



---

# Reference Guide

*Version 6.0 December 02, 2019*

Copyright for ThoughtSpot publications. © 2019 ThoughtSpot, Inc. All rights reserved.

ThoughtSpot, Inc. 910 Hermosa Ct  
Sunnyvale, CA 94085

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

<b>Keyword reference .....</b>	<b>2</b>
<b>TQL reference .....</b>	<b>10</b>
<b>tsload flag reference .....</b>	<b>23</b>
<b>tscli command reference .....</b>	<b>26</b>
<b>Date and time formats reference .....</b>	<b>71</b>
<b>Row level security rules reference .....</b>	<b>75</b>
<b>Formula function reference .....</b>	<b>87</b>
<b>Alert codes reference .....</b>	<b>103</b>
<b>User action codes reference .....</b>	<b>112</b>
<b>Frequently asked questions .....</b>	<b>114</b>

# Keyword reference

You can use keywords when asking a question to help define your search. This reference lists the various keywords. You can also see this list of keywords and examples from within the help center.

## Keywords in Other Languages

Currently, we offer the following keyword translations.

日本語	中文 (简体)	Deutsche	Español (latín)	Español (España)
Français (Canada)	Français (France)	Português (Brasil)	Português (Portugal)	Italiano
Dansk	Suomi	Svenska	Norsk	Nederland

Also, see the topic on how to set [locale preferences in your user profile](#) to control language, date, and number formats on the ThoughtSpot UI.

## General

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

# Date

Keyword	Examples
after	order date <b>after</b> 10/31/2014
before	order date <b>before</b> 03/01/2014
between ... and ...	order date <b>between</b> 01/30/2012 and 01/30/2014
daily year-over-year	growth of revenue by order date <b>daily year-over-year</b>
daily	shipments by region <b>daily</b>
day	count <b>monday</b> restaurant
day of week	revenue by <b>day of week</b> last 6 months
day of week	count shipments <b>Monday</b>
n days for each month	sales last <b>2 days for each month</b>
n days for each quarter	revenue last <b>15 days for each quarter</b>
n days for each week	total sold last <b>2 days for each week</b>
n days for each year	revenue last <b>300 days for each year</b>
growth of ... by ... daily	<b>growth of sales by</b> order date <b>daily</b>
growth of ... by ... monthly	<b>growth of sales by</b> date shipped <b>monthly</b> sales > 24000
growth of ... by ... quarterly	<b>growth of sales by</b> date shipped <b>quarterly</b>
growth of ... by ... weekly	<b>growth of sales by</b> receipt date <b>weekly</b> for pro-ski2000
growth of ... by ... yearly	<b>growth of sales by</b> date closed <b>yearly</b>
growth of ... by ...	<b>growth of sales by</b> order date
n hours for each day	sales last <b>2 hours for each day</b>

Keyword	Examples
last day by	customers <b>last day by</b> referrer
last month by	customers <b>last month by</b> day
last <i>n</i> days	visitors <b>last 7 days</b>
last <i>n</i> quarters	visitors <b>last 2 quarters</b> by month by campaign
last <i>n</i> weeks	visitors <b>last 10 weeks</b> by day
last quarter	customers <b>last quarter</b> sale > 300
last week	customers <b>last week</b> by store
last year	top 10 customers <b>last year</b> by sale by store for region west
last <i>n</i> years	visitors <b>last 5 years</b> by revenue for sum revenue > 5000
month to date	sales by product <b>month to date</b> sales > 2400
<i>month year</i>	commission by sales rep <b>February 2014</b>
<i>month</i>	commission <b>January</b>
month	revenue by <b>month</b> last year
monthly year-over-year	growth of revenue by receipt date <b>monthly year-over-year</b>
monthly	commission > 10000 <b>monthly</b>
<i>n</i> months for each quarter	cost <b>last 2 months for each quarter</b>
<i>n</i> months for each year	<b>last 8 months for each year</b>
<i>n</i> days ago	<b>sales 2 days ago</b>
<i>n</i> months ago	<b>sales 2 months ago</b> by region
<i>n</i> months	visitors <b>last 6 months</b> for homepage visits > 30 by month
<i>n</i> quarters ago	<b>sales 4 quarters ago</b> by product name contains deluxe
<i>n</i> weeks ago	<b>sales 4 weeks ago</b> by store
<i>n</i> years ago	<b>sales 5 years ago</b> by store for region west
<i>n</i> years	opportunities <b>next 5 years</b> by revenue

Keyword	Examples
next day	shipments <b>next day</b> by order
next month	appointments <b>next month</b> by day
next <i>n</i> days	shipments <b>next 7 days</b>
next <i>n</i> months	openings <b>next 6 months</b> location
next <i>n</i> quarters	opportunities <b>next 2 quarters</b> by campaign
next <i>n</i> weeks	shipments <b>next 10 weeks</b> by day
next quarter	opportunities <b>next quarter</b> amount > 30000
next week	shipments <b>next week</b> by store
next year	opportunities <b>next year</b> by sales rep
quarter to date	sales by product <b>quarter to date</b> for top 10 products by sales
quarterly year-over-year	growth of revenue by date shipped <b>quarterly year-over-year</b>
quarterly	sales <b>quarterly</b> for each product
<i>n</i> quarters for each year	<b>last 2 quarters</b> for each year
today	sales <b>today</b> by store
week to date	sales by order date <b>week to date</b> for pro-ski200
week	revenue by <b>week</b> last quarter
weekly year-over-year	growth of revenue by date shipped <b>weekly year-over-year</b>
weekly	revenue <b>weekly</b>
<i>n</i> weeks for each month	sales <b>last 3 weeks for each month</b>
<i>n</i> weeks for each quarter	<b>last 2 weeks for each quarter</b>
<i>n</i> weeks for each year	<b>last 3 weeks for each year</b>
year to date	sales by product <b>year to date</b>
year	revenue by product <b>2014</b> product name contains snowboard

Keyword	Examples
yearly	shipments by product <b>yearly</b>
yesterday	sales <b>yesterday</b> for pro-ski200 by store

## Time

Keyword	Examples
detailed	ship time <b>detailed</b>
last minute	count homepage views <b>last minute</b>
last hour	count unique visits <b>last hour</b>
<i>n</i> minutes	count visitors last <b>30 minutes</b>
<i>n</i> hours	count visitors last <b>12 hours</b>
hourly	visitors by page name <b>hourly</b>
<i>n</i> minutes ago	sum inventory by product <b>10 minutes ago</b>
<i>n</i> hours ago	sum inventory by product by store <b>2 hours ago</b>

## Text

Keyword	Examples
begins with	product name <b>begins with</b> 'pro'
contains	product name contains "alpine" description <b>contains</b> "snow shoe"
ends with	product name <b>ends with</b> 'deluxe'

Keyword	Examples
not begins with	product name <b>not begins with</b> "tom's"
not contains	product color <b>not contains</b> 'tan' product color <b>not contains</b> 'red'
not ends with	product name <b>not ends with</b> "trial"
similar to	course name <b>similar to</b> 'hand'
not similar to	course name <b>not similar to</b> 'hand'

## Number

Function	Examples
sum	<b>sum</b> revenue
average	<b>average</b> revenue by store
count	<b>count</b> visitors by site
max	<b>max</b> sales by visitor by site
min	<b>min</b> revenue by store by campaign for cost > 5000
standard deviation	<b>standard deviation</b> revenue by product by month for date after 10/31/2010
unique count	<b>unique count</b> visitor by product page last week
variance	<b>variance</b> sale amount by visitor by product for last year

## Comparative

Function	Examples
all	<b>all</b>

Function	Examples
between... and	revenue <b>between 0 and 1000</b>
vs, versus	revenue east <b>vs</b> west
>	sum sale amount by visitor by product for last year sale amount > 2000
<	unique count visitor by product by store for sale amount < 20
>=	count calls by employee lastname >= m
<=	count shipments by city latitude <= 0
=	unique count visitor by store purchased products = 3 for last 5 days
!=	sum sale amount region != canada region != mexico

## Location

Keyword	Examples
near	revenue store name county <b>near</b> san francisco
near... within <i>n</i> miles km meters	revenue store name county <b>near</b> alameda <b>within 50</b> <b>miles</b>
farther than <i>n</i> miles km meters from	average hours worked branch <b>farther than 80 km</b> from scarborough

Location keywords only work for searches where the data source includes latitude/longitude data.

## Period

Keyword	Example
quarter ( <i>date</i> )	<b>quarter</b> (purchase date)
quarter of year ( <i>date</i> )	<b>quarter of year</b> (purchase date)

month of quarter ( <i>date</i> )	<b>month of quarter</b> (purchase date)
week of year ( <i>date</i> )	<b>week of year</b> (ship date)
week of quarter ( <i>date</i> )	<b>week of quarter</b> (ship date)
week of month ( <i>date</i> )	<b>week of month</b> (ship date)
day of year ( <i>date</i> )	<b>day of year</b> (ship date)
day of quarter ( <i>date</i> )	<b>day of quarter</b> (ship date)
day ( <i>date</i> )	<b>day</b> (ship date)
day of month ( <i>date</i> )	<b>day of month</b> (order date)
day of week ( <i>date</i> )	<b>day of week</b> (order date)
hour ( <i>datetime</i> )	<b>hour</b> (timestamp)

## In

Keyword	Example
in ( <i>subsearch</i> )	<b>in</b> (top 10 store name by sales footwear)

# TQL reference

TQL is the ThoughtSpot language for entering SQL commands. This reference lists TQL commands you can use to do things like creating a schema or verifying a data load.

## About using TQL

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

Use `--query_results_apply_top_row_count <number>` flag to limit the number of result rows returned by a query. For example:

```
$ tql --query_results_apply_top_row_count 100
```

As a best practice, you should enclose object names (database, schema, table, and column) in double quotes, and column values in single quotes. When referring to objects using fully qualified object names, the syntax is:

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

To get help on SQL when using TQL, enter `help` on the command line.

You can use TQL to view and modify schemas and data in tables. Remember to add a semicolon after each command. Commands are not case sensitive but are capitalized here for readability.

**Note:** Worksheets and pinboards in ThoughtSpot are dependent upon the data in the underlying tables. Use caution when modifying tables directly. If you change or remove a schema on which those objects rely, the objects could become invalid.

## View schemas and data

Syntax	Description
SHOW DATABASES	Lists all available databases.  Examples:  SHOW DATABASES;
USE <database>	Switches the context to the specified database. This is required if queries do not use fully qualified names (database.schema.table) for specifying tables.  Examples:  USE "fruit_database";
SHOW SCHEMAS	Lists all schemas within the current database. For example:  SHOW SCHEMAS;
SHOW TABLES	Lists all tables within the current database by schema. For example:  SHOW TABLES;
SHOW TABLE <table>	Lists all the columns for a table. For example:  SHOW TABLE "locations";
SCRIPT SERVER	Generates the TQL schema for all tables in all databases on the server. For example:  SCRIPT SERVER;

Syntax	Description
<pre>SCRIPT DATABASE &lt;database&gt;</pre>	Generates the TQL schema for all tables in a database. For example:  <pre>SCRIPT DATABASE "fruit_database";</pre>

SCRIPT TABLE <table>	Generates the TQL schema for a table. For example:  <pre>SCRIPT TABLE "vendor";</pre>
-------------------------	---

Syntax	Description
	Shows specified set of table data.
<pre>SELECT     &lt;cols_or_expr&gt;     FROM     &lt;table_list&gt;         [WHERE &lt;predicates&gt;]         [GROUP BY &lt;expr&gt;]         [ORDER BY &lt;expr&gt;]</pre>	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 TSQL command line flag:  <code>--query_results apply_top_row_count</code>
	You can use the following aggregation functions:
	<ul style="list-style-type: none"> <li>• sum</li> <li>• count</li> <li>• count distinct</li> <li>• stddev</li> <li>• avg</li> <li>• variance</li> <li>• min</li> <li>• max</li> </ul>
	You can use the following date functions:
	<ul style="list-style-type: none"> <li>• absyear</li> <li>• absmonth</li> <li>• absday</li> <li>• absquarter</li> <li>• date</li> <li>• time</li> </ul>
	For example:
	<pre>SELECT TOP 10 "quantity" FROM "sales_fact";</pre>
	<pre>SELECT COUNT(*) FROM "vendor";</pre>
	<pre>SELECT "vendor", SUM("quantity") FROM "sales_fact" GROUP BY "vendor";</pre>
	<pre>SELECT "vendor", SUM("amount") FROM "vendor", "sales_fact"     WHERE "sales_fact"."vendorid" = "vendor"."vendorid"         AND "amount" &gt; 100 GROUP BY "vendor" ORDER BY "amount" DESC;</pre>
	<pre>SELECT "vendor", SUM("quantity") FROM "sales_fact" GROUP BY "vendor" LIMIT 10;</pre>

## Schema creation

Syntax	Description
<pre>CREATE DATABASE     &lt;data- base&gt;</pre>	<p>Creates a database. For example:</p> <pre>CREATE DATABASE "fruit_database";</pre>
<pre>CREATE SCHEMA &lt;schema&gt;</pre>	<p>Creates a schema within the current database. For example:</p> <pre>CREATE SCHEMA "fruit_schema";</pre>
<pre>CREATE TABLE &lt;table&gt; (&lt;column_def- itions&gt;     [&lt;con- straints&gt;]) [PARTI- TION BY HASH (&lt;num- ber&gt;) [KEY     ("&lt;col- umn&gt;")]]</pre>	<p>Creates a table with the specified column definitions and constraints.</p> <p>Use PARTITION BY HASH to shard a table across all nodes. If no KEY is specified, the table will be randomly sharded.</p> <p>Do not specify relationship constraints ( FOREIGN KEY or RELATIONSHIP ) in the CREATE TABLE statement. Instead, define these using ALTER TABLE statements at the end of your TQL script, after creating your tables. This method guarantees that tables are created before they are referenced in the constraint definitions. For example:</p> <pre>CREATE TABLE "vendor" ("vendorid" int, "name" var- char(255));  CREATE TABLE "sales_fact" ("saleid" int, "locationid" int, "vendorid" int, "quantity" int, "sale_amount" double, "fruitid" int, CONSTRAINT PRIMARY KEY("saleid")) PARTITION BY HASH(96) KEY ("saleid");</pre>

