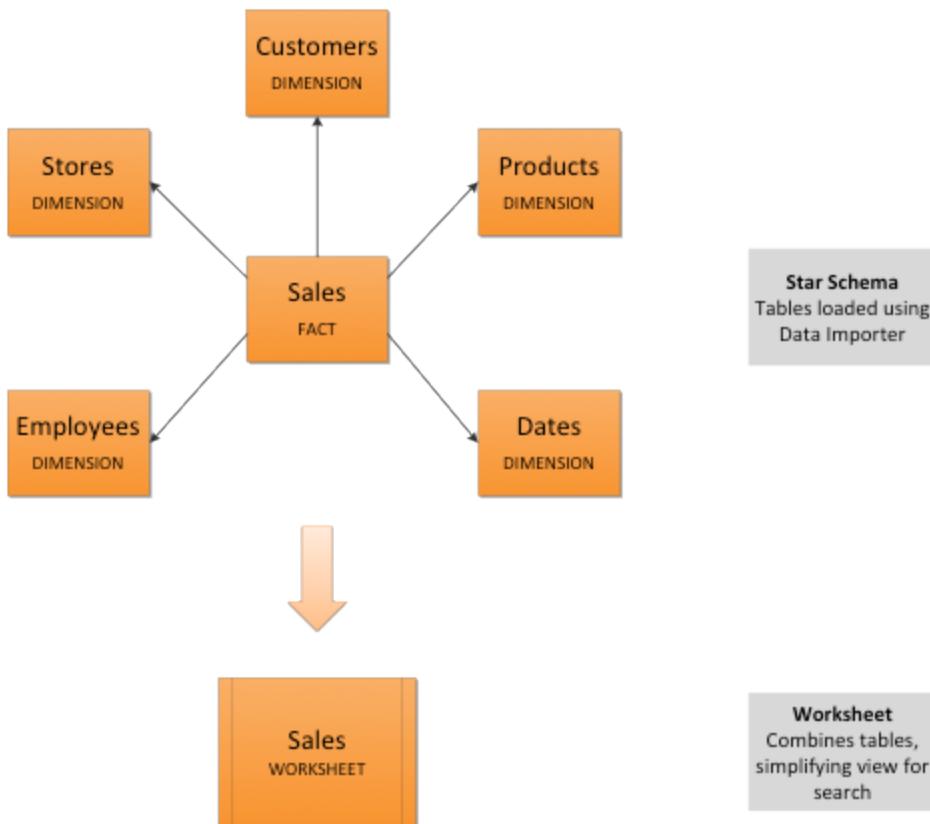
After modeling the data, create worksheets to make searching easier. For example, a sales executive might need to search for information about retail sales. The required data could be contained in several tables (sales, customers, products, stores, etc.), with foreign key relationships between them. 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.

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 [inclusion rule](#) to apply.
5. Choose the [worksheet join rule](#).
6. Select the columns to include.
7. Optionally [create formulas](#).
8. Save the worksheet.
9. [Share the worksheet with groups or users](#).

An alternative way to create a worksheet is to do a search and save it as a worksheet. See the [ThoughtSpot User Guide](#) for details on how to do this.

Create a worksheet

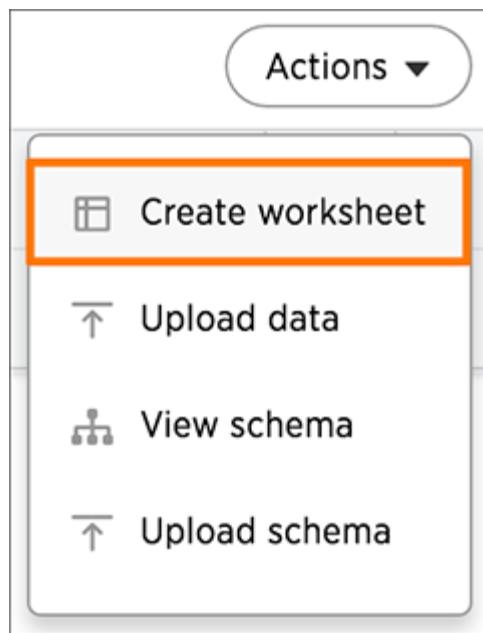
Create a worksheet to make the data easy for users to search. This process includes adding a new worksheet, after which you will choose the data sources to include in it.

To create a new worksheet:

1. Click on DATA, on the top navigation bar.



2. Click the Actions icon from the upper right side of the screen, 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 Choose Sources link.

The image shows two main sections. On the left, under the heading "Data", there is a "Choose Sources" button highlighted with an orange border. Below it is a "Search Columns" input field containing the text "olympicathletes-subset". At the bottom of this section is a "+ Add Columns" button. On the right, under the heading "US Olympic Data", there are two buttons: "Delete" and "+ Add prefix". Below these are several checkboxes for selecting columns: "Age" (checked), "Athlete Name" (checked), "Bronze Medals" (unchecked), and "Country" (unchecked).

Column	Status
Age	Selected
Athlete Name	Selected
Bronze Medals	Not Selected
Country	Not Selected

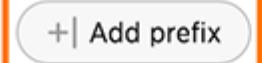
2. Check the box next to each of the sources you want to include in the worksheet.

Note that the list of sources only shows the tables on which you have view privileges.

The screenshot shows the 'Sources' interface for the 'MLB' category. On the left, there's a sidebar with a list of categories: All, Selected (4), and a list of various departments like All Hands, Board, Call Logs, Customer Usage, Engineering, Marketing, and MLB. Under MLB, there are sub-categories: PM, Recruiting, Sales, SFDC, Support, TPCH, Training, and TS Stats. At the bottom of the sidebar is a button labeled 'Explore all data'. The main area shows a grid of tables with checkboxes. The checked tables are: Batting, Fielding, Pitching, and Players. Other tables shown include Awardsplayers, Awardssharemanagers, Awardsshareplayers, Fieldingof, Fieldingpost, Halloffame, Managers, Managershalf, Pitchingpost, Schools, Schoolsplayers, Salaries, Seriespost, Teams, Teamsfranchises, and Teamshalf. There are also buttons for 'Choose the inclusion rule' (with 'Exclude empty rows' selected) and 'Choose the worksheet join rule' (with 'Apply joins progressively (recommended for most cases)' selected).

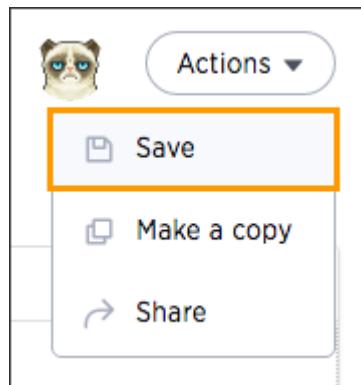
3. If you want to see what the data inside the sources looks like, click **Explore all data**.
4. Choose the [inclusion rule](#).
5. Choose the [worksheet join rule](#).
6. Click **Done** to save your changes.
7. Expand the table names under **Columns** to select the columns to add to the worksheet.
 - 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, **Ctrl+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. Click on the worksheet title to name it, and then **Save** it.
9. 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.
10. 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.

US Olympic Data

olympicathletes-subset			
<input checked="" type="checkbox"/> Age	26	30	
<input checked="" type="checkbox"/> Athlete Name	a. j. mleczko	aaror	
<input type="checkbox"/> Bronze Medals	0	0	
<input type="checkbox"/> Country	united states	unite	
<input type="checkbox"/> Gold Medals	0	0	

11. Click Actions and select Save.



12. [Share your worksheet](#), if you want other people to be able to use it.

Where to go next

- [How the inclusion rule works](#)
Use the inclusion rule to specify which data to include in a worksheet where two or more tables are joined. If you are familiar with SQL, you might think of it as a JOIN condition.
- [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.

How the inclusion rule works

Summary: Use the inclusion rule to specify which data to include in a worksheet where two or more tables are joined. If you are familiar with SQL, you might think of it as a JOIN condition.

In the case where some of the rows in a fact table do not have a match in any dimension tables, the inclusion rule determines whether or not the rows are shown. Because of this, the inclusion rule can affect the number of rows the worksheet will contain. The inclusion rule is important if some table values are empty (or NULL) or if some primary key column values in a fact table do not have a match to a foreign key in the dimension table. In these cases, how you set an inclusion rule affects the number of rows in the worksheet.

Only rows in the fact table (also known as the LEFT table) are affected by the inclusion rule. It works like this:

If you choose	Then...
Apply full outer join (Full Outer Join)	The results of both the left and right outer joins are combined, and all matched or unmatched rows from the tables on both sides are shown.
Apply left outer join (Left Outer Join)	All possible rows in the fact table are shown, regardless of whether they have a match in the dimension tables.
Apply right outer join (Right Outer Join)	All possible rows in the second table are shown, regardless of whether they have a match in the dimension tables.
Exclude empty rows (Inner Join)	Any rows that do not have a match in one of the dimension tables, won't be shown in search results.

When using **Exclude empty rows (Inner Join)**, the number of rows in the resulting worksheet can differ from the number of rows in the table when accessing it directly, because of the join condition. The worksheet acts like a materialized view. This means that it contains the results of a defined query in the form of a table.

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](#).

The answer returned when searching using a worksheet as the source can be different from the answer you get when using the table directly as a source. When using a worksheet as a source, even if you were to select fields that come from only one table in your search, any underlying joins to other tables will still be active. When using the table directly as the source, you will see every value.

This is best understood through an example.

A typical sales fact table contains a column with the employee ID of the person who made the sale. The employee ID column has a foreign key in the employee dimension table. This is the relationship used to join the two tables.

Sometimes a sale has been made directly or through a reseller, without involving a sales person. In this case, the employee ID value for the sale will be empty in the fact table. If you wanted the worksheet to include all sales, regardless of whether or not they were associated with a sales person, you would choose **Include all rows (Left Outer Join)**. If you only want the worksheet to contain sales made by employees, you would choose **Exclude empty rows (Inner Join)**.

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.

Joins and aggregated worksheets

You have the ability to join an aggregated worksheet with a table. You can join an aggregated worksheet with a system table by creating a relationship. This means aggregated worksheets behave similar to tables, and they can be used in the same way as a table, excluding any TQL manipulation. You also have the capability to create a worksheet on top of an aggregated worksheet. So aggregated worksheets can be included as tables in regular worksheets.

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 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:

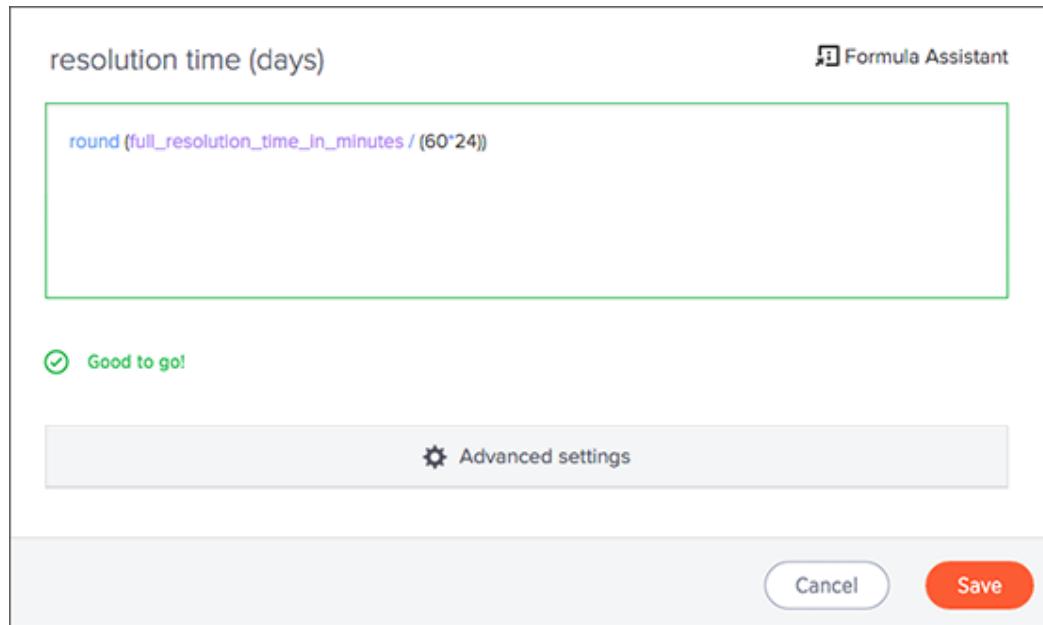
1. Create a new worksheet, or edit an existing one.
2. Click the + button next to **Formulas**.

The screenshot shows the ThoughtSpot Formula Builder interface. On the left, there's a sidebar titled "Data" with a "Choose Sources" button and a "Search Columns" input field. Below it, under "fruit_for_help", are columns: Date, Fruit, Location, Price per fruit (\$), Quantity sold, Total sale, and Vendor. There's also a "+ Add Columns" button. In the center, there's a search bar with "fruit for help" and "Delete" and "+| Add prefix" buttons. To the right, a table lists formulas and their results:

Formulas	
<input type="checkbox"/> average quantity sold	N/A
<input type="checkbox"/> fruit_for_help	
<input type="checkbox"/> Date	05/13/FY 2013
<input type="checkbox"/> Fruit	apples
<input type="checkbox"/> Location	the bronx
<input type="checkbox"/> Price per fruit (\$)	3.00
<input type="checkbox"/> Quantity sold	11
<input type="checkbox"/> Total sale	16.50
<input type="checkbox"/> Vendor	ray ratliff

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

The screenshot shows the 'resolution time (days)' formula in the main workspace. The formula is defined as `round(full_resolution_time_in_minutes / (60*24))`. Below the formula, a green checkmark indicates it is 'Good to go'. An 'Advanced settings' dialog is open, showing the following configuration:

- Data type:** Numeric
- Measure or attribute:** MEASURE
- Aggregation:** TOTAL (with options: AVG, MAX, MIN, STD DEVIATION, TOTAL, TOTAL COUNT, UNIQUE COUNT, VARIANCE)
- Value:** false
- Label:** 8
- Unit:** <=120min
- Label Unit:** <=120min

A red 'Save' button is visible on the right side of the dialog.

5. You can see a list of formula operators with examples by clicking on Formula Assistant.

The screenshot shows the same formula and settings as above, but the 'Formula Assistant' sidebar is now expanded. The sidebar lists various operators under the 'Number' category, including basic arithmetic operators (+, -, *, /), mathematical functions (abs, acos, asin, atan, atan2, cbrt, ceil, cos, cube), and aggregate functions (Aggregate, Date, Mixed). The division operator (/) is highlighted, with its description: 'Returns the result of dividing the first number by the second.' and examples: '6 / 3 = 2' and 'markup / retail price'.

6. Name the formula by clicking on its title and typing the new name. Click Save.

Edit or rename worksheet

Summary: A worksheet can be edited by anyone with the proper permissions.

You can always go into a worksheet and make changes such as adding sources and columns, adding or editing formulas, and changing column names. You can also rename a worksheet.

Edit a worksheet

To edit a worksheet:

1. Click on the DATA icon on the top navigation bar and then on Worksheets.

The screenshot shows the ThoughtSpot navigation bar with 'SEARCH', 'ANSWERS', 'PINBOARDS', 'DATA' (which is highlighted with an orange box), and 'ADMIN'. Below the navigation bar, there are tabs for 'Tables' and 'Data Sources'. Under the 'DATA' tab, there are filters for 'All', 'Yours', 'All types', and 'Worksheets' (which is highlighted with an orange box). There are also buttons for 'Delete', 'Share', and 'Apply Sticker'. A table header row shows columns for 'NAME', 'DESCRIPTION', and 'SOURCE'. Below the header, a single row is visible for 'Fruit Worksheet'.

2. Click on the name of the worksheet you want to edit from the list.
3. Click the Edit button in the upper right hand side of the screen.
4. Make your changes to the worksheet.
5. Click Actions and select Save.

The screenshot shows a 'Actions' dropdown menu with three options: 'Save' (highlighted with an orange box), 'Make a copy', and 'Share'.

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. On the right hand side, click the current name, and enter a new name.

A screenshot of the ThoughtSpot interface. On the left, there is a search bar containing the text "olympic". Below it is a table with columns "NAME" and "DESCRIPTION". A row is selected, showing "US Olympic Data" in the NAME column. On the right, a modal window titled "WORKSHEET" shows the current name "US Olympic Data" in a text input field, which is outlined in orange. Below the input field are buttons for "Columns" and "Columns". At the bottom of the modal is a table with columns "COLUMN NAME", "DESCRIPTION", and "DATA".

You can also edit column names and other details in the same way.

4. Click Done and Save.

Related information

[Change the inclusion or join rule for a worksheet](#)

Change inclusion, join, or RLS for a worksheet

Summary: As long as you have permissions to edit a worksheet, you can always go into it and set a different inclusion rule or join 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 [inclusion rule](#) for the worksheet.

if you have the **Can Administer RLS** privilege, or belong to a group who does, you can create a worksheet that bypasses the RLS rules underlying a worksheet's data. When **by pass** is set the underlying RLS allows users, that would not otherwise be unable, to see an aggregate/market view of a tables data.

Change/configure a worksheet

Before working through this procedure, make sure you are familiar with how the following affect data:

- [inclusion rule](#)
- [worksheet join rule](#)
- [role-level security \(RLS\)](#)

To configure these values for a worksheet:

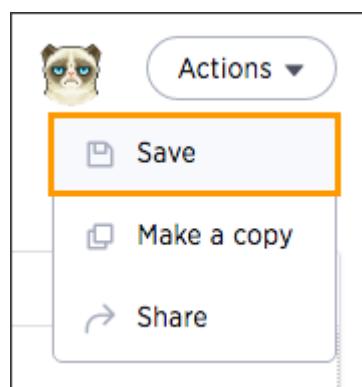
1. Click on the DATA icon on the top navigation bar and then on Worksheets.

The screenshot shows the ThoughtSpot navigation bar with the 'DATA' tab highlighted. Below the navigation bar, there are two tabs: 'Tables' and 'Data Sources'. Under the 'Tables' tab, there are filters for 'All', 'Yours', 'All types', and 'Worksheets'. The 'Worksheets' tab is highlighted with a red box. Below these filters, there are buttons for 'Delete', 'Share', and 'Apply Sticker'. A table follows, with columns for 'NAME', 'DESCRIPTION', and 'SOURCE'. A row for 'Fruit Worksheet' is visible, with a checkbox in the first column and a small thumbnail in the second column.

2. Click on the name of the worksheet you want to edit from the list.
3. Click the **Edit** button in the upper right hand side of the screen.
4. Click on the **Choose Sources** link.
5. Scroll to the bottom of the dialog.
6. Configure the inclusion rule, worksheet join rule, and RLS as needed.

The screenshot shows the 'Sources' configuration screen for a worksheet named 'Wootric Scores with Releases'. On the left, there's a sidebar with 'Data' and 'Formulas' sections. The 'Sources' section is open, showing a list of available sources under 'All' and 'Selected (1)'. The selected source is '4.4.1'. Other sources listed include 'Admin', 'All Hands', 'Alonzo Test', 'Archived -..', 'Board', 'chasm attr..', 'CMO Webi..', 'Connectio..', 'Content Te..', and 'Customer'. There are also several system tables like 'WM_CAL_FLAGS', 'wootric_responses', 'wootric_users', and 'zen_jira_mapping'. A 'Choose the inclusion rule:' section contains radio buttons for 'Apply full outer join (Full outer join)', 'Apply left outer join (Left outer join)', 'Apply right outer join (Right outer join)', and 'Exclude empty rows (Inner join)', with the last one selected. Another section for 'Choose the worksheet join rule:' has radio buttons for 'Apply joins progressively (recommended for most cases)' (selected) and 'Apply all joins'. A checkbox for 'Disable row level security for worksheet' is also present. A 'Done' button is at the top right.

7. Click Done.
8. Click Actions and select Save.



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.

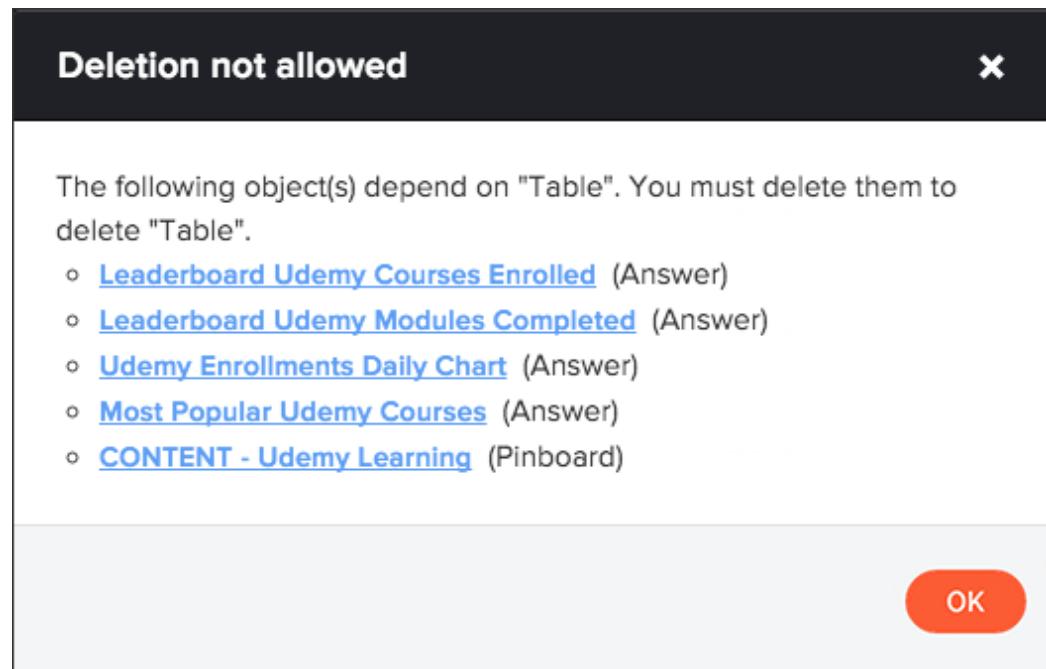


2. Find the worksheet or table you want to remove in the list, and check the box next to its name.
3. Click the Delete icon.

