PostgreSQL Export Integration

Learn more about PostgreSQL Import Integration.

You can export job results from Treasure Data to your existing PostgreSQL instance.

For sample workflows of PostgreSQL, view Treasure Boxes.

Continue to the following topics:

- Prerequisites
- Use the TD Console to Create Your Connection
    - Create a New Connection
- Configure Results Export to Your PostgreSQL Instance
    - Set the Export Result Parameters
    - Run Query and Check Results
    - Optional: Use of Scheduled Jobs for Output
    - Execute the Query
    - Optional: Configure Export Results in Workflow

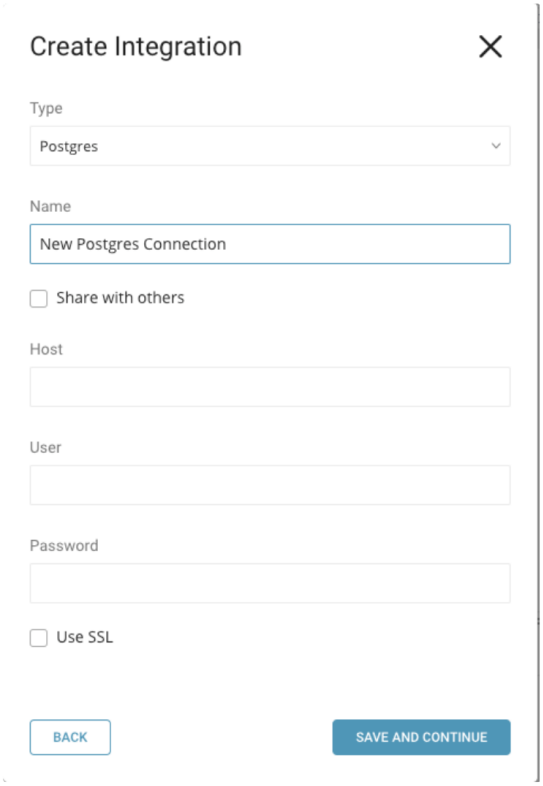
Prerequisites

- Basic knowledge of Treasure Data, including the TD Toolbelt.
- A **PostgreSQL instance**.

Use the TD Console to Create Your Connection

Create a New Connection

Complete the field values to create a new connection.



Enter the required credentials for your new PostgreSQL connection. Set the following parameters:

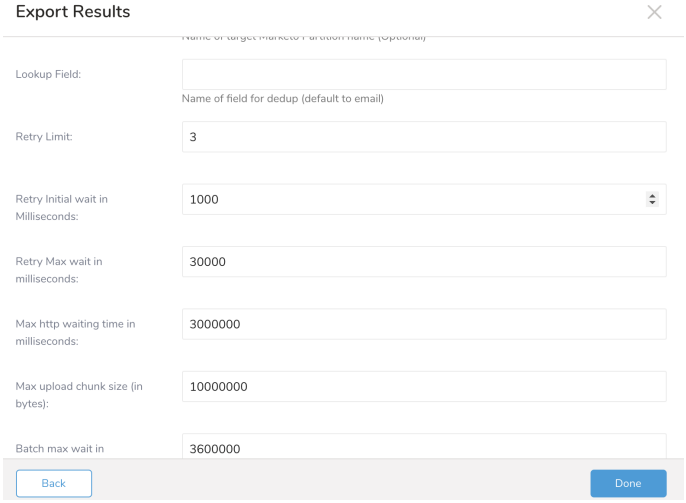
- **Host**: The host information of the source database, such as an IP address.
- **User**: Username to connect to the source database.
- **Password**: Password to connect to the source database.
- **Use SSL**: Check this box to connect using SSL
    - **Require a valid SSL certificate?:** Require that a valid SSL certificate is presented on the connection.

Configure Results Export to Your PostgreSQL Instance

Export from Treasure Data uses queries. You create or reuse a query. In the query, you configure the data connection.

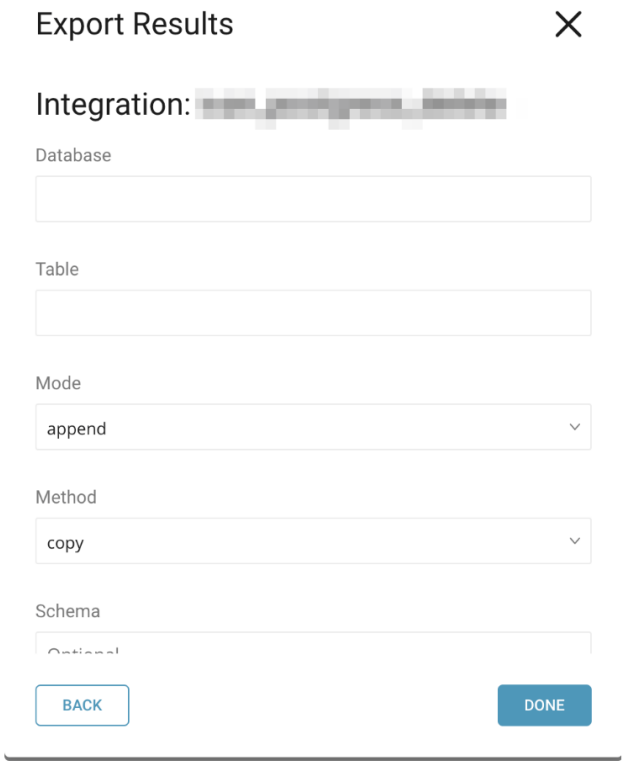
1. Complete the instructions in Creating a Destination Integration.
2. Navigate to **Data Workbench > Queries**.
3. Select a query for which you would like to export data.
4. Run the query to validate the result set.
5. Select **Export Results**.
6. Select an existing integration authentication.

