Building a Data Warehouse in Snowflake using ELTMaestro: Facts and Dimensions

01 Basic DW Concepts

- What is a Data Warehouse?
- What is a Data Mart?
- Facts and Dimensions
- ETL and ELT

What is a Data Warehouse?

- A data warehouse (DW) is a database that answers questions about a business.
- These questions are generally of the form "Tell me about all the <some noun> having <some attributes>." For example: "Tell me about all the *sales* having *timestamps between 1/31/2020 and 2/3/2020."*
- More examples:
  - A supermarket chain asks, "How much soda did we sell in zip codes 02474 and 02476 during Super Bowl weekend?"
  - An insurance company asks, "How many claims for mammograms for patients under 50 years were submitted in 2018?"

What is a Data Mart?

- A Data Mart is a subset of a data warehouse that focuses on a specific business line, or some other subdivision of the business.
- The question about soda sales might be handled by the *sales* data mart.
- The question about claims might be answered by the *claims* data mart (or, alternatively, by the *healthcare* data mart).

Facts and Dimensions

- DWs handle questions of the form “Tell me about all the <noun> having <attributes>.” The *nouns* in question are referred to as *facts*.
- *Sales* and *claims* are examples of facts.
- The *attributes* are referred to as *dimensions*. *Date* and *location* are typical examples of dimensions.
- Facts are generally stored in tables called *fact tables*. Dimensions are stored in *dimension tables*.
- *Why do we put facts and dimensions in separate tables? (Think about it!)**
- * See discussion of “Date-related queries,” below.

ETL and ELT

- **Extract, Transform, Load (ETL) and Extract, Load, Transform (ELT) are techniques for getting information from source systems into data warehouses.**
- **The source systems in question are generally the operational systems of the business.**
  - **The operational systems of a business are the systems that the business uses to carry on its basic operations.**
  - **For a retail operation, operational systems might include cash registers, customer facing web pages and the databases that support them.**
  - **For a bank, operational systems might include ATM machines and the databases that support them.**

ETL and ELT, cont.

- **ETL/ELT is comprised of all the steps involved in getting information from the operational systems to the fact and dimension tables of the warehouse.**
- **In this course, we use ELT. But we won’t be concerned with the distinction between ETL and ELT for now.**

Transformations

- The changes and rearrangements that data undergoes on its way from source systems to data warehouses are referred to as transformations.
- Some transformations are relatively straightforward; others are more complex.
- Complex transformations are typically composed of simpler transformations.

Transforming Structured Files into Relational Tables

- A simple type of transformation found in many data warehouses is a one which transforms structured files into relational tables.
- Data often arrives at the data warehouse not as relational tables but as structured text files.
- Examples of structured file formats include CSV, XML, and JSON.

01 Lab

- Assumptions
- A quick overview of the ELTMaestro UI
- First job: parsing JSON
- Adding a date dimension

Assumptions

- You have access to Snowflake and ELTMaestro for Snowflake.
- You can run SQL queries on Snowflake.

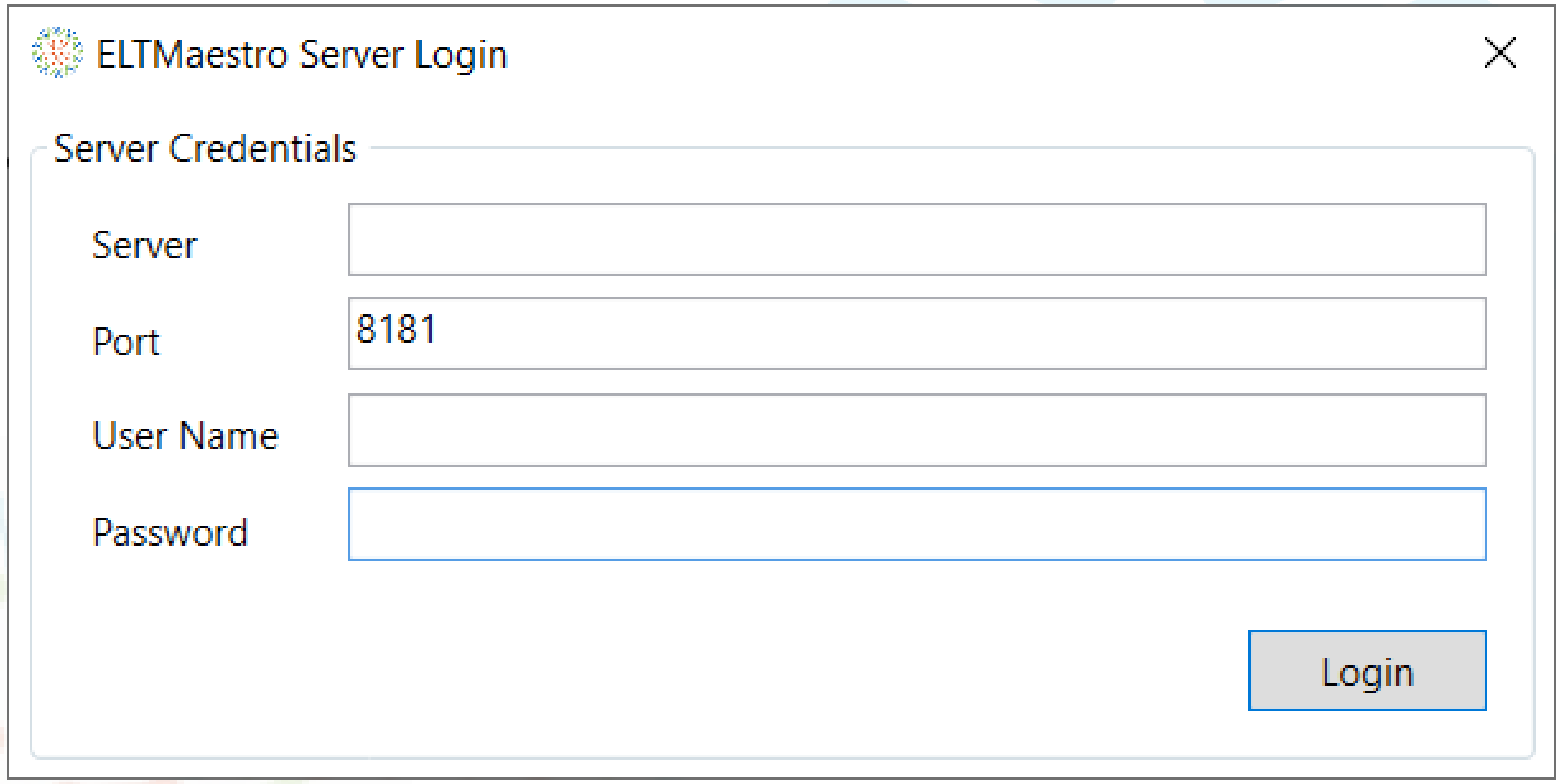
A quick overview of the ELTMaestro UI

In the next few slides, we’ll gain some initial familiarity with the ELTMaestro UI.

Important! Appending your initials

- In what follows, we'll refer to your initials as "XX." If your name is Elsa Black, for example, you should substitute "_EB" everywhere you see "_XX."
- After you register, we'll create a database named DWH_XX (where XX are your real initials).
- DWH_XX will contain the tables you need to do the course exercises; any tables you create should also go there.
- Your workflows (jobs) are stored in a common area along with those of other students. Identify the workflows *you* create by appending _XX to your workflow names – for example, name your first workflow PARSE_JORDER_XX.