A screenshot of the 'Tables' page in ThoughtSpot. The top navigation bar shows 'Tables' is selected. Below the navigation are filters: 'All' and 'Yours' under 'All types', and 'Worksheets' and 'Tables' under 'Tables'. There are also 'Share' and 'Apply Sticker' buttons. The main area shows a list of tables with columns for 'NAME', 'DESCRIPTION', and 'SOURCE'. Two tables have checkboxes next to their names: 'Udemy_user_progr..' and 'Udemy Worksheet'. The third table, 'Udemy_user_activi..', has a checked checkbox and is highlighted with a yellow background. The fourth table, 'Udemy user cours..', also has a checked checkbox. The 'Delete' button, which is highlighted with an orange border, is located at the top left of the table list area. The table names are: Udemy_user_progr.., Udemy user cours.., Udemy_user_activi.., and Udemy Worksheet.

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.

		Tables Data Sources																									
All	Yours	All types	Worksheets	Tables	Stickers	Select sticker ▾																					
<input type="text" value="udemy"/> <input type="checkbox"/> NAME DESCRIPTION		WORKSHEET > Udemy Worksheet Columns Data Relationships Dependents Row security																									
<input type="checkbox"/> Udemy_user_pr..		<table border="1"> <thead> <tr> <th>COLUMN NAME</th> <th>DEPENDENT NAME</th> <th>TYPE</th> </tr> </thead> <tbody> <tr> <td>date enrolled</td> <td>CONTENT - Udemy..</td> <td>Pinboard</td> </tr> <tr> <td>email</td> <td>CONTENT - Udemy..</td> <td>Pinboard</td> </tr> <tr> <td>number of course..</td> <td>CONTENT - Udemy..</td> <td>Pinboard</td> </tr> <tr> <td>number of modul..</td> <td>CONTENT - Udemy..</td> <td>Pinboard</td> </tr> <tr> <td>last name</td> <td>CONTENT - Udemy..</td> <td>Pinboard</td> </tr> <tr> <td>first name</td> <td>CONTENT - Udemy..</td> <td>Pinboard</td> </tr> </tbody> </table>					COLUMN NAME	DEPENDENT NAME	TYPE	date enrolled	CONTENT - Udemy..	Pinboard	email	CONTENT - Udemy..	Pinboard	number of course..	CONTENT - Udemy..	Pinboard	number of modul..	CONTENT - Udemy..	Pinboard	last name	CONTENT - Udemy..	Pinboard	first name	CONTENT - Udemy..	Pinboard
COLUMN NAME	DEPENDENT NAME	TYPE																									
date enrolled	CONTENT - Udemy..	Pinboard																									
email	CONTENT - Udemy..	Pinboard																									
number of course..	CONTENT - Udemy..	Pinboard																									
number of modul..	CONTENT - Udemy..	Pinboard																									
last name	CONTENT - Udemy..	Pinboard																									
first name	CONTENT - Udemy..	Pinboard																									
<input type="checkbox"/> Udemy_user_co..																											
<input type="checkbox"/> Udemy_user_act..																											
<input type="checkbox"/> Udemy Workshe..																											

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
Has administration privileges	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.
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

Privilege	Description
	ThoughtSpot Data Connect, if it is enabled on your cluster.
Can schedule pinboards	Can create pinboard schedules and edit their own scheduled jobs.
Has Spot IQ privilege	User can use SpotIQ's auto analyze function.
Can Administer RLS	Users in groups with this privilege (directly or indirectly) can bypass row-level security (RLS) rules. This privilege can only be assigned by a user who already Has administration privilege . Your installation configuration may enable or disable this feature. By default, it is enabled.

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 Agg WS	Modify Col. Props. ¹	Upload Data	Download Data	Share within Group	Share with All	RLS rules	CrUD Relationships	Read Relationships	See Hidden Cols	Join with Upload Data	Schema Viewer	Use Data Connect	Use Scheduler	Use Auto-Analyze
Admin	Y	Y	Y	Y	Y	Y	Y	Y ²	Y	Y	Y	Y	Y	Y	Y	Y
Can Upload Data	N	N	N	Y	N	Y	N	Y ³	Y ⁴	N	N	N	N	N	N	N
Can Download Data	N	N	N	N	Y	Y	N	N	N	N	N	N	N	N	N	N
Data Management	Y	Y	Y	Y	N	Y	N	Y ⁴	Y ⁴	Y ⁵	Y	N	Y	N	N	N
Can Auto-Analyze	N	N	N	N	N	N	N	N	Y ⁴	N	N	N	N	N	N	Y
Can Schedule	N	N	N	N	N	N	N	N	Y ⁴	N	N	N	N	N	N	N
Can Share with All	N	N	N	N	N	Y	Y	N	N	N	N	N	Y	N	N	N
None	N	N	N	N	Y	N	N	N	Y ⁴	N	N	N	N	N	N	N

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:

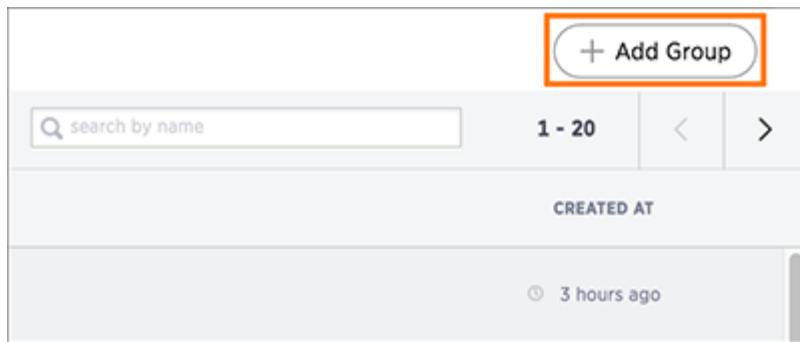
1. Log into ThoughtSpot from a browser.
2. Click on the Admin icon, on the top navigation bar.



3. In the Admin panel, click on User Management and Groups.

A screenshot of the ThoughtSpot Admin interface under the User Management and Groups section. The top navigation bar shows the Admin icon is still active. Below it, there are three tabs: User Management (which is selected and highlighted in orange), Data Management, and System Health. Under the User Management tab, there are two sub-tabs: Users and Groups, with Groups being the active one and also highlighted in orange. The main content area displays a list of groups. Each group entry includes a checkbox, a display name, and a corresponding privilege level. The first group is 'Administration Group' with the privilege 'Administrator'. The second group is 'All Hands' with the privilege 'All Hands'. A large 'Delete' button is visible at the top of the group list.

4. Click the + Add Group button on the upper right hand side of the list of groups.



- Enter the details for the new group:

Add a new group

Group name *	Creators	Manage Groups Manage Users No Groups in Group
Display name *	Creators	
Sharing visibility *	SHARABLE	
Description		
Privileges	<input type="checkbox"/> Has administration privileges <input checked="" type="checkbox"/> Can upload user data <input checked="" type="checkbox"/> Can download data <input type="checkbox"/> Can share with all users <input type="checkbox"/> Can manage data <input type="checkbox"/> Can schedule pinboards <input type="checkbox"/> Has Spot IQ privilege <input type="checkbox"/> Can Administer RLS	
<small>* Required field</small>		
		<small>Cancel</small> <small>ADD</small>

Field	Description
Group name	Enter a unique name for the group.
Display name	
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 privi-

lege 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.

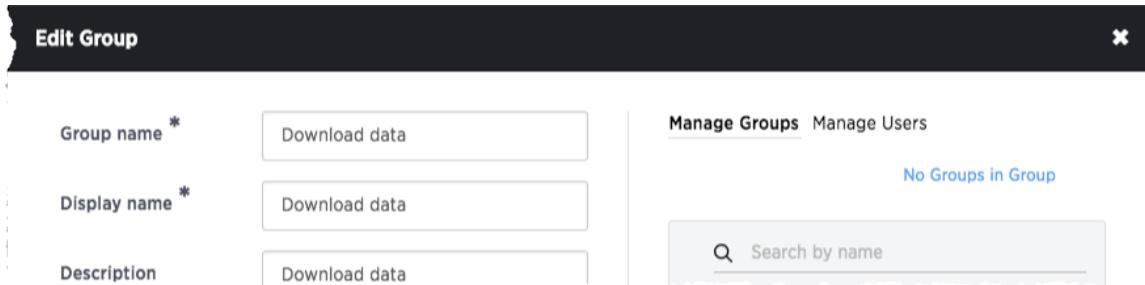
7. Click the Manage Users tab if you want to add users.

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**.

<input type="checkbox"/> DISPLAY NAME	NAME
<input type="checkbox"/> Administration Group	Administrator
<input type="checkbox"/> All Hands	All Hands

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.

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.

<input type="checkbox"/> DISPLAY NAME	NAME
<input type="checkbox"/> Abhay Bothra	abhay
<input type="checkbox"/> Abhishek	abhishek

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.

User Management Data Management System Health Style Customization		
Users	Groups	+ Add User
Add Users to Groups	Delete	
DISPLAY NAME	NAME	CREATED AT
<input checked="" type="checkbox"/>  test2	test2@thoughtspot.int	<input type="radio"/> 6 months ago
<input checked="" type="checkbox"/>  Testing	testing@thoughtspot.int	<input type="radio"/> 7 months ago
<input checked="" type="checkbox"/>  test-user	test user	<input type="radio"/> 1 month ago

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.

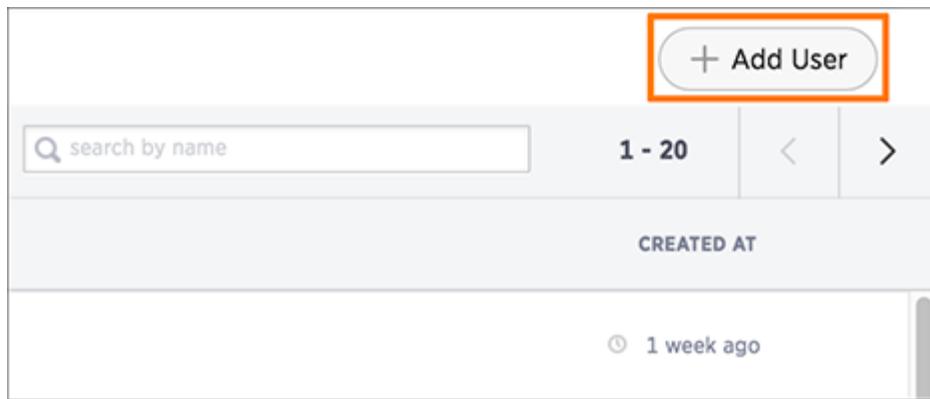
1. Log into ThoughtSpot from a browser.
2. Click on the Admin icon, on the top navigation bar.



3. In the Admin panel, click on User Management and Users.

A screenshot of the ThoughtSpot Admin panel under the User Management section. The top navigation bar shows the Admin icon is still selected. Below it, the 'User Management' tab is highlighted with a red border. Underneath, there are two tabs: 'Users' (which is also highlighted with a red border) and 'Groups'. A toolbar at the top includes buttons for 'Add Users to Groups' and 'Delete'. The main area displays a table with two rows of user data. The columns are labeled 'DISPLAY NAME' and 'NAME'. The first row shows a user named 'Abhay Bothra' with the display name 'AB' and the name 'abhay'. The second row shows a user named 'Abhishek' with the display name 'Abhishek' and the name 'abhishek'. Each row has a small profile picture and a checkbox to its left.

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:

Add a new user

Username *	<input type="text"/>	Manage Groups No Group assigned to User <input type="text"/> Search by name <input type="button"/> Clear all <input type="button"/> Select all <ul style="list-style-type: none"> <input type="checkbox"/> Administrator <input type="checkbox"/> Analyst <input type="checkbox"/> Consumer <input type="checkbox"/> DataDownloader <input type="checkbox"/> ShareWithAll <input type="checkbox"/> UserDataUploader
Display name *	<input type="text"/>	
Sharing visibility *	SHARABLE	
Change password *	<input type="text"/>	
Confirm password *	<input type="text"/>	
Email	<input type="text"/>	

* Required field

Cancel ADD

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 username you specify has to be domain qualified (e.g. <code>username@ldap.thoughtspot.com</code>), 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.

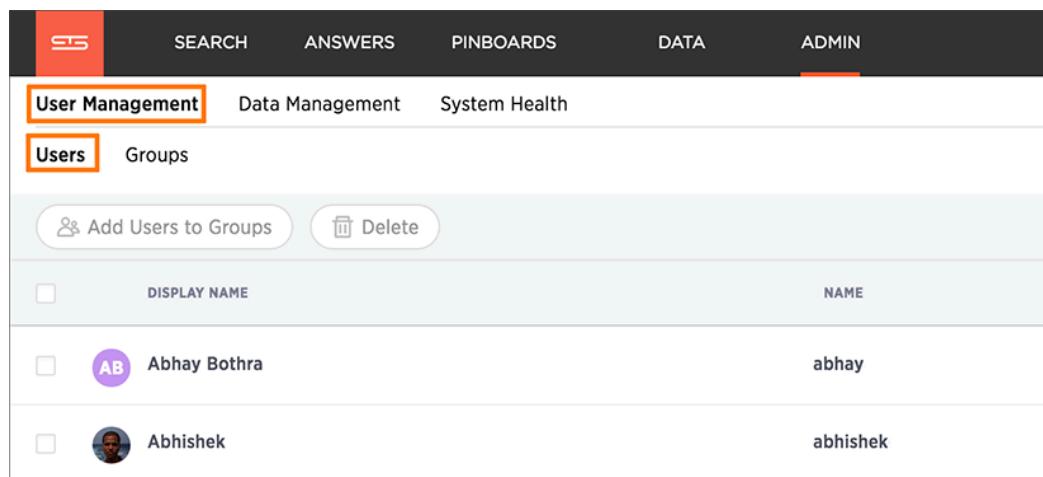
As an administrator, you can edit a user and change the groups the user belongs to. You can also edit a user to reset a user's password by entering and confirming the new password. This is useful if a user has forgotten their password, or to effectively disable an account.

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.



The screenshot shows the ThoughtSpot Admin interface with the 'User Management' tab selected. Below it, the 'Users' tab is also highlighted. There are two buttons at the top: 'Add Users to Groups' and 'Delete'. A table lists two users: 'Abhay Bothra' and 'Abhishek'. Each row has a checkbox for selecting the user and an edit icon (pencil) for editing the name.

	DISPLAY NAME	NAME
<input type="checkbox"/>  AB	Abhay Bothra	abhay
<input type="checkbox"/>  Abhishek		abhishek

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.

[CSV](#)[PDF](#)

Table visualizations have all data rows that they're supposed to have.

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 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.

Table visualizations include only the first 100 rows.

In the case of a corrupted pinboard: the PDF attachment has empty/error screenshots.

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.

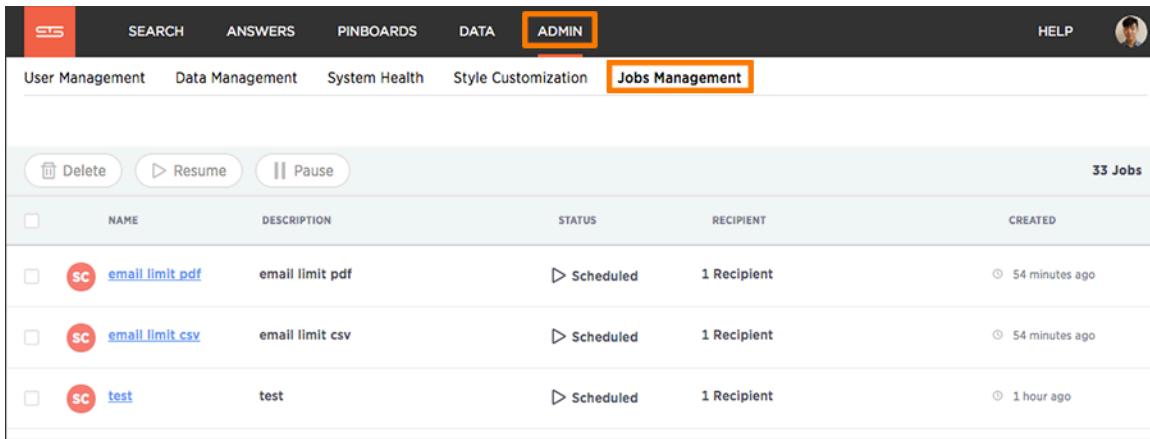
Related information

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.



The screenshot shows the ThoughtSpot Admin interface. The top navigation bar has tabs for SEARCH, ANSWERS, PINBOARDS, DATA, ADMIN (which is highlighted with a red box), and HELP. Below the navigation is a sub-navigation bar with links for User Management, Data Management, System Health, Style Customization, and Jobs Management (which is also highlighted with a red box). The main content area is titled "Jobs Management". It features a toolbar with buttons for Delete, Resume, and Pause. A table lists three scheduled jobs. The columns are: NAME, DESCRIPTION, STATUS, RECIPIENT, and CREATED. The first job is named "email limit pdf" with description "email limit pdf", status "Scheduled", 1 recipient, and created 54 minutes ago. The second job is named "email limit csv" with description "email limit csv", status "Scheduled", 1 recipient, and created 54 minutes ago. The third job is named "test" with description "test", status "Scheduled", 1 recipient, and created 1 hour ago. Each job row has a checkbox to its left.

NAME	DESCRIPTION	STATUS	RECIPIENT	CREATED
email limit pdf	email limit pdf	Scheduled	1 Recipient	54 minutes ago
email limit csv	email limit csv	Scheduled	1 Recipient	54 minutes ago
test	test	Scheduled	1 Recipient	1 hour ago

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.

User Management	Data Management	Sy	> email limit csv		
			STARTED AT	ENDED AT	STATUS
			6 minutes ago	5 minutes ago	Success
<input type="checkbox"/>		email limit pdf	11 minutes ago	10 minutes ago	Failed
<input type="checkbox"/>		email limit csv	16 minutes ago	15 minutes ago	Success
<input type="checkbox"/>		test	21 minutes ago	20 minutes ago	Success
<input type="checkbox"/>		max_jobs test	26 minutes ago	25 minutes ago	Success
<input type="checkbox"/>		header			
<input type="checkbox"/>		10.14 rls pdf	Job started at 10/14/FY 2017 14:20:00 Scheduled updates generated as expected.		
<input type="checkbox"/>		10.14 rls	Generating updates as stephanie@thoughtspot.int. SUCCESS: Create update for visualization t3 (1) of pinboard big table in format csv. SUCCESS: Create update for visualization CITY, NAME, NATION, PHONE, REGION, SUPPKEY (2) of pinboard big table in format csv. SUCCESS: Create update for visualization t2 (3) of pinboard big table in format csv. SUCCESS: Create update for visualization ADDRESS, CATEGORY, CUSTKEY, MKTSEGMENT (4) of pinboard big table in format csv. SUCCESS: Create update for visualization t1 (5) of ninnhard bin table in format csv.		
<input type="checkbox"/>		delete_creator.pdf			
<input type="checkbox"/>		delete_creator			

User Management	Data Management	Sy	> email limit pdf		
			STARTED AT	ENDED AT	STATUS
			2 minutes ago	N/A	Running
<input type="checkbox"/>		email limit pdf	7 minutes ago	3 minutes ago	Failed
<input type="checkbox"/>		email limit csv	12 minutes ago	11 minutes ago	Failed
<input type="checkbox"/>		test	22 minutes ago	17 minutes ago	Failed
<input type="checkbox"/>		max_jobs test	27 minutes ago	25 minutes ago	Failed
<input type="checkbox"/>		header			
<input type="checkbox"/>		10.14 rls pdf	Job started at 10/14/FY 2017 14:20:00 Error Code: 12700 Incident Id: f1cf72ad-cec6-4017-be26-88becc4f5fb9 Error Message: Error in generating scheduled update. Error Code: 12708 Details: Pdf for pinboard big table could not be generated. Error Code: FOOLSCAP_4017-be26-88becc4f5fb9 Error Message: Foolscap returned partial success. Failing request.		
<input type="checkbox"/>		10.14 rls			
<input type="checkbox"/>		delete_creator.pdf	Generating updates as stephanie@thoughtspot.int. FAILURE: Create update for pinboard big table in format pdf.		
<input type="checkbox"/>		delete_creator	FAILURE: Send scheduled update		

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.
--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.

