THOUGHTSPOT

Data Connect Guide

Version 3.4

September 2016

Contents

Chapter 1: Get started with ThoughtSpot Data Connect

Topics:

You can import data to ThoughtSpot from on-premise and cloud based sources using ThoughtSpot Data Connect. The initial connection to your data sources is set up by ThoughtSpot. After the connections are set up, you can do all your data imports through the ThoughtSpot application.

ThoughtSpot has partnered with Informatica to bring you ThoughtSpot Data Connect, which is a self-service data integration tool that allows you to easily pull data into ThoughtSpot for analysis. ThoughtSpot Data Connect was developed by ThoughtSpot, but it uses Informatica's REST APIs to connect to multiple end points, define import tasks and execute them. You can use ThoughtSpot Data Connect to perform extract, transform, and load operations. Data source connections may be stored, to make reconnecting seamless.

After choosing a data source, you can choose from the list of available tables to import. Then select only the columns of interest from each table. Primary key and foreign key relationships will also be imported, but only when the primary key table is imported at the same time as the foreign key table. It's easy to apply transformations and filters to the data, too. So you only import the data you want to explore. You

can schedule incremental or full recurring data loads. Everything you need to do to get your data loaded can be done within the ThoughtSpot application.

ThoughtSpot Data Connect is a premium feature, which is available at an additional cost. Please contact your ThoughtSpot representative, if you want to find out how to enable ThoughtSpot Data Connect.

Before using ThoughtSpot Data Connect

There are a few prerequisites before using ThoughtSpot Data Connect. You need to have ThoughtSpot enable the connection to Informatica, and enter the data connection credentials.

Prerequisites

- Your Informatica account will be provisioned by ThoughtSpot.

- If you want to use HTTPS with your URL, you will need to provide a signed certificate when enabling ThoughtSpot Data Connect.

- You will need outbound HTTPS (port 443 for infaagent) internet access to https://app.informaticaondemand.com and https://app2.informaticacloud.com/. Of course, you will also need any other internet connectivity and credentials required to access any cloud data sources you want to connect to, such as Salesforce. No inbound access is required. Please refer to the Informatica Cloud Security Architecture Overview whitepaper to learn how each component and service is secured. You can also ask ThoughtSpot Support for a copy of the document.

- Any user who belongs to a group that has the privilege **Has administration privileges** or **Can Manage Data** will be able to use ThoughtSpot Data Connect, if it is enabled on your cluster.

ThoughtSpot Data Connect design

ThoughtSpot Data Connect is designed to configure and connect to any on-premise or cloud data source. It quickly pulls data into ThoughtSpot through an easy to use point-and-click user interface. This diagram shows all of the required connections.
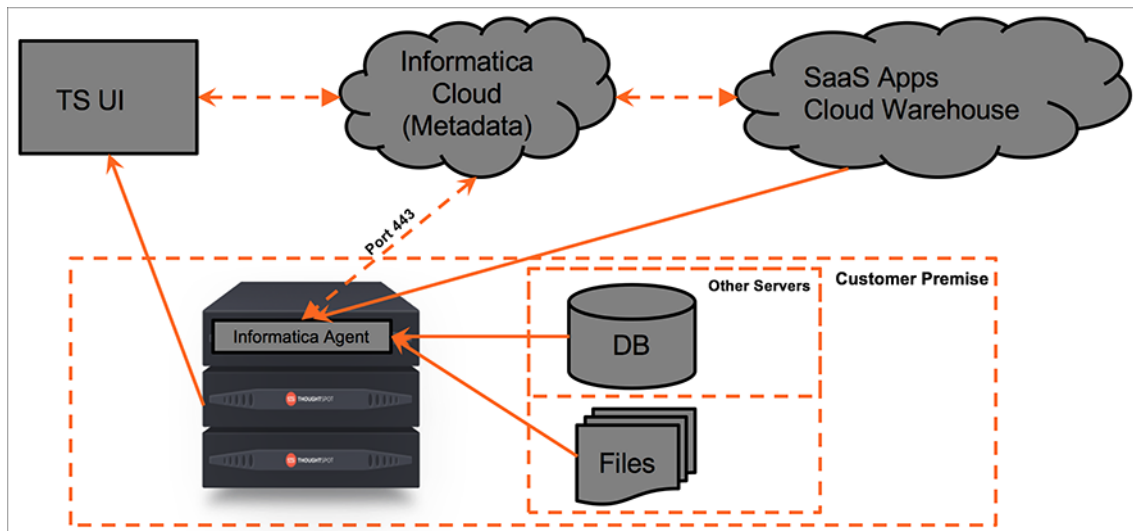
Figure 1: High-level ThoughtSpot Data Connect design diagram

Set up ThoughtSpot Data Connect

The ThoughtSpot team will assist you in the initial setup and configuration of ThoughtSpot Data Connect. You will need to provide the credentials for logging in to the various data sources you want to connect to.

Your purchase of ThoughtSpot Data Connect includes the necessary credentials for connecting to Informatica. ThoughtSpot Support will supply the Informatica credentials, if you do not already have an account. You can change the Informatica credentials later, if you would like. You need these credentials when issuing the command to enable ThoughtSpot Data Connect.

To set up ThoughtSpot Data Connect:

1. Check with ThoughtSpot Support to ensure they have provisioned an account for you with the appropriate connectivity licenses. A full list of supported connectivity licenses is available in the Data Connect Connectors Reference.

2. Enable ThoughtSpot Data Connect by completing the following steps:

    a) Log in to the Linux shell using SSH.

b) Invoke tscli to enable ThoughtSpot Data Connect, which will register with Informatica Cloud and start the infaagent service. The process will take a couple of minutes:

```
tscli etl enable-lw
  --admin_username <admin_user>
  --username <Informatica_user>
  --thoughtspot_url <URL>
  --org_id <informatica_org_id>
  [--pin_to <IP_address>]
```

If you want to use HTTPS with your URL, you will need to use a signed certificate.

The parameter *org_id* specifies the Informatica id of the organization (company). NOTE: org_id shouldn't include the prefix "Org". For example, if on Informatica cloud, the orgid is Org003XYZ, then use only 003XYZ.

The command `--pin_to` is optional. If you specify an IP to pin to, that node becomes sticky to the Informatica agent, and will always be used. This defaults to the public IP address of the localhost where this command was run.

3. You may change your user password using the following tscli command:

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

4. Now you can log in to ThoughtSpot, and begin using ThoughtSpot Data Connect.

Add a new data source

Now that you've enabled ThoughtSpot Data Connect, you will be able to add a new data source. Data sources are the source systems from which you want to import data and schedule repeat data loads.

Data sources can consist of:

• Relational database sources, like SQL Server.

- Applications, like Marketo or Salesforce.

To add a data source:

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

Figure 2: Data

3. Click the **Data Sources** tab at the top of the page, and select **+ New data source** on the upper right hand side of the panel.
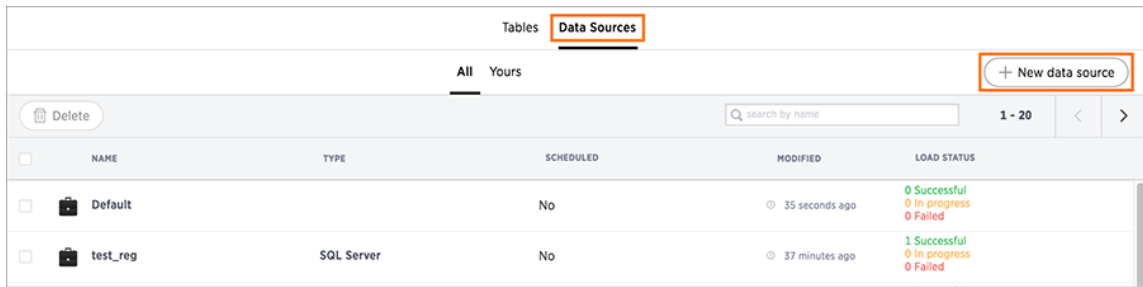

Figure 3: New data source

4. Select which type of connection you wish to use from the available list.
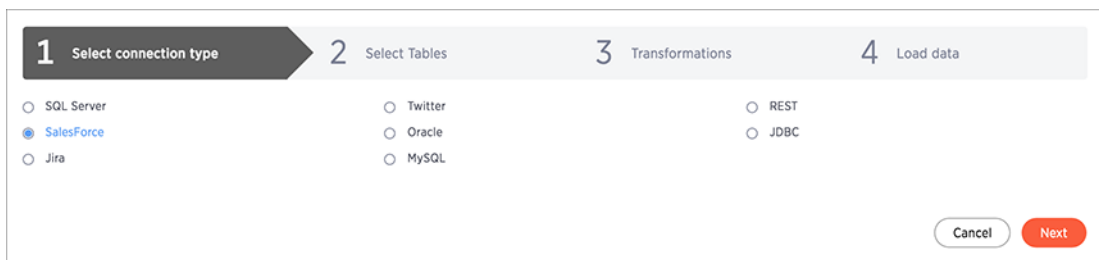5. Select a connection type and click **Next**.