1. The first thing you see when you open ELTMaestro is the Login window.



ELTMaestro Server Login

Server Credentials

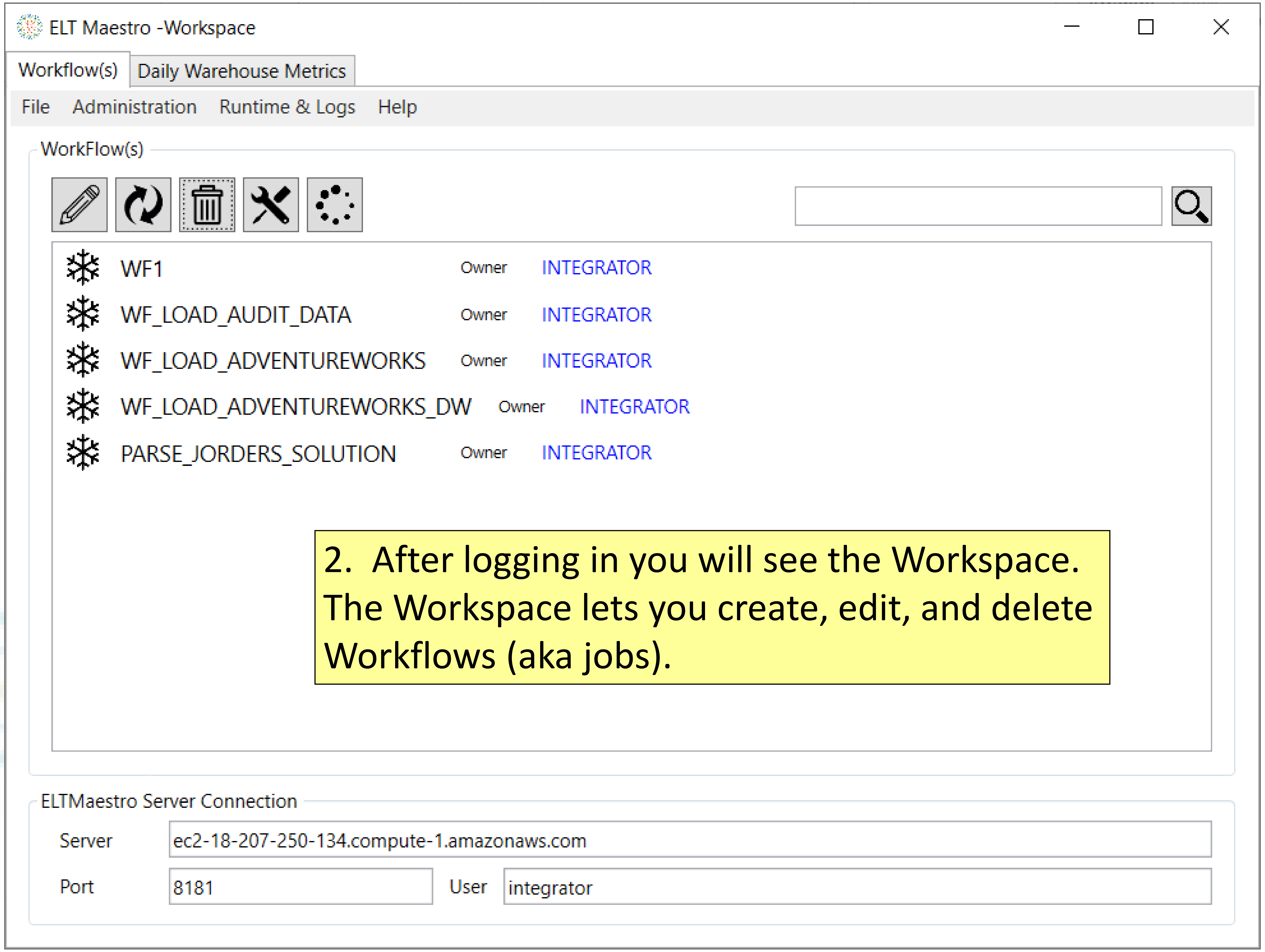
Server

Port 8181

User Name

Password

You will be provided with appropriate values for Server, Port, User Name and Password



WF1 Owner INTEGRATOR

WF_LOAD_AUDIT_DATA Owner INTEGRATOR

WF_LOAD_ADVENTUREWORKS Owner INTEGRATOR

WF_LOAD_ADVENTUREWORKS_DW Owner INTEGRATOR

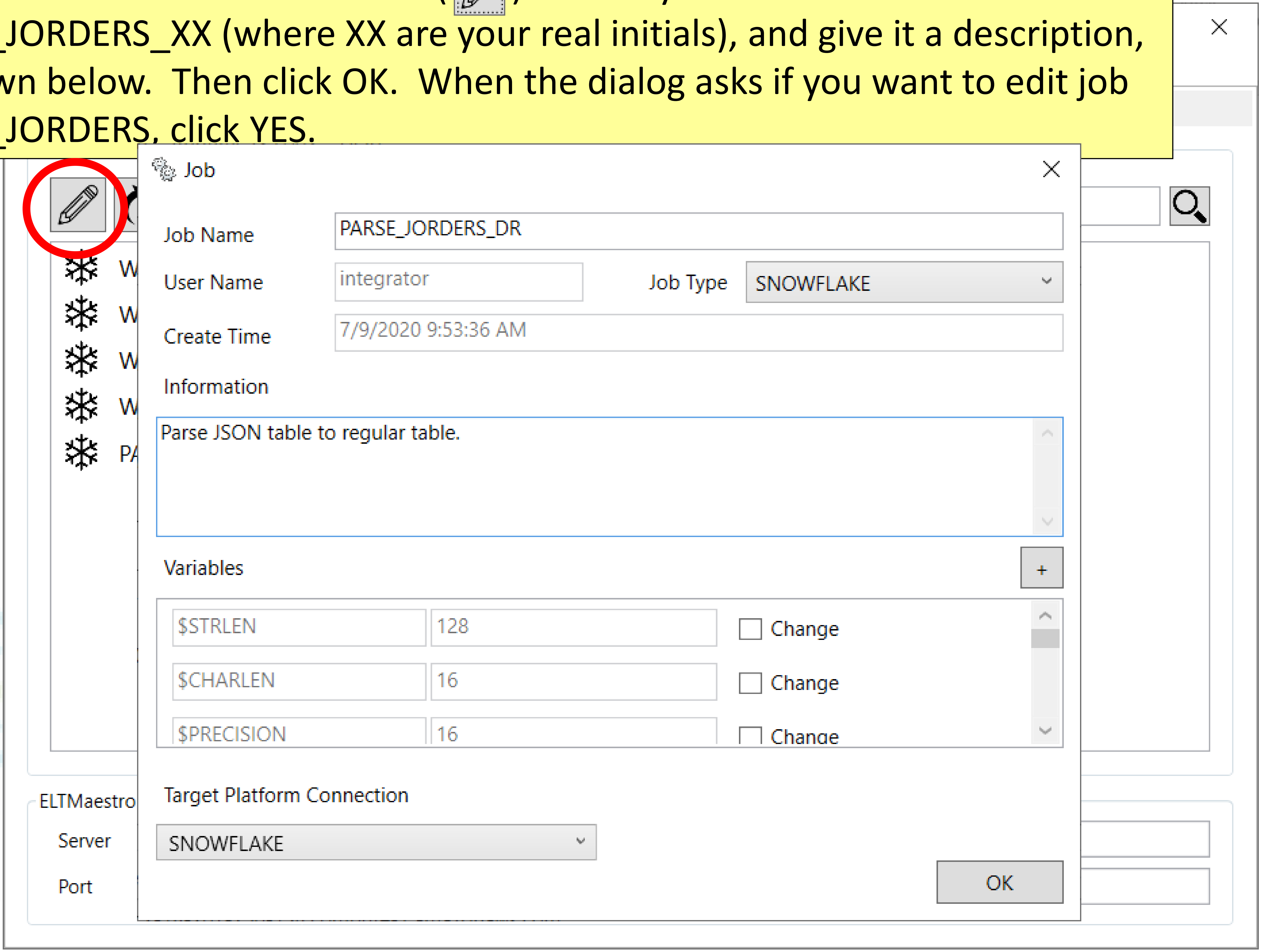
PARSE_JORDERS_SOLUTION Owner INTEGRATOR

Server ec2-18-207-250-134.compute-1.amazonaws.com

Port 8181 User integrator

2. After logging in you will see the Workspace. The Workspace lets you create, edit, and delete Workflows (aka jobs).

3. Click on Create New Workflow (✎). Name your new workflow PARSE_JORDERS_XX (where XX are your real initials), and give it a description, as shown below. Then click OK. When the dialog asks if you want to edit job PARSE_JORDERS, click YES.