Parameter	Description
--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 Agg WS	Modify Col. Props. ¹	Upload Data	Download Data	Share within Group	Share with All	RLS rules	CrUD Relationships	Read Relationships	See Hidden Cols	Join with Upload Data	Schema Viewer	Use Data Connect	Use Scheduler	Use Auto-Analyze
Admin	Y	Y	Y	Y	Y	Y	Y	Y ²	Y	Y	Y	Y	Y	Y	Y	Y
Can Upload Data	N	N	N	Y	N	Y	N	Y ³	Y ⁴	N	N	N	N	N	N	N
Can Download Data	N	N	N	N	Y	Y	N	N	N	N	N	N	N	N	N	N
Data Management	Y	Y	Y	Y	N	Y	N	Y ⁴	Y ⁴	Y ⁵	Y	N	Y	N	N	N
Can Auto-Analyze	N	N	N	N	N	N	N	N	Y ⁴	N	N	N	N	N	N	Y
Can Schedule	N	N	N	N	N	N	N	N	Y ⁴	N	N	N	N	N	N	N
Can Share with All	N	N	N	N	N	Y	Y	N	N	N	N	N	Y	N	N	N
None	N	N	N	N	Y	N	N	N	Y ⁴	N	N	N	N	N	N	N

Table notes:

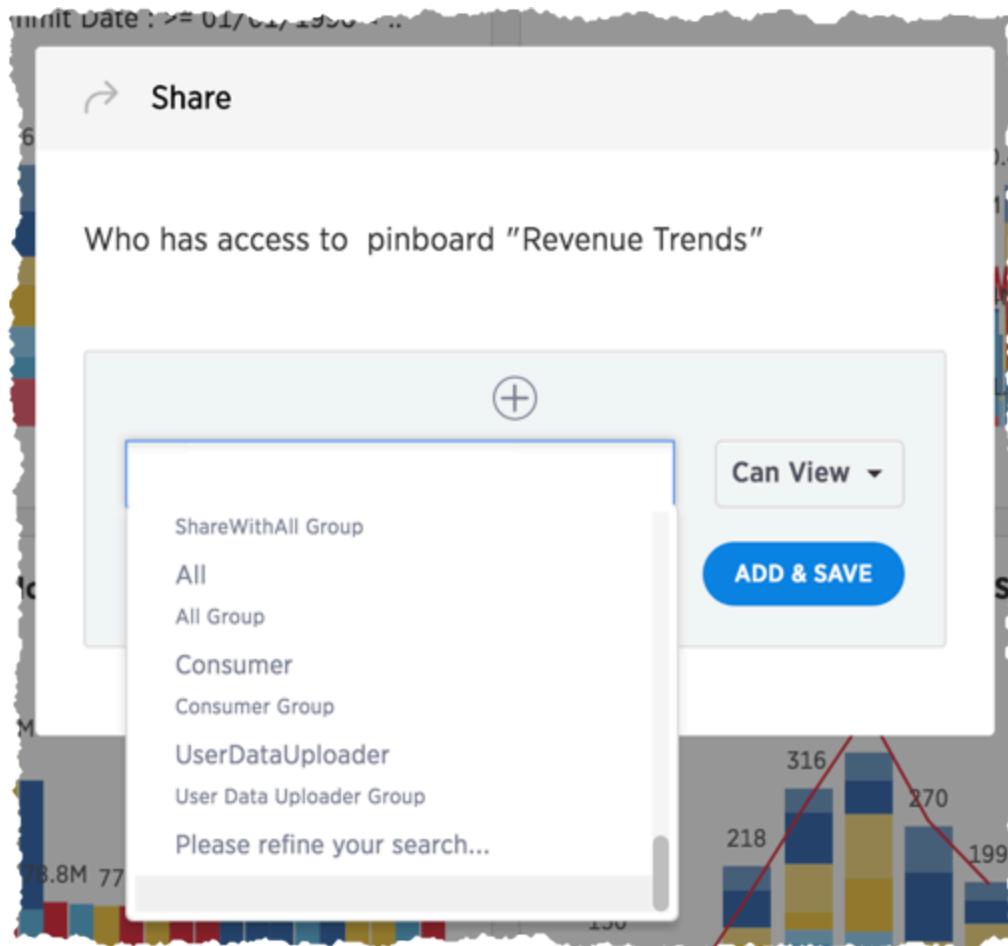
1. Applies to non-owners only.
2. Any tables.
3. Author of at least one table in relationship.
4. Only when read permission for columns used in the relationship.
5. With edit permission.

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
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
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
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
Pinboards	A pinboard of saved search results.	Anyone who can view a pinboard can share it. See Share a pinboard

When you share an object, only the users and groups that have SHAREABLE set for the **Sharing visibility** option appear on the dialog.



User 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.

Only users in the **Administrators** group can share with groups marked as NOT SHAREABLE.

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.

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.

Share a table or imported data by following these steps:

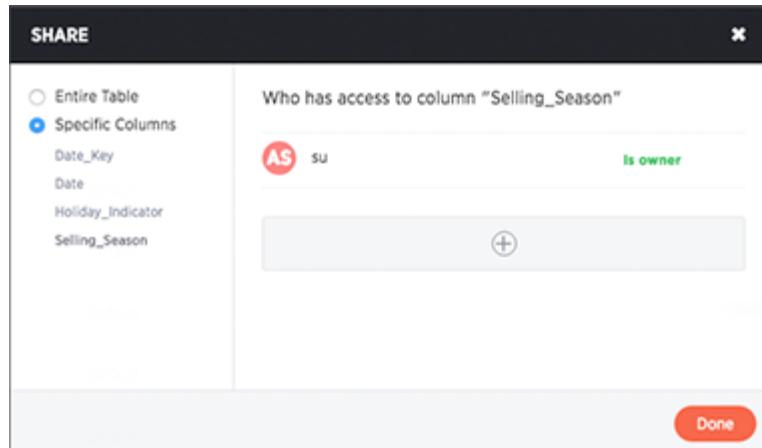
1. Click on the DATA icon in the top navigation bar.
2. Click on Tables.

NAME	DESCRIPTION	SOURCE
fish_seas_oceans		Default

3. Select one or more tables to share, and click the Share icon.

NAME	DESCRIPTION	SOURCE
Fish1 WS		
Fish Worksheet		
fish_seas_oceans		Default

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 that you want to share with.
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.

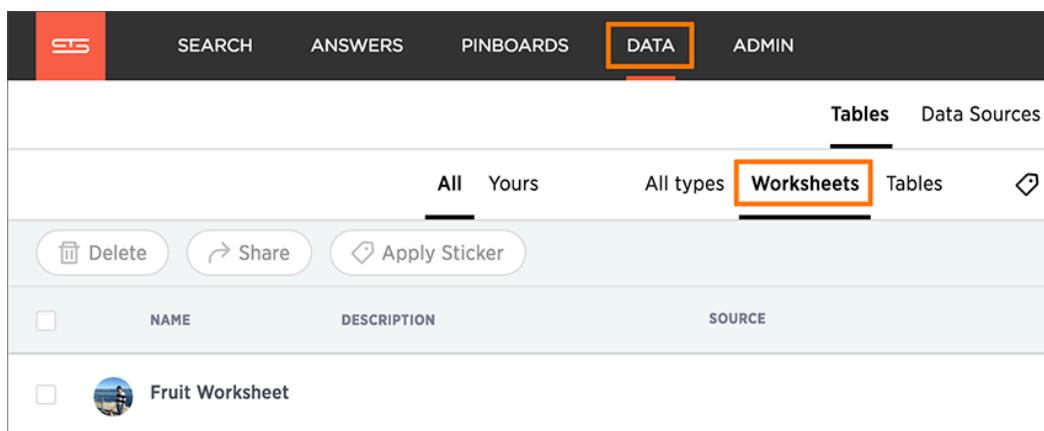
Share worksheets

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.

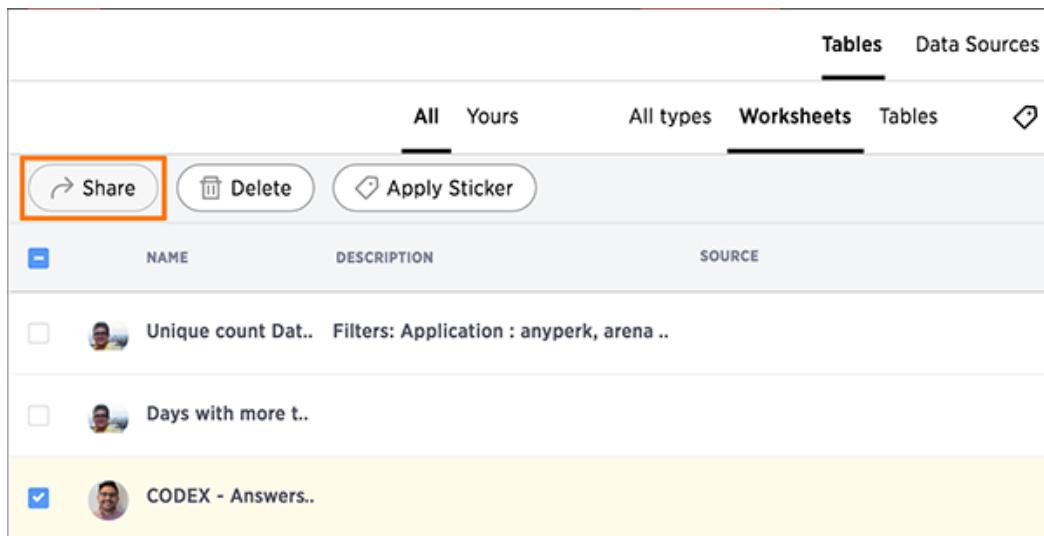
To share a pinboard:

1. Click on the DATA icon on the top navigation bar and then on Worksheets.



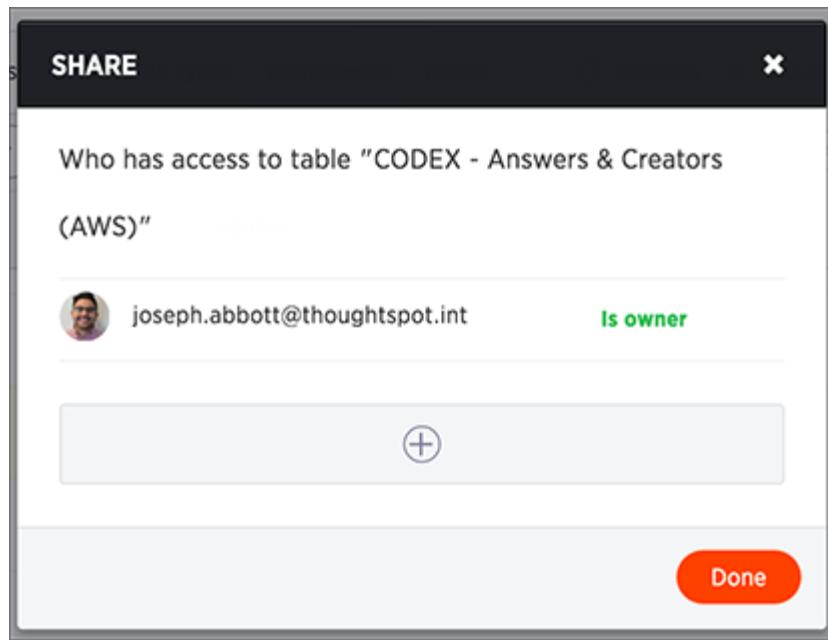
The screenshot shows the ThoughtSpot Data interface. The top navigation bar has tabs for SEARCH, ANSWERS, PINBOARDS, DATA (which is highlighted in orange), and ADMIN. Below the navigation bar, there are two main tabs: Tables and Data Sources. Under the Data Sources tab, there are sub-tabs for All, Yours, All types, Worksheets (which is highlighted in orange), Tables, and a refresh icon. Below these tabs are three buttons: Delete, Share (which is highlighted with a red box), and Apply Sticker. There are also columns for NAME, DESCRIPTION, and SOURCE. A single worksheet named "Fruit Worksheet" is listed, with a checkbox next to its name.

2. Select one or more worksheets to share, and click the Share icon.



The screenshot shows the ThoughtSpot Data interface with the Share button highlighted with a red box. The interface is identical to the previous one, but now multiple worksheets are selected for sharing. The "Fruit Worksheet" is still present, but other worksheets like "Unique count Dat.." and "Days with more t.." have checkboxes checked next to their names. The "CODEX - Answers.." worksheet has a checked checkbox and is highlighted with a yellow background.

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.

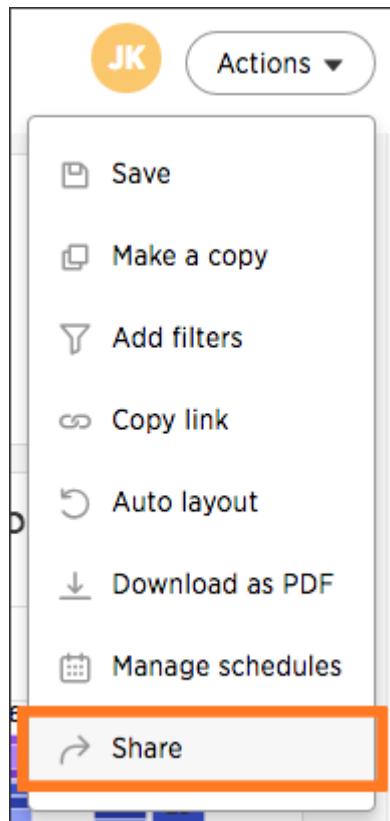
Share a pinboard

Summary: Whenever you view a pinboard you have the option of sharing it with others.

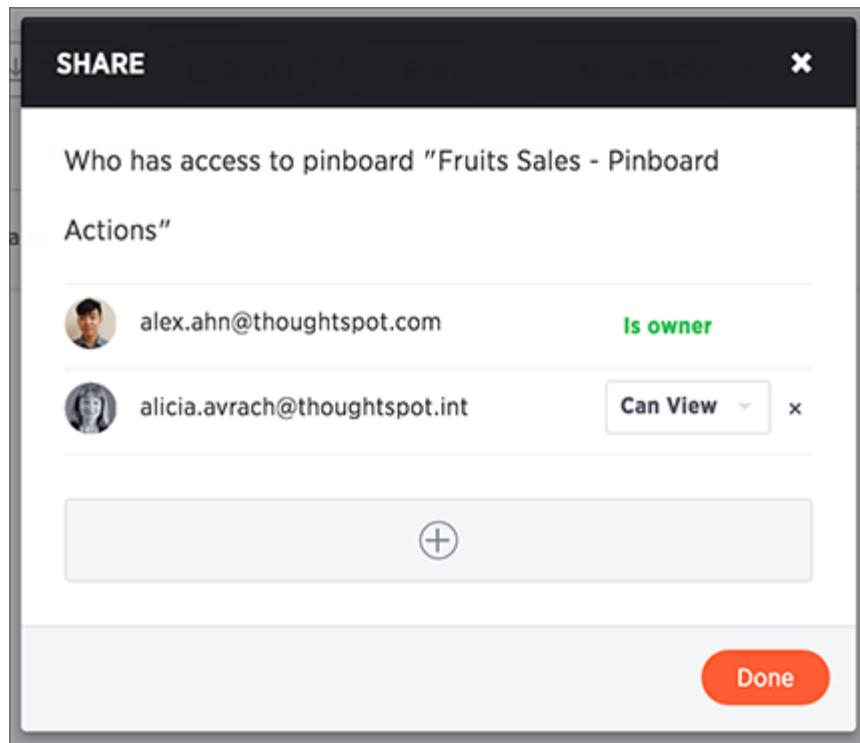
When you share a pinboard what you are really sharing is a live link to the pinboard, when you click **Share with....** So whenever someone else views it, they will see the most recently saved version with the most recent data. You do not have to be an administrator or the owner to share saved pinboards. Any user can share them, based on the access levels the user has.

To share a pinboard:

1. Configure the pinboard to look as you'll want it to appear when shared.
2. 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 will only see options available, based on your own access level. For example, if you have only View access, you cannot 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 and Save.
6. Click Add Permissions.

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)

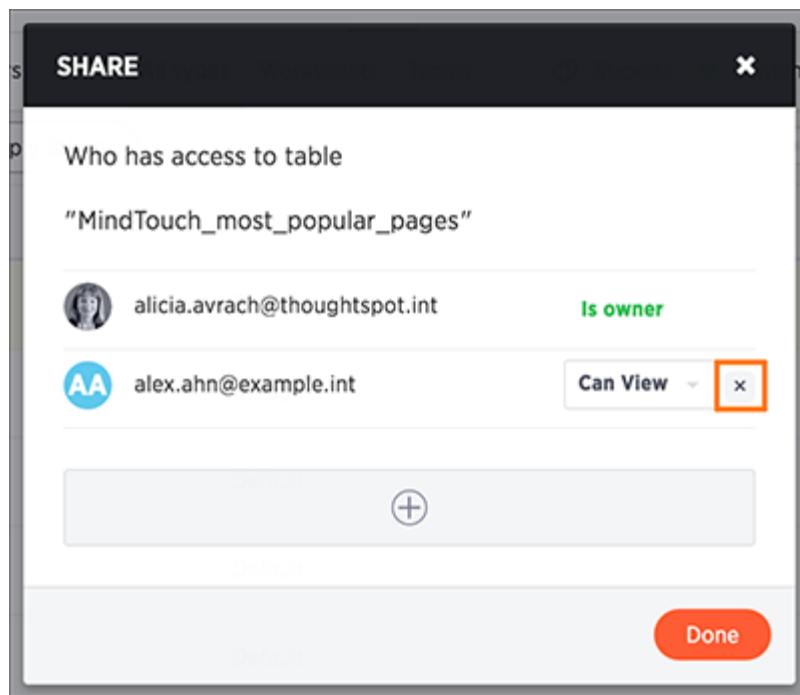
You may need to revoke access to an object (table, worksheet, or pinboard) that you have previously shared. Unsharing an object is very similar to sharing it.

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.

The screenshot shows a user interface for managing worksheets. At the top, there are tabs for 'Tables' and 'Data Sources'. Below that, there are filters for 'All', 'Yours', 'All types', 'Worksheets' (which is currently selected), and 'Tables'. There are also buttons for 'Share', 'Delete', and 'Apply Sticker'. The main area lists worksheets with columns for 'NAME', 'DESCRIPTION', and 'SOURCE'. The 'CODEX - Answers..' worksheet has a checked checkbox next to it, indicating it is selected for sharing. The 'Share' button is highlighted with an orange box.

4. Click the X next to the users and groups that you want to remove from sharing.



5. Click Done.

Row level security (RLS)

Summary: Using row level security, you can effectively customize search results and pinboards for each group that views them.

Row level security (RLS) allows you to define which groups can see individual rows in a table, based on the values in one of its columns. This RLS feature can handle thousands of groups, and allows you to set up flexible rules that are self-maintaining.

How RLS works

RLS works at the group level, not the individual user level. By default, all groups can see all rows for any table they can view. You can limit the rows a group can see by setting conditions on column values. The row level security rules you define on a table also apply to any worksheets and pinboards based on that table.

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.

RLS and administrators

If your installation has enabled the **Can Administer RLS** privilege, user groups with this privilege (directly or indirectly) can bypass row-level security (RLS) rules on worksheets. This privilege can only be assigned by a user who already **Has administration privilege**. See "[Change inclusion, join, or RLS for a worksheet](#)" for more information on how to do this.

Users with **Has administration privilege** are administrators and can see can see all data sources, and no type of row level security applies to them.

How rule-based RLS works

Summary: Use rule-based RLS to protect your data so that users see only those rows they are allowed to see based on their group membership.

Row level security works at the group level, not the individual user level. By default, all groups can see all rows for any table they can view. You can limit the rows a group can see by setting rules based on the data values contained in one or more columns. The row level security rules you define on a table also apply to any worksheets and pinboards based on that table.

For each data source (table or imported data), you will define one or more rules that govern which groups can see which data. The rules take the form of an expression which is evaluated for each row and group combination, to decide if that group can see that row. If the expression evaluates to “true”, for a particular group, they will be able to see that row.

For a list of operators and functions you can use to build these expressions see [Row level security rules reference](#).

Best practices for using Rule-Based Row Level Security

Use these best practices for Rule-Based Row Level Security:

1. Contact ThoughtSpot Support to have them disable search suggestions based on data values.

These are not hidden from users when you set row level security, so if you don't want them to ever see a search suggestion from a row they are not allowed to see, you'll need to disable the data value search suggestions.

2. Set up row level security on every table to which it applies.

It is always a possibility that a particular search will only include 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.

3. Give users access to worksheets instead of tables.

This is a general best practice in all implementations. It makes things easier for end users, because they only need to choose among a few sources, rather than every table. Also they won't have to choose five separate tables to get meaningful results. They can choose the single worksheet that combines the tables.

4. Explicitly grant access for users that should see all rows.

As soon as you create a row level security definition on a table for one group, all other groups are then blocked from seeing any rows in the table. You have to specifically grant other groups access in order for them to see any rows.

If you want to ensure that a group can always see all rows in a table, use a rule that always evaluates to “true” for that group. For example:

- if ts_groups = supergroup then true

Row level security with multiple conditions

When multiple row level security rules apply, the permissions are additive. That is, when there are multiple row level security conditions specified on a table, they are applied using an OR operator. If any of the rules applied allow a user to see a particular row, the row will be shown to that user.

If a user is a member of multiple groups, the user will be able to see all the rows that are visible to all of the groups, so the most permissive policy is used.

Set Rule-Based RLS

Summary: Explains the process for setting RLS rules.

Setting row level security is a three steps process:

1. Disable data value suggestions
2. Access the Rule Builder.
3. Define Rule-Based Row Level Security rules.

Disable data value suggestions

When you set Rule-Based Row Level Security, you need to first turn off search suggestions on data values.

When Rule-Based Row Level Security is set, it protects users from seeing data they shouldn't in worksheets and pinboards. However, the search suggestions are not filtered using Rule-Based Row Level Security, so it is possible someone could see a data value they should not in a search suggestion. Disabling suggestions on data values corrects this. Be sure and do this procedure before setting up Rule-Based Row Level Security.

Contact [ThoughtSpot Support](#), and tell them that you will be setting up Rule-Based Row Level Security. Ask them to disable data value suggestions for you. When this setting has been made, continue with the next procedure.

Access the Rule Builder

To set up rule-based row level security, you first need to access the Rule Builder. To get to the Rule Builder:

1. Click on DATA, on the top navigation bar.



2. Select a table from the list by clicking on it.

Row level security can only be set on tables and imported data, not on worksheets. The settings you make to the tables automatically apply to any worksheets or pinboards created on top of them.

3. Click **Row security** at the top right side of the page.
4. Click the **+ Add row security** button.

SYSTEM TABLE
Fruit_Cart_Large Columns Data Relationships Dependents Row security

With row security, control which groups have access to specific rows.

[+ Add row security](#)

Example



Type an expression to apply data from a column to a group.