Figure 4: Select a connection type

6. To select a connection, choose one of these options:

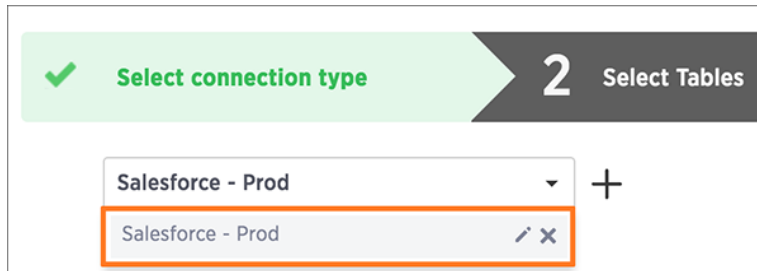- Select a pre-defined connection type, by choosing it from the **Select Connection** list:

Figure 5: Select a connection type

- Click **+** to create the connection and provide the credentials to your data source. Then click **Test & Create**:
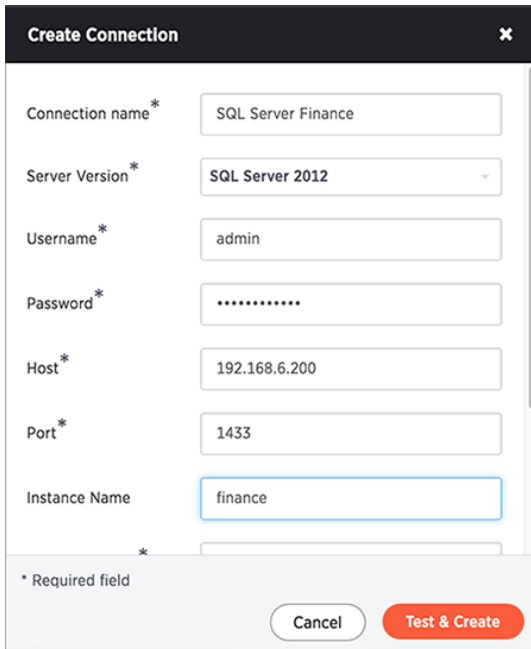

Figure 6: Create a connection

7. Choose the tables you wish to load from the list of **Available Tables**.

⊘ **Restriction:** Only the first 200 objects that occur alphabetically will be listed per schema. If you want to access a table that falls below the first 200 objects alphabetically, you must use the object search bar.
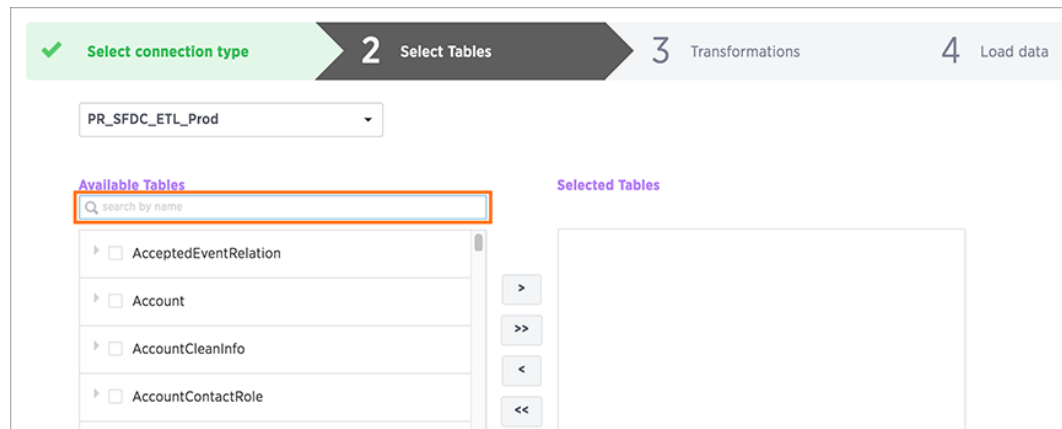
Figure 7: Object search bar

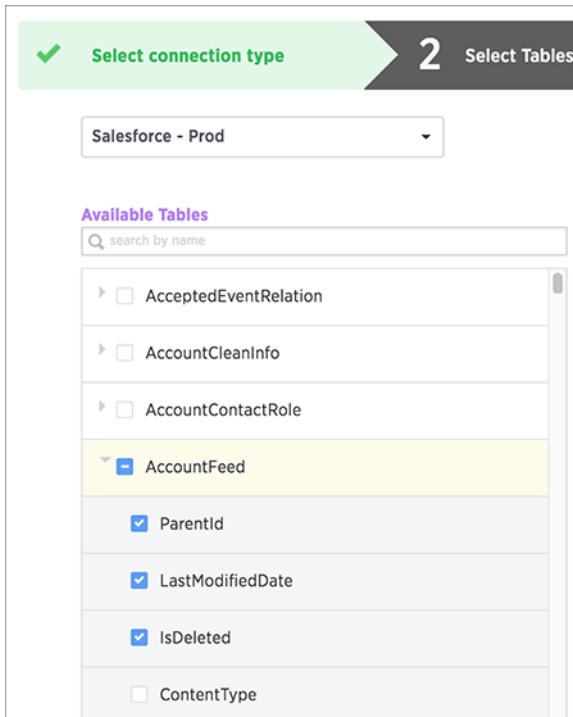Here are examples of the user and system performed search patterns for this object search bar:

- User enters KNA, then the system performs a search for %KNA%.
- User enters KNA*, then the system performs search for KNA%.
- User enters *KNA, then the system performs search for %KNA.
- User enters "KNA", then the system performs exact match search for KNA.

Select the specific columns to import.

Figure 8: Selecting tables and columns

8. When all the columns are selected, click **>** to move them to the **Selected Tables** box. Click **Next**.

9. You can optionally apply a filter to only import specific rows by clicking **New...** under **Data Filters**.

   a) Select a table to filter.

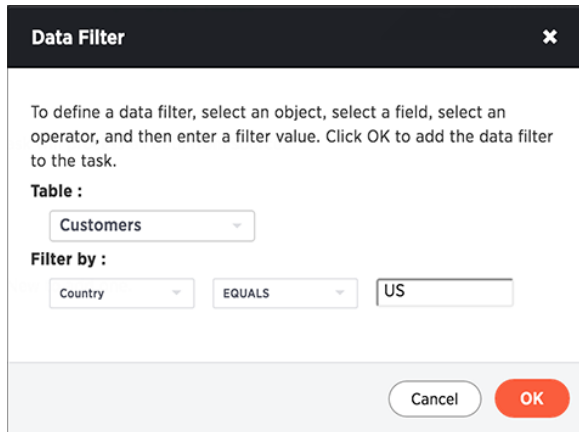   b) Select a filter by specifying a column, operator, and value.

Data Filter

To define a data filter, select an object, select a field, select an operator, and then enter a filter value. Click OK to add the data filter to the task.

Table :

Customers

Filter by :

Country   EQUALS   US

Cancel   OK

Figure 9: Apply a Data Filter

c) Click **OK**.

d) You can create as many filters as you need by clicking **+** and repeating these steps.

10. You can optionally apply data transformations and write the result to a new column by clicking **New...** under **Expressions**.

a) Give the new column an alphanumeric name (underscores are okay). You can rename it later in the ThoughtSpot application.

b) Select the table and column the data will come from.

c) Type in an expression.