Job

Job Name: PARSE_JORDERS_DR

User Name: integrator

Job Type: SNOWFLAKE

Create Time: 7/9/2020 9:53:36 AM

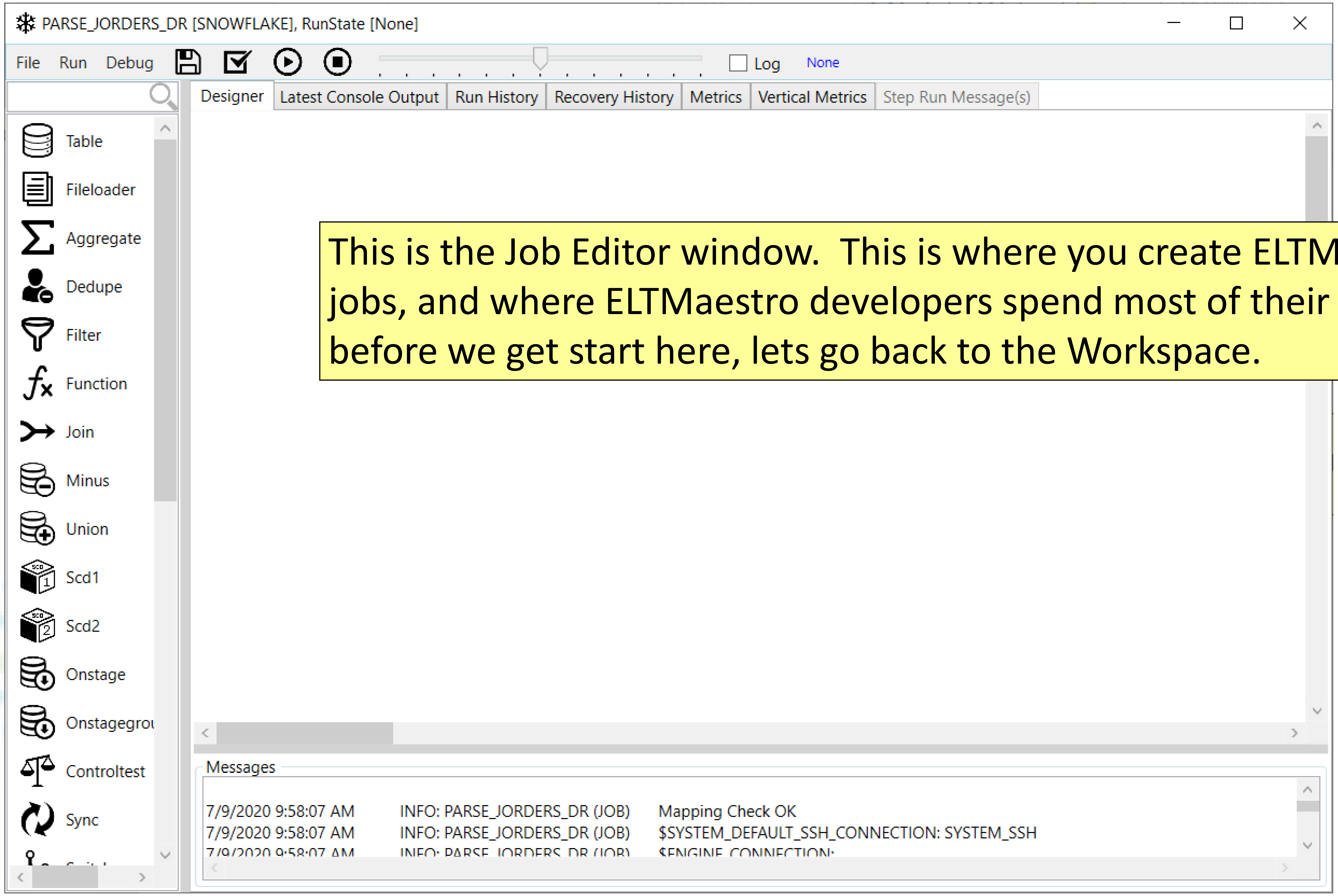
Information

Parse JSON table to regular table.

Variables

$STRLEN	128	□ Change
$CHARLEN	16	□ Change
$PRECISION	16	□ Change

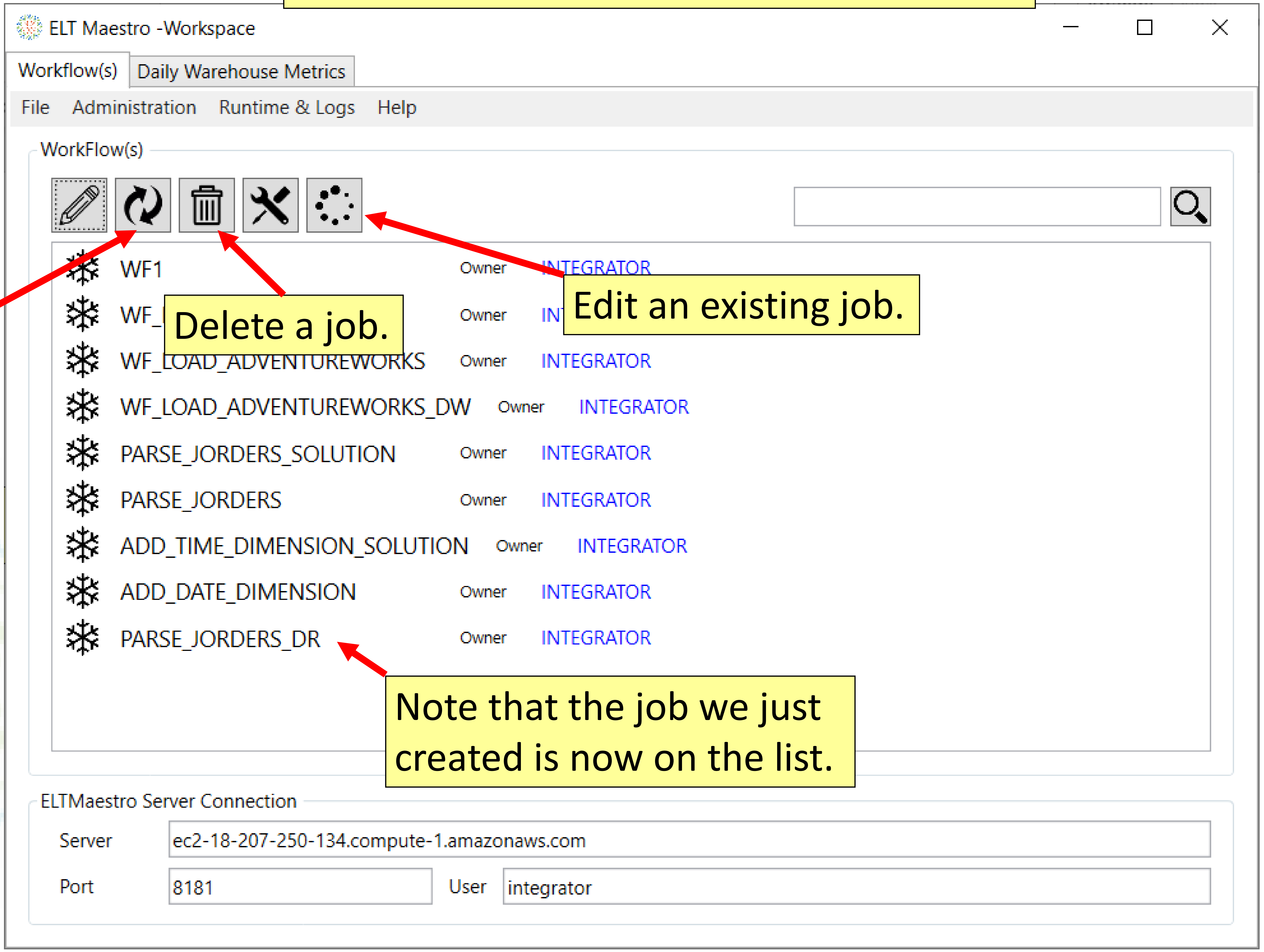
Target Platform Connection: SNOWFLAKE



```
7/9/2020 9:58:07 AM     INFO: PARSE_JORDERS_DR (JOB)    Mapping Check OK
7/9/2020 9:58:07 AM     INFO: PARSE_JORDERS_DR (JOB)    $SYSTEM_DEFAULT_SSH_CONNECTION: SYSTEM_SSH
7/9/2020 9:58:07 AM     INFO: PARSE_JORDERS_DR (JOB)    $ENGINE_CONNECTION:
```

This is the Job Editor window. This is where you create ELTMaestro jobs, and where ELTMaestro developers spend most of their time. But before we get start here, lets go back to the Workspace.