# Schema modification

Syntax	Description
<pre>DROP DATABASE       &lt;data- base&gt;</pre>	Drops a database and all of its schemas and tables. For example:  <pre>DROP DATABASE "fruit_database";</pre>
<pre>DROP SCHEMA &lt;schema&gt;</pre>	Drops a schema within the current database, and drops all of the tables in the schema. For example:  <pre>DROP SCHEMA "fruit_schema";</pre>
<pre>DROP TABLE &lt;table&gt;</pre>	Drops a table. For example:  <pre>DROP TABLE "location";</pre>
<pre>TRUNCATE TABLE &lt;table&gt;</pre>	Removes all data from a table, but preserves its metadata, including all GUIDs, relationships, etc. This can be used to force a new schema for a table without losing the metadata.  However, this operation removes all existing data from the table and must be used with caution. You must reload the data following a TRUNCATE , or all dependent objects (worksheets and pinboards) in ThoughtSpot will become invalid. For example:  <pre>TRUNCATE TABLE "location";</pre>

Syntax	Description
	Alters a table to add, drop, or rename a column.
ALTER TABLE <table> ADD   DROP   RENAME COLUMN <column>	When you add a column to an existing table, you must provide a default value to use for existing rows. For example:  ALTER TABLE "cart" ADD COLUMN "nickname" varchar(255) DE- FAULT 'no nickname';
	ALTER TABLE "cart" DROP COLUMN "nickname";
	ALTER TABLE "cart" RENAME COLUMN "nickname" TO "shortname";

ALTER TABLE <table> DROP CON- STRAINT PRIMARY KEY;	Drops the primary key from a table.  Note that if you then add a new primary key, the same upsert behavior will be applied as with adding any primary key. This can result in data deletion, so make sure you understand how the upsert will affect your data ahead of time. For example:
	ALTER TABLE "sales" DROP CONSTRAINT PRIMARY KEY;
	ALTER TABLE "sales" ADD CONSTRAINT PRIMARY KEY ("P0_num- ber");

ALTER TABLE <table> DROP CON- STRAINT [FOREIGN KEY  RELATIONSHIP] <name>;	Drops the named foreign key or relationship between two tables. For example:  ALTER TABLE "sales_fact" DROP CONSTRAINT FOREIGN KEY "FK_P0_number";
	ALTER TABLE "fruit_dim" DROP RELATIONSHIP "REL_dates";

Syntax	Description
<pre>ALTER TABLE &lt;table&gt; DROP [CONSTRAINT FOREIGN KEY [&lt;table_name&gt;]   RELATIONSHIP [WITH &lt;table_name&gt;];</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"> <li>• the name of the referenced table, for a foreign key.</li> <li>• the name of the related table, for a relationship.</li> </ul> <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>

Examples:

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

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

Drops all relationships that have wholesale\_buys as a source.

```
ALTER TABLE "wholesale_buys" DROP RELATIONSHIP;
```

Drops all foreign keys from wholesale\_buys.

```
ALTER TABLE "wholesale_buys" DROP CONSTRAINT FOREIGN KEY;
```

Syntax	Description
<pre>ALTER TABLE &lt;table&gt; [SET DI- MENSION   SET FACT [PARTITION BY     HASH [(&lt;shards&gt;)] [KEY(&lt;column&gt;)]]]</pre>	<p>Changes the partitioning on a table by doing one of:</p> <ul style="list-style-type: none"> <li>• re-sharding a sharded table</li> <li>• changing a replicated table to a sharded table</li> <li>• changing a sharded table to a replicated (unsharded) table</li> </ul> <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>
	<p>Examples of this statement:</p> <pre>ALTER TABLE "sales_fact" SET FACT PARTITION BY HASH (96) KEY     ("PO_number"); ALTER TABLE "fruit_dim" SET DIMENSION;</pre>

<pre>ALTER TABLE &lt;table&gt; MODIFY COLUMN &lt;column&gt; &lt;new_data_type&gt;;</pre>	<p>Changes the data type of a column. This can have implications on sharding and primary key behavior. See <a href="#">About data type conversion</a>. For example:</p> <pre>ALTER TABLE fact100 MODIFY COLUMN product_id int;</pre>
--	--

## Modify data

Syntax	Description
<pre>INSERT INTO &lt;table&gt; VALUES ...</pre>	<p>Inserts values into a table. Only use this for testing. Do not use <code>INSERT</code> on a production system. For example:</p> <pre>INSERT INTO "vendor" VALUES ('helen rose', 'jacob norse', 'eileen ruff', 'manny gates');</pre>

Syntax	Description
<pre>ALTER TABLE &lt;table&gt; SET LOAD PRIORITY &lt;value&gt; &lt;new_da- ta_type&gt;;</pre>	<p>Sets the load priority for a table. Load priority determines the order in which a table is loaded on a cluster restart. You can set any value from 1–100 . The system default for all tables is 50 . For example:</p> <pre>ALTER TABLE 'sales_facts' SET LOAD PRIORITY 1;</pre>
<pre>UPDATE &lt;table&gt; ... SET ... [WHERE ...]</pre>	<p>Updates rows in a table that match optionally provided predicates. Predicates have the form column = value connected by the AND keyword. Sets the column values to the specified values. For example:</p> <pre>UPDATE "location" SET "borough" = 'staten island', "city" = 'new york' WHERE "borough" = 'staten isl' AND city = 'NY';</pre>
<pre>DELETE FROM &lt;table&gt; [WHERE...]</pre>	<p>Deletes rows from a table that match optionally provided predicates. Predicates have the form column = value connected by the AND keyword. For example.</p> <pre>DELETE FROM "vendor" WHERE "name" = 'Joey Smith' AND "ven- dorid" = '19463';</pre>

## Constraints and relationships

Constraints and relationships in ThoughtSpot are used to define the relationships between tables (how they can be joined). However, constraints are not enforced, as they would be in a transactional database. You can define the following constraints when creating a table with `CREATE TABLE`, or add them to an existing table using the `ADD CONSTRAINT` syntax:

Syntax	Description
PRIMARY KEY	<p>Designates a unique, non-null value as the primary key for a table. This can be one column or a combination of columns. If values are not unique, an upsert will be performed if a row includes a primary key that is already present in the data. Some examples:</p> <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>ALTER TABLE "cart" ADD CONSTRAINT PRIMARY KEY ("cart_id");</pre> <pre>ALTER TABLE "cart" DROP CONSTRAINT PRIMARY KEY "cart_id";</pre>

#### FOREIGN KEY

Defines a relationship where the value(s) in the table are used to join to a second table. Uses an equality operator. The foreign key must match the primary key of the table that is referenced in number, column type, and order of columns.

When creating a foreign key, give it a name. You can reference the foreign key name later, if you want to remove it.

Examples of this statement:

```
ALTER TABLE "batting" ADD CONSTRAINT "FK_player" FOREIGN KEY ("playerID")
REFERENCES "players" ("playerID");
```

```
ALTER TABLE "batting" ADD CONSTRAINT "FK_lg_team" FOREIGN KEY ("lgID" , "teamID")
REFERENCES "teams" ("lgID" , "teamID");
```

```
ALTER TABLE "shipment" ADD CONSTRAINT "FK_P0_vendor" FOREIGN KEY ("po_number",
"vendor") REFERENCES "orders" ("po_number", "vendor");
```

```
ALTER TABLE "shipment" DROP CONSTRAINT "FK_P0_vendor";
```

Syntax	Description
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. When creating a relationship, give it a name. You can reference the relationship name later, if you want to remove it.</p> <p>Examples of this statement:</p> <pre>ALTER TABLE "wholesale_buys" ADD RELATIONSHIP "REL_fruit" WITH "retail_sales" AS "wholesale_buys"."fruit" = "retail_sales"."fruit" AND ("wholesale_buys"."date_order" &lt; "retail_sales"."date_sold" AND "retail_sales"."date_sold" &lt; "wholesale_buys"."expire_date");  ALTER TABLE "wholesale_buys" DROP RELATIONSHIP "REL_fruit";</pre>

## Data types

ThoughtSpot supports a simplified list of data types:

Syntax	Description	Examples
Character	<ul style="list-style-type: none"> <li>VARCHAR(<i>n</i>)</li> </ul>	Specify the maximum number of characters, as in VARCHAR(255). The size limit is 1GB for VARCHAR values.
Floating point	<ul style="list-style-type: none"> <li>DOUBLE</li> <li>FLOAT</li> </ul>	DOUBLE is recommended.
Boolean	<ul style="list-style-type: none"> <li>BOOL</li> </ul>	Can be true or false .

Syntax	Description	Examples
Integer	<ul style="list-style-type: none"><li>• INT</li><li>• BIGINT</li></ul>	<p>INT holds 32 bits.</p> <p>BIGINT holds 64 bits.</p>
Date or time	<ul style="list-style-type: none"><li>• DATE</li><li>• DATETIME</li><li>• TIMESTAMP</li><li>• TIME</li></ul>	<p>DATETIME, TIMESTAMP, and TIME are stored at the granularity of seconds</p> <p>.</p> <p>TIMESTAMP is identical to DATETIME, but is included for syntax compatibility.</p>

# tsload flag reference

For recurring data loads and for scripting loads, use `tsload` (the ThoughtSpot Loader). This reference section lists all the flags that can be used to modify the behavior of `tsload`.

## General tsload flags

Flag	Description	Notes
<code>--target_database &lt;database&gt;</code>	Specifies the pre-existing target database into which tsload should load the data.	
<code>--target_schema &lt;schema&gt;</code>	Specifies the target schema. Default is “falcon_default_schema”.	
<code>--target_table &lt;table&gt;</code>	Specifies the tables that you want to load into the database. The tables must exist in the database specified by <code>--target_database</code> .	
<code>--empty_target</code>	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 <code>--target_database</code> and <code>--target_table</code> will be deleted before this data load. To perform an “upsert” on the existing data, omit this flag or specify <code>--noempty_target</code> .
<code>--max_ignored_rows &lt;number&gt;</code>	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.
<code>--bad_records_file &lt;path_to_file&gt;/&lt;file_name&gt;</code>	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.	
<code>--date_format &lt;date_formatmask&gt;</code>	Specifies the format string for date values.	The default format is <code>yearmonthday</code> e.g. “Dec 30th, 2001” and is represented as <code>20011230</code> . Use the date format specifications supported in the <a href="#">strftime library function</a> .
<code>--date_time_format &lt;date_formatmask&gt;/&lt;time_formatmask&gt;</code>	Specifies the format string for datetime values.	The default is <code>yearmonthday hour:minute:second</code> e.g. Dec 30th, 2001 1:15:12 and is represented as <code>20011230 01:15:12</code> . Use the datetime format specifications supported in the <a href="#">strftime library function</a> .
<code>--time_format &lt;time_formatmask&gt;</code>	Specifies the format string for time values.	The default is <code>hour:minute:second</code> . Use the time format specifications supported in the <a href="#">strftime library function</a> .

Flag	Description	Notes
--v=[0 1 2 3]	Specifies the verbosity of log messages.	Provide a value for verbosity level. By default, verbosity is set to the minimum, which is 0. This value is similar to a volume control. At higher levels your log receives more messages and that log more frequently. This is used for debugging. You should not change this value unless instructed by ThoughtSpot Support.
--skip_second_fraction	Skips fractional seconds when loading data.	If supplied, the upserts logic may be affected, especially if the date time being loaded is a primary key, and the data has millisecond granularity. Load the data twice, first time as a string with a primary key, and again with second granularity date time. There is no support to store fractional seconds in the ThoughtSpot system.

## File loading tsload flags

The following flags are used when loading data from an input file:

Flag	Description	Notes
--source_file <path_to_file>/<file_name>	Specifies the location of the file to be loaded.	
--source_data_format [csv delimited]	Specifies the data file format.	Optional. The default is csv.
--field_separator "<delimiter>"	Specifies the field delimiter used in the input file.	
--trailing_field_separator	Specifies that the field separator appears after every field, including the last field per row.	Example row with trailing field separator: a,b,c,The default is false.
--null_value "<null_representation>"	Specifies how null values are represented in the input file.	These values will be converted to NULL upon loading.
--date_converted_to_epoch [true false]	Specifies whether the “date” or “datetime” values in the input file are represented as epoch values.	

Flag	Description	Notes
--boolean_representation [true_false   1_0   T_F   Y_N]	Specifies the format in which boolean values are represented in the input file.	The default is T_F. You can also use this flag to specify other values. For example, if your data used Y for true and NULL for false, you could specify: --boolean_representation Y_NULL
--has_header_row	Indicates that the input file contains a header row.	If supplied, column names in the header row are used to match column names in the target table in ThoughtSpot. If not supplied, the first row of the file is loaded as data, the same as all subsequent rows.
--escape_character "<character>"	Specifies the escape character used in the input file.	If no value is specified, the default is "(double quotes).
--enclosing_character "<character>"	Specifies the enclosing character used in the input file.	If the enclosing character is double quotes, you need to escape it, as in this example: --enclosing_character "\""
--use_bit_boolean_values = [true   false]	Specifies how boolean values are represented in the input file.	If supplied, the input CSV file uses a bit for boolean values, i.e. the false value is represented as 0x0 and true as 0x1. If omitted or set to false, boolean values are assumed to be T_F, unless you specify something else using the flag --boolean_representation [true_false   1_0   T_F   Y_N] .