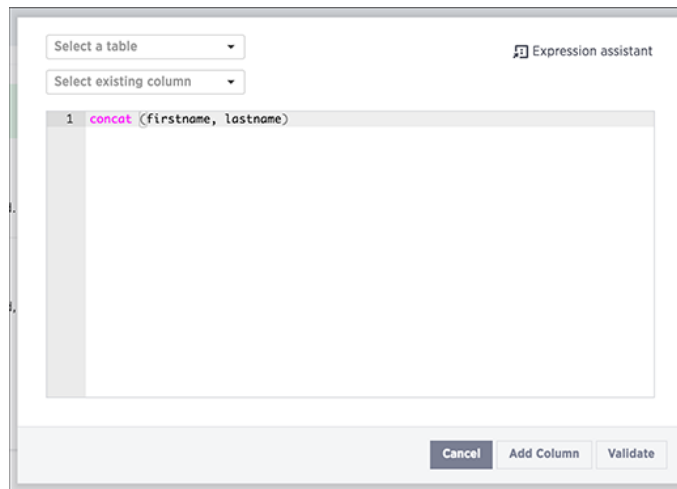You can see a list of expressions by clicking the **Expression assistant**.

Figure 10: Apply an Expression

d) Click **Add Column**.

e) Click **Validate** to validate your expression before creating the data source.

f) You can create as many expressions as you need by clicking **+** and repeating these steps.

11. Click **Next**.

12. Provide a **Source Name** for the data source.

13. Check the list of columns to be imported, to make sure it is correct. If you are loading into new or empty tables, or if you want to delete any data that already exists in the tables, click the checkbox to select **Truncate data in the existing tables.**

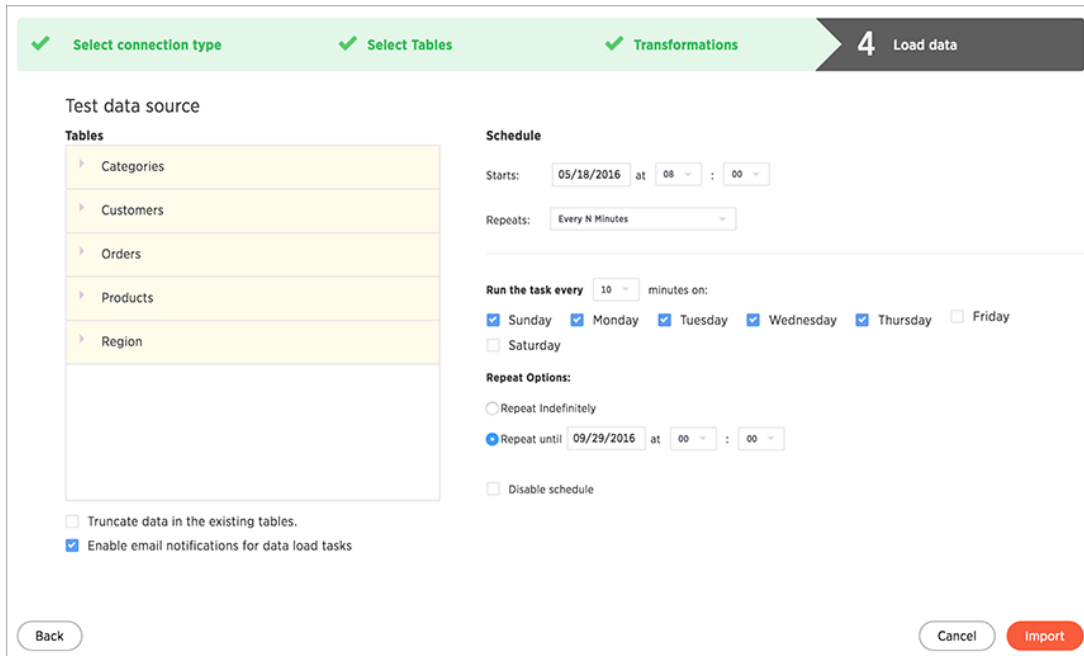14. If you want to specify repeat data loads, create a loading schedule.

Figure 11: Schedule a loading job

15. If you want to receive an email notification letting you know when your data load completes, select **Enable email notifications for data load tasks.**

> 📝 **Note:** In order to receive email notifications, you must enable **Email me sharing notifications** in your Profile Preferences. You also need to have an email set in your user properties found on the Admin User Management page. You can't send notifications to multiple email addresses.

16. Click **Import**.

17. If you want to view or change the SQL DDL (data definition language) that will create the target table, click **here**.
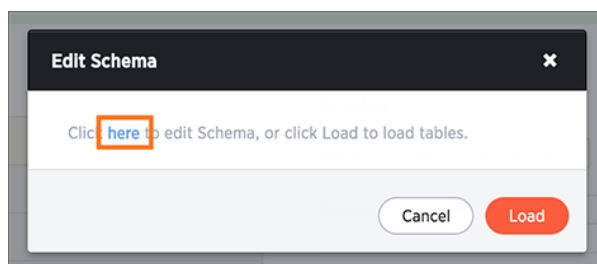
Figure 12: Edit schema here

18. Review and edit the schema as necessary. Click **Load** to load the tables.
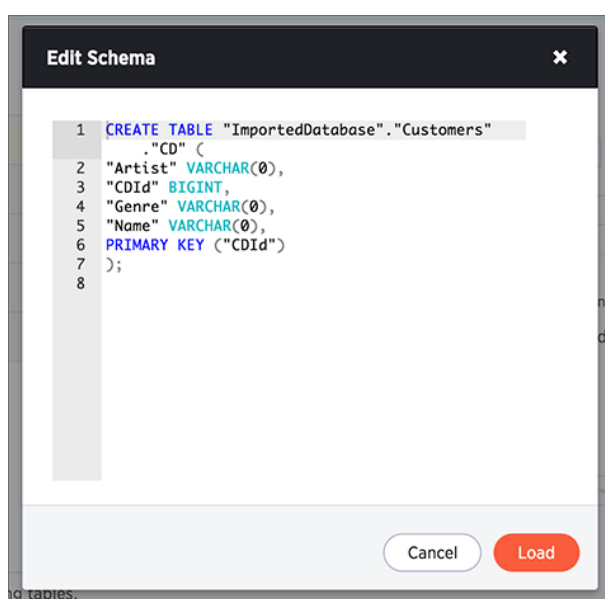
Figure 13: Edit schema view

19. Click **Load**.

Your data source will appear on the Data Sources list page, and you can find the imported tables on the Tables list page. Tables imported via ThoughtSpot Data Connect display the job name in the Source column.

About changing the schema (DDL)

You can view and edit the DDL script that gets generated by ThoughtSpot Data Connect. This allows you to review the table that will be created and make changes to the SQL that generates it.

Some common changes you can make to the DDL before you first run the data load are:

- Edit the DDL script to point to an existing table in ThoughtSpot. The columns and their order must exactly match that of the existing table. You can check the columns and their order by issuing the `SCRIPT TABLE <table_name>` command from TQL. You may need to rearranged the order of columns in the generated DDL so that it matches the order of the target table. In fact, you can even copy the output of the SCRIPT TABLE command and paste it into the DDL editing window.
- Create the table(s) in a different schema than what the default DDL suggests.
- Add sharding using the `PARTITION BY HASH (<number>) <column>` syntax.

About incremental data loads

You can create an incremental data load, with some planning ahead.

Although there is no native incremental load capability, incremental loads can be handled by adding a filter on the source data. This only works if the source has the capability of providing information on the last successful load. For example, if the source has a timestamp field that indicates when a particular source was modified, then the following filters could be used:

- modifiedDate > $LastRunDate
- modifiedTime > $LastRunTime

The $LastRunDate and $LastRunTime filters are correctly updated at runtime to reflect the date/time of the last successful job run.