More things you can do in the Workspace



ELT Maestro -Workspace

WorkFlow(s)

WF1	Owner	INTEGRATOR
WF_	Owner	IN
WF_LOAD_ADVENTUREWORKS	Owner	INTEGRATOR
WF_LOAD_ADVENTUREWORKS_DW	Owner	INTEGRATOR
PARSE_JORDERS_SOLUTION	Owner	INTEGRATOR
PARSE_JORDERS	Owner	INTEGRATOR
ADD_TIME_DIMENSION_SOLUTION	Owner	INTEGRATOR
ADD_DATE_DIMENSION	Owner	INTEGRATOR
PARSE_JORDERS_DR	Owner	INTEGRATOR

ELTMaestro Server Connection

Server: ec2-18-207-250-134.compute-1.amazonaws.com

Port: 8181

User: integrator

Refresh the list of jobs.

Delete a job.

Edit an existing job.

Note that the job we just created is now on the list.

Our first job: parsing JSON

- A lot of ETL consists of transforming data from various structured file formats into relational tables.
- In this section we will parse data represented in a popular format called JSON.

The JORDERS Table

- **The JORDERS table has a single column called ORDERS.**
- **Each row consists of a text string like the one below.**
- **This string is in a format called *JSON.***

```
{
  "o_clerk": "Clerk#000000385",
  "o_comment": "refully special platelets cajole. slyly unusual pinto be",
  "o_custkey": 63355,
  "o_orderdate": "1996-02-15",
  "o_orderkey": 5242401,
  "o_orderpriority": "5-LOW",
  "o_orderstatus": "O",
  "o_shippriority": 0,
  "o_totalprice": 230578.84
}
```

The ORDERS Table

- **We want to convert the JORDERS table to a table called ORDERS, with the 9 columns shown below**

```
O_ORDERKEY|O_CUSTKEY|O_ORDERSTATUS|O_TOTALPRICE|O_ORDERDATE|O_ORDERPRIORITY|O_CLERK        |O_SHIPPRIORITY|O_COMMENT
----------|---------|-------------|------------|-----------|---------------|---------------|--------------|-------------------------
   5242401|    63355|O            |   230578.84| 1996-02-15|5-LOW          |Clerk#000000385|             0|refully special platelets
   5242402|    98561|O            |   222665.47| 1997-02-17|1-URGENT       |Clerk#000000370|             0|eodolites wake furiously
   5242403|    63685|O            |   187295.18| 1996-07-01|2-HIGH         |Clerk#000000480|             0|ously unusual requests ar
   5242404|    91651|O            |   171004.71| 1998-05-22|5-LOW          |Clerk#000000910|             0|y express deposits nag sl
```

- **To do this we use the function PARSE_JSON.**

PARSE_JSON usage

When ORDERS takes the value

```
{
  "o_clerk": "Clerk#000000385",
  "o_comment": "refully special platelets cajole. slyly unusual pinto be",
  "o_custkey": 63355,
  "o_orderdate": "1996-02-15",
  "o_orderkey": 5242401,
  "o_orderpriority": "5-LOW",
  "o_orderstatus": "O",
  "o_shippriority": 0,
  "o_totalprice": 230578.84
}
```

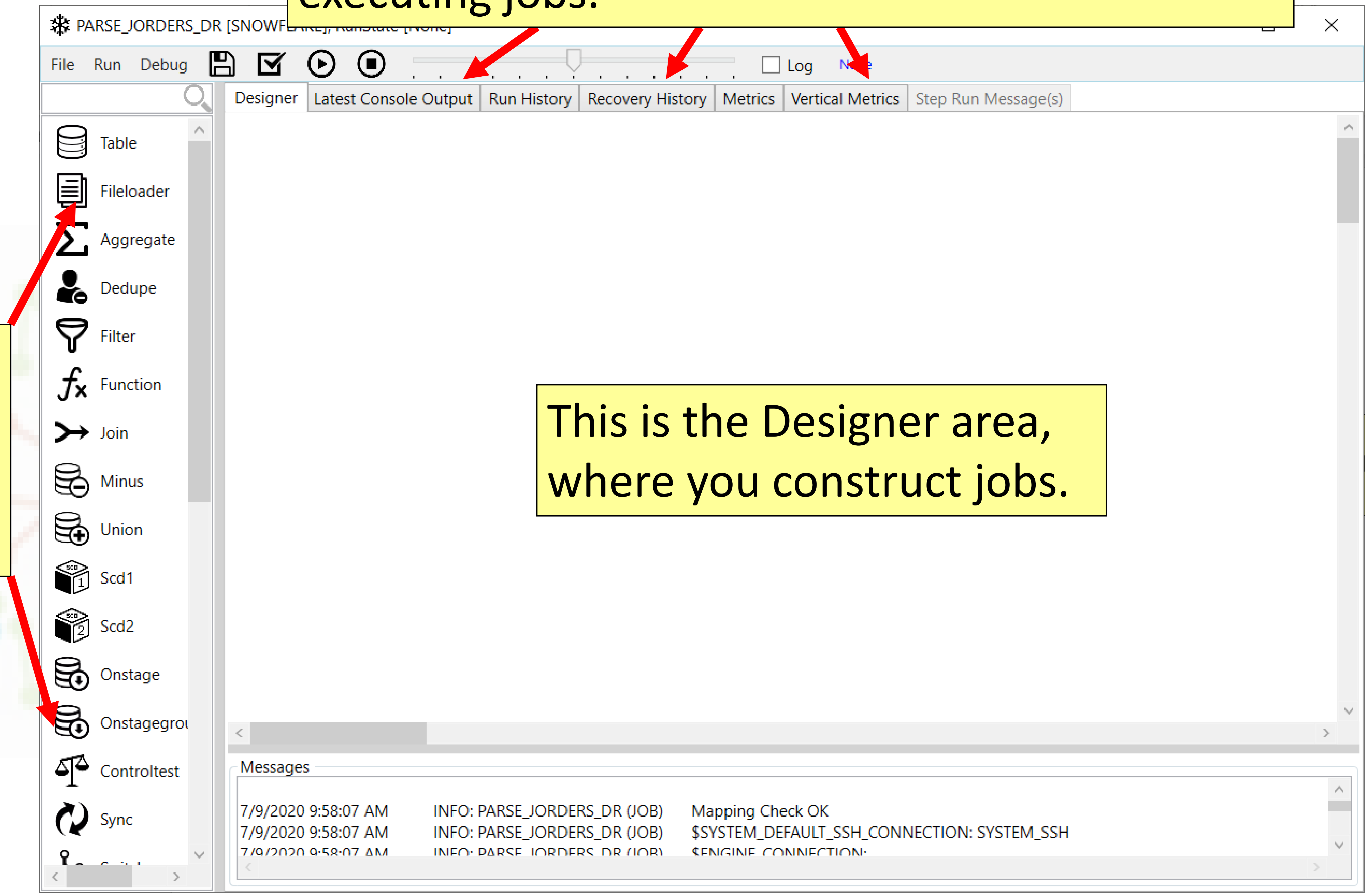
PARSE_JSON(ORDERS):o_clerk = ‘Clerk#000000385’

PARSE_JSON(ORDERS):o_custkey = 63355

PARSE_JSON(ORDERS):o_totalprice = 230578.84

etc.

Go back to the Job Editor window for the PARSE_ORDERS job. If it's not still open, open it from the Workspace by double clicking on its name, or by selecting it and then clicking the ⁙ button.