# tscli command reference

**Summary:** The ThoughtSpot command line interface, or tscli, is an administration interface for the cluster. 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.

The command returns `0` on success, and a non-zero exit codes on failure. The `tscli` command logs errors to the `stderr` directory.

## How to use the tscli command

The `tscli` command has the following syntax:

```
tscli [-h]
      [--helpfull]
      [--verbose]
      [--noautoconfig]
      [--autoconfig]
      [--yes]
      [--cluster ]
      [--zoo ]
      {access, alert, ansible, backup, backup-policy, calendar, callhome, cassandra,
       cluster, command, dr-mirror, etl, event, feature, fileserver,
       firewall, hdfs, ipsec, ldap, logs, map-tiles, monitoring, nas,
       node, notification, onboarding, patch, rpackage, saml, scheduled-pinboards, set, smtp, snapshot,
       snapshot-policy, socialproof spot, sssd, ssl, storage, support,
       tokenauthentication}
```

The `tscli` command has several subcommands, such as `alert`, `backup`, and so on.

Issue subcommands using the following format:

```
tscli [subcommand]
```

Subcommands have their own additional options and actions, such as `tscli backup create` or `tscli backup delete`.

Each subcommand may have several options.

To view help for a subcommand, type `-h` after the subcommand option:

```
tscli [subcommand] -h
```

## tscli subcommands

This section lists each subcommand and its syntax.

### access

```
tscli access [-h] {list}
```

This subcommand has the following option:

**tscli access list**

Lists objects by last access time, with the following parameters:

**--type TYPE**

Type of object, either answer or pinboard.

**--limit LIMIT**

The number of objects to fetch.

The default is 30.

**--offset OFFSET**

Offset to use to skip objects for batched results.

The default is 0.

**--ascending**

Sorts the answers by access time ascending.

The default is `True`.

**alert**

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

This subcommand has the following options:

**tscli alert count**

Lists counts of generated alerts by type.

**tscli alert info**

Lists all alerts. Add `silenced` to list only silenced alerts, `active` to list only active alerts, or `detailed` to get detailed alert information.

**tscli alert list**

Lists the generated alerts, with these parameters:

**--limit LIMIT**

Specifies the number of recent alerts to display.

**--since SINCE**

Lists all alerts raised since a specified time period, in the form of a human readable duration string, such as `4h` (4 hours) or `4m` (4 minutes).

**tscli alert off**

Disables all alerts from the cluster in the cluster's timezone.

**tscli alert on**

Enables alerts from the cluster.

**tscli alert refresh**

Refreshes alert metadata on the cluster.

**tscli alert silence --name NAME**

Silences the alert with `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 NAME
```

Unsilences the alert with `NAME`. For example, `DISK_ERROR`.

## ansible

```
tscli ansible [-h] {checkout,commit} [--local]
```

This subcommand has the following options:

```
tscli ansible checkout --host HOST
```

Checks out Ansible playbook with the target `HOST` that is running the `ts_ansible` service.

```
tscli ansible commit --host HOST
```

Commits Ansible playbooks with the target `HOST` that is running the `ts_ansible` service.

Use this subcommand to install and configure third-party software on the ThoughtSpot cluster.

For details, see these articles:

- [About third party security and monitoring software](#)
- [Installing third party security and monitoring software](#)

## backup

```
tscli backup [-h] {create,delete,ls,restore}
```

This subcommand has the following options:

```
tscli backup create [-h] [--mode {full,light,dataless}] [--type {full,incremental}]  
[--base BASE]  
[--storage_type {local,nas}] [--remote] [--no-orion-master]
```

Pulls a snapshot and saves it as a backup, with these parameters:

```
--mode {full,light,dataless}
```

Mode of backups.

The default is `full`.

**--type {full,incremental}**

Type of backup.

**Note:** `incremental` is not implemented.

The default setting is `full`.

**--base BASE**

Based snapshot name for incremental backup.

**Note:** Because `incremental` is not implemented, neither is this option.

There is no default setting.

**--storage\_type {local,nas}**

Storage type of output directory.

The default setting is `local`.

**--remote**

Takes backup through orion master.

The default setting is `True`.

**--no-orion-master**

Determines whether orion master is available during backup.

The default is `False`.

**tscli backup delete name**

Deletes the named backup.

**tscli backup ls**

Lists all backups taken by the system.

**tscli backup restore**

Restores cluster using backup, with the following parameters:

**--release RELEASE**

Restore the cluster on a specific release number.

**--disable\_rotate\_keys**

Disables cluster rotate key configurations.

The default is `False`.

**--enable\_cloud\_storage**

Determines whether to enable Cloud Storage setup.

**--heterogeneous**

Should be set for heterogeneous clusters.

The default is `False`.

## backup-policy

```
tscli backup-policy [-h] {create,delete,disable,enable,ls,show,status,update}
```

Manages the backup policy.

This subcommand has the following options:

**tscli backup-policy create**

Prompts an editor for you to edit the parameters of a new periodic backup policy, with the following parameter:

**--config CONFIG**

Specifies the text format of the periodic backup policy config.

**tscli backup-policy delete name**

Deletes the backup policy `name`.

**tscli backup-policy disable name**

Disables the policy `name`.

**tscli backup-policy enable name**

Enables the policy `name`.

**tscli backup-policy ls**

Lists backup policies.

**tscli backup-policy show name**

Shows the backup policy `name`.

**tscli backup-policy status name**

Shows the status of the backup policy `name`.

**tscli backup-policy update name**

Prompts an editor for you to edit the backup policy `name`.

## calendar

```
tscli calendar [-h] {create,delete,disable,enable,generate,ge  
t,list,update}
```

This subcommand has the following options:

**tscli calendar create**

Creates a new custom calendar, with the following parameters:

**--file\_path FILE\_PATH**

Path to the CSV file holding custom calendar data.

The default is `None`.

**--name NAME**

Custom calendar name.

The default is `None`.

**--separator SEPARATOR**

The separator used in the CSV file.

The default is `,`.

**--no-header-row**

Flag to indicate that the CSV file has no header row.

The default is `True`.

**--username USERNAME**

The admin username for ThoughtSpot login.

The default is `None`.

**tscli calendar delete**

Deletes a custom calendar table from the system, with the following parameters:

**--name NAME**

Deletes the custom calendar NAME.

The default is `None`.

**--username USERNAME**

The admin username for ThoughtSpot login.

The default is `None`.

**tscli calendar disable**

Disables custom calendar on the cluster.

**tscli calendar enable**

Enables custom calendar on the cluster.

**tscli calendar generate**

Creates a custom calendar table based on given specifications, with the following parameters:

**--name NAME**

A name to create the custom calendar CSV file with.

The default is `None`.

**--start\_date START\_DATE**

The start date to begin the custom calendar with in the form mm/dd/yyyy.

The default is `None`.

**--end\_date END\_DATE**

The end date to end the custom calendar with in the form mm/dd/yyyy.

The default is `None`.

**--calendar\_type {MONTH\_OFFSET,4-4-5,4-5-4,5-4-4}**

The type of custom calendar to create.

The default is `MONTH_OFFSET`.

**--month\_offset**

`{January,February,March,April,May,June,July,August,September,October,November,December}`

The month offset to start the year from, if the calendar is the MONTH\_OFFSET type.

The default is `January`.

**--start\_day\_of\_week**

The day the week starts on.

The default is `Sunday`.

**--quarter\_name\_prefix**

The string to prefix a quarter name with.

**--year\_name\_prefix YEAR\_NAME\_PREFIX**

The string to prefix a year name with.

**--username USERNAME**

The admin username for ThoughtSpot login.

The default is `None`.

**tscli calendar get**

Procures data of a custom calendar as a CSV file, with the following parameters:

**--name NAME**

Procures data of custom calendar `NAME`

The default is `None`.

**--username USERNAME**

Admin username for ThoughtSpot login.

The default is `None`.

**tscli calendar list**

Procures a list of custom calendars present in the cluster, with the following parameter:

**--username USERNAME**

Admin username for ThoughtSpot login.

The default is `None`.

**tscli calendar update**

Updates a custom calendar table in the system, with the following parameters:

**--file\_path FILE\_PATH**

Path to the CSV file holding custom calendar data.

The default is `None`.

**--name NAME**

Custom calendar name.

The default is `None`.

**--separator SEPARATOR**

The separator used in the CSV file.

The default is `,`.

**--no-header-row**

Flag to indicate that the CSV file has no header row.

The default is `True`.

**--username USERNAME**

The admin username for ThoughtSpot login.

The default is `None`.

## callhome

```
tscli callhome [-h] {disable,enable,generate-bundle}
```

This subcommand has the following options:

**tscli callhome disable**

Turns off the periodic call home feature.

**tscli callhome enable --customer\_name CUSTOMER\_NAME**

Enables the "call home" feature, which sends usage statistics to ThoughtSpot.

This feature is enabled by default.

The parameter `customer_name` takes the form `Shared/CUSTOMER_NAME/stats`.

The default is `None`.

**tscli callhome generate-bundle [--d D] [--since SINCE]**

Generates the callhome stats tar file, with the following parameters:

**--d D**

Destination folder for the tar file.

There is no default setting.

**--since SINCE**

Grabs `callhome` data from the specified time window in the past.

This should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

This option generates a `tar` file of the cluster metrics and writes it to the specified directory, where `SINCE` is how many days back the file must start.

There is no default setting.

## cassandra

```
tscli cassandra [-h] {backup,restore}
```

Backs up cassandra.

This subcommand has the following options:

**tscli cassandra backup**

Takes a backup of cassandra, with the following parameters:

**--keyspaces KEYSPACES**

Comma separated list of keyspaces to take a backup of.

The default is `None`.

**backup\_dir BACKUP\_DIR**

The path to the backup directory to write the backup.

The default is `None`.

**tscli cassandra restore**

Restores cassandra from a backup, with the following parameter:

**--backup\_dir BACKUP\_DIR**

The path to the backup directory to write the backup.

The default is `None`.

## cluster

```
tscli cluster [-h] abort-reinstall-os,abort-update,bucket- name,check,create,download-release,get-config,list-available-releases,list-downloaded-releases,load,reinstall-os,restore,resume-reinstall-os,resume-update,set-config,set-min-resource-spec,setup-release-host,setup-release-host-key,show-resource-spec,start,status,stop,update,update-hadoop}
```

This subcommand has the following options:

**tscli cluster abort-reinstall-os**

Aborts in-progress reinstall.

**tscli cluster abort-update**

Aborts an ongoing cluster update, if safe.

**tscli cluster bucket-name**

Returns the name of the s3 bucket associated with the cluster, if there is one.

**tscli cluster check [--path PATH] [--includes INCLUDES] [--retry RETRY] [--localhost] [--disable-events]**

Checks the status of all nodes in the cluster, with the following parameters:

**--path PATH**

Specifies the working directory of the diagnostic tool.

The default is `/usr/local/scaligent/release`.

**--includes INCLUDES**

Specifies the comma-separated component(s) to be included in the check.

The default is `all`.

**--retry RETRY**

The maximum number of retry times if the node is unreachable.

The default is `10`.

**--localhost**

Runs cluster checks only on localhost.

The default is `False`.

**--disable-events**

Disables raising configuration events.

The default is `False`.

**tscli cluster create release**

Creates a new cluster from the release file specified by adding the release number.

Used by ThoughtSpot Support when installing a new cluster. For example, `tscli cluster create 5.3.2.tar.gz`.

This command has the following parameters:

**--disable\_rotate\_keys**

Disables cluster rotate key configuration.

The default is `False`.

**--enable\_cloud\_storage {s3a,gcs}**

Determines whether to enable Cloud Storage setup, and which storage format to use.

**heterogeneous**

Should be set for heterogeneous clusters.

The default is `False`.

**tscli cluster download-release release**

Downloads the specified release to the Hadoop Distributed File System (HDFS) for later upgrading.

**tscli cluster get-config**

Gets current cluster network and time configuration. Prints JSON configuration to `stdout`.

If the system cannot be connected to all interfaces, the command returns an error but continues to function.

This command has the following parameters:

**--local**

Gets the config for the local host only.

The default is `False`.

**--nodes NODES**

A comma separated list of specified nodes to get the config for.

The default is `None`.

**tscli cluster list-available-releases**

Lists the available releases to update to on the cluster.

**tscli cluster list-downloaded-releases**

Lists the releases downloaded to the cluster.

**tscli cluster load backupdir**

Loads the state from a specified backup directory onto an existing cluster.

**tscli cluster reinstall-os**

Reinstalls OS on all nodes of the cluster, with the following parameters:

**--secondary SECONDARY**

A secondary drive for reinstall.

The default is `sdd`.

**--stdin**

Command to take JSON configuration from stdin.

**tscli cluster restore --release RELEASE backupdir**

Restores a cluster on the specified release number 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.

This command has the following parameters:

**--disable\_rotate\_keys**

Disables cluster rotate key configurations.

The default is `False`.

**--enable\_cloud\_storage {s3a,gcs}**

Determines whether to enable Cloud Storage setup.

**--heterogenous**

Should be set for heterogenous clusters.

The default is `False`.

**tscli cluster resume-reinstall-os**

Resumes in-progress reinstall.

**tscli cluster resume-update**

Resumes in-progress updates, with the following parameter:

**--ignore\_if\_unhealthy**

Comma separated list of node IPs on which upgrade is not attempted if they are found to be unhealthy. If a node outside of this list is found unhealthy, the upgrade is aborted.

The default is `None`.

**tscli cluster set-config**

Sets cluster network and time configuration. Takes JSON configuration from stdin.

This subcommand has the following parameters:

**--ipv4-only**

Only use ipv4 for node communication. Requires passing ipMap in config unless no-network-change is also specified.

The default is `False`.

**--no-network-change**

This flag ensures that a change made with set-config does not update network settings.

The default is `False`.

**tscli cluster set-min-resource-spec**

Sets the minimum resource configuration of the cluster.

**tscli cluster show-resource-spec**

Prints default or min.

**tscli cluster start**

Starts the 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 on all nodes.

This subcommand has the following option:

```
tscli command run [-h] [--nodes NODES] --dest_dir DEST_DIR [--copyfirst COPYFIRST]
[--timeout TIMEOUT] command
```

These are the parameters:

**--nodes NODES**

Space-separated IPs of nodes where to run the command.

The default setting is `all`.

**--dest\_dir DEST\_DIR**

Directory to save the files that contain the output from each node.

This is a mandatory parameter.

**--copyfirst COPYFIRST**

Copy the executable to required nodes first.

The default setting is `False`.

**--timeout TIMEOUT**

Timeout waiting for the command to finish.

The default setting is `60`.

**dr-mirror****tscli dr-mirror [-h] {start,status,stop}**

This subcommand has the following options:

```
tscli dr-mirror start
```

Starts a mirror cluster which will continuously recover from a primary cluster.

