JOB AID

Working with databases

Overview

In this module, you learned how to perform basic operations on databases in Talend Studio, specifically, how to create a table, insert data, and query data.

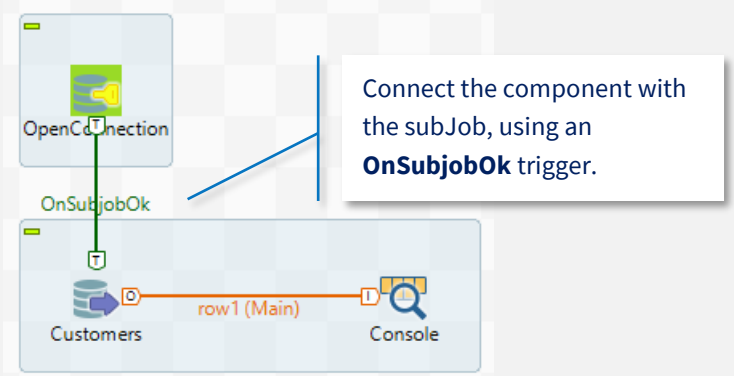
Key steps

Databases have a limited number of simultaneous connections. In a Job, a database connection can be shared between components. The connection must be closed when the Job ends.

1

Open the connection using a **tDBConnection** component.

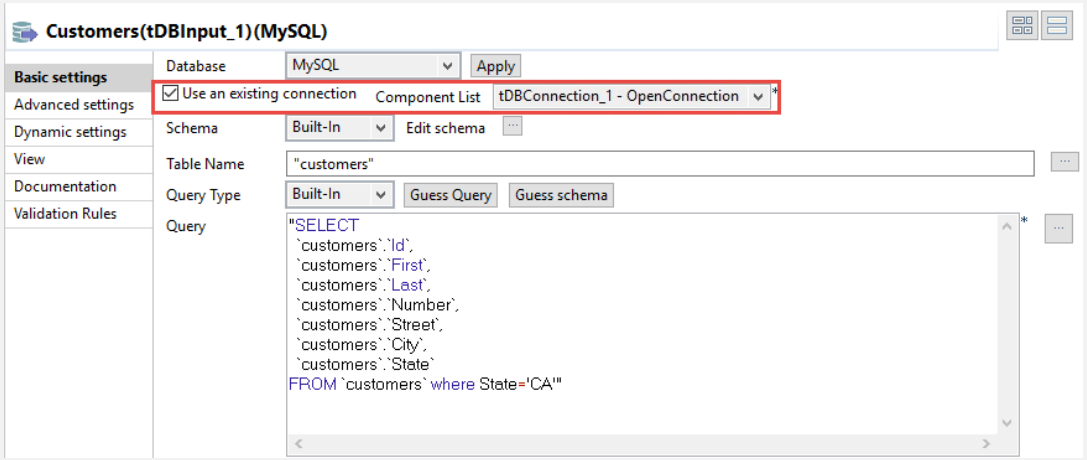
- Place the component in the **Designer**.
- Select a DB type.
- Configure the database credentials.



2

When configuring a database component, you can easily reuse the connection.

- Select the **Use an existing connection** check box.
- On the drop-down **Component List**, select the connection you created earlier.



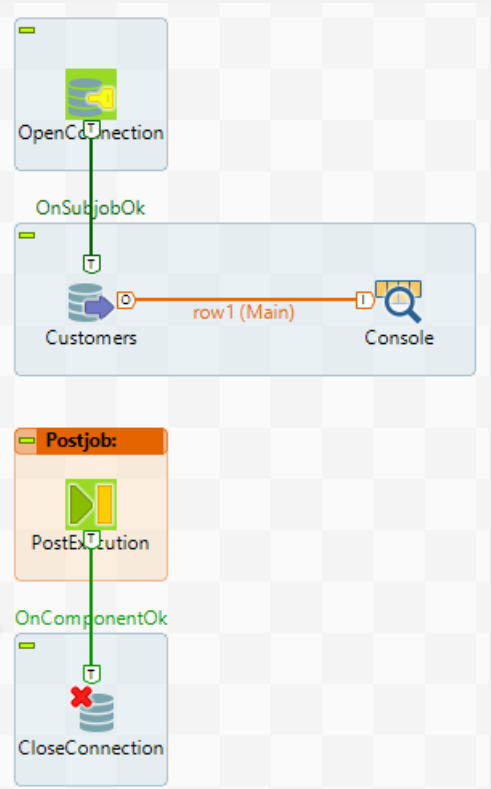
JOB AID

3

To follow best practices when working with databases, you must close the connection using a **tDBClose** component.