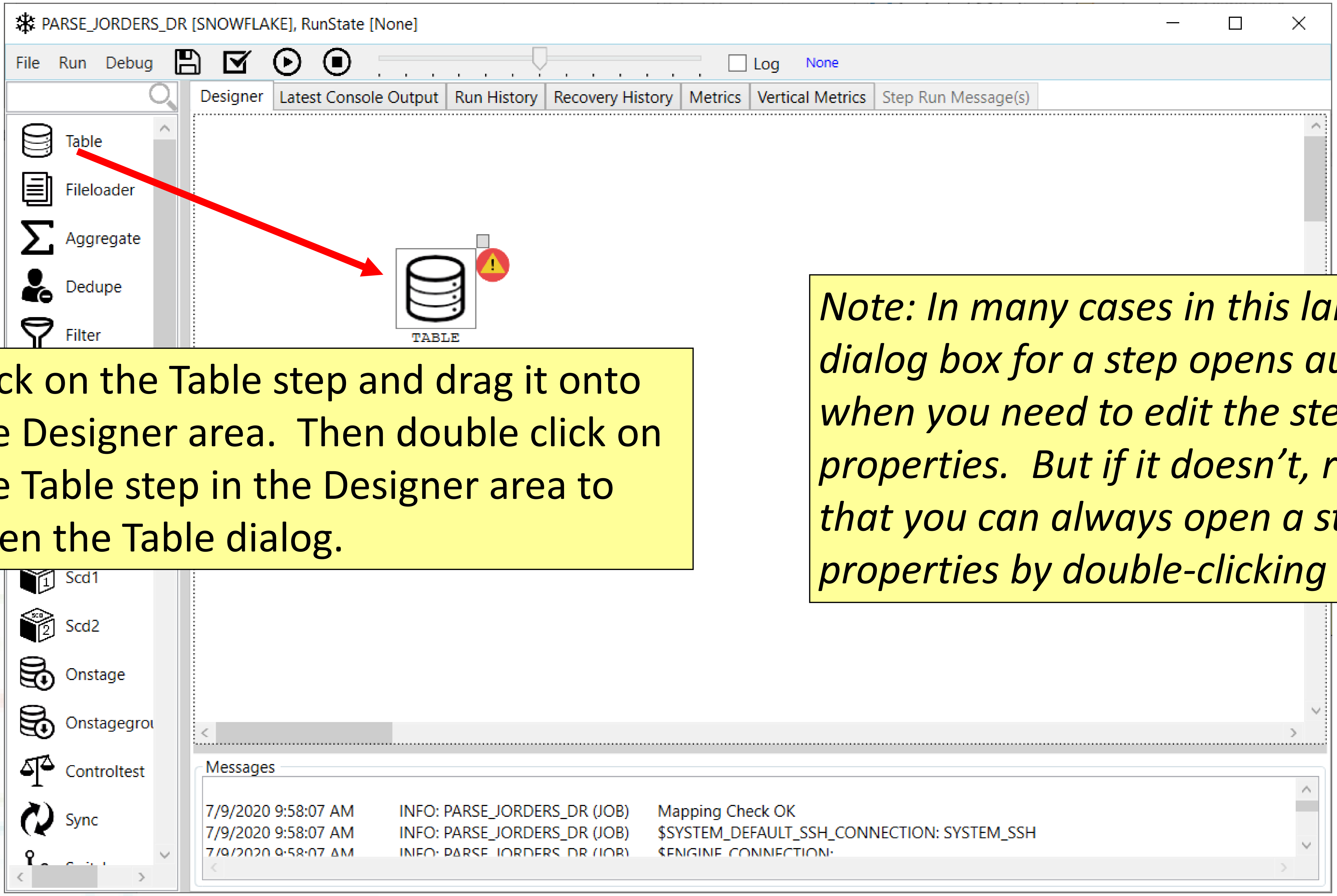


```
7/9/2020 9:58:07 AM      INFO: PARSE_JORDERS_DR (JOB)      Mapping Check OK
7/9/2020 9:58:07 AM      INFO: PARSE_JORDERS_DR (JOB)      $SYSTEM_DEFAULT_SSH_CONNECTION: SYSTEM_SSH
7/9/2020 9:58:07 AM      INFO: PARSE_JORDERS_DR (JOB)      $ENGINE_CONNECTION:
```

These other tabs help you track the progress of executing jobs.

This is the Palette, which contains *Steps,* which are operations on data.

This is the Designer area, where you construct jobs.



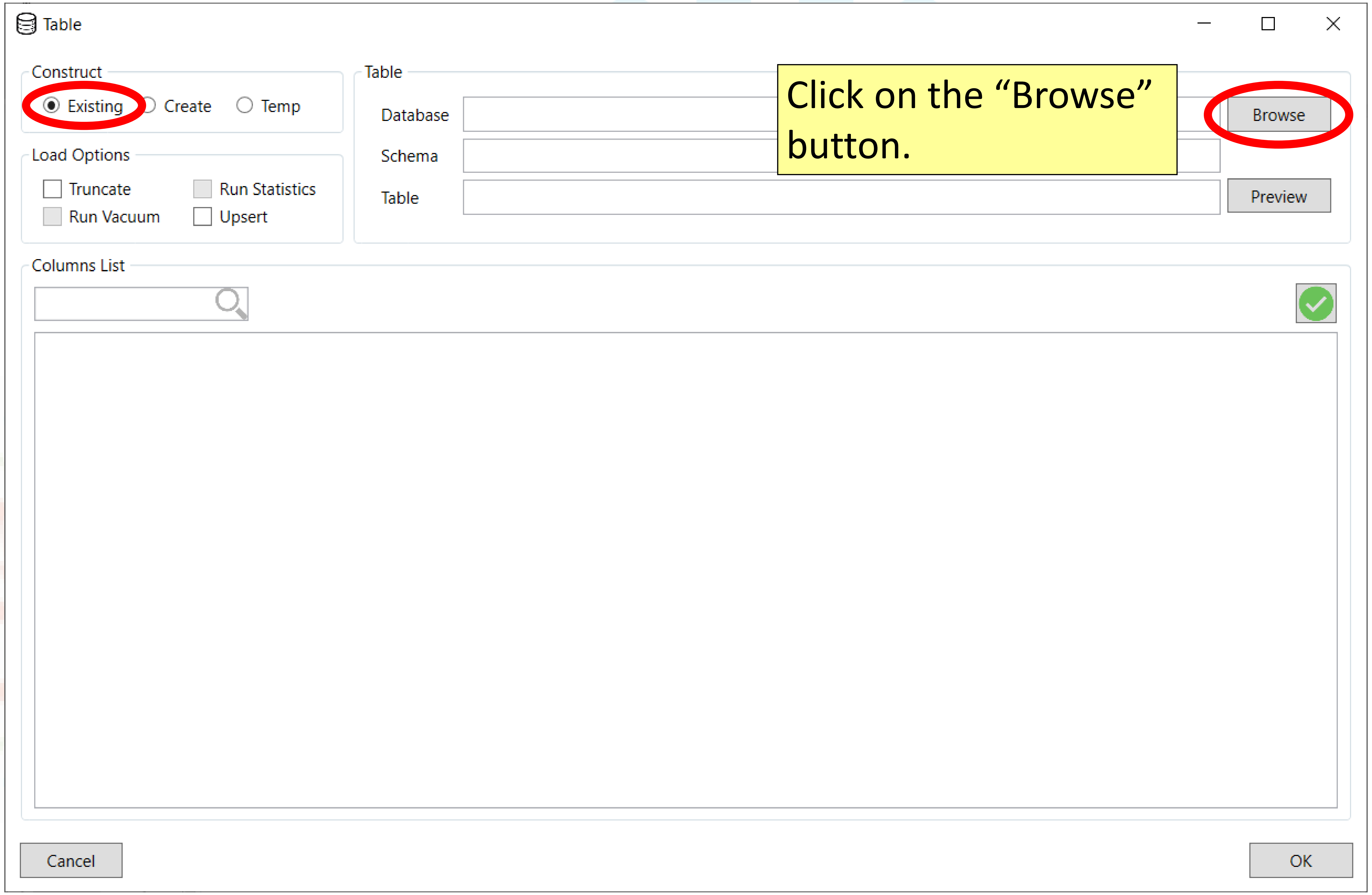
```
7/9/2020 9:58:07 AM    INFO: PARSE_JORDERS_DR (JOB)    Mapping Check OK
7/9/2020 9:58:07 AM    INFO: PARSE_JORDERS_DR (JOB)    $SYSTEM_DEFAULT_SSH_CONNECTION: SYSTEM_SSH
7/9/2020 9:58:07 AM    INFO: PARSE_JORDERS_DR (JOB)    $ENGINE_CONNECTION:
```

Click on the Table step and drag it onto the Designer area. Then double click on the Table step in the Designer area to open the Table dialog.