```
tscli dr-mirror status
```

Checks whether the current cluster is running in mirror mode.

```
tscli dr-mirror stop
```

Stops mirroring on the local cluster.

## etl

```
tscli etl [-h] {change-password,disable-lw,download-agent,enabl  
e-lw,show-lw}
```

This subcommand has the following options:

```
tscli etl change-password --admin_username admin_user --username Informatica_user
```

Changes the Informatica Cloud account password used by ThoughtSpot Data Connect.

Required parameters are:

```
--admin_username admin_user
```

Specifies the Administrator username for ThoughtSpot.

```
--username Informatica_user
```

Specifies the username for the Informatica Cloud.

```
tscli etl disable-lw
```

Disables ThoughtSpot Data Connect.

```
tscli etl download-agent
```

Downloads the ThoughtSpot Data Connect agent to the cluster.

```
tscli etl enable-lw [-h] --username USERNAME --thoughtspot_url THOUGHTSPOT_URL --  
admin_username ADMIN_USERNAME [--groupname GROUPNAME] --org_id ORG_ID [--pin_to  
PIN_TO] [--proxy_host PROXY_HOST] [--proxy_port PROXY_PORT] [--proxy_username  
PROXY_USERNAME] [--max_wait MAX_WAIT]
```

Contact ThoughtSpot Support for assistance in setting this up.

Required parameters are:

**--username *USERNAME***

Username for Informatica Cloud

**--thoughtspot\_url *THOUGHTSPOT\_URL***

URL to reach thoughtspot.

**--admin\_username *ADMIN\_USERNAME***

Admin username for ThoughtSpot

**--groupname *GROUPNAME*****--org\_id *ORG\_ID***Specifies the Informatica `id` of the company.**--pin\_to *PIN\_TO***

Specifies the IP address to pin to. If you specify an IP to pin to, that node becomes sticky to the Informatica agent, and will always be used. Defaults to the public IP address of the localhost where this command was run.

**--proxy\_host *PROXY\_HOST***

Proxy server host for network access.

**--proxy\_port *PROXY\_PORT***

Proxy server port.

**--proxy\_username *PROXY\_USERNAME***

Proxy server username.

**--max\_wait *MAX\_WAIT***

Maximum time in seconds to wait for Data Connect agent to start.

**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 and its options manage event notifications.

This subcommand has the following option:

**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]
```

The `event` subcommand accepts these optional flags:

**--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, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

**--from FROM**

Begin timestamp. Must be of the form: `yyyymmdd-HH:MM`.

**--to TO**

End timestamp. Must be of the form: `yyyymmdd-HH:MM`.

**--limit LIMIT**

Maximum number of events to fetch.

The default setting is `0`.

**--detail**

Print events in detail format. This is not tabular. Default is a tabular summary.

The default setting is `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 `|` (pipe). The event is returned if it `matchesALL`. Put single quotes around the param value to prevent undesired glob expansion.

**--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 `|` (pipe). The event is returned if it `matches ALL`. Put single quotes around the param value to prevent undesired glob expansion.

**--attributes ATTRIBUTES**

Specify attributes to match as key=value. Multiple strings to check for can be specified by separating them with `|` (pipe). The event is returned if it `matches ALL`. Put single quotes around the param value to prevent undesired glob expansion.

## feature

```
tscli feature [-h] {get-all-config}
```

This subcommand has the following option:

**tscli feature get-all-config**

Gets the configured features in a cluster. The command returns a list of features, such as custom branding, Data Connect, and call home, and informs whether they are enabled or disabled.

**fileserver**

```
tscli fileserver [-h] {configure,download-release,purge-config,show-config,upload}
```

This subcommand has the following options:

**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 have to issue this command one time, to set up the connection to the secure file server. 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, including its checksum, and verifies the integrity of release bundle.

You must specify the exact release number, such as `5.1.3`.

Before using this command for the first time, you must set up the file server connection using `tscli fileserver configure`. You can then work with a member of the ThoughtSpot Support team because a privileged user and a corresponding password must be specified to download releases.

**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 specified file to the directory specified on the secure file server.

You may optionally specify the `user` and `password` to bypass the credentials specified when configuring the file server connection with `tscli fileservice configure`. Before using this command for the first time, you must set up the file server connection using `tscli fileservice configure`.

This uses the following flags:

**--user *USER***

Username of the fileserver.

**--password *PASSWORD***

Password of the fileserver. This is required and the command prompts you for it if you do not supply it.

**--file\_name *FILE\_NAME***

Local file to upload.

**--server\_dir\_path *SERVER\_DIR\_PATH***

Directory path on fileserver. The `SERVER_DIR_PATH` parameter specifies the directory for file upload. It is based on customer name, and takes the form `/Shared/support/customer_name`.

## firewall

```
tscli firewall [-h] {close-ports,disable,enable,open-ports,stat  
us}
```

This subcommand has the following options:

**tscli firewall close-ports**

Closes specified ports through firewall on all nodes.

Accepts a comma-separated list of ports. Only closes ports that were previously opened using `open-ports`, and ignores ports that were not opened with `open-port`, or closed ports.

Some essential ports are always kept open, such as `ssh`; they are not affected by this command or by `open-ports`.

**tscli firewall disable**

Disable firewall.

**tscli firewall enable**

Enable firewall.

**tscli firewall open-ports --ports *ports***

Opens specified ports through a firewall on all nodes.

Accepts a comma-separated list of ports.

Ignores open ports.

Some essential ports are always kept open, such as `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 option:

#### **tscli hdfs leave-safemode**

Command to get HDFS namenodes out of `safemode`.

## ipsec

```
tscli ipsec [-h] {disable,enable,status}
```

This subcommand has the following options:

#### **tscli ipsec disable**

Disable IPSec

#### **tscli ipsec enable**

Enable IPSec

#### **tscli ipsec status**

Show IPSec status on all nodes

## ldap

```
tscli ldap [-h] {add-cert,configure,purge-configuration}
```

This subcommand has the following options:

**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.

**tscli ldap purge-configuration**

Purges (removes) any existing LDAP configuration.

## logs

```
tscli logs [-h] {collect,runcmd}
```

Manages the logging behavior.

This subcommand has the following options:

```
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 it is interpreted through `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`.

**--since SINCE**

Grab logs from this time window in the past. Should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day). (default: None)

**--from FROM**

Timestamp where collection begins; must be of the form `yyyymmdd-HH:MM`.

**--to TO**

Timestamp where collection ends; must be of the form `yyyymmdd-HH:MM`.

**--out OUT**

Tarball path for writing logs from each node.

The default setting is `/tmp/logs.tar.gz`.

**--maxsize MAXSIZE**

Only fetch logs if size is smaller than this value. Can be specified in megabytes or gigabytes, such as `100MB`, `10GB`.

**--sizeonly**

Do not collect logs. Just report the size.

The default setting is `False`.

**--nodes NODES**

Comma separated list of nodes from where to collect logs. Skip this to use all nodes.

```
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.

Accepts these optional flags:

**--cmd CMD**

Unix-Command to be run on the selected logs. Use single quotes to escape spaces etc. Note the language used to specify CMDSTR has following rules.

- A logfile and its corresponding result file can be referred by keywords `SRCFILE` and `DSTFILE`. For example, `cp SRCFILE DSTFILE`.
- Without any reference to `DSTFILE` in CMDSTR, `DSTFILE` will be appended to CMDSTR for output redirection. For example, `du -sch SRCFILE` gets auto-transtalted to `du -sch SRCFILE > DSTFILE`.
- Without any reference to `SRCFILE`, content of log is streamed to CMDSTR by pipe. For example, `tail -n100 | grep ERROR` gets auto-transtalted to `cat SRCFILE | tail -n100 | grep ERROR > DSTFILE`.

**--include INCLUDE**

Comma-separated list of logs to include,each entry is either a "selector" or a glob for matching files.

Selectors must be one of `all`, `orion`, `system`, `ts`.

Anything that starts with `/` (forward slash) is assumed to be a glob pattern and interpreted through `find(1)`. Other entries are ignored.

**TIP:** use single quotes around the parameter value to prevent undesired glob expansion.

The default setting is `all`.

#### `--exclude EXCLUDE`

Comma separated list of logs to exclude. Applies to the list selected by `--include`. Params are interpreted just like in `--include`.

#### `--since SINCE`

Grab logs from this time window in the past. Should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

#### `--from FROM`

Timestamp where collection begins; must be of the form `yyyymmdd-HH:MM`.

#### `--to TO`

Timestamp where collection ends; must be of the form `yyyymmdd-HH:MM`.

#### `--outfile OUTFILE`

File path for printing all results. By default printed to `stdout`.

#### `--outdir OUTDIR`

Directory path for writing results with original directory structure from each node. Used as an alternative to printing output to `outfile/stdout`.

#### `--cmd_infmt CMD_INFMT`

Specify if the input file should be compressed or uncompressed before running `CMD`. `C` for compressed, `U` for uncompressed. Don't use this flag if `CMD` works on both.

#### `--cmd_outfmt CMD_OUTFMT`

Specify if `OUTFILE` generated by `CMD` should be compressed or uncompressed. `C` for compressed, `U` for uncompressed. Don't use this flag if output file is of the same format as the input file.

#### `--nodes NODES`

Comma separated list of nodes where to run command. Skip this to use all nodes.

## map-tiles

```
tscli map-tiles [-h] {disable,enable,status}
```

This subcommand has the following options:

```
tscli map-tiles enable [-h] [--online] [--offline] [--tar TAR] [--md5 MD5]
```

Enables ThoughtSpot's map tiles, used when constructing geomap charts.

If you don't have internet access, you must download the map tiles tar and md5 files, and append the following to the `tscli` command:

**--online**

Download `maptiles` tar from internet.

The default setting is `True`

**--offline**

Using `maptiles` tar from local disk.

The default setting is `False`

**--tar TAR**

Specified tar file for map-tiles.

**--md5 MD5**

Specified md5 file for map-tiles.

**tscli map-tiles disable**

Disable map-tiles functionality.

**tscli map-tiles status**

Check whether map-tiles are enabled.

## monitoring

```
tscli monitoring [-h] {set-config,show-config}
```

This subcommand has the following options:

```
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.

The `monitoring` subcommand accepts the following optional flags:

**--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.

The default setting is False .

**tscli monitoring show-config**

Shows the monitoring configuration.

**nas**

```
tscli nas [-h] {ls,mount-cifs,mount-nfs,unmount}
```

This subcommand has the following options:

**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.

Accepts the following optional flags:

**--server SERVER**

IP address or DNS name of CIFS service.

For example, 10.20.30.40 .

**--path\_on\_server PATH\_ON\_SERVER**

Filesystem path on the CIFS source server to mount NAS.

For example, `/a`.

The default setting is `/` (forward slash).

**--mount\_point MOUNT\_POINT**

Directory on all cluster nodes where to mount the NFS filesystem on the target.

If this directory does not exist, the command creates it. If this directory already exists, the command uses it for mounting.

For example, `/mnt/external`.

**--username USERNAME**

Username to connect to the CIFS filesystem

**--password PASSWORD**

CIFS password for `--username`

**--uid UID**

The `UID` that owns all files or directories on the mounted filesystem when the server does not provide ownership information.

See `man mount.cifs` for more details.

The default setting is `1001`.

**--gid GID**

The `GID` that owns all files or directories on the mounted filesystem when the server does not provide ownership information.

See `man mount.cifs` for more details.

The default is `1001`.

**--options OPTIONS**

Other command-line options to forward to the `mount.cifs` command.

The default setting is `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.

Accepts the following optional flags:

**--server SERVER**

IP address or DNS name of NFS service. For example, `10.20.30.40`.

**--path\_on\_server PATH\_ON\_SERVER**

Filesystem path on the NFS source server.

For example, `/a/b/c/d`.

The default setting is `/`.

**--mount\_point MOUNT\_POINT**

Directory on all cluster nodes of the target system.

If this directory does not exist, the command creates it. If this directory already exists, the command uses it for mounting.

For example, `/mnt/external`.

**--options OPTIONS**

Command-line options to mount.

The default setting is `noexec`.

**--protocol PROTO**

One of `nfs` or `nfs4`.

The default is `nfs`.

**tscli nas umount [-h] --dir DIR**

Unmounts all devices from the specified directory, `DIR`.

This command returns an error if nothing is currently mounted on this directory through `tscli nas mount`.

## node

```
tscli node [-h] {check,ls,reinstall-os,resume-reinstall-os,stat,us}
```

This subcommand has the following options:

```
tscli node check [-h] [--select {reinstall-preflight}] [--secondary SECONDARY]
```

Run checks per node.

Accepts the following flags:

```
--select {reinstall-preflight}
```

Select the type of node check

The default setting is `reinstall-preflight`.