For example, let's say you have a group called "EastSales" and a Group called "WestSales". You only want users in these groups to see rows with "East" and "West" respectively. Therefore, you would type:

```
concat(region, "Sales") = ts__groups
```

This way, any row with "East" in the data will be seen if user belongs to "EastSales", "West" will be seen if the belong to "WestSales"... etc.

Groups	This table
EastSales	Customer Region
WestSales	ABC East
	XYZ West

- The Rule Builder will appear, where you can define row level security rules.

Enter Rule name Rule Assistant

start typing your formula here

[Cancel](#) [Save](#)

Define Rule-Based Row Level Security rules

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. To define a row level security rule:

- Open the Rule Builder.
- Use the Rule Builder window to enter a row security rule.

You'll type in an expression, which gets evaluated for every row and group combination. If the rule evaluates to true for a particular row and group, that group will be able to see that

row. Use the variable `ts_groups` to refer to the group name. For example, the expression `ts_groups = location` would allow users to only see rows where the value in the location column was the same as their group name.

Notice how this type of security rule is self-maintaining. If you were to later add additional locations, the rule will still work, as long as users are placed in the group that matches their location.

Set RLS by Location

`ts_groups = location`

Rule Assistant

Good to go!

3. Use formulas if you want to define more complex expressions.

You can see a list of available operators by clicking on Rule Assistant.

Set RLS by Location

`ts_groups = location`

Rule Assistant

Good to go!

Cancel Save

concat(region, "Sales") = ts_groups

This way, any row with "East" in the data will be seen if the user belongs to "WestSales" etc.

Groups	This table
EastSales	Customer Region
WestSales	ABC East
	XYZ West

ts_groups

Returns the 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`

4. Use the suggestions to build a valid expression.

When the expression is valid, you'll see a green indicator at the bottom of the Rule Builder. As you type, you'll see suggestions for formula syntax, variables, and column names. Using these suggestions can make defining an expression easier, particularly when it comes to specific syntax, like enclosing parameters in parenthesis. And if you can't remember the exact column

name or variable you want to use, the suggestions can help.

5. Click **Save**.

The rule you created will be shown in the list of rules.

6. Click on its name to view or edit the rule. You can also add more rules by clicking + Add.

To test your rule, create a search that includes the column(s) you used in your expression, save it to a pinboard, and share it with all users. Log in as users in different groups, and make sure they are seeing the appropriate rows in the pinboard you created.

Bypass RLS on a worksheet

Administrators can bypass the RLS rules set on a the table at the worksheet level. Bypassing the underlying RLS rules allows users, that would not otherwise be unable, to see an aggregate/market view of a tables data. See how to [“Change inclusion, join, or RLS for a worksheet”](#) in this documentation for more information.

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.

System monitoring

System monitoring tools in ThoughtSpot include the Control Center, system log files and out-of-the-box system monitoring pinboards.

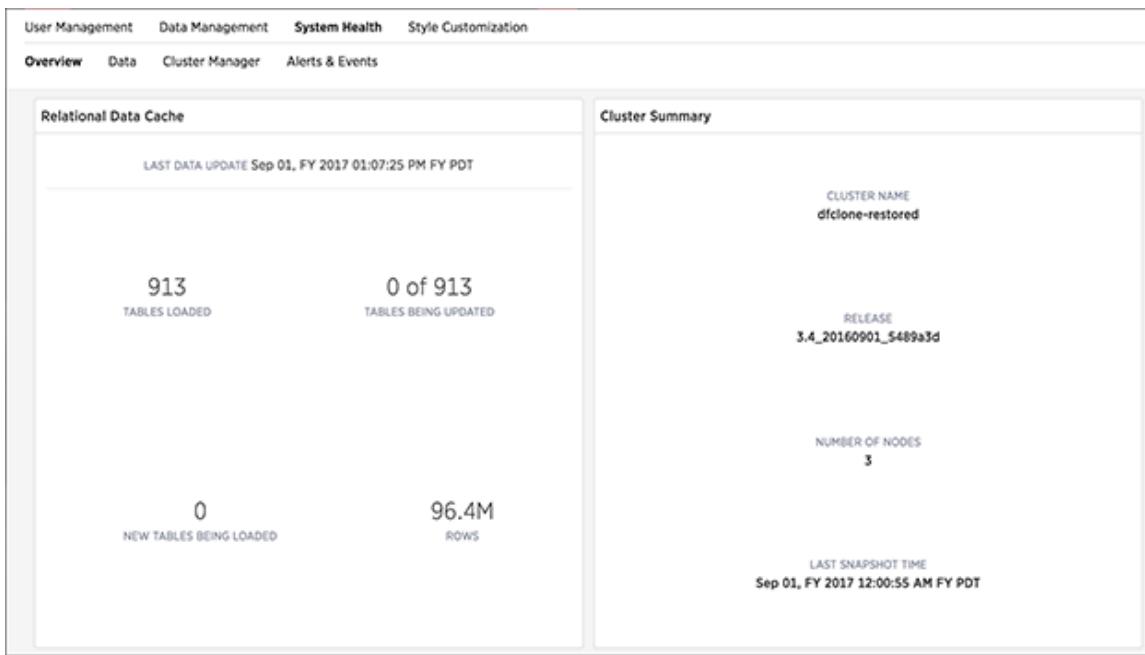
System Health Center

The ThoughtSpot application includes a System Health Center, for easy monitoring of usage and cluster health, including alerts. You can view the System Health Center by clicking on the **Admin** icon and then clicking **System Health**.

The screenshot shows the ThoughtSpot Control Center interface. At the top, there is a navigation bar with the ThoughtSpot logo, followed by links for SEARCH, ANSWERS, PINBOARDS, DATA, and ADMIN. The ADMIN link is highlighted with an orange border. Below the navigation bar, there are several tabs: User Management, Data Management, System Health (which is also highlighted with an orange border), and Style Customization. Under the System Health tab, there are four sub-tabs: Overview, Data, Cluster Manager, and Alerts & Events. The main content area has two sections. The left section, titled "Relational Data Cache", displays the message "LAST DATA UPDATE Sep 01, FY 2017 01:07:25 PM FY PDT". Below this, it shows "913 TABLES LOADED". The right section, titled "Cluster Summary", shows "0 of 913 TABLES BEING UPDATED".

Overview

The **Overview** tab shows a summary of overall cluster status, usage and capacity information, configuration changes, and critical alerts.



Data

The **DATA** section shows all the stored tables with details on the last update time, time taken for auto-indexing, number of rows, etc.

The **Database** section has the following values:

Section	Status	Meaning
Database	READY	The data has been loaded.
	IN PROGRESS	The data is still being loaded.
	STALE	The data is not up to date.
	ERROR	The table is invalid. Call Customer Support.
Search	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 in-

Section	Status	Meaning
		dexed, 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.

Table Information		
DATABASE	USER SCHEMA	NAME
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-a6c0991e-462d-4343-8f30-13a1a1a1a1a1
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-96a40275-7427-4343-8f30-13a1a1a1a1a1
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-bcda2191-cd6c-4343-8f30-13a1a1a1a1a1
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-9dc1bfc7-2d27-4343-8f30-13a1a1a1a1a1
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-29546f4d-84ac-4343-8f30-13a1a1a1a1a1
thoughtspot_analytics	falcon_default_sche..	candidates
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-4343525d-261b-4343-8f30-13a1a1a1a1a1
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-0694fd57-fadf-4343-8f30-13a1a1a1a1a1
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-51435761-0aac-4343-8f30-13a1a1a1a1a1
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-0fb9daec-5230-4343-8f30-13a1a1a1a1a1
dw	falcon_default_sche..	fact_lead_transitions
FalconUserDataDataBase	FalconUserDataSch..	USERDATA-12b3cf23-0de5-4343-8f30-13a1a1a1a1a1

(showing rows 1 - 14 of 913.)

Cluster Manager

The Cluster Manager section show detailed information about a cluster including latency over time, snapshot status, installed release, node functions, and logs.

Cluster Details		Cluster Logs		
PROPERTY	CONTENT	TYPE	RELEASE	
Cluster Name	dflclone-316	Feb 08, 2016 08:12:35 am PST	UPDATE	3.2_20160208_fa1495
Cluster ID	dflcid	Feb 07, 2016 09:01:10 pm PST	UPDATE	3.2_20160207_f91480
Release	3.2_20160208_fa1495	Feb 07, 2016 07:10:17 pm PST	UPDATE	3.2_20160207_6476f4
Last Update Time	Feb 08, 2016 08:12:35 am PST	Feb 07, 2016 01:31:03 pm PST	UPDATE	3.2_20160207_3e7a6d
Zookeeper Servers	192.168.6.154.2181,192.168.6.155.2181,192.168.6.156.2181	Feb 06, 2016 08:45:03 pm PST	UPDATE	3.2_20160206_c473465
HDFS Name Nodes	192.168.6.154.8020,192.168.6.155.8020	Feb 06, 2016 09:15:50 am PST	UPDATE	3.2_20160206_d93f46
Cluster Alert Email		Feb 01, 2016 02:19:11 pm PST	UPDATE	3.2_20160205_40942ed
Periodic Snapshot...	disabled	Feb 04, 2016 12:10:29 pm PST	UPDATE	3.2_20160204_221fb60
		Feb 04, 2016 12:10:29 pm PST	UPDATE	3.2_20160204_221fb60

(showing rows 1 - 8 of 8)

List of Snapshots			Average Latency Last 2 Weeks (sec)		
TIME	NAME	REASON	Avg Total (sec)	Avg Min (sec)	Avg Max (sec)
Feb 06, 2016 08:02:25 am PST	pre_3.2_20160208_fa1495....	Automatic pre update snapshot	0.30	0.25	0.35
Feb 07, 2016 08:49:55 pm PST	pre_3.2_20160207_f91480....	Automatic pre update snapshot	0.30	0.25	0.35
Feb 07, 2016 06:58:48 pm PST	pre_3.2_20160207_6476f4....	Automatic pre update snapshot	0.30	0.25	0.35
Feb 07, 2016 01:20:18 pm PST	pre_3.2_20160207_b7c0ead....	Automatic pre update snapshot	0.30	0.25	0.35
Feb 06, 2016 08:33:58 pm PST	pre_3.2_20160206_c473465....	Automatic pre update snapshot	0.30	0.25	0.35
Feb 06, 2016 09:04:42 am PST	pre_3.2_20160206_d93f46....	Automatic pre update snapshot	0.30	0.25	0.35
Feb 05, 2016 02:29:42 pm PST	pre_3.2_20160205_40942ed....	Automatic pre update snapshot	0.30	0.25	0.35
Feb 05, 2016 01:08:41 pm PST	pre_3.2_20160205_40942ed....	Automatic pre update snapshot	0.30	0.25	0.35

(showing rows 1 - 10 of 20)

Average Latency Last 2 Weeks (sec)

Daily (Timeline) for 2016

Events and Alerts

The Events and Alerts section shows notifications, alerts, and an audit trail of cluster configuration changes..

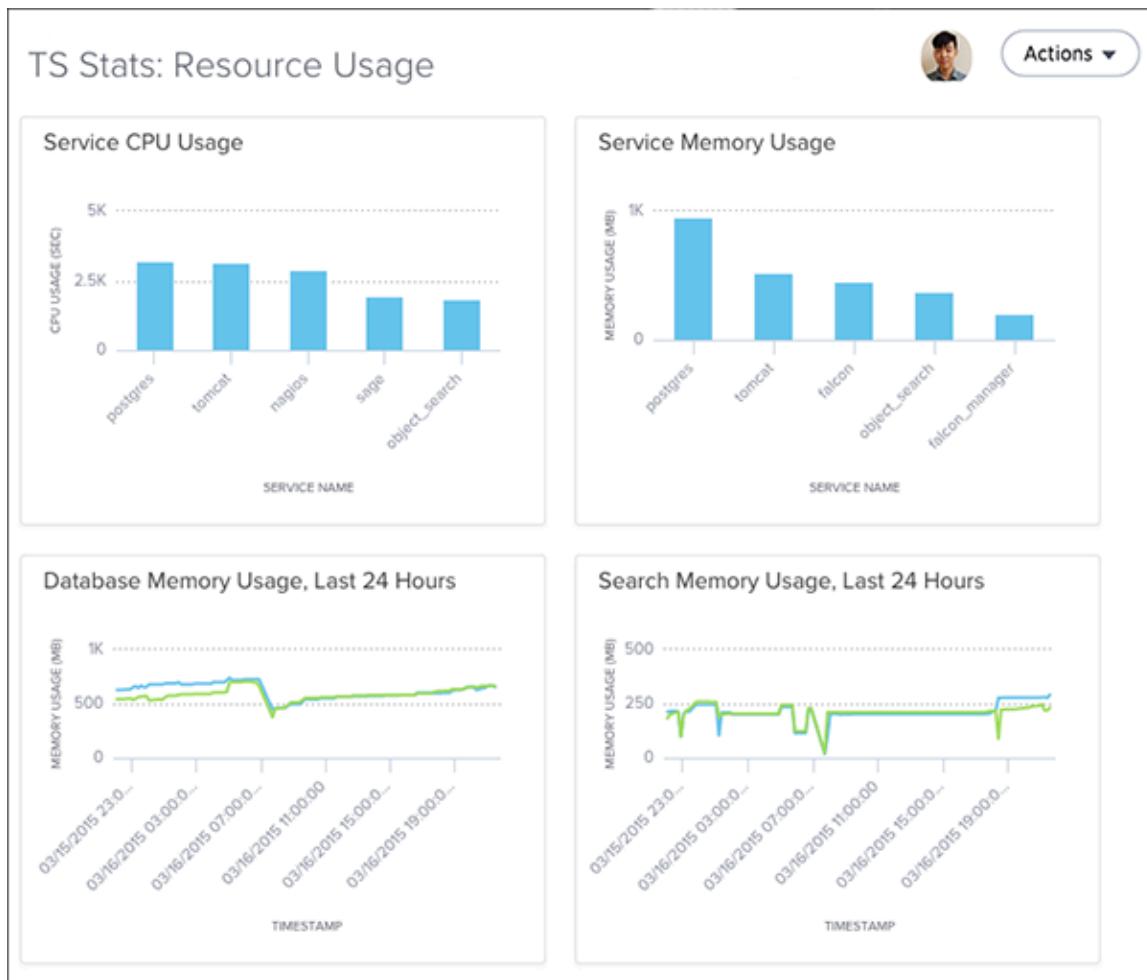
Configuration Events

ClusterConfiguration tscli service add-javaopt tomcat.tomcat D orion.customBrandingFontCustomization..	3 minutes ago
ClusterConfiguration tscli service add-javaopt tomcat.tomcat D orion.customBrandingEnabled true	3 minutes ago
ClusterConfiguration tscli service add-javaopt tomcat.tomcat D orion.defaultQuarterStartMonth 2	1 hour ago
ClusterConfiguration tscli service delete-javaopt tomcat.tomcat D orion.defaultQuarterStartMonth	2 hours ago
ClusterConfiguration tscli etl enable-lw --username priyanka.shriram@thoughtspot.com --admin_userna..	2 hours ago

System monitoring pinboards

There are several system monitoring pinboards in ThoughtSpot that include information about the system

status and resource usage. The information in these pinboard is updated hourly from an internal database that collects monitoring statistics.



The monitoring pinboards can only be viewed by users with the administrator privilege. They are based on worksheets, which administrators can view, but not modify. The underlying tables are protected system tables that cannot be accessed directly. However, you can create new monitoring reports from the worksheets.

The worksheets for system monitoring are:

- Ts: bi server
- Ts: database
- Ts: loader
- Ts: search
- Ts: service resources

Here is a list of the system monitoring pinboards:

- **TS Stats: Usage** Helps you see how much the system is being used. Shows search and pinboard activity by user and by date. Examples include:
 - Questions asked by user
 - Questions asked by date
 - Pinboard impressions
- **TS Stats: Suggestions** Helps you monitor the performance statistics for the suggestions provided in the search bar. Shows the number and latency of suggestions given over time. Examples include:
 - Suggestion volume over time
 - Suggestion latency over time
- **TS Stats: Queries** Helps you monitor database performance over time by showing query volume,

- latency, and any errors. Examples include:
 - Query latency by size of response
 - Average vs. maximum query latency
 - Database queries and errors
- **TS Stats: Resource Usage** Helps you monitor cluster resources by showing memory and CPU usage per component:
 - Service
 - Database
 - Search
 - Host
 - Aggregate (all)

Examples include:

- CPU usage per component over time
- Memory usage per component over time
- Aggregate memory usage over time

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:

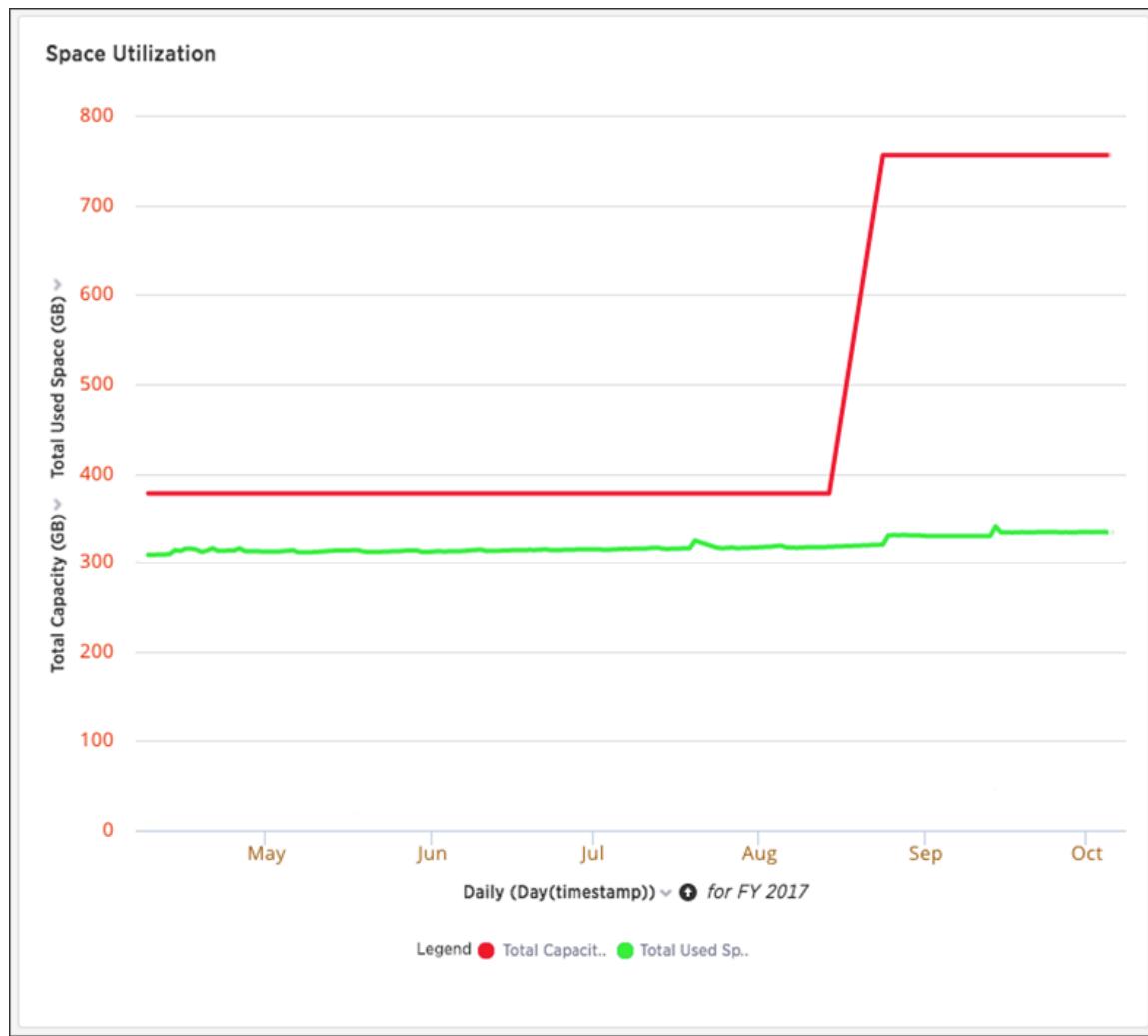
- `/export/logs/orion`
- `/export/logs/oreo`
- `/export/logs/hadoop`
- `/export/logs/zookeeper`

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](#).

About the Space Utilization chart

The Space Utilization chart is one of the available charts for you to use when checking the cluster overview. You can find the chart in the [Overview](#) section of the System Health center. This line chart displays the total used space, which consists of raw uncompressed data, including replication.



The x-axis is by time. It allows you to zoom in and see daily or hourly data. The y-axis measures the size in GB. 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. The increase in the red line at the end of the period indicates the addition of extra hardware, resulting in increased capacity.

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.