You can use filters on any other column of any datatype, if it provides a way to identify the modified rows. Even if the source does not have this capability, you may still be able to do an incremental load. As long as a primary key exists on the target ThoughtSpot table, the data will be loaded using an upsert. This means that for a given row being loaded, if the primary key already exists in the target table, that row will be updated with the data in the new row. If a row
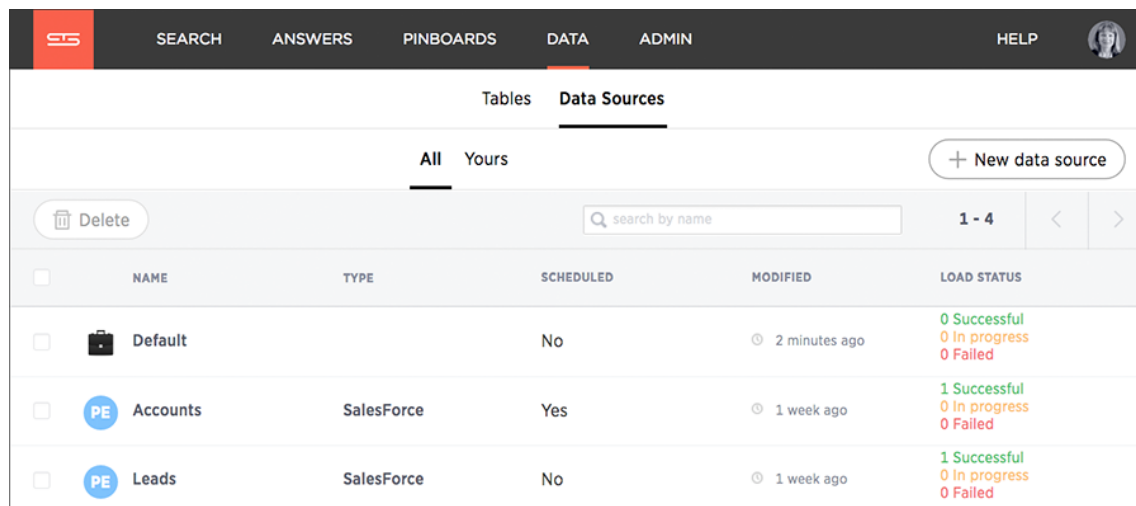
with that primary key does not exist, the new row will be inserted. This is how ThoughtSpot handles data loads into a table that already contains rows, and it is the equivalent to an incremental load. If you are relying on the upsert method, you'll want to run a COMPACT TABLE command in ThoughtSpot SQL Command Line (TQL) to reduce the size of the table at intervals (usually weekly).

Another technique that can effectively free up space would be to run a post-job script that deletes data older than a particular date. This keeps the data fresh and ensures that old data, which may not be valuable, is not taking up memory.

Data sources management

After you have successfully loaded your tables, you can check on the status of the load, make edits, and more.

You can manage your new data sources by clicking on the name of your source on the **Data Sources** page.

Figure 14: Data sources management view

Once you have selected a data source, you have the option to perform a number of editing and maintenance tasks.

Schedule, edit, and refresh

You can select a table row and click one of the following three buttons to make changes:

- **Schedule**: Opens a Scheduler menu where you can make changes to the loading schedule.
- **Edit**: Takes you back to edit the data source details, such as loading another table or truncating an existing one.
- **Refresh**: Refreshes the data load status.
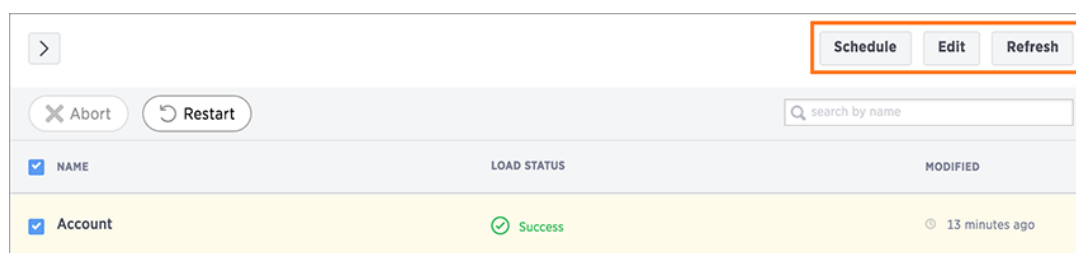
Figure 15: Schedule, edit, and refresh data sources

Restart the load

Selecting a source and clicking the **Restart** icon will immediately reload the data.
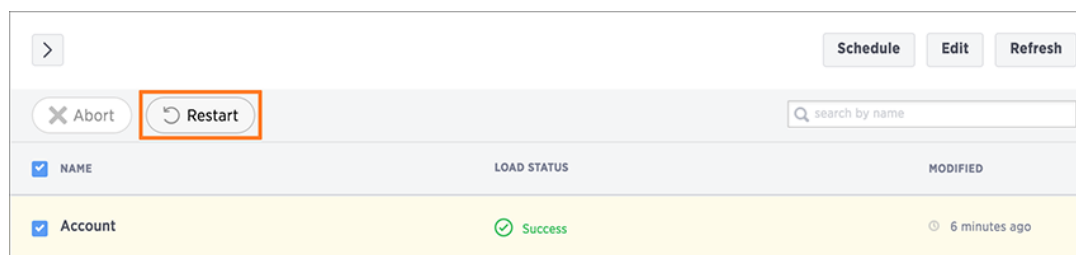
Figure 16: Restart the data source load

View log files

You can download the session logs for a data load run. Select a table row, and click **View session logs** at the bottom of the page. View the log file to troubleshoot any failed loads or to simply check the log queries.
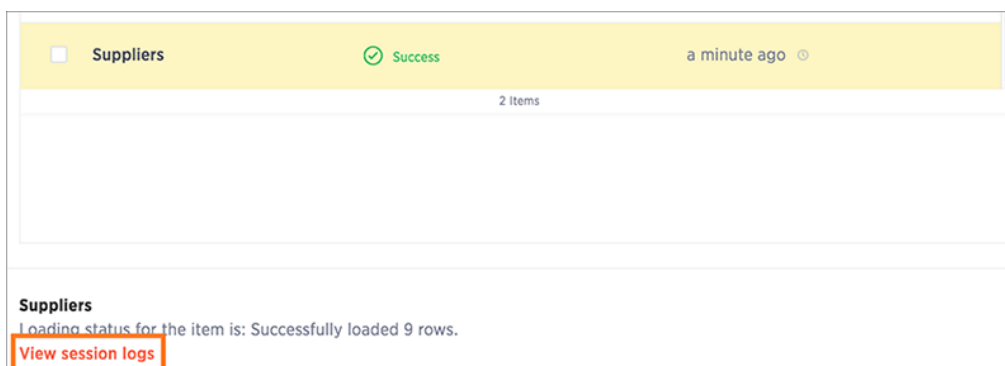
Figure 17: View session logs

Delete a data source

You can delete a data source by selecting it in the **Data Sources** page, then clicking the **Delete** icon.

Figure 18: Delete a data source

⚠️ **Attention:** Do not delete the "Default" data source. It does not correspond to any connection in ThoughtSpot Data Connect. However, this source is assigned to default datasets.

Chapter 2: About making changes to Data Connect setup

Topics:

-
-
-
-
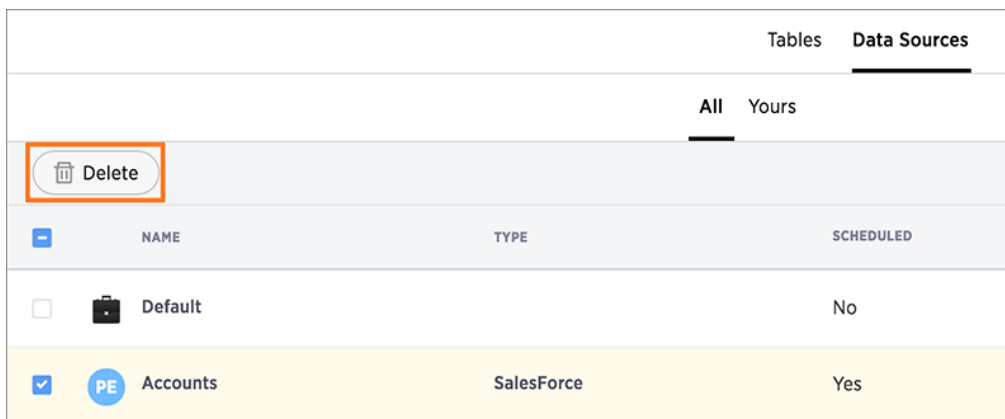-
-
-
-

After setting up ThoughtSpot Data Connect, you may want to make changes to the scheduling, data sources, or even the schema you are importing into in ThoughtSpot. These best practices describe how to make such changes.

Changes to data sources

If your company migrates from one source database or application to another you may find that you need to redo the data import using the new system. You can do this without losing any data that has already been loaded into ThoughtSpot, but you need to be careful not to accidentally delete an existing table that was created by the old data source. Create the new data source, but choose to retain the table when you delete the old data source. You should also stop any scheduled jobs associated with the old data source before deletion.

- Delete a connection
- Delete a data source

Changes to scheduled data imports

It's easy to change scheduled data imports to run at a different time or frequency through the Data Connect web interface. You may also want to pause a

data import because the source system will be offline for maintenance. You can then resume it, when the maintenance has been completed.