To close the connection even if the Job fails, **tDBClose** must be called from a **tPostJob** component.

- Add a **tPostJob** component below the last component.
- Add a **tDBClose** component below the **tPostJob** component.
- Configure the **tDBClose** component:
  - Select the database type.
  - On the **Component List**, select the connection you created earlier.



To connect **tPostJob** to **tDBClose**, use an **OnComponentOk** trigger.

When using a tDBOutput component with a shared connection, data is not saved to the database unless you use a specific component to commit.

To orchestrate a Job with data validation, use:

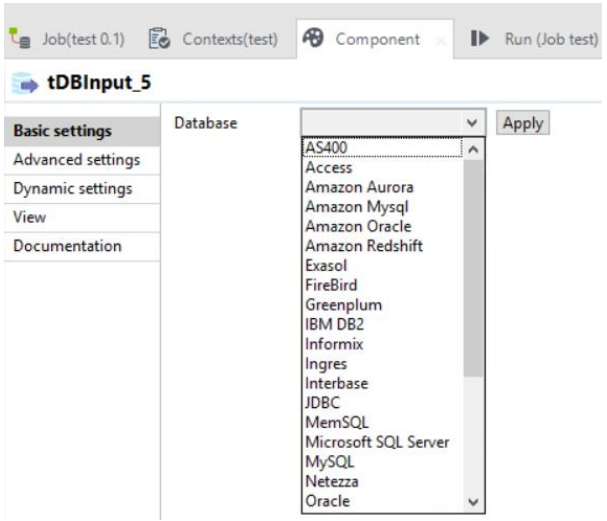
- **tDBCommit** triggered with an On Subjob OK trigger
- **tDBRollBack** triggered with an On Subjob Error trigger

JOB AID

Tips

Regardless of the database type, working with a database means connecting to the database, accessing the data using the data model (table schema), and performing operations on the data.

Generic components



Studio provides several unified database components for a large set of database types, including Amazon Redshift, Oracle, and MySQL.

Database components are generic. You must select a database type before using a component because the properties to configure are slightly different for each database type.

Table schemas

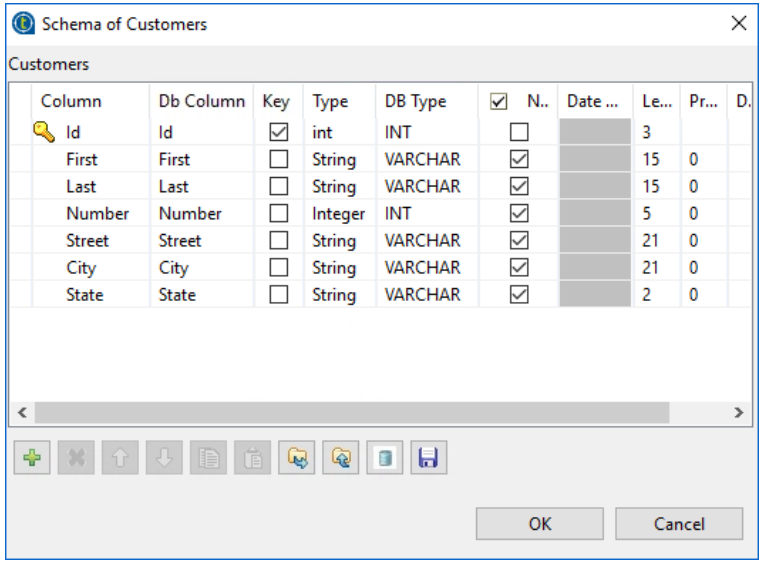


Table schemas define the data structure. A schema has several columns, and each column is assigned a specific data type, such as integer (INT), sequence of characters (VARCHAR), or date (DATE).

The Type column corresponds to the Java types of the data flow.