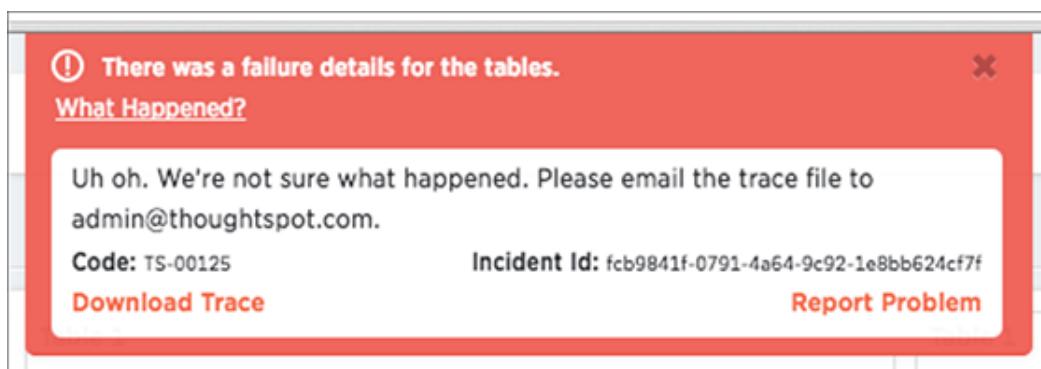
3. Change directories to the directory where you wrote the log bundle.
4. Issue the command to send the log bundle to ThoughtSpot Support:

```
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.

1. 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.

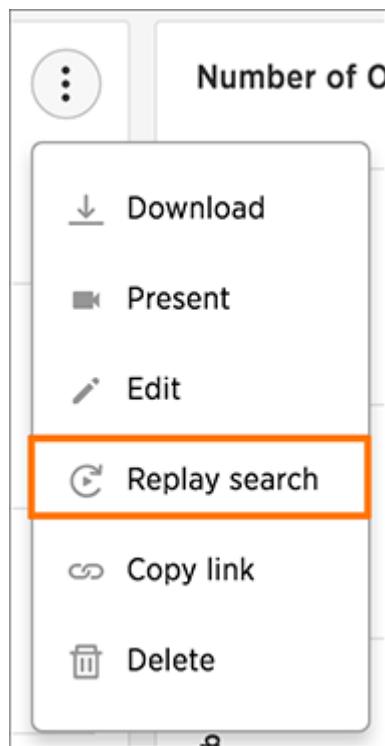
Set up recording for Replay Search

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.

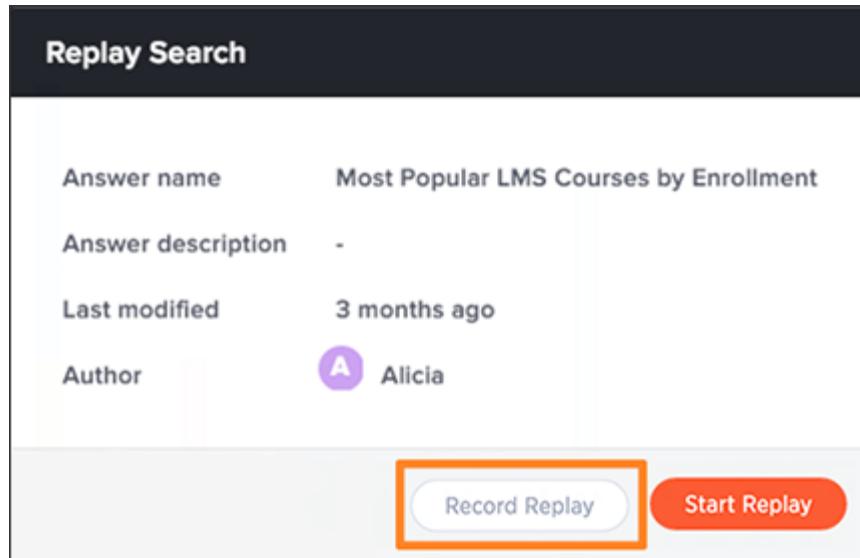
To record a search replay using ThoughtSpot:

1. While viewing a chart or table in ThoughtSpot, click the Replay Search icon.



2. Click the Record Replay button.

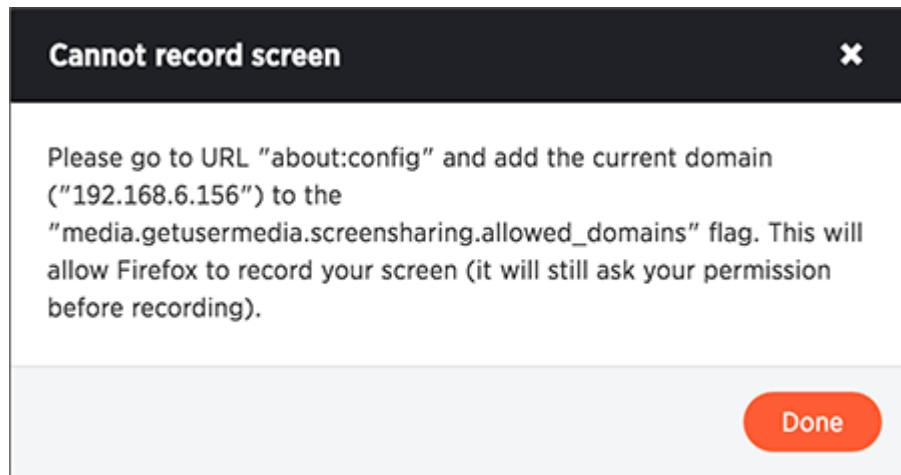
If you do not see the button, you must log in as a user with administrator privileges.



A message will display, showing a URL and a domain or an IP address.

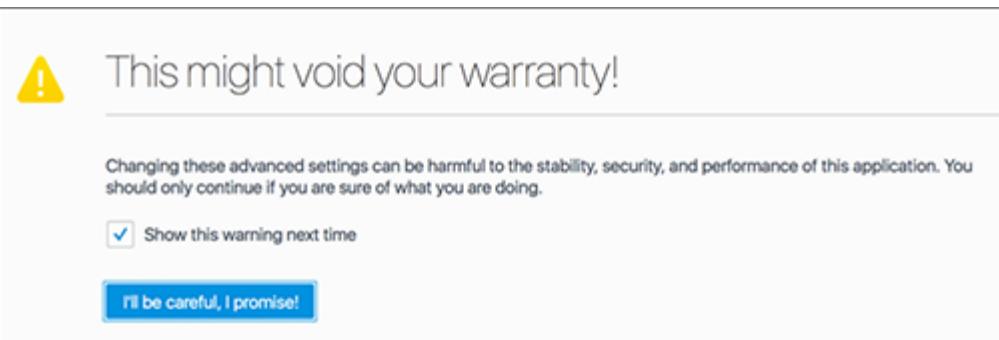
3. Make note of both of these items.
4. Open a new browser tab and go to the URL shown in the message (for example, "about:config").

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!".

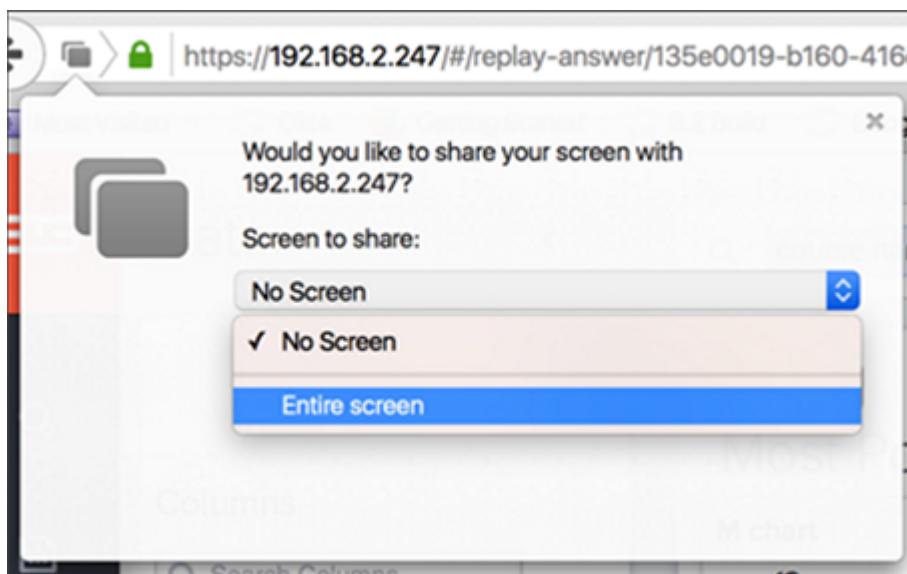


6. Find the setting for media.getusermedia.screensharing.allowed_domains, and add the domain used by ThoughtSpot.

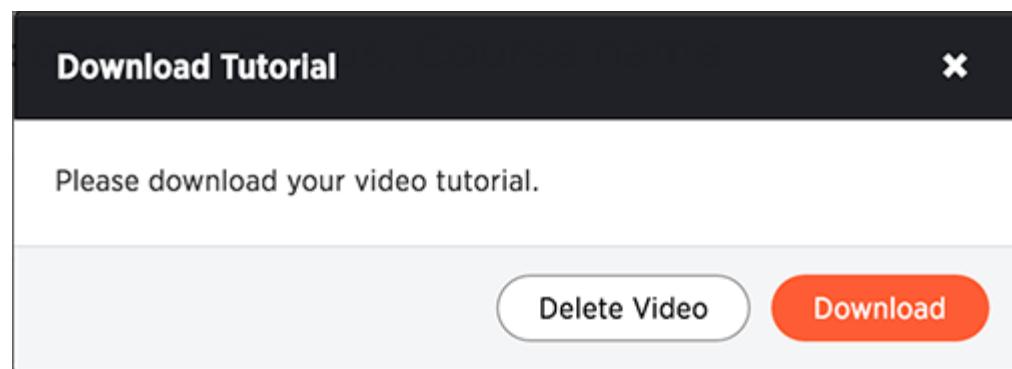
This domain will be the same one you made note of from the Cannot record screen message.

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,clicospark.com,*
media.getusermedia.screensharing.enabled	default	boolean	true
media.gmp-gmpopenh264.abl	user set	string	x86_64-gcc3-u-i386-x86_64
media.gmp-gmpopenh264.lastUpdate	user set	integer	1454453226

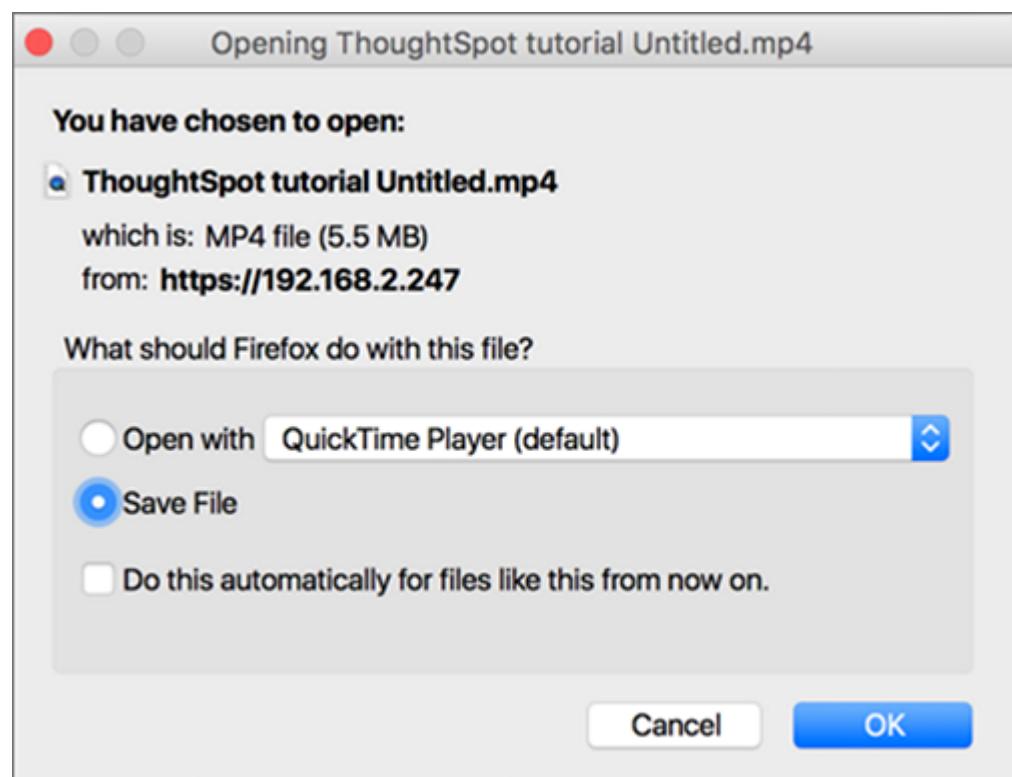
7. If you see a message asking if you'd like to share your screen with the IP address or domain name of ThoughtSpot, select Entire screen.



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.



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 enabled by default a snapshot configuration. 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.	<ul style="list-style-type: none">• Restore a cluster to a prior

Stored	In the cluster's HDFS	<p>state.</p> <ul style="list-style-type: none"> • 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.
Advantages	<ul style="list-style-type: none"> • Can be taken on or restored to a running cluster • Creation and restore is fast 	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • 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. <p> </td> </tr> </table> You should never restore from a snapshot or backup yourself, instead contact ThoughtSpot Support for help.</p>

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 backup policy. For backups, there is no such policy but you may want to create one or several of your own configurations. This section is useful for understanding how 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 (one) hour with the midnight offset of 0 (zero).

```
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 every 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 <snapshot_name> <reason>
```

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.

```
$ 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: true
```

This policy takes a snapshot every hour starting at midnight on Sunday. It retains the snapshots from the last three hour intervals and two snapshots from two of the previous 4 hour intervals. That means, there are 5 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`.

CAUTION: 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 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.
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 your 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). The deletion takes place before the backup that replaces it. This means that if you set a bucket retention of 1 the system stores a single backup at any one time. And, more importantly, the only backup would be deleted before the next is taken. For this reason, you should always set the number of

backups to be greater than 2 (two), to ensure you have at least one backup available in the case of a failure while taking a backup.

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.	<code>tscli backup-policy ls</code>
Show a backup policy.	<code>tscli backup-policy show <name></code>
Check the status of a policy.	<code>tscli backup-policy status <name></code>
Change an existing policy.	<code>tscli backup-policy update <name></code>
Disable or enable an existing policy.	<code>tscli backup-policy disable or enable</code>
Delete a policy	<code>tscli backup-policy delete <name></code>

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 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.
- [Change 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.

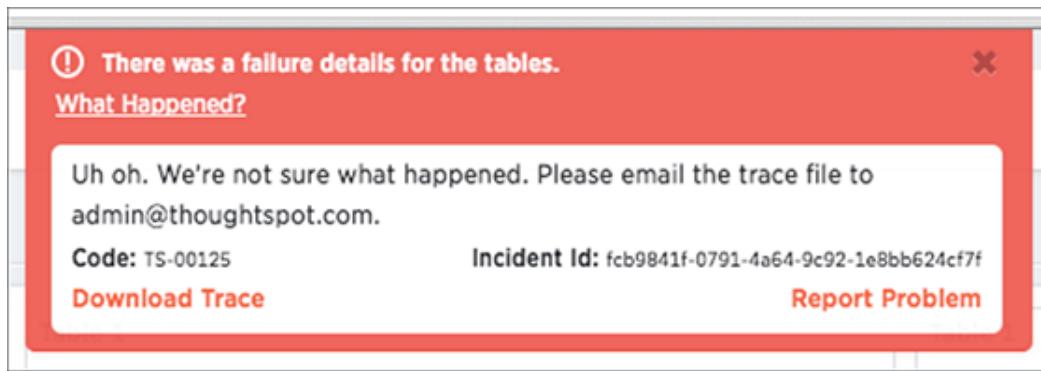
Get logs

For troubleshooting on specific incidents or cluster problems, getting a log bundle can help.

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 /tmp/debug.tar.gz
```

This command collects logs from a specific time window:

```
$ tscli logs collect --include system,orion --from 20150520-12:00:00 --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 *hadoop*,*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.

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_dir_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.

Change 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 `tq1` is run.

If you need to change the timezone, for example when moving a ThoughtSpot appliance to another data center [contact ThoughtSpot Support](#), and they will change the timezone for you.

Note that if the timezone was not set correctly when shipped, it may be necessary to have ThoughtSpot Support reset it. This can be true even if the file `/etc/timezone` lists the correct timezone. Sometimes the timezone that is listed is not the active timezone and it needs to be reset.

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
```

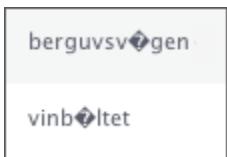
- You can choose to ignore the message, and access ThoughtSpot without SSL.

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

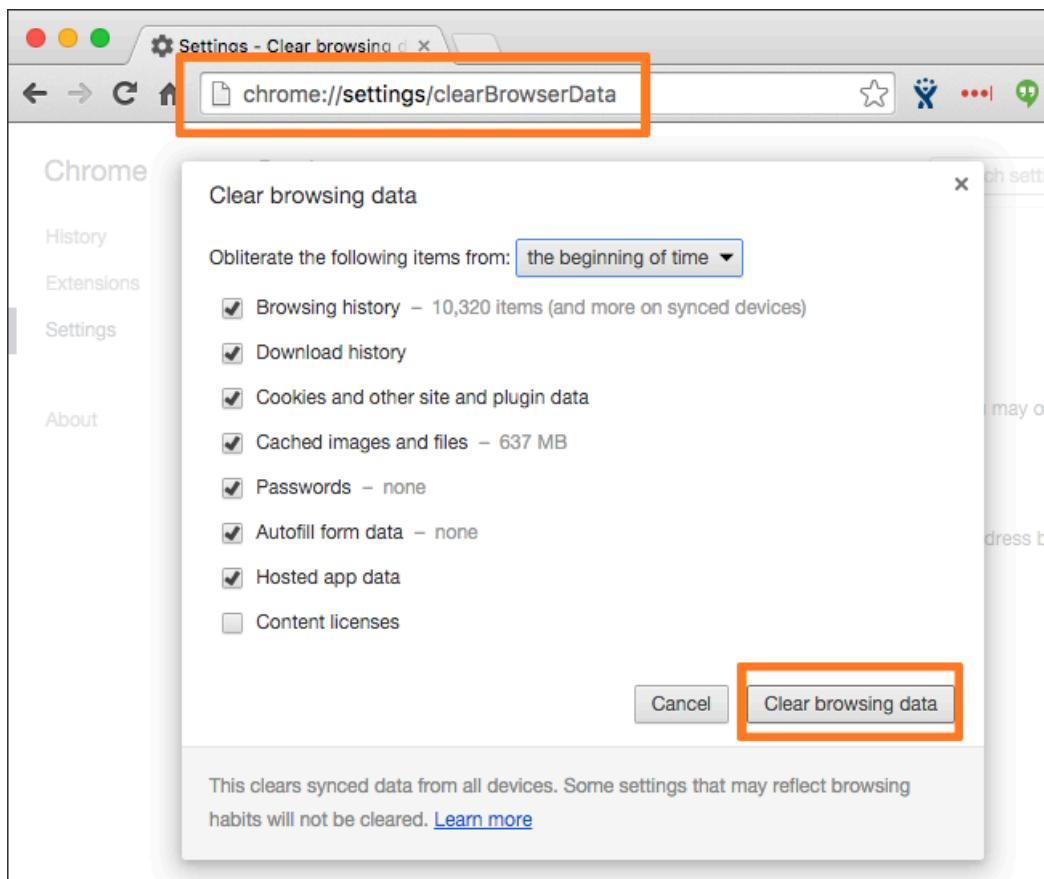
You might occasionally see unexpected behavior that is due to the Web browser caching information from ThoughtSpot. In this case, clearing the browser cache and reloading the page should resolve the problem. You can usually resolve these situations by clearing the browser cache:

- 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.

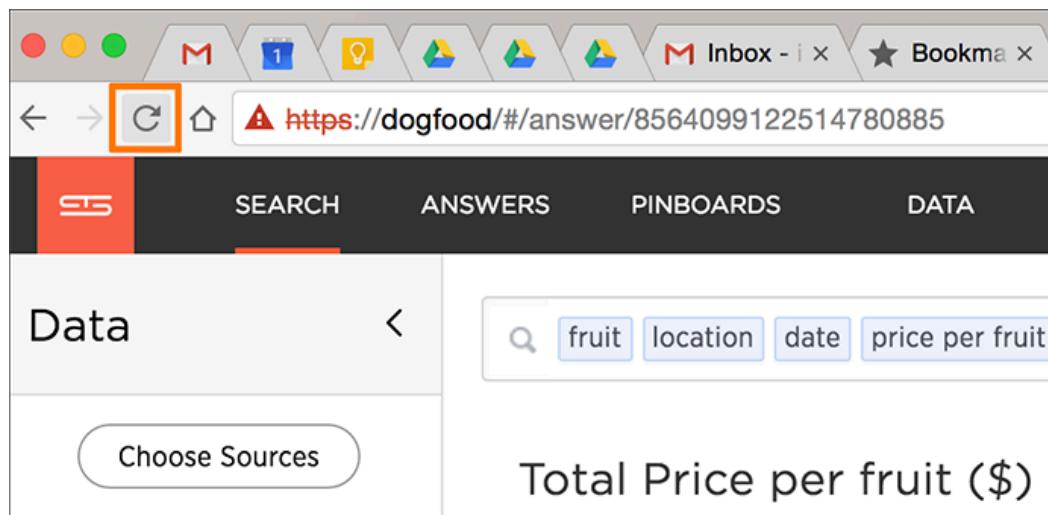
To resolve any of these situations, clear the browser cache:

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:



4. 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.

The screenshot shows two adjacent interfaces. On the left is the 'Data' interface, which includes a 'Choose Sources' button, a search bar for columns, and a list of available sources: 'Udemy_user_activity_report', 'Udemy_user_course_activi..', and 'Udemy_user_progress_rep..'. Below these are buttons for '+ Add Columns' and '+ Add Rows'. A formula named 'weekday' is selected and highlighted with a red box. On the right is the 'Udemy Worksheet' interface, showing a table with data from the 'Udemy_user_activity_report' source. The table has three columns: 'date joined', 'minutes video watched', and 'number of courses assigned'. The data rows include: date joined (FY 2016-08-27 17:00:00), minutes video watched (1,159.00), number of courses assigned (0); date joined (FY 2017-05-18 17:13:47), minutes video watched (1,159.00), number of courses assigned (0); etc.

- 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**.

The screenshot shows the 'Formula Assistant' for the 'weekday' formula. The formula is displayed as 'day_number_of_week (date of last visit (utc))'. A tooltip provides a detailed description: 'Returns the number (1-7) of the day in a week from the given date with 1 being Monday and 7 being Sunday.' Examples provided are: 'day_number_of_week (01/30/2015) = 6' and 'day_number_of_week (shipped)'. A green checkmark icon indicates that the formula is valid.