- Stop a scheduled job from running

Changes to the Schema in ThoughtSpot

It is very common to need to change what you're importing into ThoughtSpot for various reasons. For example the source database may have changed, because you've implemented a new application. Or you might simply change your mind about the best way to model the data that was brought in through Data Connect.

- Change sharding for a data source
- Change a data type for a data source
- Add a column to an existing data source
- Remove a column from an existing data source

Edit a connection

You can edit a connection used in a data source.

You can safely edit a connection when you create a new data source or edit an existing one.

To edit a connection:

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

Figure 19: Data

3. Click the **Data Sources** tab at the top of the screen.
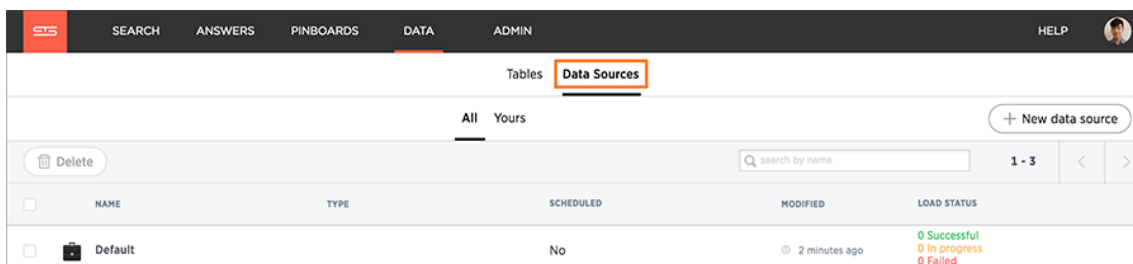
Figure 20: Data Sources tab

4. Select **+ New data source** on the upper right hand side of the panel.

Figure 21: New data source

5. Select the connection type of interest and click **Next**.

Figure 22: Select a connection type

6. In the **Select Tables** step, click the **Select a connection** dropdown and browse the list for the connection you wish to edit.

a) Click **Edit connection** on the right of your selected connection to open the Edit connection dialog box.
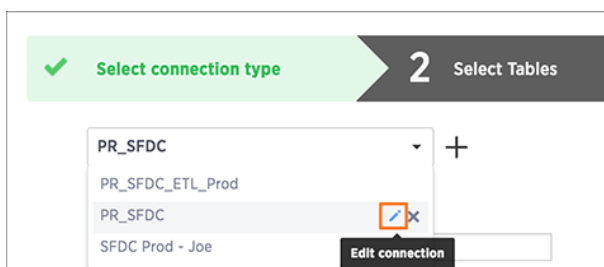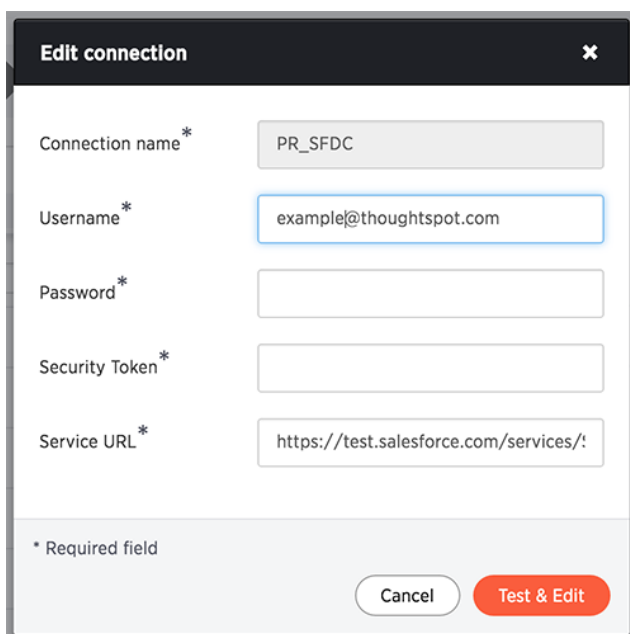
Figure 23: Edit a connection

7. Edit the connection details, then click **Test & Edit**.

Figure 24: Edit connection dialog box

Delete a connection

You can delete a connection, but only once you've resolved each of its dependencies.

A connection is not tied to a task, and can therefore be called in multiple data sources. So in order to delete a connection, you must first delete all of the sources and tasks that use that connection. Once the connection is not used in any data source, you will be able to delete it.

To delete a connection:

1. Log in to ThoughtSpot from a browser.

2. Click on **Data**, on the top navigation bar.

Figure 25: Data

3. Click the **Data Sources** tab at the top of the screen.
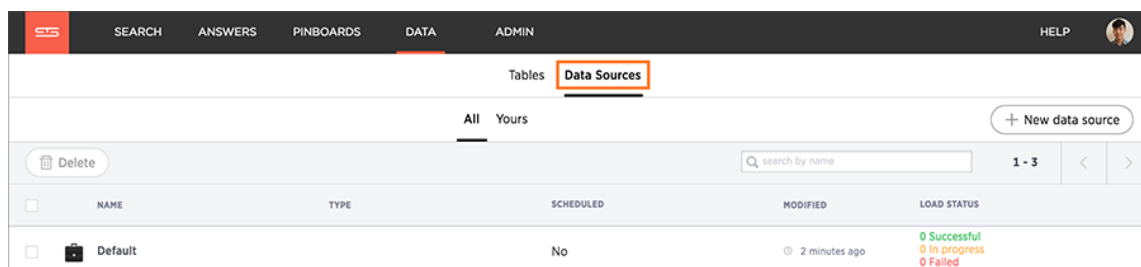
Figure 26: Data Sources tab

4. Delete any dependent data sources.

   To delete a data source, see Delete a data source.

5. Select **+ New data source** on the upper right hand side of the panel.
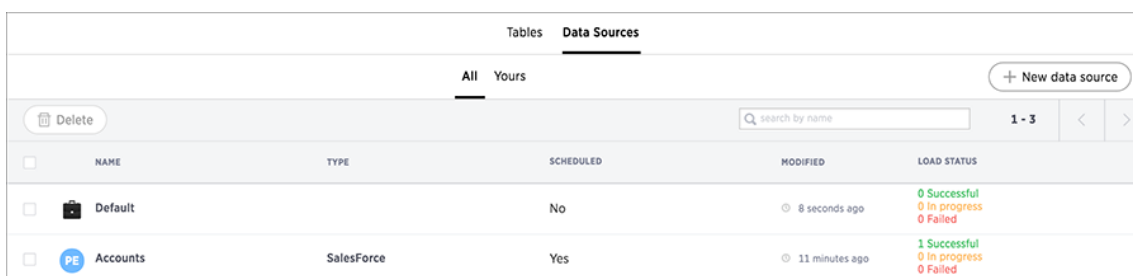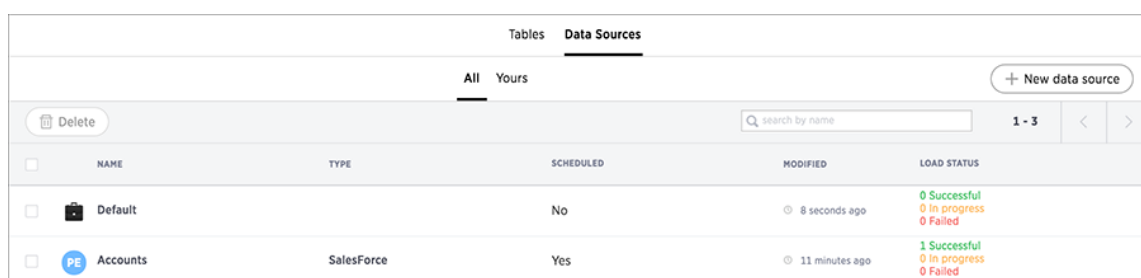