```
--secondary SECONDARY
```

Secondary drive for `reinstall-preflight`.

The default setting is `sdd`.

```
tscli node ls [-h] [--type {all,healthy,not-healthy}]
```

Filter by node state.

The default setting is `all`.

```
tscli node reinstall-os [-h] [--secondary SECONDARY] [--cluster]
```

Reinstall OS on a node.

Accepts the following flags:

```
--secondary SECONDARY
```

Secondary drive for reinstall.

The default setting is `sdd`.

```
--cluster
```

The node part of a cluster.

The default setting is `False`.

```
tscli node resume-reinstall-os
```

Resume in-progress reinstall

## onboarding

```
tscli onboarding [-h] {configure,purge-configuration}
```

Onboarding helps application administrators to bulk update user information. In particular, it configures various in-app email options.

This subcommand has the following options:

```
tscli onboarding --help
```

Prints help for the onboarding configuration

```
tscli onboarding configure
```

Configures the onboarding through series of steps.

Asks the user to provide information necessary for onboarding-related functionality, such as the following:

1. Company name
2. Product name
3. Should welcome emails be enabled?
  - Send welcome emails to new users
  - Support email
  - Custom message to include in emails
  - URL of the ThoughtSpot instance
  - URL of the ThoughtSpot documentation

```
tscli onboarding purge-configuration
```

This command removes all onboarding configuration.

## patch

```
tscli patch [-h] {apply,ls,resume-apply,resume-rollback,rollbac  
k}
```

This subcommand has the following options:

```
tscli patch apply [-h] [release]
```

Apply the patch on an existing cluster.

Accepts the following flag:

```
release
```

The relative path to the patch tar ball.

```
tscli patch ls [-h] [--applied] [--rolled_back] [--service SERVICE] [--md5 MD5] [--
```

**history]**

Lists the patches currently applied.

Accepts the following flags:

**--applied**

Show only the patches applied since last full release.

The default setting is `False`.

**--rolled\_back**

Show only the patches rolled back since last full release.

The default setting is `False`.

**--service SERVICE**

Show patches filtered by service.

The default setting is `None`.

**--md5 MD5**

Shows the details of the patch specified.

The default setting is `None`.

**--history**

Shows the history of all patch apply/rollback release.

The default setting is `False`.

**tscli patch resume-apply [-h]**

Resume patch apply

**tscli patch resume-rollback [-h]**

Resume patch roll-backup

**tscli patch rollback [-h]**

Rollback the patch from an existing cluster

## rpackage

```
tscli rpackage [-h] {add,delete,list}
```

Manages R packages available to SpotIQ.

This subcommand has the following options:

```
tscli rpackage add [-h] [--repo REPO] [--timeout TIMEOUT] [--dest_dir DEST_DIR] [--nodes NODES] package_name
```

Command to add an R package, `package_name`, to the cluster.

Accepts the following flags:

**--repo REPO**

Specify the url of a specific repository to download packages.

**-timeout REPO**

Timeout waiting for the R Package to be installed (default: 60)

**--dest\_dir REPO**

Directory where output of this command will be placed

**--nodes NODES**

Space-separated list of IPs for nodes where to run the command.

The default setting is `all`.

```
tscli rpackage add [-h] [--timeout TIMEOUT] [--dest_dir DEST_DIR] [--nodes NODES] package_name
```

Command to delete an installed R package from the cluster.

Accepts the following flags:

**--timeout REPO**

Timeout waiting before removing the R package.

The default is 60

**--dest\_dir REPO**

Directory where to save the output of this command.

**--nodes NODES**

Space-separated list of node IPs where to run the command.

The default setting is `all`.

```
tscli rpackage list [-h] [--detailed]
```

List all R packages installed on the cluster.

## saml

```
tscli saml [-h] {configure,purge-configuration}
```

This subcommand has the following options:

```
tscli saml configure [-h]
```

Configures SAML.

```
tscli saml purge-configuration
```

Purges any existing SAML configuration.

To see a list of prerequisites, refer to [Configure SAML](#).

## scheduled-pinboards

```
tscli scheduled-pinboards [-h] {disable,enable}
```

This subcommand has the following options:

```
tscli scheduled-pinboards disable [-h]
```

Disable scheduled pinboards for this cluster.

```
tscli scheduled-pinboards enable [-h]
```

Enables scheduled pinboards, which is disabled in prod clusters by default.

**ⓘ Note:** When you enable scheduled pinboards, you should also configure a whitelist of intended email domains. Contact ThoughtSpot Support for help on how to configure a whitelist.

## smtp

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

This subcommand has the following options:

**tscli smtp remove-mailfromname**

Removes current cluster mail from name.

**tscli smtp remove-mailname**

Removes current cluster mail name.

**tscli smtp remove-relayhost**

Removes current cluster relay host.

**tscli smtp remove-saslcredentials**

Clears SASL credentials and disables SMTP AUTH.

**tscli smtp reset-canonical-mapping**

Deletes the current postmap mapping.

**tscli smtp set-canonical-mapping [-h] new\_key new\_value**

Sets a new Postmap mapping.

**tscli smtp set-mailfromname mailfromname**

Sets the name and an email address from where email alerts are sent for the cluster.

**tscli smtp set-mailname mailname**

Sets the mailname and a domain from 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.

Accepts the following flag:

**--force FORCE**

Set even if relay host is not accessible.

The default setting is `False`.

**tscli smtp set-saslcredentials**

Sets SASL credentials and enables SMTP AUTH

**tscli smtp show-canonical-mapping**

Shows the current postmap mapping.

**tscli smtp show-mailfromname**

Shows the mailname, from which email alerts are sent, for the cluster.

**tscli smtp show-mailname**

Shows the mailname, where email alerts are sent, for the cluster.

**tscli smtp show-relayhost**

Shows the for SMTP (email) sent from the cluster.

If there a relay host is not configured, the command returns `NOT FOUND`.

## snapshot

```
tscli snapshot [-h] {backup,create,delete,ls,pin,restore,unpin,update-ttl}
```

To learn more about snapshots and backups, see the [Understand the backup strategies](#) documentation.

This subcommand has the following options:

```
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.

Accepts the following flags:

**--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.

The default setting is `full`.

```
--base BASE
```

Based snapshot name for incremental backup.

Incremental backup not implemented yet.

```
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 specified `name` and `reason`.

This command does not accept `.` (periods). It does accept `-` (dashes or hyphens).

The `ttl` parameter is the number of days after which this snapshot is automatically deleted. A value of `-1` disables automatic deletion.

```
tscli snapshot pin [-h] name
```

Pins a snapshot so it cannot be deleted or garbage collected.

```
tscli snapshot delete [-h] name
```

Deletes the named snapshot.

```
tscli snapshot ls [-h]
```

Lists available snapshots.

```
tscli snapshot restore [-h] [--allow_release_change] [--only_service_state] name
```

Restores cluster to an existing snapshot.

Accepts the following flags:

```
--allow_release_change
```

Allow restoration to a snapshot at a different release. (default: `False`)

```
--only_service_state
```

Restore only service state. (default: `False`)

```
tscli snapshot unpin [-h] name
```

Unpin a snapshot so it can be deleted or garbage-collected

```
tscli snapshot update-ttl [-h] [--disable DISABLE] name ttl
```

Updates manual snapshot garbage collection policy.

Accepts the following flags:

**name**

Specifies which snapshot to update.

**ttl**

Extends the manual snapshot

**ttl**

This is the "time-to-live" value.

Use a positive value to increase **ttl**. Use negative values 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 has the following options:

```
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.

**spot**

```
tscli spot [-h] {enable}
```

Enables Spot integration.

This subcommand has the following option:

```
tscli spot enable [-h] --token TOKEN --thoughtspot_url THOUGHTSPOT_URL [--cache_timeout CACHE_TIMEOUT]
```

The `spot` subcommand accepts the following optional flags:

**--token TOKEN**

Slack authorization token for Spot bot. This is required. You receive this token when your Slack administrator adds the Spot application.

**--thoughtspot\_url THOUGHTSPOT\_URL**

URL for the ThoughtSpot application.

This is required.

**--cache\_timeout CACHE\_TIMEOUT**

Internal cache timeout.

The default setting is `60000`.

**ssl**

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

This subcommand manages the SSL configuration.

To use SSL, the following ports must be open:

- 443

- 80

This subcommand has the following options:

```
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.

```
tscli ssl add-valid-hosts [-h] VALID_HOSTS
```

Enables host validation for the specified host(s). Helps improve security. This feature is for all customers that have or are planning to enable SSL. Multiple hosts must be separated by a comma (,).

Examples:

1. If you want to make sure the valid host is **cluster1.corp.example.com**, you would run the command:

```
tscli ssl add-valid-hosts cluster1.corp.example.com
```

2. If you want to allow all hosts which have the suffix **corp.example.com**, you would run the command:

```
tscli ssl add-valid-hosts *.corp.example.com
```

. This wild card should be used within the hostname.

3. If you want to allow multiple valid hosts, for example both **\*.corp.example.com** and **cluster1**, you would run the command:

```
tscli ssl add-valid-hosts *.corp.thoughtspot.com,cluster1
```

## sssd

```
tscli sssd {enable, disable, set-sudo-group, clear-sudo-group}
```

This subcommand uses system security services daemon (SSSD), and has the following options:

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

Enables system Active Directory (AD) user access on a single node. You will be prompted for password credentials. The user must have permission to join a computer or VM to the domain.

```
tscli sssd disable
```

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

Allows `sudo` permissions for AD group.

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

Clears any set AD sudo group.

For more about setting up Active Directory access, see [Enable Active Directory based access](#).

## storage

```
tscli storage [-h] {gc,df}
```

This subcommand has the following options:

```
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 finishes, you can restart the cluster with `tscli cluster start`.

This command frees space in the following directories:

```
/tmp
/usr/local/scaligent/logs/
/export/logs/orion
/export/logs/oreo
/export/logs/hadoop
/export/logs/zookeeper
cores
```

The `storage` subcommand accepts these optional flags:

**--log\_age *LOG\_AGE***

Deletes logs older than these many hours. Use a non-zero value, because zero deletes all temporary files, including the ones that are closed temporarily, while they are passed from one component to the next.

The default setting is `4`.

**--force**

Forces deletion of all logs and temporary files regardless of age. This must only be run on a stopped cluster.

The default setting is `False`.

**--localhost\_only**

If used, only the logs on the localhost will be removed. If not specified, the command acts on the entire cluster.

**tscli storage df [--mode disk|hdfs]**

Checks the disk usage on the relevant mounts. Returns output similar to the Linux system command `df -h directory`.

## support

```
tscli support [-h]
{bundle, restart-remote, rm-admin-email, rm-admin-phone, rm-
feedback-email,
set-admin-email, set-admin-phone, set-debug-ui-password, se
t-feedback-email,
set-remote, show-admin-email, show-admin-phone, show-feedba
ck-email,
show-remote, start-remote, stop-remote}
```

This subcommand has the following options:

```
tscli support bundle [-h] [--include INCLUDE] [--exclude EXCLUDE] [--list_selectors]
[--since SINCE] [--from FROM] [--to TO] [--out OUT] [--nodes NODES]
```

#### --include INCLUDE

Comma-separated list of selectors to include. Each entry is either a "selector" or a glob for matching files. To see the list of valid selectors, run this command with `--list_selectors`. You may also specify `all` to get all selectors and logs, and `basic` to get only the basic selectors.

Selectors can be used for logs collection: `all`, `orion`, `system`, `ts`, or the name of a service.

Anything that starting with `/` (forward slash) is assumed to be a glob pattern, and it is interpreted through `find(1)`. Other entries are ignored.

**TIP:** Use single quotes around the param value to prevent undesired glob expansion. Use `all` to collect all selectors and all logs.

The default setting is `all_but_logs`.

#### --exclude EXCLUDE

Comma-separated list of selectors to exclude. Applies to the list selected by `--include`. Params are interpreted in the same manner as in `--include`.

Use the special keyword `logs` to exclude logs collection altogether.

There is no default setting.

#### --list\_selectors

List the selectors available for `--include` and `--exclude`, and then exit.

The default setting is `False`.

#### --since SINCE

Grabs logs from this time window in the past. Should be a human-readable duration string, such as `4h` (4 hours), `30m` (30 minutes), `1d` (1 day).

There is no default setting.

#### --from FROM

Timestamp when collection begins. Must be of the form: `yyyymmdd-HH:MM`.

There is no default setting.

**--to *T0***

Timestamp when collection ends. Must be of the form: `yyyymmdd-HH:MM`.

There is no default setting.

**--out *OUT***

Tarball path for dumping the support bundle.

The default setting is `/tmp/support_bundle.tar.gz`.

**--nodes *NODES***

Comma separated list of nodes from where to collect logs. Skip this to use all nodes.

There is no default setting.

**tscli support restart-remote**

Restarts remote support.

**tscli support rm-admin-email**

Removes the email address for contacting the customer administrator. Replaces it with the default ThoughtSpot Support email address.

**tscli support rm-feedback-email**

Removes the email address for product feedback. Replaces it with the default ThoughtSpot Support email address

**tscli support rm-admin-phone**

Removes the phone number for contacting the customer administrator. Replaces it with the default ThoughtSpot Support phone number

**tscli support rm-feedback-email**

Removes the email for sending feedback out of the system. To set a blank email address, issue the command `tscli support set-feedback-email`.

**tscli support set-admin-email *email***

Sets the email address for contacting the customer administrator. To display a blank email address, issue the command `tscli support set-admin-email`.

**tscli support set-feedback-email *email***

Sets the email address for sending feedback. To display a blank email address, issue the command `tscli support set-feedback-email`.