7. Define any additional Export Results details. In your export integration content review the integration parameters. For example, your Export Results screen might be different, or you might not have additional details to fill out:



8. Select **Done**.
9. Run your query.
10. Validate that your data moved to the destination you specified.

Set the Export Result Parameters



Database name: The name of the database you are transferring data to. (Ex. `your_database_name`)

Table: The table to which you would like to export the data.

Output mode. Different methods to upload the data.

- **Append** (default): The **append** mode is the **default** that is used when no mode option is provided in the URL. In this mode, the query results are appended to the table. If the table does not exist, it is created. This mode is atomic.
- **Replace**: The **replace** mode consists of replacing the entire content of an existing table with the resulting output of the query. If the table does not exist yet, a new table is created. The replace mode achieves **atomicity** (so that a consumer of the table always has consistent data) by performing the following three steps in a **single transaction**:
    1. Create a temporary table;
    2. Write to the temporary table;
    3. Replace the existing table with the temporary table using ALTER TABLE RENAME.
- **Truncate:** the system first truncates the existing table, then inserts the query results. If the table does not exist yet, a new table is created. This mode is atomic.
- **Update:** a row is inserted unless it would cause a duplicate value in the columns specified in the “unique” parameter: in such case, an update is performed instead. The “unique” parameter is required when using the update mode. This mode is atomic.

Insert Method. This option controls how the data is written into the Postgres table. The default method is **copy**; it is also the recommended method for most situations.

- **Copy** (default): Data is first stored in a temporary file on the server, then written to Postgres using a COPY transaction. This method is faster than INSERT, so it is useful when handling a large amount of data.
- **Insert**: Data is written to Postgres using ‘INSERT’ statements. This is the most reliable and compatible method and it is recommended for most situations.

Schema: Defines the schema where the target table is located. If not specified, the default schema is to be used. The default schema depends on the user’s “search_path” setting but it is usually “public”.

Foreign Data Wrapper: This option controls whether or not a data wrapper is used to store the data. The default is none and should work in most instances.

- **None** (default) - No foreign-data wrapper.
- **Cstore** - used when columnar storage is required/enabled on the destination table.

Run Query and Check Results

Run the query with **Output Results** selected. If the query completes successfully, you see the results in the PostgreSQL database and table that you specified when entering the transfer details.

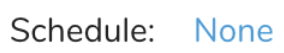
Optional: Use of Scheduled Jobs for Output

You can use Scheduled Jobs with Result Output, to periodically write the output result to a PostgreSQL instance that you specify.

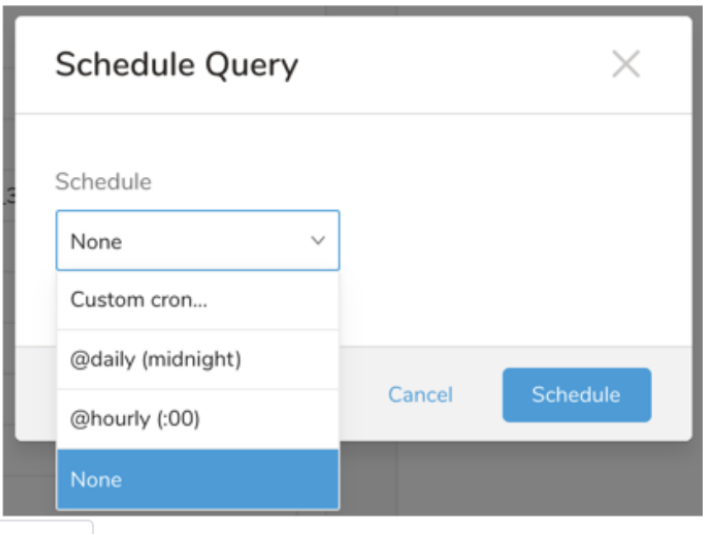
Navigate to **Data Workbench > Queries**.

Create a new query or select an existing query.

Next to **Schedule**, select None.