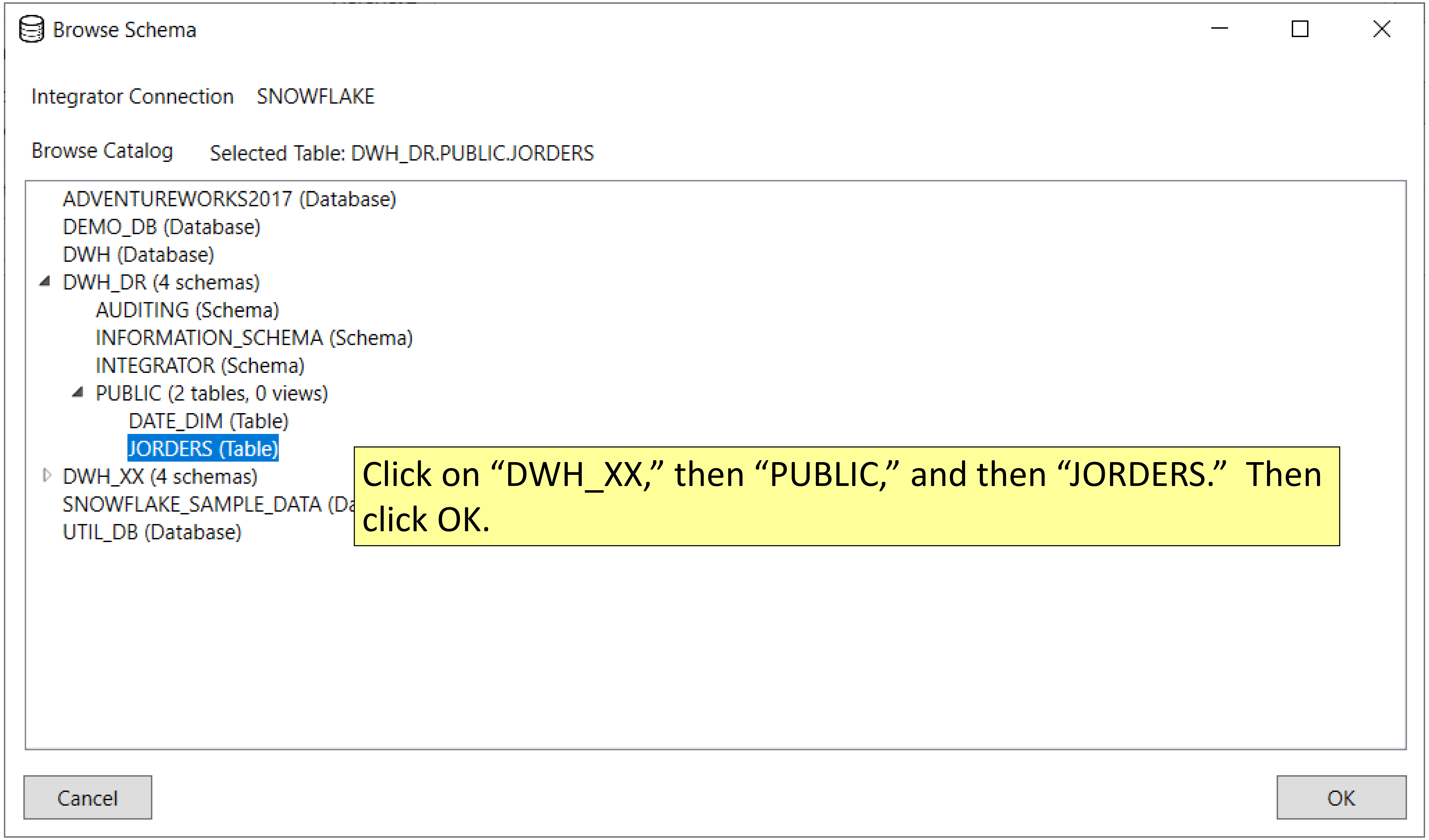
Note: In many cases in this lab, the dialog box for a step opens automatically when you need to edit the steps properties. But if it doesn't, remember that you can always open a step's properties by double-clicking on it.

This is the Table dialog.

The “Existing” radio button should be selected, meaning we are reading from an existing table.



Click on the “Browse” button.



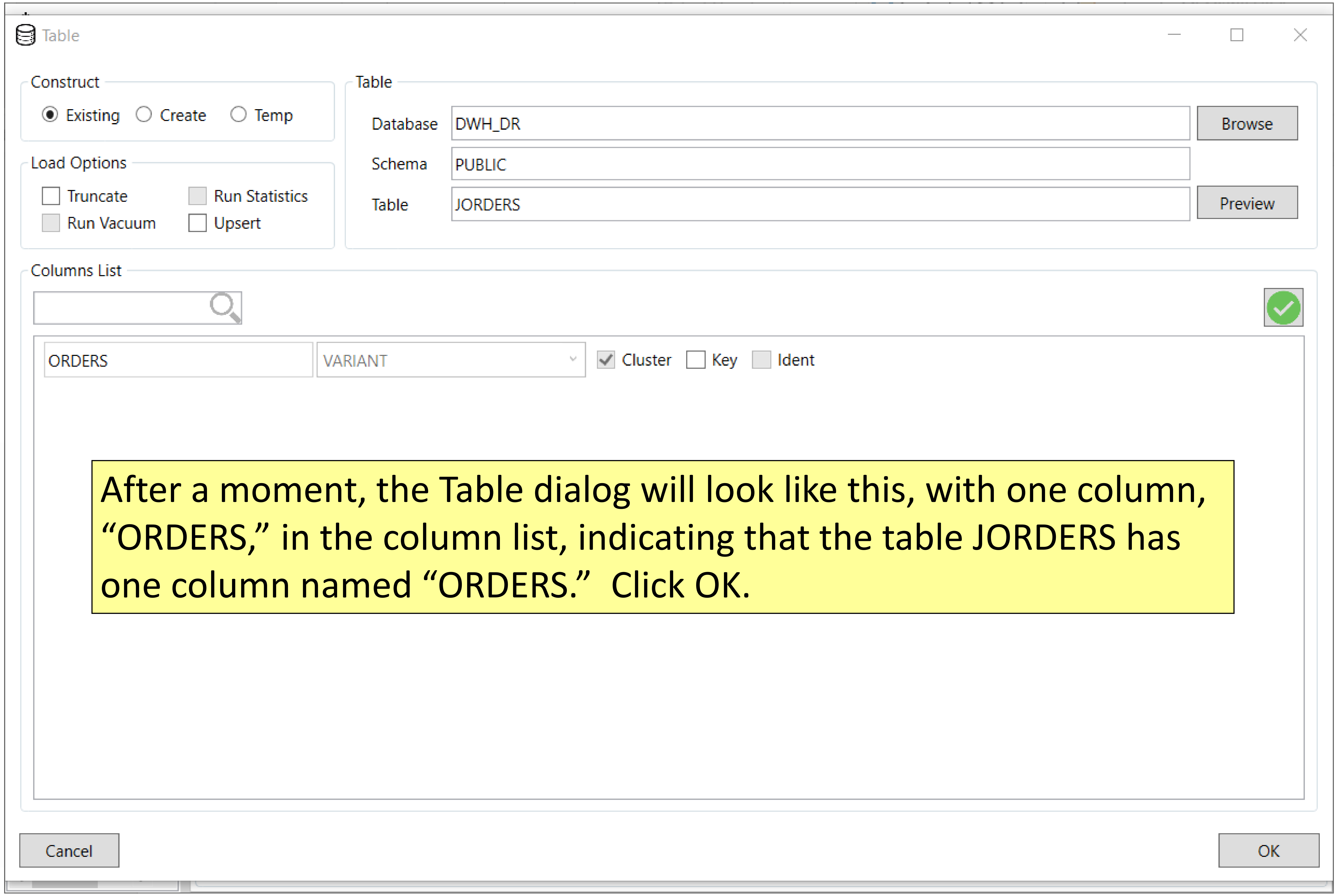
Browse Schema

Integrator Connection SNOWFLAKE

Browse Catalog Selected Table: DWH_DR.PUBLIC.JORDERS

- ADVENTUREWORKS2017 (Database)
- DEMO_DB (Database)
- DWH (Database)
- DWH_DR (4 schemas)
  - AUDITING (Schema)
  - INFORMATION_SCHEMA (Schema)
  - INTEGRATOR (Schema)
  - PUBLIC (2 tables, 0 views)
    - DATE_DIM (Table)
    - JORDERS (Table)
- DWH_XX (4 schemas)
- SNOWFLAKE_SAMPLE_DATA (Da
- UTIL_DB (Database)

Click on “DWH_XX,” then “PUBLIC,” and then “JORDERS.” Then click OK.