- Make your changes, and saving the formula by clicking **Save**.
- Save the worksheet by clicking **Save**.
- 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:

1. Run the following command to find the number of tables that are building and their names

```
tscli cluster status --mode table
```

2. You may notice that a few small tables are taking up a lot of time to be built. However, this could simply be due to the deceptively large size of the table. There is also the chance it could be due to a high data version issue. In order 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;' | tq1
```

3. If there is a large number of rows in the table, proceed to shard the table.
4. 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:

```
tq1> 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 many 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.

About this reference

This Reference section contains the commands and their syntax for all the command line tools in ThoughtSpot.

Included in this guide are:

- [Keyword reference](#) lists the available keywords to use in your search. These are also listed in the Help Center, which is available from [Help](#) on the top navigation bar in ThoughtSpot.
- [tscli command reference](#) lists the ThoughtSpot Command Line Interface commands.
- [Formula reference](#) lists the available formula operators and functions. These are also listed in the Formula Assistant, which is available from the place in ThoughtSpot where you build formulas.
- [Date and time formats reference](#) lists the accepted date, time, and timestamp formats that you can use when uploading data through the Web interface or using the ThoughtSpot Loader.
- [Row level security rules reference](#) lists the operators for building row level security rules.
- [TQL reference](#) lists the SQL commands that are supported in TQL.
- [ThoughtSpot Loader flag reference](#) lists the options for loading data with tsload.

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.

General keywords

Keyword	Examples
top	<ul style="list-style-type: none">• top sales rep by count sales for average revenue >10000• sales rep average revenue for each region top
bottom	<ul style="list-style-type: none">• bottom revenue average revenue by state• customer by revenue for each sales rep bottom
n	<ul style="list-style-type: none">• top 10 sales rep revenue
n	<ul style="list-style-type: none">• bottom 25 customer by revenue for each sales rep
sort by	<ul style="list-style-type: none">• revenue by state sort by average revenue• revenue by customer sort by region

Date keywords

Keyword	Examples
after	<ul style="list-style-type: none"> order date after 10/31/2014
before	<ul style="list-style-type: none"> order date before 03/01/2014
between ... and ...	<ul style="list-style-type: none"> order date between 01/30/2012 and 01/30/2014
day of week	<ul style="list-style-type: none"> revenue by day of week last 6 months
week	<ul style="list-style-type: none"> revenue by week last quarter
month	<ul style="list-style-type: none"> revenue by month last year
daily	<ul style="list-style-type: none"> shipments by region daily
weekly	<ul style="list-style-type: none"> revenue weekly
monthly	<ul style="list-style-type: none"> commission > 10000 monthly
quarterly	<ul style="list-style-type: none"> sales quarterly for each product
yearly	<ul style="list-style-type: none"> shipments by product yearly
day of week</var>	<ul style="list-style-type: none"> count shipments Monday

Keyword	Examples
month</var>	<ul style="list-style-type: none"> commission January
month year</var>	<ul style="list-style-type: none"> commission by sales rep February 2014
year</var>	<ul style="list-style-type: none"> revenue by product 2014 product name contains snowboard
yesterday	<ul style="list-style-type: none"> sales yesterday for pro -ski200 by store
week to date	<ul style="list-style-type: none"> sales by order date week to date for pro-ski200
month to date	<ul style="list-style-type: none"> sales by product month to date sales > 2400
quarter to date	<ul style="list-style-type: none"> sales by product quarter to date for top 10 products by sales
year to date	<ul style="list-style-type: none"> sales by product year to date
last day	<ul style="list-style-type: none"> customers last day by referrer
last week	<ul style="list-style-type: none"> customers last week by store
last month	<ul style="list-style-type: none"> customers last month by day
last quarter	<ul style="list-style-type: none"> customers last quarter sale >300

Keyword	Examples
last year	<ul style="list-style-type: none"> top 10 customers last year by sale by store for region west
n days	<ul style="list-style-type: none"> visitors last 7 days
n weeks	<ul style="list-style-type: none"> visitors last 10 weeks by day
n months	<ul style="list-style-type: none"> visitors last 6 months for homepage visits > 30 by month
n quarters	<ul style="list-style-type: none"> visitors last 2 quarters by month by campaign
n years	<ul style="list-style-type: none"> visitors last 5 years by revenue for sum revenue >5000
growth of ... by ...	<ul style="list-style-type: none"> growth of sales by order date
growth of ... by ... daily	<ul style="list-style-type: none"> growth of sales by order date daily
growth of ... by ... monthly	<ul style="list-style-type: none"> growth of sales by date shipped monthly sales > 24000
growth of ... by ... weekly	<ul style="list-style-type: none"> growth of sales by receipt date weekly for proski2000
growth of ... by ... quarterly	<ul style="list-style-type: none"> growth of sales by date shipped quarterly

Keyword	Examples
growth of ... by ... yearly	<ul style="list-style-type: none"> • growth of sales by date closed yearly
daily year-over-year	<ul style="list-style-type: none"> • growth of revenue by order date daily year-over-year
weekly year-over-year	<ul style="list-style-type: none"> • growth of revenue by date shipped weekly year-over-year
monthly year-over-year	<ul style="list-style-type: none"> • growth of revenue by receipt date monthly year-over-year
quarterly year-over-year	<ul style="list-style-type: none"> • growth of revenue by date shipped quarterly year-over-year
<i>n</i> days ago	<ul style="list-style-type: none"> • sales 2 days ago
<i>n</i> weeks ago	<ul style="list-style-type: none"> • sales 4 weeks ago by store
<i>n</i> months ago	<ul style="list-style-type: none"> • sales 2 months ago by region
<i>n</i> quarters ago	<ul style="list-style-type: none"> • sales 4 quarters ago by product name contains deluxe
<i>n</i> years ago	<ul style="list-style-type: none"> • sales 5 years ago by store for region west

Keyword	Examples
today	<ul style="list-style-type: none"> sales today by store
next day	<ul style="list-style-type: none"> shipments next day by order
next week	<ul style="list-style-type: none"> shipments next week by store
next month	<ul style="list-style-type: none"> appointments next month by day
next quarter	<ul style="list-style-type: none"> opportunities next quarter amount > 30000
next year	<ul style="list-style-type: none"> opportunities next year by sales rep
n days	<ul style="list-style-type: none"> shipments next 7 days
n weeks	<ul style="list-style-type: none"> shipments next 10 weeks by day
n months	<ul style="list-style-type: none"> openings next 6 months location
n quarters	<ul style="list-style-type: none"> opportunities next 2 quarters by campaign
n years	<ul style="list-style-type: none"> opportunities next 5 years by revenue

Time keywords

Keyword	Examples
detailed	<ul style="list-style-type: none"> ship time detailed

Keyword	Examples
last minute	<ul style="list-style-type: none"> count homepage views last minute
last hour	<ul style="list-style-type: none"> count unique visits last hour
<i>n</i> minutes	<ul style="list-style-type: none"> count visitors last 30 minutes
<i>n</i> hours	<ul style="list-style-type: none"> count visitors last 12 hours
hourly	<ul style="list-style-type: none"> visitors by page name hourly
<i>n</i> minutes ago	<ul style="list-style-type: none"> sum inventory by product 10 minutes ago
<i>n</i> hours ago	<ul style="list-style-type: none"> sum inventory by product by store 2 hours ago

Text keywords

Keyword	Examples
begins with	<ul style="list-style-type: none"> product name begins with 'pro'
contains	<ul style="list-style-type: none"> product name contains "alpine" description contains "snow shoe"
ends with	<ul style="list-style-type: none"> product name ends with 'deluxe'

Keyword	Examples
not begins with	<ul style="list-style-type: none"> product name not begins with "tom's"
not contains	<ul style="list-style-type: none"> product color not contains 'tan' product color not contains 'red'
not ends with	<ul style="list-style-type: none"> product name not ends with "trial"
similar to	<ul style="list-style-type: none"> course name similar to 'hand'
not similar to	<ul style="list-style-type: none"> course name not similar to 'hand'

Number keywords

Function	Examples
sum	<ul style="list-style-type: none"> sum revenue
average	<ul style="list-style-type: none"> average revenue by store
count	<ul style="list-style-type: none"> count visitors by site
max	<ul style="list-style-type: none"> max sales by visitor by site
min	<ul style="list-style-type: none"> min revenue by store by campaign for cost > 5000
standard deviation	<ul style="list-style-type: none"> standard deviation revenue by product by month for date after 10/31/2010
unique count	<ul style="list-style-type: none"> unique count visitor by product page last week

Function	Examples
variance	<ul style="list-style-type: none"> • variance sale amount by visitor by product for last year

Filter keywords

Function	Examples
between... and	<ul style="list-style-type: none"> • revenue between 0 and 1000
>	<ul style="list-style-type: none"> • sum sale amount by visitor by product for last year sale amount > 2000
<	<ul style="list-style-type: none"> • unique count visitor by product by store for sale amount < 20
>=	<ul style="list-style-type: none"> • count calls by employee lastname >= m
<=	<ul style="list-style-type: none"> • count shipments by city latitude <= 0
=	<ul style="list-style-type: none"> • unique count visitor by store purchased products = 3 for last 5 days
!=	<ul style="list-style-type: none"> • sum sale amount region != canada region != mexico

Location keywords

Keyword	Examples
near	<ul style="list-style-type: none"> revenue store name county near san francisco
near... within <i>n</i> miles km meters	<ul style="list-style-type: none"> revenue store name county near alameda within 50 miles
farther than <i>n</i> miles km meters from	<ul style="list-style-type: none"> 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 keywords

Keyword	Example
quarter (<i>date</i>)	quarter (commit date)</td></tr>
month of quarter (<i>date</i>)	month of quarter (commit date)</td></tr>
week of year (<i>date</i>)	week of year (commit date)</td></tr>
week of quarter (<i>date</i>)	week of quarter (commit date)</td></tr>
week of month (<i>date</i>)	week of month (commit date)</td></tr>
day of year (<i>date</i>)	day of year (commit date)</td></tr>
day of quarter (<i>date</i>)	day of quarter (commit date)</td></tr>

day (<i>date</i>)	day (order date)</td> </tr>
day of week (<i>date</i>)	day of week (order date)</td> </tr>
hour (<i>datetime</i>)	hour (timestamp)</td> </tr> </table>

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.

TQL commands

You can use TQL either [through the ThoughtSpot application's web interface](#) or the [command line interface](#) in the Linux shell.

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.

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.

When referring to objects using fully qualified object names, the syntax is:

```
"database"."schema"."table"
```

As a best practice, you should enclose object names (database, schema, table, and column) in double quotes, and column values in single quotes.

Basic commands

Syntax	Description	Examples
help	Displays command help.	TQL> help

View schemas and data

Syntax	Description	Examples
SHOW DATABASES	Lists all available databases.	TQL> SHOW DATABASES;
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.	TQL> USE "fruit_database";
SHOW SCHEMAS	Lists all schemas within the current database.	TQL> SHOW SCHEMAS;
SHOW TABLES	Lists all tables within the current database by schema.	TQL> SHOW TABLES;
SHOW TABLE <table>	Lists all the columns for a table.	TQL> SHOW TABLE "locations";

Syntax	Description	Examples
SCRIPT SERVER	Generates the TQL schema for all tables in all databases on the server.	TQL> SCRIPT SERVER;
SCRIPT DATABASE <database>	Generates the TQL schema for all tables in a database.	TQL> SCRIPT DATABASE "fruit_database";
SCRIPT TABLE <table>	Generates the TQL schema for a table.	TQL> SCRIPT TABLE "vendor";
<pre>SELECT <cols_or_expr> FROM <table_list> [WHERE <predicates>] [GROUP BY <expr>] [ORDER BY <expr>]</pre>	<p>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:</p> <p>--query_results _apply_top_row_count</p> <p>You can use the following aggregation functions:</p> <ul style="list-style-type: none"> • sum • count • count distinct • stddev • avg • variance • min • max <p>You can use the following date functions:</p> <ul style="list-style-type: none"> • absyear • absmonth • absday • absquarter • date • time 	TQL> SELECT TOP 10 "quantity" FROM "sales_fact"; TQL> SELECT COUNT(*) FROM "vendor"; TQL> SELECT "vendor", SUM("quantity") FROM "sales_fact" GROUP BY "vendor"; TQL> SELECT "vendor", SUM("amount") FROM "vendor", "sales_fact" WHERE "sales_fact"."vendorid" = "vendor"."vendorid" AND "amount" > 100 GROUP BY "vendor" ORDER BY "amount" DESC; TQL> SELECT "vendor", SUM("quantity") FROM "sales_fact" GROUP BY "vendor" LIMIT 10;

Schema creation

Syntax	Description	Examples
CREATE DATABASE <database>	Creates a database.	TQL> CREATE DATABASE "fruit_database";
CREATE SCHEMA <schema>	Creates a schema within the current database.	TQL> CREATE SCHEMA "fruit_schema";
<pre>CREATE TABLE <table> (<column_definitions> [<constraints>])</pre>	<p>Creates a table with the specified column definitions and constraints.</p> <p>Use PARTITION BY HASH to shard a table across</p>	TQL> CREATE TABLE "vendor" ("vendorid" int, "name" varchar(255));

Syntax	Description	Examples
[PARTITION BY HASH (<number>) [KEY ("<column>")]]	<p>all nodes. If no KEY is specified, the table will be randomly sharded.</p> <p>Note that you can specify relationship constraints (FOREIGN KEY or RELATIONSHIP) in the CREATE TABLE statement. But it is recommended to define these using ALTER TABLE statements at the end of your TQL script, after creating your tables. This works better in scripts, because it guarantees that tables are created before they are referenced in the constraint definitions.</p>	<pre>TQL> 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");</pre>

Schema modification

Syntax	Description	Examples
DROP DATABASE <database>	Drops a database and all of its schemas and tables.	<pre>TQL> DROP DATABASE "fruit_database";</pre>
DROP SCHEMA <schema>	Drops a schema within the current database, and drops all of the tables in the schema.	<pre>TQL> DROP SCHEMA "fruit_schema";</pre>
DROP TABLE <table>	Drops a table.	<pre>TQL> DROP TABLE "location";</pre>
ALTER TABLE <table> ADD DROP RENAME COLUMN <column>	<p>Alters a table to add, drop, or rename a column.</p> <p>When you add a column to an existing table, you must provide a default value to use for existing rows.</p>	<pre>TQL> ALTER TABLE "cart" ADD COLUMN "nickname" varchar(255) DEFAULT 'no nickname'; TQL> ALTER TABLE "cart" DROP COLUMN "nickname"; TQL> ALTER TABLE "cart" RENAME COLUMN "nickname" TO "shortname";</pre>
TRUNCATE TABLE <table>	<p>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.</p> <p>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.</p>	<pre>TQL> TRUNCATE TABLE "location";</pre>

Syntax	Description	Examples
<pre>ALTER TABLE <table> DROP CONSTRAINT PRIMARY KEY;</pre>	<p>Drops the primary key from a table.</p> <p>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.</p>	<pre>TQL> ALTER TABLE "sales" DROP CONSTRAINT PRIMARY KEY;</pre> <pre>TQL> ALTER TABLE "sales" ADD CONSTRAINT PRIMARY KEY ("PO_number");</pre>
<pre>ALTER TABLE <table> DROP [FOREIGN KEY RELATIONSHIP] <name>;</pre>	<p>Drops the named foreign key or relationship between two tables.</p>	<pre>TQL> ALTER TABLE "sales_fact" DROP FOREIGN KEY "FK_PO_number";</pre> <pre>TQL> ALTER TABLE "fruit_dim" DROP RELATIONSHIP "REL_dates";</pre>
<pre>ALTER TABLE <table> DROP [CONSTRAINT FOREIGN KEY [<table_name>] RELATIONSHIP [WITH <table_name>];</pre>	<p>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.</p> <p>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:</p> <ul style="list-style-type: none"> the name of the referenced table, for a foreign key. the name of the related table, for a relationship. <p>If you drop a foreign key without specifying the referenced table, all foreign keys from the table you are altering will be dropped.</p>	<pre>TQL> ALTER TABLE "shipments" DROP CONSTRAINT FOREIGN KEY "orders";</pre> <pre>TQL> ALTER TABLE "wholesale_buys" DROP RELATIONSHIP WITH "retail_sales";</pre> <pre>/* Drops all relationships that have wholesale_buys as a source. */</pre> <pre>TQL> ALTER TABLE "wholesale_buys" DROP RELATIONSHIP;</pre> <pre>/* Drops all foreign keys from wholesale_buys. */</pre> <pre>TQL> ALTER TABLE "wholesale_buys" DROP CONSTRAINT FOREIGN KEY;</pre>
<pre>ALTER TABLE <table> [SET DIMENSION SET FACT [PARTITION BY HASH [(<shards>)] [KEY(<column>)]]]</pre>	<p>Changes the partitioning on a table by doing one of:</p> <ul style="list-style-type: none"> re-sharding a sharded table changing a replicated table to a sharded table changing a sharded table to a replicat- 	<pre>TQL> ALTER TABLE "sales_fact" SET FACT PARTITION BY HASH (96) KEY ("PO_number");</pre> <pre>TQL> ALTER TABLE</pre>

Syntax	Description	Examples
	<p>ed (unsharded) table</p> <p>By default, ThoughtSpot does not shard dimension tables.</p> <p>To change the partitioning on a table, or to change a dimension table to a sharded table, use <code>ALTER TABLE...SET FACT PARTITION BY HASH...;</code></p> <p>To make a sharded table into a dimension table (replicated on every node), use <code>ALTER TABLE...SET DIMENSION;</code> command.</p>	<pre>"fruit_dim" SET DIMENSION;</pre>
<code>ALTER TABLE <table> MODIFY COLUMN <column> <new_data_type>;</code>	Changes the data type of a column. This can have implications on sharding and primary key behavior. See About data type conversion .	<pre>TQL> ALTER TABLE fact100 MODIFY COLUMN product_id int;</pre>

Modify Schema

Syntax	Description	Examples
<code>INSERT INTO <table> VALUES ...</code>	Inserts values into a table. Only use this for testing. Do not use INSERT on a production system.	<pre>TQL> INSERT INTO "vendor" VALUES 'helen rose', 'jacob norse', 'eileen ruff', 'manny gates';</pre>
<code>UPDATE <table> ... SET ... [WHERE ...]</code>	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.	<pre>TQL> UPDATE "location" SET "borough" = 'staten island', "city" = 'new york' WHERE "borough" = 'staten isl' AND city = 'NY';</pre>
<code>DELETE FROM <table> [WHERE...]</code>	Deletes rows from a table that match optionally provided predicates. Predicates have the form column = value connected by the AND keyword.	<pre>TQL> DELETE FROM "vendor" WHERE "name" = 'Joey Smith' AND "vendorid" = '19463';</pre>

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	Examples
PRIMARY KEY	<p>Designates a unique, non-null value as the primary key for a table. This can be one column or a combination of columns.</p> <p>If values are not unique, an upsert will be performed if a row includes a primary key that is already present in the data.</p>	<pre>CREATE TABLE "schools" ("schoolID" varchar(15), "schoolName" varchar(255), "schoolCity" varchar(55), "schoolState" varchar(55), "schoolNick" varchar(55), CONSTRAINT PRIMARY KEY ("schoolID")) ;</pre> <pre>TQL> ALTER TABLE "cart" ADD CONSTRAINT PRIMARY KEY ("cart_id");</pre> <pre>TQL> ALTER TABLE "cart" DROP CONSTRAINT PRIMARY KEY "cart_id";</pre>
FOREIGN KEY	<p>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.</p> <p>When creating a foreign key, give it a name. You can reference the foreign key name later, if you want to remove it.</p>	<pre>TQL> ALTER TABLE "batting" ADD CONSTRAINT "FK_player" FOREIGN KEY ("playerID") REFERENCES "players" ("playerID");</pre> <pre>TQL> ALTER TABLE "batting" ADD CONSTRAINT "FK_lg_team" FOREIGN KEY ("lgID" ,"teamID") REFERENCES "teams" ("lgID" , "teamID");</pre> <pre>TQL> ALTER TABLE "shipment" ADD CONSTRAINT "FK_PO_vendor" FOREIGN KEY ("po_number", "vendor") REFERENCES "orders" ("po_number", "vendor");</pre> <pre>TQL> ALTER TABLE "shipment" DROP CONSTRAINT "FK_PO_vendor";</pre>
RELATIONSHIP	<p>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.</p> <p>You may add multiple relationships between tables.</p> <p>When creating a relationship, give it a name. You can reference the relationship name later, if you want to remove it.</p>	<pre>TQL> 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" <</pre>

Syntax	Description	Examples
		<pre>"wholesale_buys"."expire_date"); TQL> ALTER TABLE "wholesale_buys" DROP RELATIONSHIP "REL_fruit";</pre>

Flags

The `--query_results_apply_top_row_count <number>` flag can be used with TQL to limit the number of result rows returned by a query. For example:

```
$ tq1 --query_results_apply_top_row_count 100
```

Data types

ThoughtSpot supports a simplified list of data types:

Syntax	Description	Examples
Character	<ul style="list-style-type: none"> • <code>VARCHAR(<i>n</i>)</code> 	Specify the maximum number of characters, as in <code>VAR-CHAR(255)</code> . The size limit is 1GB for <code>VARCHAR</code> values.
Floating point	<ul style="list-style-type: none"> • <code>DOUBLE</code> • <code>FLOAT</code> 	<code>DOUBLE</code> is recommended.
Boolean	<ul style="list-style-type: none"> • <code>BOOL</code> 	Can be <code>true</code> or <code>false</code> .
Integer	<ul style="list-style-type: none"> • <code>INT</code> • <code>BIGINT</code> 	<p><code>INT</code> holds 32 bits.</p> <p><code>BIGINT</code> holds 64 bits.</p>
Date or time	<ul style="list-style-type: none"> • <code>DATE</code> • <code>DATETIME</code> • <code>TIMESTAMP</code> • <code>TIME</code> 	<p><code>DATETIME</code>, <code>TIMESTAMP</code>, and <code>TIME</code> are stored at the granularity of seconds.</p> <p><code>TIMESTAMP</code> is identical to <code>DATETIME</code>, but is included for syntax compatibility.</p>

ThoughtSpot loader flag reference

For recurring data loads and for scripting loads, use the ThoughtSpot Loader (tsload). 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 strftime 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 strftime 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 strftime library function .
--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

Flag	Description	Notes
		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.
--dateConvertedToEpoch [true false]	Specifies whether the "date" or "datetime" values in the input file are represented as epoch values.	
--booleanRepresentation [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:--booleanRepresentation Y_NULL
--has_header_row	Indicates that the input file contains a header row.	If supplied, the first row of the file is ignored. If not supplied, the first row of the file is loaded as data.
--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:

Flag	Description	Notes
--use_bit_boolean_values = [true false]	Specifies how boolean values are represented in the input file.	--enclosing_character "\\" 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 nodes, 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] [--identity_file identity_file]
      {alert,backup,backup-policy,callhome,cluster,command,dr-mirror,etl,event,
       feature,fileserver,firewall,hdfs,ldap,logs,map-tiles,monitoring,nas,node,
       os,saml,scheduled-pinboards,smtp,snapshot,snapshot-policy,spot,ssl,storage,
       support}
```

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.

alert subcommand

```
tscli alert [-h] {count,info,list,off,on,refresh,silence,status,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.
- `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 subcommand

```
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 different 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,show,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 Support every six hours via the secure file server. Before using this command for the first time, you need to set up the file server connection using `tscli fileserver configure`.

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.

cluster

```
tscli cluster [-h]
{abort-reinstall-os,check,create,get-config,load,reinstall-os,report,restore,resume-reinstall-os,resume-update,set-config,set-min-resource-spec,show-resource-spec,start,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.

entity

```
tscli entity [-h] {pack} ...
```

Creates a serialized, dataless object file for testing, troubleshooting, or migration. You can use this command with answer, pinboard, or aggregated/unaggregated worksheet objects.