Figure 27: New data source

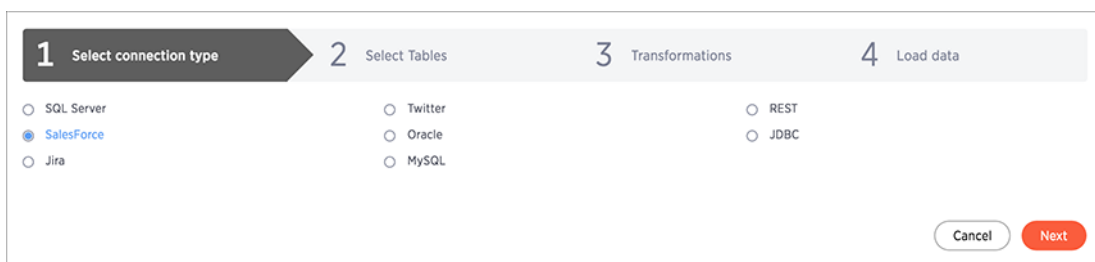6. Select the connection type of interest and click **Next**.
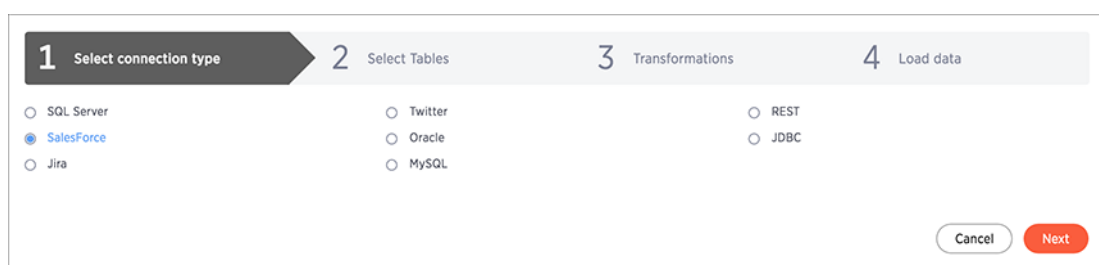
Figure 28: Select a connection type

7. In the **Select Tables** step, click the **Select a connection** dropdown and browse the list for the connection you wish to delete.

   a) Click the **x** on the right of your selected connection to delete it.
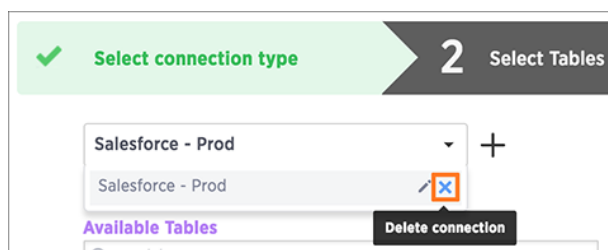
Figure 29: Delete a connection

Delete a data source

If you want to delete a data source, make sure and read this section so you don't inadvertently delete data in ThoughtSpot.

First, if there are any recurring data load jobs for the data source you are deleting, follow the steps in Stop a scheduled job from running.

Whenever you delete a data source, you also have the option to drop or retain the table it loads into in the target system (ThoughtSpot). This is useful when you make changes to a table and create a new data source that loads into the table. So, you can safely delete any data source that references a target table that you don't want to lose.

To delete a data source:

1. Log in to ThoughtSpot from a browser.

2. Click on **Data**, on the top navigation bar.

Figure 30: Data

3. Click the **Data Sources** tab at the top of the screen.
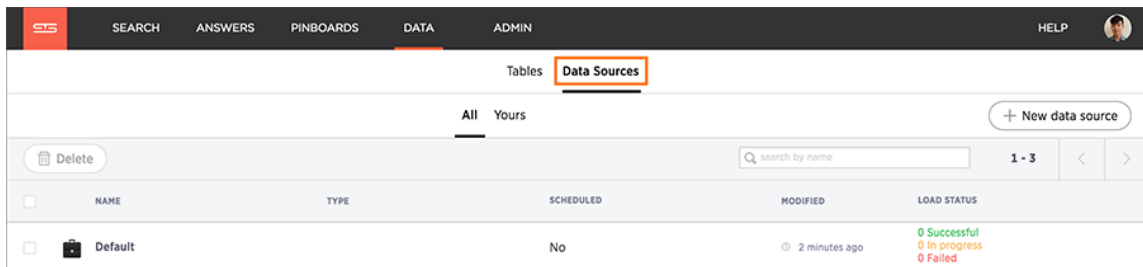
Figure 31: Data Sources tab

4. Find the data source you want to delete, and click on its name.

5. Click **Edit**. This will bring you to the setup screens for a new data source. Step through all the screens by clicking **Next**, until you reach the final page.

6. Click the **Import** button.

7. Click the **here** link to review the TQL script that created the target table in ThoughtSpot.
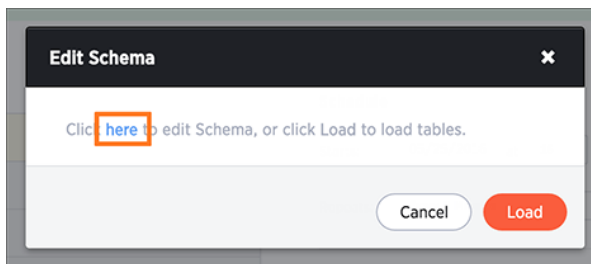
Figure 32: Edit schema here link

8. Take a look at the table name and its schema. Decide if the table is one you want to keep in ThoughtSpot.

9. Go back to the data sources listing, click the checkbox next to the data source, and delete it.

Figure 33: Delete a data source

10.In the Confirm delete prompt, select the checkbox if you want to delete the tables associated with the data source. Leave the checkbox unselected if you have decided to keep the tables.

Figure 34: Delete or retain associated tables

Stop a scheduled job from running

You may want to stop a scheduled job from running for a time, or even permanently. Use this procedure to pause or stop a job.

There are a few conditions when it may be useful to stop or pause a scheduled job:

• The source system will be offline for maintenance.

• You are migrating the data to a new source system, and don't want to delete the original job.

• There is a problem with the job, and you don't want it to run while you are figuring out how to solve it.

- ThoughtSpot will be offline for scheduled maintenance or an upgrade.

- You want to delete the data source on which the scheduled job runs.

To stop a scheduled job from running:

1. Log in to ThoughtSpot from a browser.

2. Click on **Data**, on the top navigation bar.

Figure 35: Data

3. Click the **Data Sources** tab at the top of the screen.
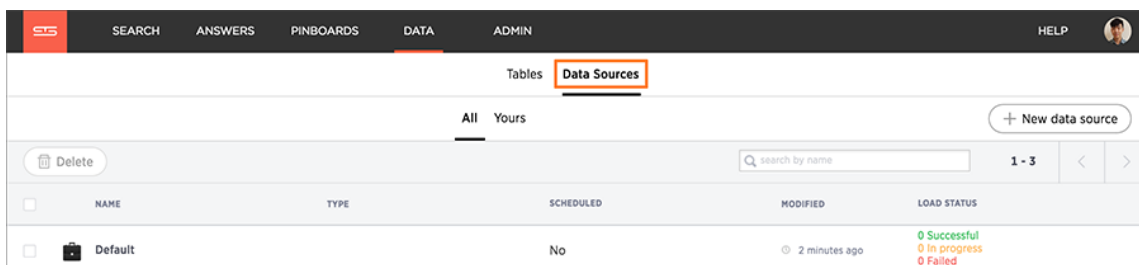
Figure 36: Data Sources tab

4. Click on the data source for the scheduled job you want to pause.

5. Click **Schedule** to edit the schedule.

6. Stop the job by selecting **Disable schedule**.
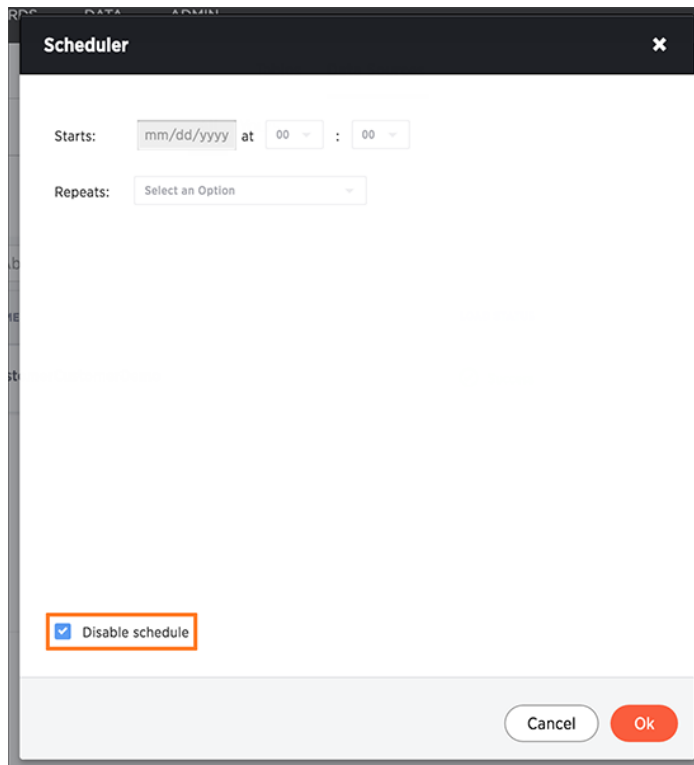
Figure 37: Stop a recurring data load

7. Click **OK**.

Change sharding for a data source

Use this procedure to change the sharding on a data source after it has been set up and the load has run. You can still change the number of shards or make it a replicated table instead.