**tscli support set-admin-phone *phone\_number***

Sets the phone number for contacting the customer administrator. Specify a phone number using any value, such as `+1 800-508-7008 Ext. 1`. 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, such as `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-feedback-email
```

Shows the email address for product feedback, if set

```
tscli support show-admin-phone
```

Shows the phone number for customer administrator, if set

```
tscli support show-remote
```

Shows the status and configuration of remote support

```
tscli support start-remote
```

Starts remote support

```
tscli support stop-remote
```

Stops remote support

## tokenauthentication

```
tscli cli tokenauthentication [-h] {disable,enable}
```

This subcommand has the following options:

```
tscli cli tokenauthentication enable
```

Generates a token

```
tscli cli tokenauthentication disable
```

Purges token login configuration

# Date and time formats reference

This is a reference 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 an 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 through `tsload`

When loading through the `tsload` command you must specify `date` and `timestamp` formats using the format specifications defined in the [strftime 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 through 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 through an ETL tool

Data loads through ETL uses ODBC or JDBC connections. After you extract the data from the source but before you load it into ThoughtSpot, you must transform any date or timestamp columns into a format that is valid for ThoughtSpot. After the data transformation completes, there is no requirement for explicit data masking. See the data integration guide for more information on loading data through 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

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

# 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>

Function	Description	Examples
to_string	Returns the input as a text string. To convert a date to a string, specify the date format you want to use.	<code>to_string (45 + 1) = '46'</code> <code>to_string (revenue - cost)</code> <code>to_string (date, ('%m/%d/%y'))</code>

## Date functions

Function	Description	Examples
add_days	Returns the result of adding the specified number of days to the given date.	<code>add_days (01/30/2015, 5) = 02/04/2015</code> <code>add_days (invoiced, 30)</code>
add_minutes	Returns the result of adding the specified number of minutes to input date/date-time/time.	<code>add_minutes (01/30/2015 00:10:20, 5) = 01/30/2015 00:11:20</code> <code>add_minutes (invoiced, 30)</code>
add_months	Returns the result of adding the specified number of months to the given date.	<code>add_months (01/30/2015, 5) = 06/30/2015</code> <code>add_months (invoiced_date, 5)</code>
add_seconds	Returns the result of adding the specified number of seconds to the given date.	<code>add_seconds (01/30/2015 00:00:00, 5) = 06/30/2015 00:00:05</code> <code>add_seconds (invoiced_date, 5)</code>
add_weeks	Returns the result of adding the specified number of weeks to the given date.	<code>add_weeks (01/30/2015, 2) = 02/13/2015</code> <code>add_weeks (invoiced_date, 2)</code>
add_years	Returns the result of adding the specified number of years to the given date.	<code>add_years (01/30/2015, 5) = 01/30/2020</code> <code>add_years (invoiced_date, 5)</code>
date	Returns the date portion of a given date.	<code>date (home_visit)</code>

Function	Description	Examples
day	Returns the number (1-31) of the day for the given date.	day (01/15/2014) = 15 day (date ordered)
day_number_of_quarter	Returns the number of the day in a quarter for a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_quarter (01/30/2015) = 30 day_number_of_quarter (01/30/2015, 'fiscal') = 91
day_number_of_week	Returns the number (1-7) of the day in a week for a given date with 1 being Monday and 7 being Sunday.	day_number_of_week(01/15/2014) = 3 day_number_of_week (shipped)
day_number_of_year	Returns the number (1-366) of the day in a year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_year (01/30/2015) = 30 day_number_of_year ( 01/30/2015, 'fiscal' ) = 275 day_number_of_year (invoiced)
day_of_week	Returns the day of the week for the given date.	day_of_week (01/30/2015) = Friday day_of_week (serviced)
diff_days	Subtracts the second date from the first date and returns the result in number of days, rounded down if not exact.	diff_days (01/15/2014, 01/17/2014) = -2 diff_days (purchased, shipped)

Function	Description	Examples
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 from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	month_number (09/20/2014) = 9 month_number (09/20/2014, 'fiscal') = 5 month_number (purchased)
month_number_of_quarter	Returns the month (1-3) number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	month_number_of_quarter (02/20/2018) = 2 month_number_of_quarter (02/20/2018, 'fiscal') = 1
now	Returns the current timestamp.	now ()

Function	Description	Examples
quarter_number	Returns the number (1-4) of the quarter associated with the given date. Add an optional second parameter to specify 'fiscal' or 'calendar' dates. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	<code>quarter_number ( 04/14/2014 ) = 2</code> <code>quarter_number ( 04/14/2014, 'fiscal' ) = 4</code> <code>quarter_number ( shipped )</code>
start_of_month	Returns MMM yyyy for the first day of the month. Your installation configuration can override this setting so that it returns a different format such as MM/dd/yyyy . Speak with your ThoughtSpot administrator for information on doing this.	<code>start_of_month ( 01/31/2015 ) = Jan FY 2015</code> <code>start_of_month ( shipped )</code>
start_of_quarter	Returns the date for the first day of the quarter for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	<code>start_of_quarter ( 04/01/2014 ) = Apr 2014</code> <code>start_of_quarter ( 04/01/2014, 'fiscal' ) = Feb 2014</code> <code>start_of_quarter ( sold )</code>
start_of_week	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>

Function	Description	Examples
start_of_year	Returns the date for the first day of the year for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	start_of_year (04/01/2014) returns Jan 2014 start_of_year (04/01/2014, 'fiscal') returns May 2013 start_of_year (joined)
time	Returns the time portion of a given date.	time (3/1/2002 10:32) = 10:32 time (call began)
week_number_of_month	Returns the week number for the given date in a month.	week_number_of_month(03/23/2017) = 3
week_number_of_quarter	Returns the week number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	week_number_of_quarter (04/03/2017) = 1 week_number_of_quarter (04/03/2017, 'fiscal') = 10
week_number_of_year	Returns the week number for the given date in a year. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	week_number_of_year (01/17/2014) = 3 week_number_of_year ( 01/17/2014, 'fiscal') = 38

Function	Description	Examples
year	Returns the year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01. Per standard convention, the fiscal year is defined by the year-end date.)	year (01/15/2014) = 2014 year (12/15/2013, 'fiscal') = 2014 year (date ordered)

## Mixed functions

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

Function	Description	Examples
!=	Returns true if the first value is not equal to the second value.	3 != 2 = true revenue != 1000000
<	Returns true if the first value is less than the second value.	3 < 2 = false revenue < 1000000
<=	Returns true if the first value is less than or equal to the second value.	1 <= 2 = true revenue <= 1000000
=	Returns true if the first value is equal to the second value.	2 = 2 = true revenue = 1000000
>	Returns true if the first value is greater than the second value.	3 > 2 = true revenue > 1000000
>=	Returns true if the first value is greater than or equal to the second value.	3 >= 2 = true revenue >= 1000000
greatest	Returns the larger of the values.	greatest (20, 10) = 20 greatest (q1 revenue, q2 revenue)
least	Returns the smaller of the values.	least (20, 10) = 10 least (q1 revenue, q2 revenue)

# Number functions

Function	Description	Examples
*	Returns the result of multiplying both numbers.	$3 * 2 = 6$ price * taxrate
+	Returns the result of adding both numbers.	$1 + 2 = 3$ price + shipping
-	Returns the result of subtracting the second number from the first.	$3 - 2 = 1$ revenue - tax
/	Returns the result of dividing the first number by the second.	$6 / 3 = 2$ markup / retail price
<sup>^</sup>	Returns the first number raised to the power of the second.	$3 ^ 2 = 9$ width ^ 2
abs	Returns the absolute value.	abs (-10) = 10 abs (profit)
acos	Returns the inverse cosine in degrees.	acos (0.5) = 60 acos (cos-satellite-angle)
asin	Returns the inverse sine (specified in degrees).	asin (0.5) = 30 asin (sin-satellite-angle)
atan	Returns the inverse tangent in degrees.	atan (1) = 45 atan (tan-satellite-angle)
atan2	Returns the inverse tangent in degrees.	atan2 (10, 10) = 45 atan2 (longitude, latitude)
cbrt	Returns the cube root of a number.	cbrt (27) = 3 cbrt (volume)
ceil	Returns the smallest following integer.	ceil (5.9) = 6 ceil (growth rate)
cos	Returns the cosine of an angle (specified in degrees).	cos (63) = 0.45 cos (beam angle)
cube	Returns the cube of a number.	cube (3) = 27 cube (length)
exp	Returns Euler's number (~2.718) raised to a power.	exp (2) = 7.38905609893 exp (growth)

Function	Description	Examples
exp2	Returns 2 raised to a power.	exp2 (3) = 8 exp2 (growth)
floor	Returns the largest previous integer.	floor (5.1) = 5 floor (growth rate)
ln	Returns the natural logarithm.	ln (7.38905609893) = 2 ln (distance)
log10	Returns the logarithm with base 10.	log10 (100) = 2 log10 (volume)
log2	Returns the logarithm with base 2 (binary logarithm).	log2 (32) = 5 log2 (volume)
mod	Returns the remainder of first number divided by the second number.	mod (8, 3) = 2 mod (revenue, quantity)
pow	Returns the first number raised to the power of the second number.	pow (5, 2) = 25 pow (width, 2)
random	Returns a random number between 0 and 1.	random () = .457718 random ()
round	Returns the first number rounded to the second number (the default is 1).	round (35.65, 10) = 40 round (battingavg, 100) round (48.67, .1) = 48.7
safe_divide	Returns the result of dividing the first number by the second. If the second number is 0, returns 0 instead of NaN (not a number).	safe_divide (12, 0) = 0 safe_divide (total_cost, units)
sign	Returns +1 if the number is greater than zero, -1 if less than zero, 0 if zero.	sign (-250) = -1 sign (growth rate)
sin	Returns the sine of an angle (specified in degrees).	sin (35) = 0.57 sin (beam angle)
spherical_distance	Returns the distance in km between two points on Earth.	spherical_distance (37.465191, -122.153617, 37.421962, -122.142174) = 4,961.96 spherical_distance (start_latitude, start_longitude, start_latitude, start_longitude)

Function	Description	Examples
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'
		<b>Important:</b> Not available for row level security (RLS) formulas.
if...then...else	Conditional operator.	if (3 > 2) then 'bigger' else 'not bigger' if (cost > 500) then 'flag' else 'approve'
ifnull	Returns the first value if it is not null, otherwise returns the second.	ifnull (cost, 'unknown')
isnull	Returns true if the value is null.	isnull (phone)
not	Returns true if the condition is false, otherwise returns false.	not (3 > 2) = false not (state = 'texas')
or	Returns true when either condition is true, otherwise returns false.	(1 = 5) or (3 > 2) = true state = 'california' or state ='oregon'

## Text functions

Function	Description	Examples
concat	Returns two or more values as a concatenated text string. Use single quotes around each literal string, not double quotes.	concat ( 'hay' , 'stack' ) = 'haystack' concat (title, ' ', first_name , ' ', last_name)
contains	Returns true if the first string contains the second string, otherwise returns false.	contains ('broomstick', 'room') = true contains (product, 'trial version')
edit_distance	Accepts two text strings. Returns the edit distance (minimum number of operations required to transform one string into the other) as an integer. Works with strings under 1023 characters.	edit_distance ('attorney', 'atty') = 4 edit_distance (color, 'red')
edit_distance_with_cap	Accepts two text strings and an integer to specify the upper limit cap for the edit distance (minimum number of operations required to transform one string into the other). If the edit distance is less than or equal to the specified cap, returns the edit distance. If it is higher than the cap, returns the cap plus 1. Works with strings under 1023 characters.	edit_distance_with_cap ('pokemon go', 'minecraft pixelmon', 3) = 4 edit_distance_with_cap (event, 'burning man', 3)
similar_to	Accepts a document text string and a search text string. Returns true if relevance score (0-100) of the search string with respect to the document is greater than or equal to 20. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document.	similar_to ('hello world', 'hello swirl') = true similar_to (current team, drafted by)

Function	Description	Examples
similarity	Accepts a document text string and a search text string. Returns the relevance score (0-100) of the search string with respect to the document. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document. If the two strings are an exact match, returns 100.	similarity ('where is the burning man concert', 'burning man') = 46 similarity (tweet1, tweet2)
spells_like	Accepts two text strings. Returns true if they are spelled similarly and false if they are not. Works with strings under 1023 characters.	spells_like ('thouhgtspot', 'thoughtspot') = true spells_like (studio, distributor)
strlen	Returns the length of the text.	strlen ('smith') = 5 strlen (lastname)
strpos	Returns the numeric position (starting from 0) of the first occurrence of the second string in the first string, or -1 if not found.	strpos ('haystack_with_needles', 'needle') = 14 strpos (complaint, 'lawyer')
substr	Returns the portion of the given string, beginning at the location specified (starting from 0), and of the given length.	substr ('persnickety', 3, 7) = snicket substr (lastname, 0, 5)

## Variables

These variables can be used in your expressions.

Function	Description	Examples
ts_groups	Returns a list of all the groups the current logged in user belongs to. For any row, if the expression evaluates to true for any of the groups, the user can see that row.	ts_groups = 'east'