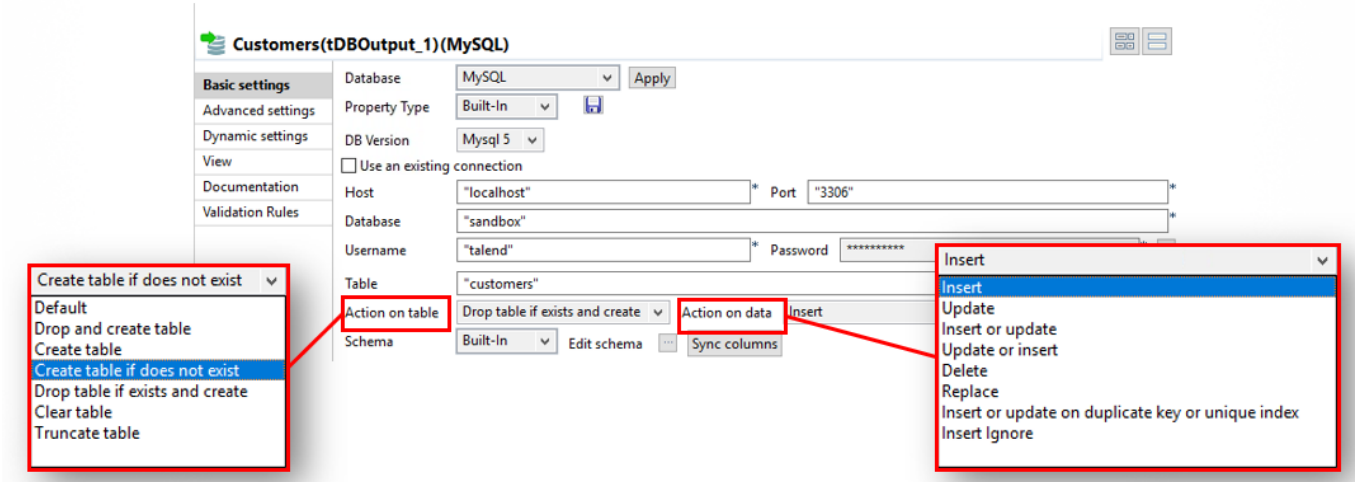
The DB Type column corresponds to the column types in the database table.

To configure the primary key column, select the **Key** check box.

For the other columns, you can select the Nullable attribute to indicate that NULL values are accepted, which is useful for missing or unknown values.

JOB AID

tDBOutput

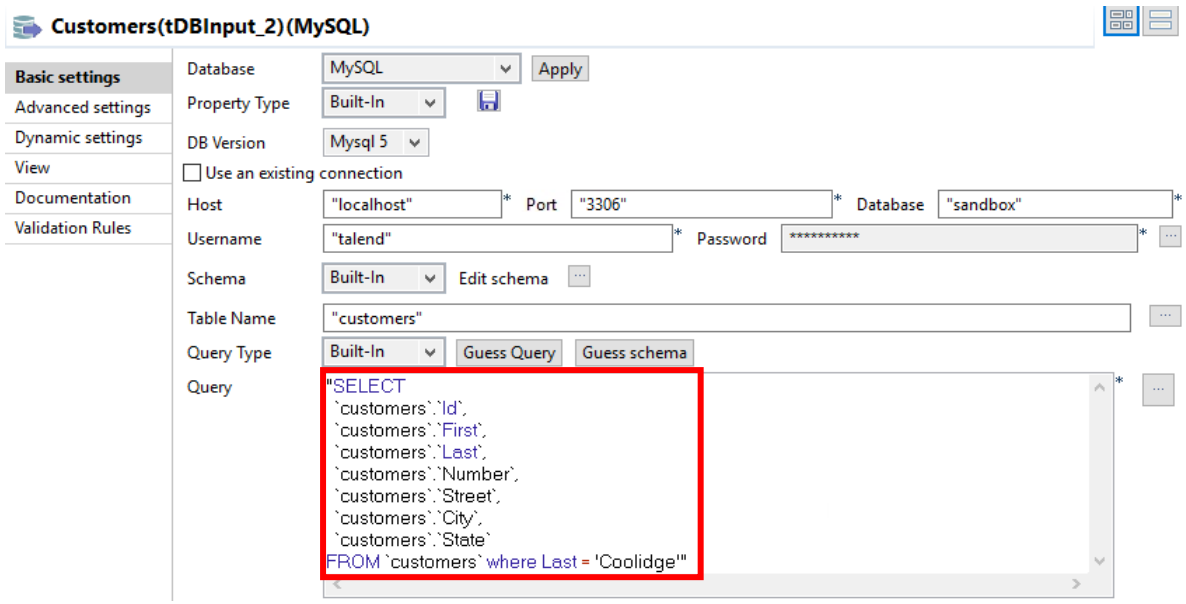


Use the **tDBOutput** component to load data into a database table.

The component schema must match the table schema.

Make sure to correctly select an action in the **Action on table** and **Action on data** lists.

tDBInput



Use the **tDBInput** component to read data from a table.

Use the **Guess Query** button to automatically generate a SQL query based on the schema definition. You can customize it using SQLBuilder.

Use the **Guess schema** button: when the database connection settings, table name, and query are selected, the table schema is automatically retrieved from the database.

JOB AID

Table creation best practices

- Being able to create and drop a table is very convenient in the development phase. Do not use these options in a production environment.
- Production databases must be managed by a database administrator only.
- The database administrator must have full control of the production database. Using their own set of administrative tools, they have more options when creating tables.