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

After you've set up a Data Source, you may decide that you want the table it is loading into to have a different type of sharding. For example, if you specified 96 shards, but then you expand your cluster, you may want to specify a large

number or shards. You can even change a sharded table to a replicated table, or vice versa, by using this procedure.

First, you'll modify the table sharding, using TQL. Note that this operation changes the target table itself. You do not need to change the DDL that was generated when you first configured the data source. That DDL executes only one time, to build the empty table into which the data gets loaded. So if you change the sharding on the table in ThoughtSpot, you do not need to change the DDL that got generated when you first set up the data source. Likewise, if the load is scheduled to recur, the sharding that is defined on the target table will be used in the next load. Sharding is a one time setup, and does not need to be kept in sync with the data source DDL.

To change the sharding on a Data Source:

1. Log in to the Linux shell using SSH.

2. Launch TQL.

   ```
   $ tql
   ```

3. Designate the database you'll be working in:

   ```
   TQL> use <database_name>;
   ```

4. Find the name of the table you want to change.

   You can use the TQL command SHOW TABLES; to see a list of tables.

   To see the current sharding on the table, use SCRIPT TABLE <table_name>;

5. Issue the command to change the sharding using this syntax:

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

   For example:

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

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

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

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

6. The next time the data source load runs, it will use the new sharding automatically.

Change a data type for a data source

When a data type used in a data source has changed, or you decide that the data should come into ThoughtSpot with a different data type, you can change the data type, even if you've already loaded data into the target table.

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

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

The procedure to make this change to an existing data source has two main parts:

- Alter the table definition in the target table using TQL.

- Create a new data source, and edit the DDL to match the target table exactly.

To change the data type of an existing data source:

1. Log in to the Linux shell using SSH.

2. Launch TQL.

```
$ tql
```

3. Designate the database you'll be working in:

```
TQL> use <database_name>;
```

4. Find the name of the table you want to change.

You can use the TQL command SHOW TABLES; to see a list of tables.

To see the current sharding on the table, use SCRIPT TABLE <table_name>;

5. Issue the command to change the data type of a column using this syntax:

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

For example, to change "account_id" with data type INT to data type VARCHAR:

```
TQL> ALTER TABLE foodmart.account
     MODIFY COLUMN account_id VARCHAR(10);
```

You must use the fully qualified name of the old table when adding the column via TQL. To find that you can look at the DDL for the data source job itself.

6. Run the SCRIPT TABLE command to get the new DDL to create the table.

```
TQL> SCRIPT TABLE <table>;
```

Copy the output of the command. Replace any VARCHAR(<*number*) column definitions with VARCHAR(0), to match the DDL that Data Connect generates. This is the DDL that you will use in your new data source.

7. Create a new data source.

Be sure to choose the correct columns to match the new target table columns definitions. When you reach the step about editing the generated schema DDL, paste in the DDL that was output by the `SCRIPT TABLE` command.

8. Run the data load and verify that everything is working as it should be.

9. If the old data source was running as a recurring load, stop it from running.

Add a column to an existing data source

You can add a column to an existing data source using this procedure. This will work even if the data load has already been run in the past.

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

Reasons you may want to add a column to an existing data source might include:

• A new column has been added to the source table, and you want to include it in the data import.

• You've decided to include an additional column from the source.

Initial DDL did not include a column, but you wanted to pull it in later in a subsequent load. Or it could be a new column added to the source.

The procedure to make this change to an existing data source has two main parts:

• Alter the table definition in the target table using TQL.

• Create a new data source, and edit the DDL to match the target table exactly.

1. Log in to the Linux shell using SSH.

2. Launch TQL.

```
$ tql
```

3. Designate the database you'll be working in:

```
TQL> use <database_name>;
```

4. Find the name of the table you want to change.

   You can use the TQL command SHOW TABLES; to see a list of tables.

   To see the current sharding on the table, use SCRIPT TABLE <table_name>;

5. Issue the command to change the data type of a column using this syntax:

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

   For example, to change "account_id" with data type INT to data type VARCHAR:

```
TQL> ALTER TABLE foodmart.account
     ADD COLUMN account_id VARCHAR(10);
```

   You must use the fully qualified name of the old table when adding the column via TQL. To find that you can look at the DDL for the data source job itself.

6. Run the SCRIPT TABLE command to get the new DDL to create the table.

```
TQL> SCRIPT TABLE <table>;
```

   Copy the output of the command. Replace any VARCHAR(<*number*) column definitions with VARCHAR(0), to match the DDL that Data Connect generates. This is the DDL that you will use in your new data source.

7. Create a new data source.

   Be sure to choose the correct columns to match the new target table columns definitions. When you reach the step about editing the generated schema DDL, paste in the DDL that was output by the SCRIPT TABLE command.

8. Run the data load and verify that everything is working as it should be.

9. If the old data source was running as a recurring load, stop it from running.

Remove a column from an existing data source

You can remove a column from a data source, even if its data load has been run in the past, by using this procedure.

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

The procedure to make this change to an existing data source has two main parts:

- Alter the table definition in the target table using TQL.
- Create a new data source, and edit the DDL to match the target table exactly.

1. Log in to the Linux shell using SSH.

2. Launch TQL.

   ```
   $ tql
   ```

3. Designate the database you'll be working in:

   ```
   TQL> use <database_name>;
   ```

4. Find the name of the table you want to change.

   You can use the TQL command `SHOW TABLES;` to see a list of tables.

   To see the current sharding on the table, use `SCRIPT TABLE <table_name>;`

5. Issue the command to remove the column from the table using this syntax:

   ```
   TQL> ALTER TABLE <schema>.<table>
        DROP COLUMN <column>;
   ```

   For example, to drop the column "account_id" from the table "account" in the schema "foodmart":

   ```
   TQL> ALTER TABLE foodmart.account
        DROP COLUMN account_id;
   ```

You must use the fully qualified name of the old table when adding the column via TQL. To find that you can look at the DDL for the data source job itself.

6. Run the `SCRIPT TABLE` command to get the new DDL to create the table.

```
TQL> SCRIPT TABLE <table>;
```

Copy the output of the command. Replace any VARCHAR(<*number*) column definitions with VARCHAR(0), to match the DDL that Data Connect generates. This is the DDL that you will use in your new data source.

7. Create a new data source.

Be sure to choose the correct columns to match the new target table columns definitions. When you reach the step about editing the generated schema DDL, paste in the DDL that was output by the `SCRIPT TABLE` command.

8. Run the data load and verify that everything is working as it should be.

9. If the old data source was running as a recurring load, stop it from running.

Chapter 3: Data Connect reference

Topics:

- Login credentials for administration
- Data Connect best practices
- Data Connect connectors reference

This section contains a reference for ThoughtSpot Data Connect.

Login credentials

This section is included for convenience, so that if you are setting up Data Connect, and are not familiar with how to log in to ThoughtSpot from a browser or in the Linux shell, you can find the information easily.

Best practices

This section suggests some best practices, introduces some troubleshooting techniques, and lists things to watch out for.

Data Connect connectors reference

This section contains a list of Tier 1 and Tier 2 connectors supported by ThoughtSpot Data Connect.

Login credentials for administration

You will need administrative permissions to perform the actions discussed in this guide. You can access ThoughtSpot via SSH at the command prompt and from a Web browser.

There are two separate default administrator users, an operating system user that you type in at the Linux shell prompt, and an application user for access through a browser. Make sure you use the correct login and password for the method you are using to log in. Passwords are case sensitive.

Table 1: Default administrative user credentials

Login Type	User	Access Method	Password
OS user	admin	Access remotely via SSH from the command prompt on a client machine.	Contact ThoughtSpot to obtain the default password.
Application user	tsadmin	Access through a Web browser.	Contact ThoughtSpot to obtain the default password.