- `tscli entity pack [-h] --id ID [--outdir FULLPATHNAME]` Packs object metadata and schema into a file. Packed filenames have the format XXX.YYY where XXX is ### ? optional arguments:
 - `--id ID` The required ID of the object to pack. IDs are found in the URL of an answer, pinboard, or aggregated/unaggregated worksheet. For example, the ID for a pinboard <http://thoughtspot.com:8088/#/pinboard/>

- 061457a2-27bc-43a9-9754-0cd873691bf0/ is
- 061457a2-27bc-43a9-9754-0cd873691bf0.
- o --outdir FULLPATHNAME Directory where the command places the packed object.

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:
- o --admin_username *admin_user* specifies the Administrator username for ThoughtSpot.
 - o --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:

- o --username *USERNAME* Username for Informatica Cloud (default: None)
- o --thoughtspot_url *THOUGHTSPOT_URL* URL to reach thoughtspot. (default: None)
- o --admin_username *ADMIN_USERNAME* Admin username for ThoughtSpot (default: None)
- o --groupname *GROUPNAME*
- o --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
- o --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.
- o --proxy_host *PROXY_HOST* Proxy server host for network access (default:)
- o --proxy_port *PROXY_PORT* Proxy server port (default:)
- o --proxy_username *PROXY_USERNAME* Proxy server username (default:)
- o --max_wait *MAX_WAIT* Maximum time in seconds to wait for Data Connect agent to 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-config,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, status}
```

- `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*`
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.
```

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: yyyy-mm-dd-HH:MM (default: None)
- `--to TO` Timestamp where collection ends, must be of the form: yyyy-mm-dd-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 than 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-translated 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-translated 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: yyyy-mm-dd-HH:MM (default: None)
- **--to TO** Timestamp where collection ends, must be of the form: yyyy-mm-dd-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}
```

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.

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,status}
```

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)

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.

smtp

```
tscli smtp [-h]
{reset-canonical-mapping, set-canonical-mapping, set-mailfromname, set-mailname, set-relayhost, show-canonical-mapping, show-mailfromname, show-mailname, show-relayhost}
```

This subcommand takes supports the following actions:

- `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 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,restore,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` parameter is the number of days after which this snapshot will be automatically deleted. A value of `-1` disables automatic deletion.

- `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 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}
```

This subcommand supports the following actions:

```
tscli spot [-h] {enable} Enables Spot integration.
```

ssl

```
tscli ssl [-h]
{add-cert,clear-min-tls-version,off,on,rm-cert,set-min-tls-version,status,tls-status}
```

status Shows whether of SSL authentication is enabled or disabled for the ThoughtSpot application.
tls-status Prints the status of TLS support

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.

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]
{restart-remote,rm-admin-email,rm-admin-phone,set-admin-email,set-admin-phone,set-remote,show-admin-email,show-admin-phone,show-remote,start-remote,stop-remote}
```

This subcommand supports the following actions:

- 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-admin-phone Removes the phone number for contacting the customer administrator. Replaces it with the default ThoughtSpot Support phone number.
- 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-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. `tunnel.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-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.

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
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

Format mask	Description
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, if 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
<code>to_bool</code>	Returns the input as a boolean (true or false).	<code>to_bool (0) = false</code> <code>to_bool (married)</code>
<code>to_date</code>	Accepts a date represented as an integer or text string, and a second string parameter that can include strftime date formatting elements. Replaces all the valid strftime date formatting elements with their string counterparts and returns the result. Does not accept epoch formatted dates as input.	<code>to_date (date_sold, '%Y-%m-%d')</code>
<code>to_double</code>	Returns the input as a double.	<code>to_double ('3.14') = 3.14</code> <code>to_double (revenue * .01)</code>
<code>to_integer</code>	Returns the input as an integer.	<code>to_integer ('45') + 1 = 46</code> <code>to_integer (price + tax - cost)</code>
<code>to_string</code>	Returns the input as a text string.	<code>to_string (45 + 1) = '46'</code> <code>to_string (revenue - cost)</code>

Date functions

Function	Description	Examples
<code>add_days</code>	Returns the result of adding the specified number of	<code>add_days (01/30/2015, 5) = 02/04/2015</code> <code>add_days (invoiced, 30)</code>

Function	Description	Examples
	days to the given date.	
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_week	Returns the number (1-7) of the day in a week for the given date with 1 being Monday and 7 being Sunday.	day_number_of_week (01/30/2015) = 6 day_number_of_week (shipped)
day_number_of_year	Returns the number (1-366) of the day in a year for the given date.	day_number_of_year (01/30/2015) = 30 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)
month_number	Returns the number (1-12) of the month for the given date.	month_number (09/20/2014) = 9 month_number (purchased)
now	Returns the current time-stamp.	now ()
start_of_month	Returns `MMM yyyy` for	start_of_month (01/31/2015) = Jan

Function	Description	Examples
	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 if you .	FY 2015 <code>start_of_month (shipped)</code>
<code>start_of_quarter</code>	Returns the date for the first day of the quarter for the given date.	<code>start_of_quarter (09/18/2015) = Q3 FY 2015</code> <code>start_of_quarter (sold)</code>
<code>start_of_week</code>	Returns the date for the first day of the week for the given date.	<code>start_of_week (06/01/2015) = 05/30/2015 Week</code> <code>start_of_week (emailed)</code>
<code>start_of_year</code>	Returns the date for the first day of the year for the given date.	<code>start_of_year (02/15/2015) = FY 2015</code> <code>start_of_year (joined)</code>
<code>time</code>	Returns the time portion of a given date.	<code>time (3/1/2002 10:32) = 10:32</code> <code>time (call began)</code>
<code>year</code>	Returns the year from the given date.	<code>year (01/15/2014) = 2014</code> <code>year (date ordered)</code>

Mixed functions

These functions can be used with text and numeric data types.

Function	Description	Examples
<code>!=</code>	Returns true if the first value is not equal to the second value.	<code>3 != 2 = true</code> <code>revenue != 1000000</code>
<code><</code>	Returns true if the first value is less than the second value.	<code>3 < 2 = false</code> <code>revenue < 1000000</code>
<code><=</code>	Returns true if the first value is less than or equal to the second value.	<code>1 <= 2 = true</code> <code>revenue <= 1000000</code>
<code>=</code>	Returns true if the first value is equal to the second value.	<code>2 = 2 = true</code> <code>revenue = 1000000</code>
<code>></code>	Returns true if the first value is greater than the second value.	<code>3 > 2 = true</code> <code>revenue > 1000000</code>
<code>>=</code>	Returns true if the first value is greater	<code>3 >= 2 = true</code>

Function	Description	Examples
	than or equal to the second value.	<code>revenue >= 1000000</code>
<code>greatest</code>	Returns the larger of the values.	<code>greatest (20, 10) = 20</code> <code>greatest (q1 revenue, q2 revenue)</code>
<code>least</code>	Returns the smaller of the values.	<code>least (20, 10) = 10</code> <code>least (q1 revenue, q2 revenue)</code>

Number functions

Function	Description	Examples
<code>?</code>	Returns the result of multiplying both numbers.	<code>3 * 2 = 6</code> <code>price * taxrate</code>
<code>+</code>	Returns the result of adding both numbers.	<code>1 + 2 = 3</code> <code>price + shipping</code>
<code>-</code>	Returns the result of subtracting the second number from the first.	<code>3 - 2 = 1</code> <code>revenue - tax</code>
<code>/</code>	Returns the result of dividing the first number by the second.	<code>6 / 3 = 2</code> <code>markup / retail price</code>
<code>^</code>	Returns the first number raised to the power of the second.	<code>3 ^ 2 = 9</code> <code>width ^ 2</code>
<code>abs</code>	Returns the absolute value.	<code>abs (-10) = 10</code> <code>abs (profit)</code>
<code>acos</code>	Returns the inverse cosine in degrees.	<code>acos (0.5) = 60</code> <code>acos (cos-satellite-angle)</code>
<code>asin</code>	Returns the inverse sine (specified in degrees).	<code>asin (0.5) = 30</code> <code>asin (sin-satellite-angle)</code>
<code>atan</code>	Returns the inverse tangent in degrees.	<code>atan (1) = 45</code> <code>atan (tan-satellite-angle)</code>
<code>atan2</code>	Returns the inverse tangent in degrees.	<code>atan2 (10, 10) = 45</code> <code>atan2 (longitude, latitude)</code>
<code>cbrt</code>	Returns the cube root of a number.	<code>cbrt (27) = 3</code> <code>cbrt (volume)</code>
<code>ceil</code>	Returns the smallest following integer.	<code>ceil (5.9) = 6</code> <code>ceil (growth rate)</code>

Function	Description	Examples
<code>cos</code>	Returns the cosine of an angle (specified in degrees).	<code>cos (63) = 0.45</code> <code>cos (beam angle)</code>
<code>cube</code>	Returns the cube of a number.	<code>cube (3) = 27</code> <code>cube (length)</code>
<code>exp</code>	Returns Euler's number (~2.718) raised to a power.	<code>exp (2) = 7.38905609893</code> <code>exp (growth)</code>
<code>exp2</code>	Returns 2 raised to a power.	<code>exp2 (3) = 8</code> <code>exp2 (growth)</code>
<code>floor</code>	Returns the largest previous integer.	<code>floor (5.1) = 5</code> <code>floor (growth rate)</code>
<code>ln</code>	Returns the natural logarithm.	<code>ln (7.38905609893) = 2</code> <code>ln (distance)</code>
<code>log10</code>	Returns the logarithm with base 10.	<code>log10 (100) = 2</code> <code>log10 (volume)</code>
<code>log2</code>	Returns the logarithm with base 2 (binary logarithm).	<code>log2 (32) = 5</code> <code>log2 (volume)</code>
<code>mod</code>	Returns the remainder of first number divided by the second number.	<code>mod (8, 3) = 2</code> <code>mod (revenue, quantity)</code>
<code>pow</code>	Returns the first number raised to the power of the second number.	<code>pow (5, 2) = 25</code> <code>pow (width, 2)</code>
<code>random</code>	Returns a random number between 0 and 1.	<code>random () = .457718</code> <code>random ()</code>
<code>round</code>	Returns the first number rounded to the second number (the default is 1).	<code>round (35.65, 10) = 40</code> <code>round (battingavg, 100)</code>
<code>safe_divide</code>	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).	<code>safe_divide (12, 0) = 0</code> <code>safe_divide (total_cost, units)</code>
<code>sign</code>	Returns +1 if the number is greater than zero, -1 if less than zero, 0 if zero.	<code>sign (-250) = -1</code> <code>sign (growth rate)</code>
<code>sin</code>	Returns the sine of an angle (specified in degrees).	<code>sin (35) = 0.57</code> <code>sin (beam angle)</code>

Function	Description	Examples
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'
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.	<pre>concat ('hay' , 'stack') = 'haystack' concat (last_name , first_name)</pre>
contains	Returns true if the first string contains the second string, otherwise returns false.	<pre>contains ('broomstick', 'room') = true contains (product, 'trial version')</pre>
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.	<pre>edit_distance ('attorney', 'atty') = 4 edit_distance (color, 'red')</pre>
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.	<pre>edit_distance_with_cap ('pokemon go', 'minecraft pixelmon', 3) = 4 edit_distance_with_cap (event, 'burning man', 3)</pre>
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.	<pre>similar_to ('hello world', 'hello swirl') = true similar_to (current team, drafted by)</pre>
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	<pre>similarity ('where is the burning man concert', 'burning man') = 46 similarity (tweet1, tweet2)</pre>

Function	Description	Examples
	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.	
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.	<code>spells_like ('thouhgtpot', 'thoughtspot') = true spells_like (studio, distributor)</code>
strlen	Returns the length of the text.	<code>strlen ('smith') = 5 strlen (lastname)</code>
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.	<code>strpos ('haystack_with_needles', 'needle') = 14 strpos (complaint, 'lawyer')</code>
substr	Returns the portion of the given string, beginning at the location specified (starting from 0), and of the given length.	<code>substr ('persnickety', 3, 7) = snicket substr (lastname, 0, 5)</code>

Variables

These variables can be used in your expressions.

Function	Description	Examples
ts_groups	Returns the list 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.	<code>ts_groups = east</code>

Formula reference

ThoughtSpot allows you to create derived columns in worksheets using formulas. This reference lists the various operators and functions you can use to create formulas.

You can also see this list of operators and examples from within the Formula Builder by selecting **Formula Assistant**.

Aggregate functions

These functions can be used to aggregate data.

Function	Description	Examples
average	Returns the average of all the values of a column.	average (revenue)
count	Returns the number of rows in the table containing the column.	count (product)
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_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)

Function	Description	Examples
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)
min	Returns the minimum value of a column.	min (revenue)
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)

Function	Description	Examples
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.	<code>moving_sum (revenue, 1, 1, order date)</code>
stddev	Returns the standard deviation of all values of a column.	<code>stddev (revenue)</code>
sum	Returns the sum of all the values of a column.	<code>sum (revenue)</code>
unique count	Returns the number of unique values of a column.	<code>unique count (customer)</code>
variance	Returns the variance of all the values of a column.	<code>variance (revenue)</code>

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).	<code>to_bool (0) = false to_bool (married)</code>
to_date	Accepts a date represented as an integer or text string, and a second string parameter that can include strftime date formatting elements. Replaces all the valid strftime date formatting elements with their string counterparts and returns the result. Does not accept epoch formatted dates as input.	<code>to_date (date_sold, '%Y-%m-%d')</code>
to_double	Returns the input as a double.	<code>to_double ('3.14') = 3.14 to_double (revenue * .01)</code>
to_integer	Returns the input as an integer.	<code>to_integer ('45') + 1 = 46 to_integer (price + tax - cost)</code>
to_string	Returns the input as a text string.	<code>to_string (45 + 1) = '46' to_string (revenue - cost)</code>

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_week	Returns the number (1-7) of the day in a week for the given date with 1 being Monday and 7 being Sunday.	day_number_of_week (01/30/2015) = 6 day_number_of_week (shipped)
day_number_of_year	Returns the number (1-366) of the day in a year for the given date.	day_number_of_year (01/30/2015) = 30 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
<code>month_number</code>	Returns the number (1-12) of the month for the given date.	<code>month_number (09/20/2014) = 9</code> <code>month_number (purchased)</code>
<code>now</code>	Returns the current time-stamp.	<code>now ()</code>
<code>start_of_month</code>	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 if you .	<code>start_of_month (01/31/2015) = Jan FY 2015</code> <code>start_of_month (shipped)</code>
<code>start_of_quarter</code>	Returns the date for the first day of the quarter for the given date.	<code>start_of_quarter (09/18/2015) = Q3 FY 2015</code> <code>start_of_quarter (sold)</code>
<code>start_of_week</code>	Returns the date for the first day of the week for the given date.	<code>start_of_week (06/01/2015) = 05/30/2015 Week</code> <code>start_of_week (emailed)</code>
<code>start_of_year</code>	Returns the date for the first day of the year for the given date.	<code>start_of_year (02/15/2015) = FY 2015</code> <code>start_of_year (joined)</code>
<code>time</code>	Returns the time portion of a given date.	<code>time (3/1/2002 10:32) = 10:32</code> <code>time (call began)</code>
<code>year</code>	Returns the year from the given date.	<code>year (01/15/2014) = 2014</code> <code>year (date ordered)</code>

Mixed functions

These functions can be used with text and numeric data types.

Function	Description	Examples
<code>!=</code>	Returns true if the first value is not equal to the second value.	<code>3 != 2 = true</code> <code>revenue != 1000000</code>
<code><</code>	Returns true if the first value is less than the second value.	<code>3 < 2 = false</code> <code>revenue < 1000000</code>
<code><=</code>	Returns true if the first value is less than or equal to the second value.	<code>1 <= 2 = true</code> <code>revenue <= 1000000</code>

Function	Description	Examples
=	Returns true if the first value is equal to the second value.	<code>2 = 2 = true</code> <code>revenue = 1000000</code>
>	Returns true if the first value is greater than the second value.	<code>3 > 2 = true</code> <code>revenue > 1000000</code>
>=	Returns true if the first value is greater than or equal to the second value.	<code>3 >= 2 = true</code> <code>revenue >= 1000000</code>
greatest	Returns the larger of the values.	<code>greatest (20, 10) = 20</code> <code>greatest (q1 revenue, q2 revenue)</code>
least	Returns the smaller of the values.	<code>least (20, 10) = 10</code> <code>least (q1 revenue, q2 revenue)</code>

Number functions

Function	Description	Examples
?	Returns the result of multiplying both numbers.	<code>3 * 2 = 6</code> <code>price * taxrate</code>
+	Returns the result of adding both numbers.	<code>1 + 2 = 3</code> <code>price + shipping</code>
-	Returns the result of subtracting the second number from the first.	<code>3 - 2 = 1</code> <code>revenue - tax</code>
/	Returns the result of dividing the first number by the second.	<code>6 / 3 = 2</code> <code>markup / retail price</code>
^	Returns the first number raised to the power of the second.	<code>3 ^ 2 = 9</code> <code>width ^ 2</code>
abs	Returns the absolute value.	<code>abs (-10) = 10</code> <code>abs (profit)</code>
acos	Returns the inverse cosine in degrees.	<code>acos (0.5) = 60</code> <code>acos (cos-satellite-angle)</code>
asin	Returns the inverse sine (specified in degrees).	<code>asin (0.5) = 30</code> <code>asin (sin-satellite-angle)</code>
atan	Returns the inverse tangent in degrees.	<code>atan (1) = 45</code> <code>atan (tan-satellite-angle)</code>
atan2	Returns the inverse tangent in degrees.	<code>atan2 (10, 10) = 45</code> <code>atan2 (longitude, latitude)</code>

Function	Description	Examples
<code>cbrt</code>	Returns the cube root of a number.	<code>cbrt (27) = 3</code> <code>cbrt (volume)</code>
<code>ceil</code>	Returns the smallest following integer.	<code>ceil (5.9) = 6</code> <code>ceil (growth rate)</code>
<code>cos</code>	Returns the cosine of an angle (specified in degrees).	<code>cos (63) = 0.45</code> <code>cos (beam angle)</code>
<code>cube</code>	Returns the cube of a number.	<code>cube (3) = 27</code> <code>cube (length)</code>
<code>exp</code>	Returns Euler's number (~2.718) raised to a power.	<code>exp (2) = 7.38905609893</code> <code>exp (growth)</code>
<code>exp2</code>	Returns 2 raised to a power.	<code>exp2 (3) = 8</code> <code>exp2 (growth)</code>
<code>floor</code>	Returns the largest previous integer.	<code>floor (5.1) = 5</code> <code>floor (growth rate)</code>
<code>ln</code>	Returns the natural logarithm.	<code>ln (7.38905609893) = 2</code> <code>ln (distance)</code>
<code>log10</code>	Returns the logarithm with base 10.	<code>log10 (100) = 2</code> <code>log10 (volume)</code>
<code>log2</code>	Returns the logarithm with base 2 (binary logarithm).	<code>log2 (32) = 5</code> <code>log2 (volume)</code>
<code>mod</code>	Returns the remainder of first number divided by the second number.	<code>mod (8, 3) = 2</code> <code>mod (revenue, quantity)</code>
<code>pow</code>	Returns the first number raised to the power of the second number.	<code>pow (5, 2) = 25</code> <code>pow (width, 2)</code>
<code>random</code>	Returns a random number between 0 and 1.	<code>random () = .457718</code> <code>random ()</code>
<code>round</code>	Returns the first number rounded to the second number (the default is 1).	<code>round (35.65, 10) = 40</code> <code>round (battingavg, 100)</code>
<code>safe_divide</code>	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).	<code>safe_divide (12, 0) = 0</code> <code>safe_divide (total_cost, units)</code>
<code>sign</code>	Returns +1 if the number is greater than 0, -1 if less than 0, and 0 if equal to 0.	<code>sign (-250) = -1</code>