ts_username	Returns the user with the matching neame.	ts_username != 'mark'

# Formula function reference

ThoughtSpot allows you to create derived columns in worksheets using formulas. You create these columns by building formulas using the **Formula Assistant**. An individual formula is constructed from *n* combination of operators and functions.

This reference lists the various operators and functions you can use to create formulas.

## Operators

Operator	Description	Examples
and	Returns true when both conditions are true, otherwise returns false.	(1 = 1) and (3 > 2) = true lastname = 'smith' and state ='texas'
		<b>ⓘ Note:</b> Not available for row level security (RLS) formulas.
if...then...else	Conditional operator.	if (3 > 2) then 'bigger' else 'not bigger' if (cost > 500) then 'flag' else 'approve'
ifnull	Returns the first value if it is not null, otherwise returns the second.	ifnull (cost, 'unknown')
isnull	Returns true if the value is null.	isnull (phone)
not	Returns true if the condition is false, otherwise returns false.	not (3 > 2) = false not (state = 'texas')
or	Returns true when either condition is true, otherwise returns false.	(1 = 5) or (3 > 2) = true state = 'california' or state ='oregon'

## Aggregate functions (group aggregate)

These functions can be used to aggregate data.

Function	Description	Examples
average	Returns the average of all the values of a column.	average (revenue)
average_if	Returns the average of all the columns that meet a given criteria.	average_if(city = "San Francisco", revenue)
count	Returns the number of rows in the table containing the column.	count (product)
count_if	Returns the number of rows in the table containing the column.	count_if(region = 'west', region)
cumulative_average	Takes a measure and one or more attributes. Returns the average of the measure, accumulated by the attribute(s) in the order specified.	cumulative_average (revenue, order date, state)
cumulative_max	Takes a measure and one or more attributes. Returns the maximum of the measure, accumulated by the attribute(s) in the order specified.	cumulative_max (revenue, state)
cumulative_min	Takes a measure and one or more attributes. Returns the minimum of the measure, accumulated by the attribute(s) in the order specified.	cumulative_min (revenue, campaign)
cumulative_sum	Takes a measure and one or more attributes. Returns the sum of the measure, accumulated by the attribute(s) in the order specified.	cumulative_sum (revenue, order date)

Function	Description	Examples
group_aggregate	<p>Takes a measure and, optionally, attributes and filters. These can be used to aggregate measures with granularities and filters different from the terms/columns used in the search. Especially useful for comparison analysis.</p> <p>This formula takes the form: group_aggregate (&lt; aggregation (measure) &gt;, &lt; groupings &gt;, &lt; filters &gt;)</p> <p>Lists can be defined with {} and optional list functions query_groups or query_filters , which by default specify the lists or filters used in the original search. Plus (+) or (-) can be used to add or exclude specific columns for query groups.</p>	<pre>group_aggregate (sum (revenue) , {ship mode, date} , {} )</pre> <pre>group_aggregate (sum (revenue) , {ship mode , date}, {day_of_week (date) = 'friday'} )</pre> <pre>group_aggregate (sum (revenue) , query_groups() , query_filters() )</pre> <pre>group_aggregate (sum (revenue) , query_groups() + {date} , query_filters() )</pre>
group_average	Takes a measure and one or more attributes. Returns the average of the measure grouped by the attribute(s).	group_average (revenue, customer region, state)
group_count	Takes a measure and one or more attributes. Returns the count of the measure grouped by the attribute(s).	group_count (revenue, customer region)
group_max	Takes a measure and one or more attributes. Returns the maximum of the measure grouped by the attribute(s).	group_max (revenue, customer region)
group_min	Takes a measure and one or more attributes. Returns the minimum of the measure grouped by the attribute(s).	group_min (revenue, customer region)
group_stddev	Takes a measure and one or more attributes. Returns the standard deviation of the measure grouped by the attribute(s).	group_stddev (revenue, customer region)
group_sum	Takes a measure and one or more attributes. Returns the sum of the measure grouped by the attribute(s).	group_sum (revenue, customer region)
group_unique_count	Takes a measure and one or more attributes. Returns the unique count of the measure grouped by the attribute(s).	group_unique_count (product , supplier)

Function	Description	Examples
group_variance	Takes a measure and one or more attributes. Returns the variance of the measure grouped by the attribute(s).	group_variance (revenue, customer region)
max	Returns the maximum value of a column.	max (sales)
max_if	Returns the maximum value among columns that meet a criteria.	max_if( (revenue > 10) , customer region )
min	Returns the minimum value of a column.	min (revenue)
min_if	Returns the minimum value among columns that meet a criteria.	min_if( (revenue < 10) , customer region )
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)

Function	Description	Examples
<code>moving_min</code>	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.	<code>moving_min (defects, 3, 1, product)</code>
<code>moving_sum</code>	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>
<code>rank</code>	Returns the rank for the current row. Identical values receive an identical rank. Takes an aggregate input for the first argument. The second argument specifies the order, 'asc'   'desc' .	<code>rank (sum (revenue) , 'asc' )</code> <code>rank (sum (revenue) , 'desc' )</code>
<code>rank_percentile</code>	Returns the percentile rank for the current row. Identical values are assigned an identical percentile rank. Takes an aggregate input for the first argument. The second argument specifies the order, 'asc'   'desc' .	<code>rank_percentile (sum (revenue) , 'asc' )</code> <code>rank_percentile (sum (revenue) , 'desc' )</code>
<code>stddev</code>	Returns the standard deviation of all values of a column.	<code>stddev (revenue)</code>
<code>stddev_if</code>	Returns a standard deviation values filtered to meet a specific criteria.	<code>stddev_if( (revenue &gt; 10) , (revenue/10.0) )</code>
<code>sum</code>	Returns the sum of all the values of a column.	<code>sum (revenue)</code>
<code>sum_if</code>	Returns sum values filtered by a specific criteria.	<code>sum_if(region='west', revenue)</code>

Function	Description	Examples
unique_count	Returns the number of unique values of a column.	unique_count (customer)
unique_count_if	Returns the number of unique values of a column provided it meets a criteria.	unique_count_if( (revenue > 10) , order date )
variance	Returns the variance of all the values of a column.	variance (revenue)
variance_if	Returns the variance of all the values of a column provided it meets a criteria..	variance_if( (revenue > 10) , (revenue/10.0) )

## Conversion functions

These functions can be used to convert data from one data type to another. Conversion to or from date data types is not supported.

Function	Description	Examples
to_bool	Returns the input as a boolean (true or false).	to_bool (0) = false to_bool (married)
to_date	Accepts a date represented as an integer or text string, and a second string parameter that can include 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.	to_date (date_sold, '%Y-%m-%d')
to_double	Returns the input as a double.	to_double ('3.14') = 3.14 to_double (revenue * .01)
to_integer	Returns the input as an integer.	to_integer ('45') + 1 = 46 to_integer (price + tax - cost)
to_string	Returns the input as a text string. To convert a date to a string, specify the date format you want to use.	to_string (45 + 1) = '46' to_string (revenue - cost) to_string (date, ('%m/%d/%y'))

## Date functions

Function	Description	Examples
add_days	Returns the result of adding the specified number of days to the given date.	<code>add_days (01/30/2015, 5) = 02/04/2015</code> <code>add_days (invoiced, 30)</code>
add_minutes	Returns the result of adding the specified number of minutes to input date/date-time/time.	<code>add_minutes (01/30/2015 00:10:20, 5) = 01/30/2015 00:11:20</code> <code>add_minutes (invoiced, 30)</code>
add_months	Returns the result of adding the specified number of months to the given date.	<code>add_months (01/30/2015, 5) = 06/30/2015</code> <code>add_months (invoiced_date, 5)</code>
add_seconds	Returns the result of adding the specified number of seconds to the given date.	<code>add_seconds (01/30/2015 00:00:00, 5) = 06/30/2015 00:00:05</code> <code>add_seconds (invoiced_date, 5)</code>
add_weeks	Returns the result of adding the specified number of weeks to the given date.	<code>add_weeks (01/30/2015, 2) = 02/13/2015</code> <code>add_weeks (invoiced_date, 2)</code>
add_years	Returns the result of adding the specified number of years to the given date.	<code>add_years (01/30/2015, 5) = 01/30/2020</code> <code>add_years (invoiced_date, 5)</code>
date	Returns the date portion of a given date.	<code>date (home_visit)</code>
day	Returns the number (1-31) of the day for the given date.	<code>day (01/15/2014) = 15</code> <code>day (date_ordered)</code>

Function	Description	Examples
day_number_of_quarter	Returns the number of the day in a quarter for a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_quarter (01/30/2015) = 30 day_number_of_quarter (01/30/2015, 'fiscal') = 91
day_number_of_week	Returns the number (1-7) of the day in a week for a given date with 1 being Monday and 7 being Sunday.	day_number_of_week(01/15/2014) = 3 day_number_of_week (shipped)
day_number_of_year	Returns the number (1-366) of the day in a year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	day_number_of_year (01/30/2015) = 30 day_number_of_year ( 01/30/2015, 'fiscal' ) = 275 day_number_of_year (invoiced)
day_of_week	Returns the day of the week for the given date.	day_of_week (01/30/2015) = Friday day_of_week (serviced)
diff_days	Subtracts the second date from the first date and returns the result in number of days, rounded down if not exact.	diff_days (01/15/2014, 01/17/2014) = -2 diff_days (purchased, shipped)
diff_time	Subtracts the second date from the first date and returns the result in number of seconds.	diff_time (01/01/2014, 01/01/2014) = -86,400 diff_time (clicked, submitted)

Function	Description	Examples
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 from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	month_number (09/20/2014) = 9 month_number (09/20/2014, 'fiscal') = 5 month_number (purchased)
month_number_of_quarter	Returns the month (1-3) number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	month_number_of_quarter (02/20/2018) = 2 month_number_of_quarter (02/20/2018, 'fiscal') = 1
now	Returns the current timestamp.	now ()

Function	Description	Examples
quarter_number	Returns the number (1-4) of the quarter associated with the given date. Add an optional second parameter to specify 'fiscal' or 'calendar' dates. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	<code>quarter_number ( 04/14/2014 ) = 2</code> <code>quarter_number ( 04/14/2014, 'fiscal' ) = 4</code> <code>quarter_number ( shipped )</code>
start_of_month	Returns MMM yyyy for the first day of the month. Your installation configuration can override this setting so that it returns a different format such as MM/dd/yyyy . Speak with your ThoughtSpot administrator for information on doing this.	<code>start_of_month ( 01/31/2015 ) = Jan FY 2015</code> <code>start_of_month ( shipped )</code>
start_of_quarter	Returns the date for the first day of the quarter for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	<code>start_of_quarter ( 04/01/2014 ) = Apr 2014</code> <code>start_of_quarter ( 04/01/2014, 'fiscal' ) = Feb 2014</code> <code>start_of_quarter ( sold )</code>
start_of_week	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>

Function	Description	Examples
start_of_year	Returns the date for the first day of the year for the given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	start_of_year (04/01/2014) returns Jan 2014 start_of_year (04/01/2014, 'fiscal') returns May 2013 start_of_year (joined)
time	Returns the time portion of a given date.	time (3/1/2002 10:32) = 10:32 time (call began)
week_number_of_month	Returns the week number for the given date in a month.	week_number_of_month(03/23/2017) = 3
week_number_of_quarter	Returns the week number for the given date in a quarter. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	week_number_of_quarter (04/03/2017) = 1 week_number_of_quarter (04/03/2017, 'fiscal') = 10
week_number_of_year	Returns the week number for the given date in a year. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01.)	week_number_of_year (01/17/2014) = 3 week_number_of_year ( 01/17/2014, 'fiscal') = 38

Function	Description	Examples
year	Returns the year from a given date. Add an optional second parameter to specify whether a 'fiscal' or 'calendar' year is used to calculate the result. Default is 'calendar'. (In examples, start of fiscal year is set to May 01. Per standard convention, the fiscal year is defined by the year-end date.)	year (01/15/2014) = 2014 year (12/15/2013, 'fiscal') = 2014 year (date ordered)

## Mixed functions

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

Function	Description	Examples
!=	Returns true if the first value is not equal to the second value.	3 != 2 = true revenue != 1000000
<	Returns true if the first value is less than the second value.	3 < 2 = false revenue < 1000000
<=	Returns true if the first value is less than or equal to the second value.	1 <= 2 = true revenue <= 1000000
=	Returns true if the first value is equal to the second value.	2 = 2 = true revenue = 1000000
>	Returns true if the first value is greater than the second value.	3 > 2 = true revenue > 1000000
>=	Returns true if the first value is greater than or equal to the second value.	3 >= 2 = true revenue >= 1000000
greatest	Returns the larger of the values.	greatest (20, 10) = 20 greatest (q1 revenue, q2 revenue)
least	Returns the smaller of the values.	least (20, 10) = 10 least (q1 revenue, q2 revenue)

# Number functions

Function	Description	Examples
*	Returns the result of multiplying both numbers.	$3 * 2 = 6$ price * taxrate
+	Returns the result of adding both numbers.	$1 + 2 = 3$ price + shipping