Log in to ThoughtSpot from a browser

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

Before accessing ThoughtSpot, you need:

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

  Supported Web browsers include:

Table 2: Supported browsers

Browser	Version	Operating System
Google Chrome	20 and above	• Windows 7 or greater • Linux • MacOS
Mozilla Firefox	14 and above	• Windows 7 or greater • Linux • MacOS
Internet Explorer	10 and 11	• Windows 7 or greater

To log in to ThoughtSpot from a browser:

1. Open the browser and type in the Web address for ThoughtSpot:

   `http://<hostname_or_IP>`

2. Enter your username and password and click **Enter Now**.

Log in to the Linux shell using SSH

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

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

1. Log in to a client machine and open a command prompt.

2. Issue the SSH command, specifying the IP address or hostname of the ThoughtSpot instance:

   `ssh admin@<hostname_or_IP>`

3. Enter the password for the admin user.

Data Connect best practices

This section includes some tips and best practices for using ThoughtSpot Data Connect.

Status reporting

You can see the status of a data load job by looking at the list of data sources. The data load status gets updated every 30 seconds.

Figure 38: Data load status

To see loading details for each table that makes up a data load, click on the name of the data load.

Figure 39: Data load status details

View log files

You can download the session logs for a data load run. Access the link to download the log on the data load status details page for the session. Check the log files for a successful or failed run, and fix the cause of the failure before resubmitting the job.

Figure 40: Download session log

Import relationships

Relationships (primary key/foreign key) can be imported from the source database. For this to work, the primary and foreign key(s) need to be made up of only one column each. That is, they cannot be composite keys. You will also need to include all the tables whose relationships you want to bring over into ThoughtSpot within a single data source.

Edit the DDL script to choose a different database, schema, or table

You can edit the SQL DDL (data definition language) script generated by ThoughtSpot Data Connect to point to an existing table in ThoughtSpot. You can then load to that table and/or create the table in a different database/schema than what the default DDL suggests. To do this, the column names and the order of the column names must match exactly with the table you are importing into in ThoughtSpot.

No need to import a column mentioned in an expression

It is not necessary to import a column that is mentioned in an expression. For example, let's say you have a source table "sales_fact" with a column "revenue".

You may want to use that column as part of an expression that calculates gross margin. But maybe you don't want to include the column "revenue" in the target table. You can unselect the column "revenue" in the table selection page, and then add your gross margin expression that references the column "revenue", even though it isn't part of the data to be loaded. Your expression will still work.

Internet connectivity

The only internet access the needed for ThoughtSpot Data Connect to operate is outbound access to https://app.informaticaondemand.com and https://app2.informaticacloud.com/ (port 443). Of course, it also needs any other internet connectivity required to access any cloud data sources, like Salesforce. No inbound access is needed.

Renaming columns and table

You can rename columns and tables after a data load, using the ThoughtSpot web application. Select Data and click on the name of the table. Then click on a column to rename it.

In older versions of ThoughtSpot Data Connect, you could not use characters other than alphanumeric and underscores. Now, you can change a column name to use those, for example you could change "percent_gross_margin" to "% gross margin".

Troubleshooting

To view details about error conditions, you can look at the trace file that is generated when there is a red bar error.

Possible issues that can occur are:

- Cannot connect to data source.
- Authentication was rejected.
- Schema being loaded does not match schema receiving the load.

Best practices

Unless you're trying to bring in relationships between tables with a data source, the recommendation is to create one data source load job per table. Doing this makes your data sources more flexible for if/when you need to change something (add a column, etc.)

If you want to add a foreign key, do it using the TQL command ALTER TABLE, after the load has completed. This is often easier than trying to get relationships to come in with a data load, and it works even if the relationship you want to create does not exist in the source system. If use this method, you'll need to manually create the relationships using a TQL script, which you can run afterward the first data load.

Derived column names

When you create a derived column, further activities in ThoughtSpot Data Connect will show you the target column name, since there is no analogous column in the source.

Compact tables to save memory

If you're doing incremental loads into existing tables, you may want to automate a script that runs the TQL command COMPACT TABLE either after each incremental load, or periodically (like weekly, for a nightly load). This is because when upserts are performed, space from the old rows is not released. The delta between each row that gets upserted is retained and takes up memory.

Expressions

You can validate your expressions when using ThoughtSpot Data Connect. Make sure to click **Validate** after entering your expression. This prevents you from entering an invalid expression that will fail at runtime.

When using the expression builder `concat`, you need to use single quotes instead of double quotes around each of the strings.

Connection parameters

Informatica provides documentation on each of its connectors. This contains information on the connection properties and what they mean for each connector. You can see the connection properties in ThoughtSpot Data Connect by mousing over the key value pairs, however, you won't be able to see their meanings. For that, you should refer to the Informatica documentation.

Data Connect connectors reference

This is a list of all the connectors that are supported in ThoughtSpot Data Connect. Up to four Tier-1 Connectors and up to 10 Tier-2 Connectors can be enabled at a time.

Tier-1 connectors

These connectors provide connectivity to your most important enterprise applications:

- ACORD (2012)
- ACORD XML (2012)
- Amazon Dynamo DB
- Amazon RDS - Aurora
- Amazon RedShift
- Amazon S3
- Amazon SQS
- AMQP
- ANSI X12 EDI
- Apache ActiveMQ
- DTCC

- EMC Greenplum

- FpML 5.4

- HIPAA

- HIX

- HL7 v2

- IBM Netezza

- IBM WebSphere AppServer

- IBM WebSphere MQ

- Marketo (V2)

- Microsoft Azure Blob

- Micro

- soft Azure Bus

- Microsoft Azure DW

- Microsoft Azure SQL DB

- Microsoft Dynamics AX 2012

- Microsoft Dynamics CRM

- Microsoft Dynamics NAV 2009, 2013

- Microsoft Word

- MSMQ

- NACHA

- Netsuite

- Oracle CRM On Demand

- Oracle Fusion Middleware

- Oracle InterfaceTable

- Oracle JD Edwards EnterpriseOne

- Oracle PeopleSoft 9.x

- Oracle Siebel

- Salesforce CRM (Sales and Service Cloud)

- Salesforce Marketing Cloud

- Salesforce Wave Analytics

- SAP BAPI

- SAP Business One

- SAP Business Works

- SAP Hana

- SAP IDOC

- SAP Industry Solutions

- SAP Table Reader/ Writer

- SAS

- SEPA

- SWIFT

- Tableau V2

- Teradata

- TIBCO

- UDO

- UN/EDIFACT

- UNIFI (ISO 20022)

- Veeva

- Vertica

- WebMethods

- WebSphere MQ

Tier-2 connectors

These connectors provide connectivity to practically any enterprise data source:

- Adobe PDF

- Amazon RDS - MS SQL Server

- Amazon RDS - My SQL

- Amazon RDS - Oracle

- Apache Hadoop HDFS

- Atlassian Jira

- Avature

- Bazaarvoice

- Birst

- Box

- Coupa

- CVent

- Dropbox

- File I/O

- File Listener and R/W

- FileList

- FileProcessor

- Flat File

- FTP/sFTP

- Google API

- IBM DB2

- IBM Informix

- Informatica Rev

- Intuit Quickbooks

- Java POJO

- JBoss AppServer

- Jboss Messaging

- JDBC

- JMS

- JSON

- LDAP

- Linkedin

- Microsoft Dynamics GP 2010

- Microsoft PowerPoint

- Microsoft Sharepoint (Active Directory)

- Microsoft SQLServer

- Microsoft team foundation server 2010

- MySQL

- NetSuite FileCabinet

- NetSuite OpenAir

- OData (Consumer)

- ODBC Generic

- Oracle BigMachines

- Oracle DB

- Oracle EBS 12.x

- Oracle Eloqua BULK

- Oracle Eloqua SOAP

- Oracle Essbase

- Oracle Taleo Business

- Oracle Taleo Enterprise

- PostgreSQL

- Progress OpenEdge

- REST Services

- Salesforce Chatter

- SAP Concur

- SOAP WebServices

- StrikeIron

- Sugar CRM

- Sybase ASE

- Sybase IQ

- Twitter

- Ultra Messaging

- Vindicia

- Workday V1

- Xactly

- XML source/ target

- Zendesk

- Zoho

- Zuora

- Zuora Aqua