Function	Description	Examples
	than zero, -1 if less than zero, 0 if zero.	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'
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.	<pre>concat ('hay' , 'stack') = 'haystack' concat (last_name , first_name)</pre>
contains	Returns true if the first string contains the second string, otherwise returns false.	<pre>contains ('broomstick', 'room') = true contains (product, 'trial version')</pre>
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.	<pre>edit_distance ('attorney', 'atty') = 4 edit_distance (color, 'red')</pre>
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.	<pre>edit_distance_with_cap ('pokemon go', 'minecraft pixelmon', 3) = 4 edit_distance_with_cap (event, 'burning man', 3)</pre>
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.	<pre>similar_to ('hello world', 'hello swirl') = true similar_to (current team, drafted by)</pre>
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	<pre>similarity ('where is the burning man concert', 'burning man') = 46 similarity (tweet1, tweet2)</pre>

Function	Description	Examples
	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.	
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.	<pre>spells_like ('thouhgtspot', 'thoughtspot') = true spells_like (studio, distributor)</pre>
strlen	Returns the length of the text.	<pre>strlen ('smith') = 5 strlen (lastname)</pre>
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.	<pre>strpos ('haystack_with_needles', 'needle') = 14 strpos (complaint, 'lawyer')</pre>
substr	Returns the portion of the given string, beginning at the location specified (starting from 0), and of the given length.	<pre>substr ('persnickety', 3, 7) = snicket substr (lastname, 0, 5)</pre>

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-100	INFO	Success. {1} has been added to {2}. # {1} - name of visualization # {2} - {name/link to pinboard}	None	None
TS-101	ERROR	Failure adding {1} to {2}	Visualization could not be added to {2} # {1} - name of visualization # {2} - name/link to pinboard	None
TS-102	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-103	INFO	Success. Visualization has been deleted from {1}. 1 - name/link to pinboard	None	None
TS-104	ERROR	Failure deleting visual from {1}	Visualization could not be deleted from the pinboard. 1 - name/link to pinboard	None
TS-105	ERROR	Failure deleting visual from {1} due to corruption	Visualization could not be deleted from {1} as the pinboard has one or more	Please try again after removing the invalid visualization(s) from the pin-

Code	Severity	Summary	Details	Action
			invalid visualizations. 1 - name/link to pinboard	board
TS-106	INFO	Success. {1} created successfully. 1 - name/link to pin-board	None	None
TS-107	ERROR	Failure creating {1}. 1 - name/link to pin-board	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-108	INFO	Sticker created successfully.	None	None
TS-109	ERROR	Failure creating the sticker.	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-110	INFO	Sticker deleted successfully.	None	None
TS-111	ERROR	Failure deleting sticker.	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-112	INFO	Pinboards deleted successfully.	None	None
TS-113	ERROR	Failure deleting pin-boards	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-114	INFO	Answers deleted successfully.	None	None
TS-115	ERROR	Failure deleting answers	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-116	INFO	Tables deleted successfully.	None	None
TS-117	ERROR	Failure deleting tables	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None

Code	Severity	Summary	Details	Action
minEmail}.				
TS-118	INFO	Relationship created successfully.	None	None
TS-119	ERROR	Failure creating relationship	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-120	INFO	Relationship updated successfully.	None	None
TS-121	ERROR	Failure updating the relationship	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-122	INFO	Relationship deleted successfully.	None	None
TS-123	ERROR	Failure deleting the relationship	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-124	ERROR	Failure fetching details for table	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-125	ERROR	Failure fetching details for the tables	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-126	ERROR	Failure fetching details for datasource	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-127	ERROR	Failure fetching details for datasources	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-128	ERROR	Failure fetching details for metadata items	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None

Code	Severity	Summary	Details	Action
TS-129	ERROR	Failure opening the answer	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-130	ERROR	Failure opening the pinboard	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-131	ERROR	Failure opening the worksheet	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-132	INFO	Table saved successfully.	None	None
TS-133	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-134	INFO	Visualization update successful	None	None
TS-135	ERROR	Visualization failed to update	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-136	INFO	{1} saved 1 - name of answer	None	None
TS-137	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-138	INFO	{1} saved 1 - name of pinboard / link	None	None
TS-139	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-140	INFO	{1} saved 1 - name of worksheet	None	None
TS-141	ERROR	{1} could not be	Uh oh. We're not sure	None

Code	Severity	Summary	Details	Action
		saved 1 - name of worksheet	what happened. Please email the trace file to {adminEmail}.	
TS-142	INFO	{1} saved 1 - name of answer	None	None
TS-143	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-144	INFO	{1} saved 1 - name/link to pinboard	None	None
TS-145	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-146	INFO	Worksheet saved	None	None
TS-147	ERROR	Worksheet could not be saved	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-148	INFO	Sticker updated	None	None
TS-149	ERROR	The sticker could not be updated	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-150	INFO	Successfully assigned sticker	None	None
TS-151	ERROR	The sticker could not be assigned	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-152	INFO	Successfully unassigned sticker	None	None
TS-153	ERROR	The sticker could not be unassigned	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None

Code	Severity	Summary	Details	Action
TS-154	ERROR	Failed to fetch metadata list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-155	ERROR	Failed to fetch table list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-156	ERROR	Failed to fetch relationship list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-157	ERROR	Failed to fetch answer list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-158	ERROR	Failed to fetch pin-board list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-159	ERROR	Failed to fetch worksheet list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-160	ERROR	Failed to fetch aggregated worksheet list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-161	ERROR	Failed to fetch imported data list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-162	ERROR	Failed to fetch system table list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-163	ERROR	Failed to DB view list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-164	ERROR	Failed to fetch data source list	Uh oh. We're not sure what happened. Please	None

Code	Severity	Summary	Details	Action
			email the trace file to {adminEmail}.	
TS-165	ERROR	Failed to fetch column list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-166	ERROR	Failed to label list	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-167	ERROR	Failed to fetch answer	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-168	ERROR	Failed to fetch worksheet	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-169	INFO	Aggregated worksheet {1} created 1 - name of aggregated worksheet	None	None
TS-170	ERROR	Failure creating Aggregated Worksheet.	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-171	INFO	{1} updated 1 - name of aggregated worksheet	None	None
TS-172	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-173	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-174	ERROR	Comments cannot be fetched	Failed to save client state	None
TS-175	ERROR	Comment cannot	Uh oh. We're not sure	None

Code	Severity	Summary	Details	Action
		be created	what happened. Please email the trace file to {adminEmail}.	
TS-176	ERROR	Comment cannot be updated	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-177	ERROR	Comment cannot be deleted	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-178	INFO	Rule saved successfully	None	None
TS-179	ERROR	Rule could not be saved	We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-180	INFO	Rule deleted successfully	None	None
TS-181	ERROR	Rule could not be deleted	We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-182	INFO	Item deleted successfully.	None	None
TS-183	ERROR	Item could not be deleted.	We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-184	INFO	Related link created successfully.	None	None
TS-185	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-186	INFO	Related link updated successfully.	None	None
TS-187	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 ad-	None

Code	Severity	Summary	Details	Action
ministrator, {adminEmail}.				
TS-188	INFO	Related link deleted successfully.	None	None
TS-189	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-190	INFO	Related link detail fetched successfully.	None	None
TS-191	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-500	ERROR	Failed to fetch leaf level data	Failed to fetch leaf level data.	None
TS-501	ERROR	Failed to fetch excel data	Failed to fetch excel data.	None
TS-502	ERROR	Failed to fetch visualization data	Failed to fetch visuzliation data.	None
TS-503	ERROR	Failed to fetch visualizations data	Failed to fetch data for visualizations.	None
TS-504	ERROR	Failed to fetch chart data	Failed to fetch table data.	None
TS-505	ERROR	Failed to fetch table data	Failed to fetch table data.	None
TS-506	ERROR	Failed to fetch worksheet data	Failed to fetch worksheet data.	None
TS-507	ERROR	Failed to fetch filter data	Failed to fetch filter data.	None

Code	Severity	Summary	Details	Action
TS-508	ERROR	Failed to fetch headline data	Failed to fetch filter data.	None
TS-509	ERROR	Failed to fetch natural query	Failed to fetch natural query.	None
TS-510	INFO	File upload successful	None	None
TS-511	ERROR	Failed to upload file	Failed to upload	None
TS-512	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-700	ERROR	Failure fetching table dependents	Failed to fetch dependents for the table.	None
TS-701	ERROR	Failure fetching column dependents	Failed to fetch dependents for the column.	None
TS-702	ERROR	Failure fetching incomplete items	Failed to fetch incomplete items.	None

Admin Service Errors (800 - 899)

Code	Severity	Summary	Details	Action
TS-800	ERROR	Failure fetching MemCache stats	Failed to fetch MemCache stats.	None
TS-801	ERROR	Failure MemCache Clear	Failed to clear MemCache.	None
TS-802	ERROR	Failure searching from MemCache	Failed to search from MemCache.	None
TS-803	ERROR	Failure fetching Loggers	Failed to fetch Loggers.	None
TS-804	ERROR	Failure setting LogLevel	Failed to set Log Level.	None
TS-805	ERROR	Failure getting debug info	Failed to get debug info.	None

Code	Severity	Summary	Details	Action
TS-806	INFO	Memcache cleared successfully	None	None
TS-807	INFO	Log level set successfully	None	None
TS-808	ERROR	Failed to report problem	None	None
TS-809	INFO	Problem reported successfully	None	None

Permissions Errors (900 - 999)

Code	Severity	Summary	Details	Action
TS-900	ERROR	Failure fetching table permissions	Failed to fetch table permissions.	None
TS-901	ERROR	Failure fetching answer permissions	Failed to fetch answer permissions.	None
TS-902	ERROR	Failure fetching pinboard permissions	Failed to fetch pinboard permissions.	None
TS-903	ERROR	Failure getting metadata permissions	Failed to get metadata permissions.	None

Import Data Errors (1000 - 1099)

Code	Severity	Summary	Details	Action
TS-1000	ERROR	Data caching failed	Data caching failed.	None
TS-1001	ERROR	Read Columns failed.	Failed to read columns.	None
TS-1002	ERROR	Failed to read keys.	Failed to read keys.	None
TS-1003	ERROR	Failed to read relationships.	Failed to read relationships.	None
TS-1004	ERROR	Failed to load data.	Failed to load data.	None
TS-1005	ERROR	Failed to create table.	Failed to create table.	None
TS-1006	ERROR	Failed to fetch data rows.	Failed to fetch data rows.	None
TS-1007	ERROR	Failed to delete files.	Failed to fetch data rows.	None
TS-1008	ERROR	Failed to abort create table.	Failed to abort create table.	None

Code	Severity	Summary	Details	Action
TS-1009	ERROR	Failed to create schema.	Failed to create schema.	None
TS-1010	ERROR	Failed to fetch table models.	Failed to fetch table models.	None
TS-1011	ERROR	Failed to fetch sample values.	Failed to fetch sample values.	None

Scheduled Jobs Errors (1100 - 1199)

Code	Severity	Summary	Details	Action
TS-1100	INFO	The list of jobs.	None	Please click 'Report Problem' to email a report to your administrator.
TS-1110	INFO	Successfully created job.	None	None
TS-1111	ERROR	The job could not be created.	None	Please click 'Report Problem' to email a report to your administrator.
TS-1112	INFO	Successfully updated job.	None	None
TS-1113	ERROR	The job could not be updated.	None	Please click 'Report Problem' to email a report to your administrator.
TS-1114	INFO	Successfully deleted jobs.	None	None
TS-1115	ERROR	The job could not be deleted.	None	Please click 'Report Problem' to email a report to your administrator.
TS-1116	INFO	The job was paused.	None	None
TS-1117	ERROR	The job could not be paused.	None	Please click 'Report Problem' to email a report to your administrator.
TS-1118	INFO	The job was resumed	None	None
TS-1119	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-1200	ERROR	Failed to fetch users list	Failed to fetch users list	None
TS-1201	ERROR	Failed to fetch groups list	Failed to fetch groups list	None
TS-1202	ERROR	Failed to fetch users and groups list	Failed to fetch users and groups list	None
TS-1203	ERROR	Successfully created user	Successfully created user	None
TS-1204	ERROR	Failed to create user	Failed to create user	None
TS-1205	ERROR	Successfully created group	Successfully created group	None
TS-1206	ERROR	Failed to create group	Failed to create group	None
TS-1207	ERROR	Successfully updated user	Successfully updated user	None
TS-1208	ERROR	Failed to update user	Failed to update user	None
TS-1209	ERROR	Successfully updated users	Successfully updated users	None
TS-1210	ERROR	Failed to update users	Failed to update users	None
TS-1211	ERROR	Successfully updated group	Successfully updated group	None
TS-1212	ERROR	Failed to update group	Failed to update group	None
TS-1213	ERROR	Successfully updated password	Successfully updated password	None
TS-1214	ERROR	Failed to update password	Failed to update password	None
TS-1215	ERROR	Successfully deleted users	Successfully deleted users	None
TS-1216	ERROR	Failed to delete users	Failed to delete users	None
TS-1217	ERROR	Successfully deleted groups	Successfully deleted groups	None
TS-1218	ERROR	Failed to delete groups	Failed to delete groups	None
TS-1219	ERROR	Successfully assigned users to groups	Successfully assigned users to groups	None
TS-1220	ERROR	Failed to assign users to groups	Failed to assign users to groups	None

Code	Severity	Summary	Details	Action
TS-1221	ERROR	Failed to fetch profile pic	Failed to fetch profile pic	None
TS-1222	INFO	Successfully uploaded profile pic	None	None
TS-1223	ERROR	Failed to upload profile pic	Failed to upload profile pic	None
TS-1224	ERROR	Successfully assigned groups to group	Failed to assign user to group	None
TS-1228	ERROR	Successfully created role	Successfully created role	None
TS-1229	ERROR	Failed to create role	Failed to create role	None
TS-1230	ERROR	Successfully deleted role	Successfully deleted role	None
TS-1231	ERROR	Failed to delete role	Failed to delete role	None
TS-1232	ERROR	Successfully updated role	Successfully updated role	None
TS-1233	ERROR	Failed to update role	Failed to update role	None

Session Service Errors (1400 - 1599)

Code	Severity	Summary	Details	Action
TS-1400	ERROR	Failed to fetch session info	Failed to fetch session info	None
TS-1401	ERROR	Failed to login	Uh oh. We're not sure what happened. Please email the trace file to {adminEmail}.	None
TS-1402	ERROR	Failed to logout	Failed to logout	None
TS-1403	ERROR	Failed to save client state	Failed to save client state	None
TS-1404	ERROR	Failed to fetch login config	Failed to fetch login config	None
TS-1405	ERROR	Failed to fetch slack config	Failed to fetch slack config	None
TS-1406	ERROR	Health check failed	Health check failed	None
TS-1407	ERROR	Failed to fetch health portal token	Failed to fetch health portal token	None
TS-1408	ERROR	The health portal release	Uh oh. We're not sure what happened.	None

Code	Severity	Summary	Details	Action
		name could not be retrieved	Please email the trace file to {adminEmail}.	

Data Management Service Errors (1600 - 1799)

Code	Severity	Summary	Details	Action
TS-1600	ERROR	Failed to fetch data source types	Failed to fetch data source types	None
TS-1601	ERROR	Failed to fetch data source sample values	Failed to fetch data source sample values	None
TS-1602	ERROR	Failed to delete data source	Failed to delete data source	None
TS-1603	ERROR	Failed to execute DDL	Failed to execute DDL	None
TS-1604	ERROR	Failed to update schedule	Failed to update schedule	None
TS-1605	ERROR	Failed to reload tasks	Failed to reload tasks	None
TS-1606	ERROR	Failed to stop tasks	Failed to stop tasks	None
TS-1607	ERROR	Failed to get creation DDL	Failed to get creation DDL	None
TS-1608	ERROR	Failed to load from data source	Failed to load from data source	None
TS-1609	ERROR	Failed to create connection to data source	Failed to create connection to data source	None
TS-1610	ERROR	Failed to create data source	Failed to create data source	None
TS-1611	ERROR	Failed to connect to data source	Failed to connect to data source	None
TS-1612	ERROR	Failed to get data source connection field info	Failed to get data source connection field info	None
TS-1613	ERROR	Failed to get connection list for data source	Failed to get connection list for data source	None
TS-1614	ERROR	Failed to get connection attributes for data source	Failed to get connection attributes for data source	None
TS-1615	ERROR	Failed to get connections	Failed to get connections to data	None

Code	Severity	Summary	Details	Action
to data source		source		
TS-1616	ERROR	Failed to fetch data source config	Failed to fetch data source config	None
TS-1617	ERROR	Failed to parse sql.	Failed to parse sql.	None
TS-1618	ERROR	Failed to execute sql.	Failed to execute sql.	None
TS-1619	INFO	Successfully created connection to data source	None	None
TS-1620	INFO	Successfully updated data upload schedule	None	None
TS-1621	ERROR	Failed to execute sql.	Please check the failing command, executed {1} statements successfully.	None
TS-1622	ERROR	Lightweight data-cache disabled	Lightweight data-cache disabled	None
TS-1623	INFO	Selected tables were queued for loading.	Selected tables were queued for loading.	None
TS-1624	ERROR	DataType conversion error.	No mapping found for source datatype to ThoughtSpot datatype.	None
TS-1625	INFO	Successfully reload task started.	None	None
TS-1626	INFO	Successfully connected to data source.	None	None
TS-1627	INFO	Successfully created data source.	None	None
TS-1628	INFO	Successfully stopped the tasks.	None	None
TS-1629	INFO	Successfully deleted the connection.	None	None
TS-1630	ERROR	There was an error deleting this connection.	None	None
TS-1631	INFO	Successfully executed the DDL.	None	None

Cluster Status Service Errors (1800 - 1899)

Code	Severity	Summary	Details	Action
TS-1800	WARNING	Failed to fetch cluster information from search service.	None	None
TS-1801	WARNING	Failed to fetch table detail information from search service.	None	None
TS-1802	WARNING	Failed to fetch cluster information from database service.	None	None
TS-1803	WARNING	Failed to fetch table detail information from database service.	None	None
TS-1804	WARNING	Failed to fetch cluster information from cluster management service.	None	None
TS-1805	WARNING	Failed to fetch detail information from cluster management service.	None	None
TS-1806	WARNING	Failed to fetch log from cluster management service.	None	None
TS-1807	WARNING	Failed to fetch snapshot list from cluster management service.	None	None
TS-1808	WARNING	Failed to fetch cluster information from alert management service.	None	None
TS-1809	WARNING	Failed to fetch cluster information from event service.	None	None
TS-1810	WARNING	Failed to fetch alerts information from alert management service.	None	None
TS-1811	WARNING	Failed to fetch events information from alert management service.	None	None
TS-1812	INFO	Thanks for your feedback!	None	None
TS-1813	WARNING	Sorry! Unable to submit the feedback at this moment!	None	None
TS-1814	INFO	Successfully exported objects. File can be found at {1}.	None	None

Code	Severity	Summary	Details	Action
TS-1815	ERROR	Sorry! Unable to export objects at this moment!	What happened? {1}.	None
TS-1816	INFO	Successfully imported objects	None	None
TS-1817	ERROR	Sorry! Unable to import objects at this moment!	What happened? {1}.	None
TS-1818	INFO	Successfully deleted data source object(s).	None	None

Callosum API Errors (9000 - 9199)

Code	Severity	Summary	Details	Action
TS-9000	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-9001	ERROR	Uh oh. We're not sure what happened.	Please email the trace file to {adminEmail}.	None
TS-9002	ERROR	Could not authorize user	Try logging in again	None
TS-9003	ERROR	Uh oh. We're not sure what happened.	Please email the trace file to {adminEmail}.	None
TS-9004	WARNING	Still loading data, come back soon	None	None
TS-9005	ERROR	Uh oh. We're having trouble getting data for this request.	Please email the trace file to {adminEmail}.	None
TS-9006	ERROR	Uh oh. We're having trouble getting data for this request.	Please email the trace file to {adminEmail}.	None
TS-9007	ERROR	Uh oh. We're having trouble getting data for this request.	Please email the trace file to {adminEmail}.	None
TS-9008	ERROR	Something went	Uh oh. We're not sure what hap-	None

Code	Severity	Summary	Details	Action
		wrong with your search	pened. Please email the trace file to {adminEmail}.	
TS-9009	ERROR	The calculation engine has timed out. Please try again.	Please email the trace file to {adminEmail}.	None
TS-9010	ERROR	Cannot open Object	Object cannot be opened due to errors in some of its dependencies	None
TS-Blink Generated Errors (9500 - 9599)				
TS-9500	WARNING	Cannot connect to the calculation engine. Please try again soon.	None	None
TS-9501	WARNING	The calculation engine has timed out. Please try again.	None	None
TS-9502	WARNING	Cannot connect to the search engine. Please try again soon.	None	None
TS-9503	WARNING	The search engine has timed out. Please try again.	None	None
TS-9504	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-9505	WARNING	We're still indexing this data, try again soon	None	None
TS-9506	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-9507	ERROR	ThoughtSpot is unreachable.	None	None

Code	Severity	Summary	Details	Action
Please try again soon				

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 unavailable.	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 cannot be found in store	Object with Id: {0} of type: {1} not found. # {0} – identity of the object # {1} – type of object	None
TS-13004	ERROR	The object is in an invalid state	Object with Id: {0} of type: {1} in invalid state. # {0} – identity of the object # {1} – type of object	None
TS-13005	ERROR	Object already	Object with Id: {0} of	None

Code	Severity	Summary	Details	Action
		exists	type: {1} already exists. # {0} - identity of the object # {1} - type of object	
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 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
TS-13018	ERROR	Cyclic relationship	Detected cycles: {0}. # {0} - cycle details	None

Code	Severity	Summary	Details	Action
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