Table

Construct

● Existing ○ Create ○ Temp

Load Options

□ Truncate □ Run Statistics
□ Run Vacuum □ Upsert

Table

Database: DWH_DR

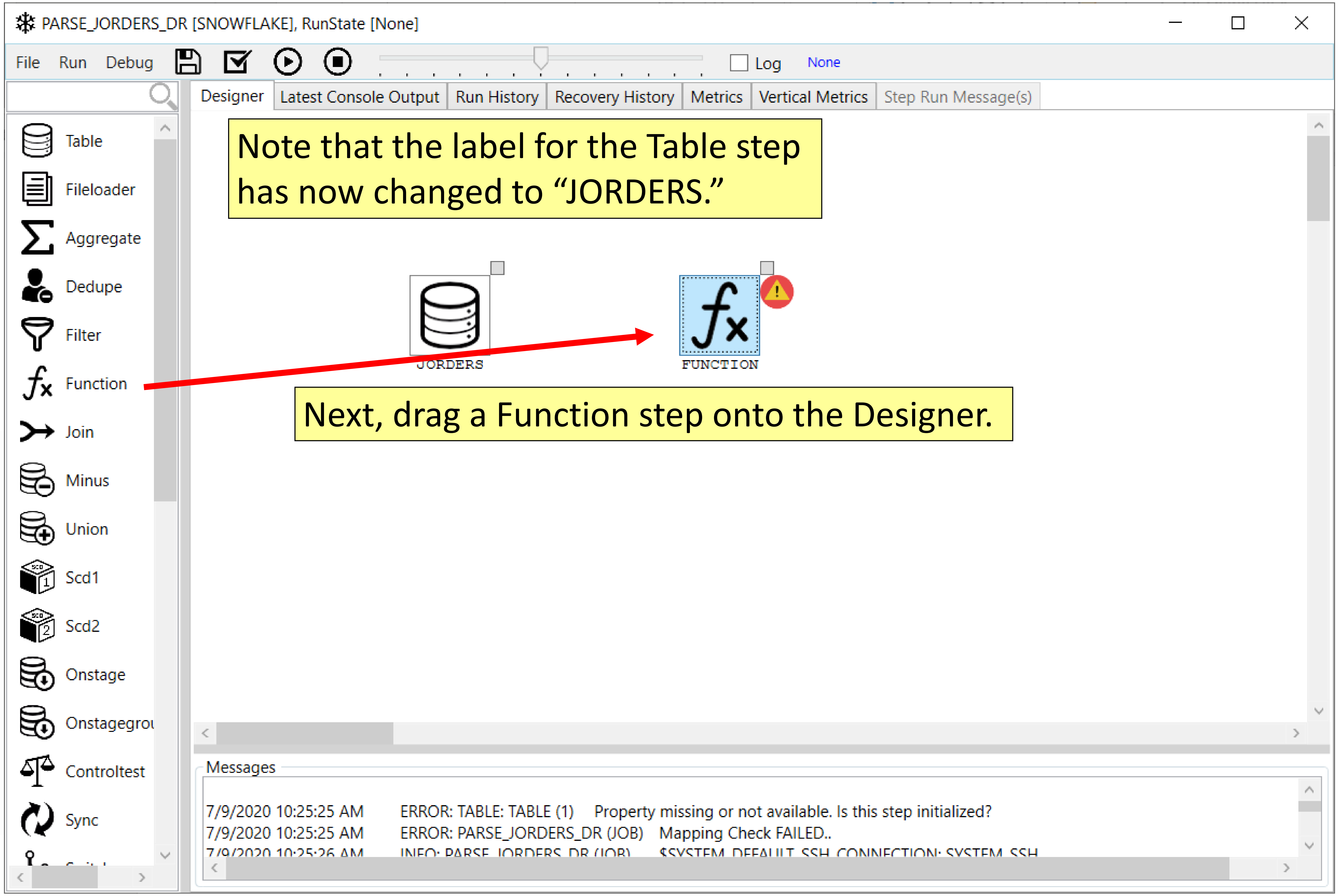
Schema: PUBLIC

Table: JORDERS

Columns List

ORDERS | VARIANT | ☑ Cluster □ Key □ Ident

After a moment, the Table dialog will look like this, with one column, “ORDERS,” in the column list, indicating that the table JORDERS has one column named “ORDERS.” Click OK.

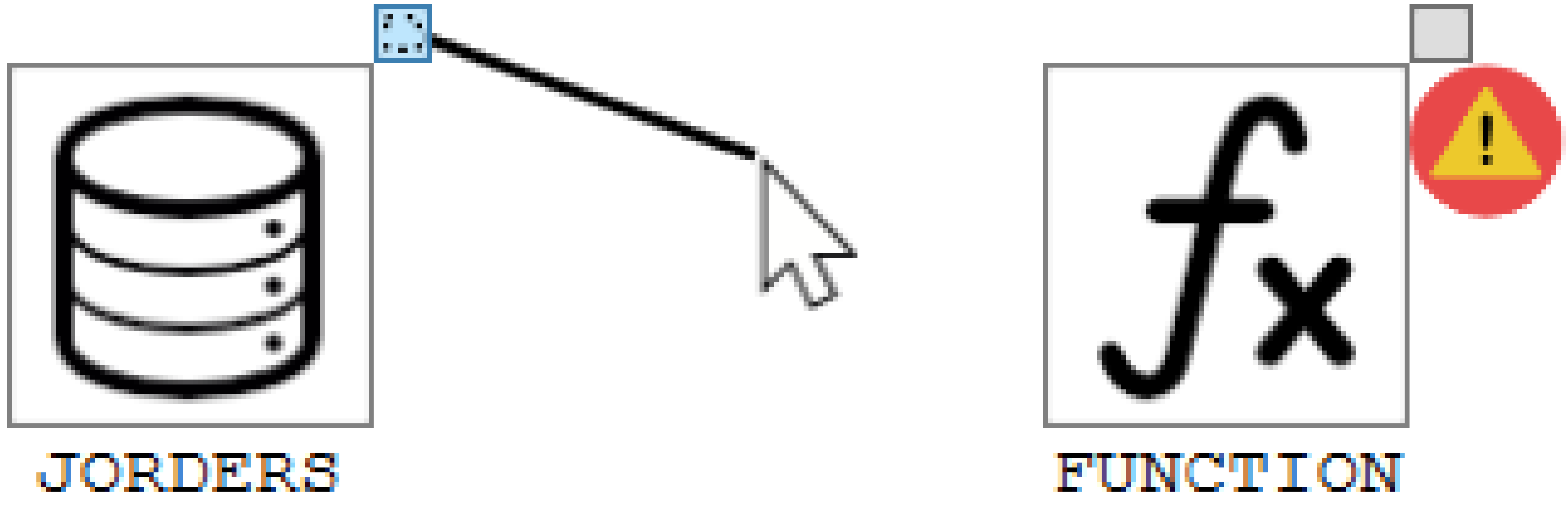


Note that the label for the Table step has now changed to “JORDERS.”