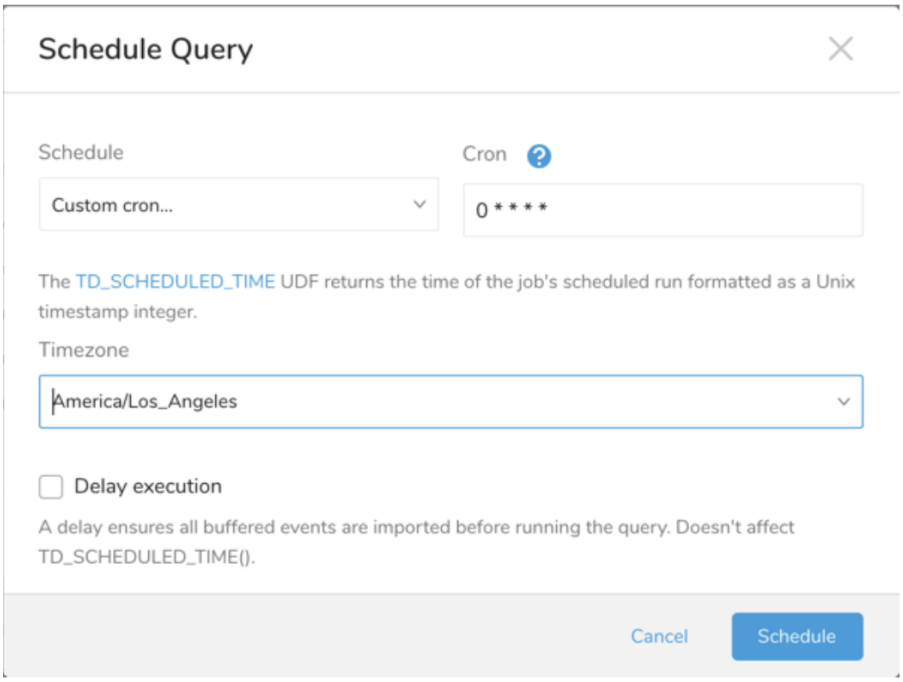


In the drop-down, select one of the following schedule options:



Drop-down Value	Description
Custom cron...	Review Custom cron... details.
@daily (midnight)	Run once a day at midnight (00:00 am) in the specified time zone.
@hourly (:00)	Run every hour at 00 minutes.
None	No schedule.

Custom cron... Details



Cron Value	Description
`0 * * * *`	Run once an hour.
`0 0 * * *`	Run once a day at midnight.
`0 0 1 * *`	Run once a month at midnight on the morning of the first day of the month.
""	Create a job that has no scheduled run time.

```
 *    *    *    *    *
 -    -    -    -    -
 |    |    |    |    |
 |    |    |    |    +----- day of week (0 - 6) (Sunday=0)
 |    |    |    +---------- month (1 - 12)
 |    |    +--------------- day of month (1 - 31)
 |    +-------------------- hour (0 - 23)
 +------------------------- min (0 - 59)
```

The following named entries can be used:

- Day of Week: sun, mon, tue, wed, thu, fri, sat.
- Month: jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec.

A single space is required between each field. The values for each field can be composed of:

Field Value	Example	Example Description
A single value, within the limits displayed above for each field.		
A wildcard `'*'` to indicate no restriction based on the field.	`'0 0 1 * *'`	Configures the schedule to run at midnight (00:00) on the first day of each month.
A range `'2-5'`, indicating the range of accepted values for the field.	`'0 0 1-10 * *'`	Configures the schedule to run at midnight (00:00) on the first 10 days of each month.
A list of comma-separated values `'2,3,4,5'`, indicating the list of accepted values for the field.	`0 0 1,11,21 * *'`	Configures the schedule to run at midnight (00:00) every 1st, 11th, and 21st day of each month.

A periodicity indicator `'*/5'` to express how often based on the field's valid range of values a schedule is allowed to run.	`'30 */2 1 * *'`	Configures the schedule to run on the 1st of every month, every 2 hours starting at 00:30. `'0 0 */5 * *'` configures the schedule to run at midnight (00:00) every 5 days starting on the 5th of each month.
A comma-separated list of any of the above except the `'*'` wildcard is also supported `'2,*/5,8-10'`.	`'0 0 5,*/10,25 * *'`	Configures the schedule to run at midnight (00:00) every 5th, 10th, 20th, and 25th day of each month.

(Optional) You can delay the start time of a query by enabling the Delay execution.

Execute the Query

Save the query with a name and run, or just run the query. Upon successful completion of the query, the query result is automatically imported to the specified container destination.

> Scheduled jobs that continuously fail due to configuration errors may be disabled on the system side after several notifications.

Optional: Configure Export Results in Workflow

Within Treasure Workflow, you can specify the use of this data connector to output data.

```
timezone: UTC

_export:
  td:
    database: sample_datasets

+td-result-output-postgresql:
  td>: queries/sample.sql
  result_connection: your_connections_name
  result_settings:
   database: database_name
   table: table_name
   mode: append
   set_role:new role
```

Read about using data connectors in a workflow to export data. See an example workflow.