-	Returns the result of subtracting the second number from the first.	$3 - 2 = 1$ revenue - tax
/	Returns the result of dividing the first number by the second.	$6 / 3 = 2$ markup / retail price
<sup>^</sup>	Returns the first number raised to the power of the second.	$3 ^ 2 = 9$ width ^ 2
abs	Returns the absolute value.	abs (-10) = 10 abs (profit)
acos	Returns the inverse cosine in degrees.	acos (0.5) = 60 acos (cos-satellite-angle)
asin	Returns the inverse sine (specified in degrees).	asin (0.5) = 30 asin (sin-satellite-angle)
atan	Returns the inverse tangent in degrees.	atan (1) = 45 atan (tan-satellite-angle)
atan2	Returns the inverse tangent in degrees.	atan2 (10, 10) = 45 atan2 (longitude, latitude)
cbrt	Returns the cube root of a number.	cbrt (27) = 3 cbrt (volume)
ceil	Returns the smallest following integer.	ceil (5.9) = 6 ceil (growth rate)
cos	Returns the cosine of an angle (specified in degrees).	cos (63) = 0.45 cos (beam angle)
cube	Returns the cube of a number.	cube (3) = 27 cube (length)
exp	Returns Euler's number (~2.718) raised to a power.	exp (2) = 7.38905609893 exp (growth)

Function	Description	Examples
exp2	Returns 2 raised to a power.	exp2 (3) = 8 exp2 (growth)
floor	Returns the largest previous integer.	floor (5.1) = 5 floor (growth rate)
ln	Returns the natural logarithm.	ln (7.38905609893) = 2 ln (distance)
log10	Returns the logarithm with base 10.	log10 (100) = 2 log10 (volume)
log2	Returns the logarithm with base 2 (binary logarithm).	log2 (32) = 5 log2 (volume)
mod	Returns the remainder of first number divided by the second number.	mod (8, 3) = 2 mod (revenue, quantity)
pow	Returns the first number raised to the power of the second number.	pow (5, 2) = 25 pow (width, 2)
random	Returns a random number between 0 and 1.	random () = .457718 random ()
round	Returns the first number rounded to the second number (the default is 1).	round (35.65, 10) = 40 round (battingavg, 100) round (48.67, .1) = 48.7
safe_divide	Returns the result of dividing the first number by the second. If the second number is 0, returns 0 instead of NaN (not a number).	safe_divide (12, 0) = 0 safe_divide (total_cost, units)
sign	Returns +1 if the number is greater than zero, -1 if less than zero, 0 if zero.	sign (-250) = -1 sign (growth rate)
sin	Returns the sine of an angle (specified in degrees).	sin (35) = 0.57 sin (beam angle)
spherical_distance	Returns the distance in km between two points on Earth.	spherical_distance (37.465191, -122.153617, 37.421962, -122.142174) = 4,961.96 spherical_distance (start_latitude, start_longitude, start_latitude, start_longitude)

Function	Description	Examples
sq	Returns the square of a numeric value.	sq (9) = 81 sq (width)
sqrt	Returns the square root.	sqrt (9) = 3 sqrt (area)
tan	Returns the tangent of an angle (specified in degrees).	tan (35) = 0.7 tan (beam angle)

## Text functions

Function	Description	Examples
concat	Returns two or more values as a concatenated text string. Use single quotes around each literal string, not double quotes.	concat ('hay', 'stack') = 'haystack' concat (title, ' ', first_name, ' ', last_name)
contains	Returns true if the first string contains the second string, otherwise returns false.	contains ('broomstick', 'room') = true contains (product, 'trial version')
edit_distance	Accepts two text strings. Returns the edit distance (minimum number of operations required to transform one string into the other) as an integer. Works with strings under 1023 characters.	edit_distance ('attorney', 'atty') = 4 edit_distance (color, 'red')
edit_distance_with_cap	Accepts two text strings and an integer to specify the upper limit cap for the edit distance (minimum number of operations required to transform one string into the other). If the edit distance is less than or equal to the specified cap, returns the edit distance. If it is higher than the cap, returns the cap plus 1. Works with strings under 1023 characters.	edit_distance_with_cap ('pokemon go', 'minecraft pixelmon', 3) = 4 edit_distance_with_cap (event, 'burning man', 3)

Function	Description	Examples
<code>similar_to</code>	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.	<code>similar_to ('hello world', 'hello swirl') = true</code> <code>similar_to (current team, drafted by)</code>
<code>similarity</code>	Accepts a document text string and a search text string. Returns the relevance score (0-100) of the search string with respect to the document. Relevance is based on edit distance, number of words in the query, and length of words in the query which are present in the document. If the two strings are an exact match, returns 100.	<code>similarity ('where is the burning man concert', 'burning man') = 46</code> <code>similarity (tweet1, tweet2)</code>
<code>spells_like</code>	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 ('thouhgtspot', 'thoughtspot') = true</code> <code>spells_like (studio, distributor)</code>
<code>strlen</code>	Returns the length of the text.	<code>strlen ('smith') = 5</code> <code>strlen (lastname)</code>
<code>strpos</code>	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</code> <code>strpos (complaint, 'lawyer')</code>
<code>substr</code>	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</code> <code>substr (lastname, 0, 5)</code>

# Alerts code reference

This reference identifies the messages that can appear in the **System Health > Overview > Critical Alerts** and in the **Alerts** dashboard.

## Informational alerts

### TASK\_TERMINATED

**Msg:** Task {{.Service}}.{{.Task}} terminated on machine {{.Machine}}

**Type:** INFO

This alert is raised when a task terminates.

### DISK\_ERROR

**Msg:** Machine {{.Machine}} has disk errors

**Type:** INFO

Raised when a machine has disk errors.

### ZK\_AVG\_LATENCY

**Msg:** Average Zookeeper latency is more than {{.Num}} msec

**Type:** INFO

Raised when average Zookeeper latency is above a threshold.

### ZK\_MAX\_LATENCY

**Msg:** Max Zookeeper latency is more than {{.Num}} msec

**Type:** INFO

Raised when max Zookeeper latency is above a threshold.

### ZK\_MIN\_LATENCY

**Msg:** Min Zookeeper latency is more than {{.Num}} msec

**Type:** INFO

Raised when min Zookeeper latency is above a threshold.

### ZK\_OUTSTANDING\_REQUESTS

**Msg:** Number of outstanding Zookeeper requests exceeds {{.Num}}

**Type:** INFO

Raised when there are too many outstanding Zookeeper requests.

### ZK\_NUM\_WATCHERS

**Msg:** Number of Zookeeper watchers exceeds {{.Num}}

**Type:** INFO

Raised when there are too many Zookeeper watchers.

### MASTER\_ELECTION

**Msg:** {{.Machine}} elected as Orion Master

**Type:** INFO

Raised when a new Orion Master is elected.

### PERIODIC\_BACKUP

**Msg:** {{.Process}} periodic backup for policy {{.Name}} failed.

**Type:** INFO

Raised when periodic backup fails.

## PERIODIC\_SNAPSHOT

**Msg:** {{.Process}} periodic snapshot {{.Name}} failed.

**Type:** INFO

Raised when a periodic snapshot fails.

## HDFS\_CORRUPTION

**Msg:** HDFS root directory is in a corrupted state.

**Type:** INFO

Raised when HDFS root directory is corrupted.

## APPLICATION\_INVALID\_STATE

**Msg:** {{.Service}}.{{.Task}} on {{.Machine}} at location {{.Location}}

**Type:** INFO

Raised when Application raises invalid state alert.

## UPDATE\_START

**Msg:** Starting update of ThoughtSpot cluster {{.Cluster}}

**Type:** INFO

Raised when update starts.

## UPDATE\_END

**Msg:** Finished update of ThoughtSpot cluster {{.Cluster}} to release {{.Release}}

**Type:** INFO

Raised when update completes.

## Errors

### TIMELY\_JOB\_RUN\_ERROR

**Msg:** Job run {{.Message}}

**Type:** ERROR

Raised when a job run fails.

### TIMELY\_ERROR

**Msg:** Job manager {{.Message}}

**Type:** ERROR

Raised when a job manager runs into an inconsistent state.

## Warnings

### DISK\_SPACE

**Msg:** Machine {{.Machine}} has less than {{.Perc}}% disk space free

**Type:** WARNING

Raised when a disk is low on available disk space. Valid only in the 3.2 version of ThoughtSpot.

### ROOT\_DISK\_SPACE

**Msg:** Machine {{.Machine}} has less than {{.Perc}}% disk space free on root partition

**Type:** WARNING

Raised when a machine is low on available disk space on root partition.

## BOOT\_DISK\_SPACE

**Msg:** Machine {{.Machine}} has less than {{.Perc}}% disk space free on boot partition

**Type:** WARNING

Raised when a machine is low on available disk space on boot partition.

## UPDATE\_DISK\_SPACE

**Msg:** Machine {{.Machine}} has less than {{.Perc}}% disk space free on update partition

**Type:** WARNING

Raised when a machine is low on available disk space on update partition.

## EXPORT\_DISK\_SPACE

**Msg:** Machine {{.Machine}} has less than {{.Perc}}% disk space free on export partition

**Type:** WARNING

Raised when a machine is low on available disk space on export partition.

## HDFS\_NAMENODE\_DISK\_SPACE

**Msg:** Machine {{.Machine}} has less than {{.Perc}}% disk space free on HDFS namenode drive

**Type:** WARNING

Raised when a machine is low on available disk space on HDFS namenode drive.

## MEMORY

**Msg:** Machine {{.Machine}} has less than {{.Perc}}% memory free

**Type:** WARNING

Raised when a machine is low on free memory.

## OS\_USERS

**Msg:** Machine {{.Machine}} has more than {{.Num}} logged in users

**Type:** WARNING

Raised when a machine has too many users logged in.

## OS\_PROCS

**Msg:** Machine {{.Machine}} has more than {{.Num}} processes

**Type:** WARNING

Raised when a machine has more too many processes.

## SSH

**Msg:** Machine {{.Machine}} doesn't have an active SSH server

**Type:** WARNING

Raised when a machine has more than 600 processes.

## DISK\_ERROR\_EXTERNAL

**Msg:** Machine {{.Machine}} has disk errors

**Type:** WARNING

Raised when more than 2 disk errors happen in a day.

### ZK\_FD\_COUNT

**Msg:** Zookeeper has more than {{.Num}} open file descriptors

**Type:** WARNING

Raised when there are too many open Zookeeper files.

### ZK\_EPHEMERAL\_COUNT

**Msg:** Zookeeper has more than {{.Num}} ephemeral files

**Type:** WARNING

Raised when there are too many Zookeeper ephemeral files.

### HOST\_DOWN

**Msg:** {{.Machine}} is down

**Type:** WARNING

Raised when a host is down.

### TASK\_UNREACHABLE

**Msg:** {{.ServiceDesc}} on {{.Machine}} is unreachable over HTTP

**Type:** WARNING

Raised when a task is unreachable over HTTP.

### TASK\_NOT\_RUNNING

**Msg:** {{.ServiceDesc}} is not running

**Type:** WARNING

Raised when a service task is not running on any machine in the cluster.

## Critical alerts

### TASK\_FLAPPING

**Msg:** Task {{.Service}}.{{.Task}} terminated {{.\_actual\_num\_occurrences}} times in last {{.\_earliest\_duration\_str}}

**Type:** CRITICAL

This alert is raised when a task is crashing repeatedly. The service is evaluated across the whole cluster.

So, if a service crashes 5 times in a day across all nodes in the cluster, this alert is generated.

### OREO\_TERMINATED

**Msg:** Oreo terminated on machine {{.Machine}}

**Type:** CRITICAL

This alert is raised when the Oreo daemon on a machine terminates due to an error. This typically happens due to an error accessing Zookeeper, HDFS, or a hardware issue.

### HDFS\_DISK\_SPACE

**Msg:** HDFS has less than {{.Perc}}% space free

**Type:** CRITICAL

Raised when a HDFS cluster is low on total available disk space.

### ZK\_INACCESSIBLE

**Msg:** Zookeeper is not accessible

**Type:** CRITICAL

Raised when Zookeeper is inaccessible.

## PERIODIC\_BACKUP\_FLAPPING

**Msg:** Periodic backup failed {{.\_actual\_num\_occurrences}} times in last  
{{.\_earliest\_duration\_str}}

**Type:** CRITICAL

This alert is raised when a periodic backup failed repeatedly.

## PERIODIC\_SNAPSHOT\_FLAPPING

**Msg:** Periodic snapshot failed {{.\_actual\_num\_occurrences}} times in last  
{{.\_earliest\_duration\_str}}

**Type:** CRITICAL

This alert is raised when periodic snapshot failed repeatedly.

## APPLICATION\_INVALID\_STATE\_EXTERNAL

**Msg:** {{.Service}}.{{.Task}} on {{.Machine}} at location {{.Location}}

**Type:** CRITICAL

Raised when Application raises invalid state alert.

# User action code reference

This reference identifies the user action codes that can appear in the **System Health** pages and in logs or other reports.

answer_unsaved	User makes a change to tokens in the search bar.
answer_saved	User opens an existing saved answer and makes changes to tokens in the search bar.
answer_pinboard_context	User opens an existing saved pinboard, edits a context viz and makes a change to tokens in the search bar.
answer_aggregated_worksheet	User opens an existing saved aggregated worksheet and makes changes to tokens in the search bar.
answer_upgrade	Requests made for the sole purpose of upgrade.
pinboard_view	User opens an existing saved pinboard.
pinboard_filter	User adds, removes or applies values to a pinboard filter.
pinboard_ad_hoc	User drills down in a pinboard viz.
data_chart_config	Request for new data being generated following a chart config change.
data_show_underlying_row	Request to show underlying data for a data row(s).
data_export	Request to export data.
pinboard_tspublic_runtime_filter	Request to TSPublic/pinboarddata with runtime filters.
answer_aggregated_worksheet_save	User updates aggregated worksheet.
answer_add_new_filter	User adds a filter using the UI.
data_show_underlying_viz	Request to show underlying data for a data row(s).
answer_view	User opens an existing, saved answer.

answer_viz_context_view	User opens an existing saved pinboard, edits a context viz.
pinboard_insight_view	User opens SpotIQ tab pinboards.
pinboard_admin_view	User opens admin tab pinboards.
pinboard_embed_view	User opens embed pinboard from a URL.
pinboard_homepage_view	On loading of homepage pinboard.
pinboard_learn_view	On loading learn pinboard.
pinboard_tspublic_no_runtime_filter	Request to TSPublic/pinboard data without run-time filters.

# Frequently asked questions

## Where can I find the version of ThoughtSpot I am using?

Users with administrative privileges can see this displayed on the **Admin > System Health > Overview** page.

## I'm not seeing certain columns/values in the drop-down, why?

It could be the index has not built with the latest data or something is causing the column to be dropped.

- Verify the the column is available using the **Data** page.
- View the table columns and check the **INDEX TYPE** value. If it is set to `DONT_INDEX`, change it.
- Check the column's **INDEX PRIORITY** — make sure it is `1`.

To learn more about modeling data see [modeling data](#) in this documentation.