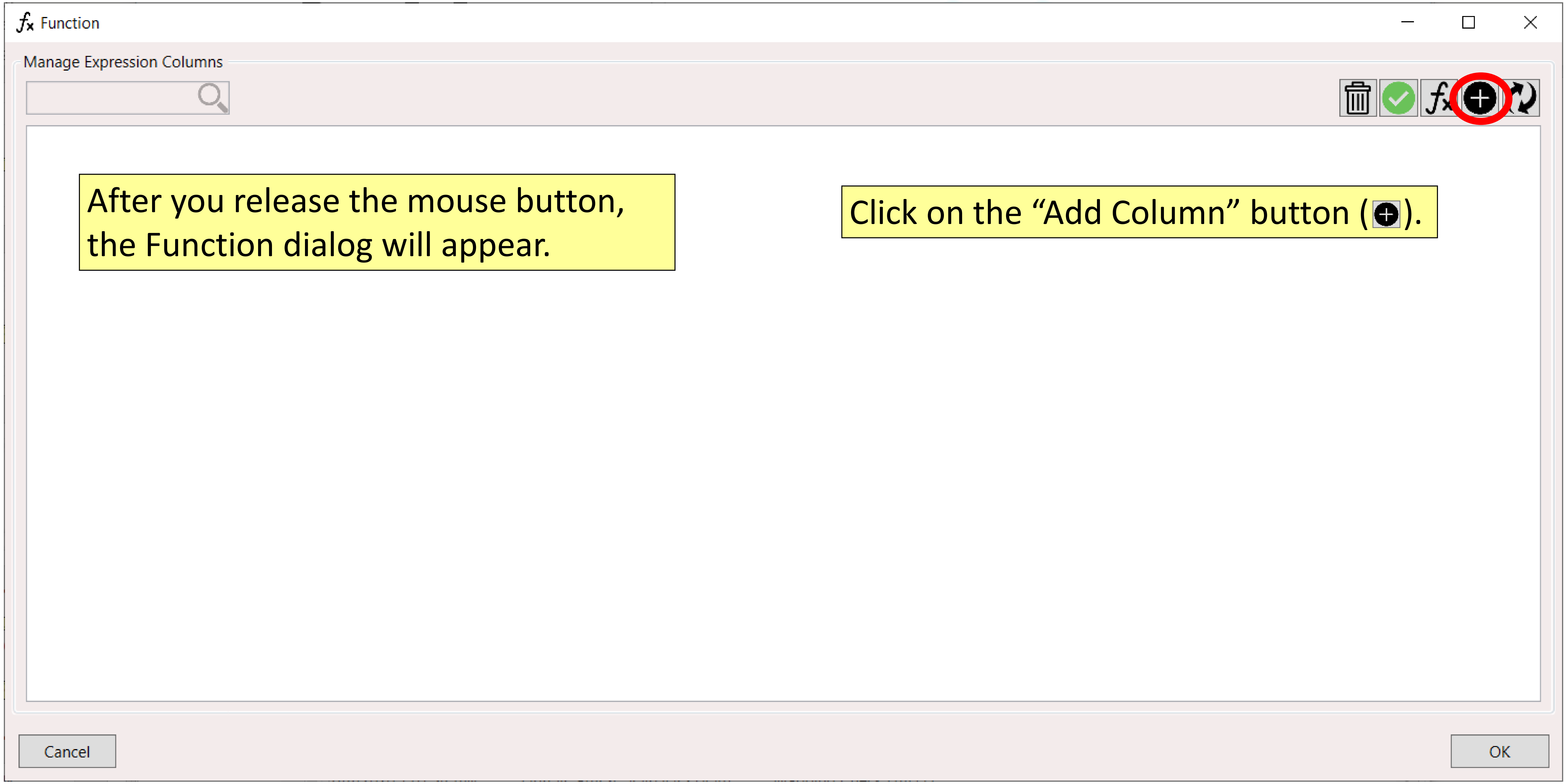
Next, drag a Function step onto the Designer.

```
7/9/2020 10:25:25 AM     ERROR: TABLE: TABLE (1)     Property missing or not available. Is this step initialized?
7/9/2020 10:25:25 AM     ERROR: PARSE_JORDERS_DR (JOB)     Mapping Check FAILED..
7/9/2020 10:25:26 AM     INFO: PARSE_JORDERS_DR (JOB)     $SYSTEM_DEFAULT_SSH_CONNECTION: SYSTEM_SSH
```

Click in the small box in the upper right corner of the Table step, and, holding down the mouse button, drag the mouse to the Function step.

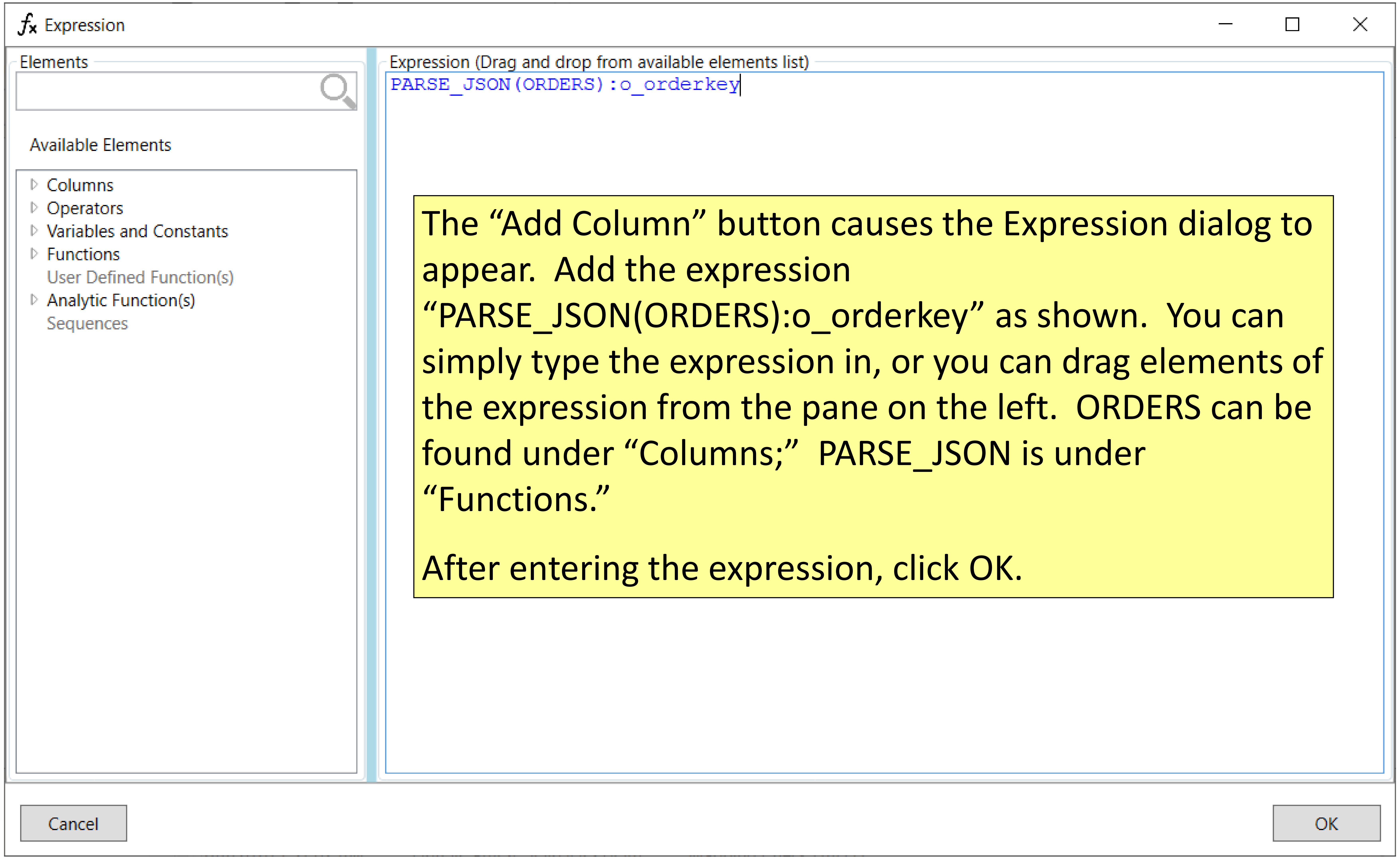


When the mouse is inside the function step (so that the line extending from the JORDERS Table step touches the Function step), release the mouse button.



After you release the mouse button, the Function dialog will appear.

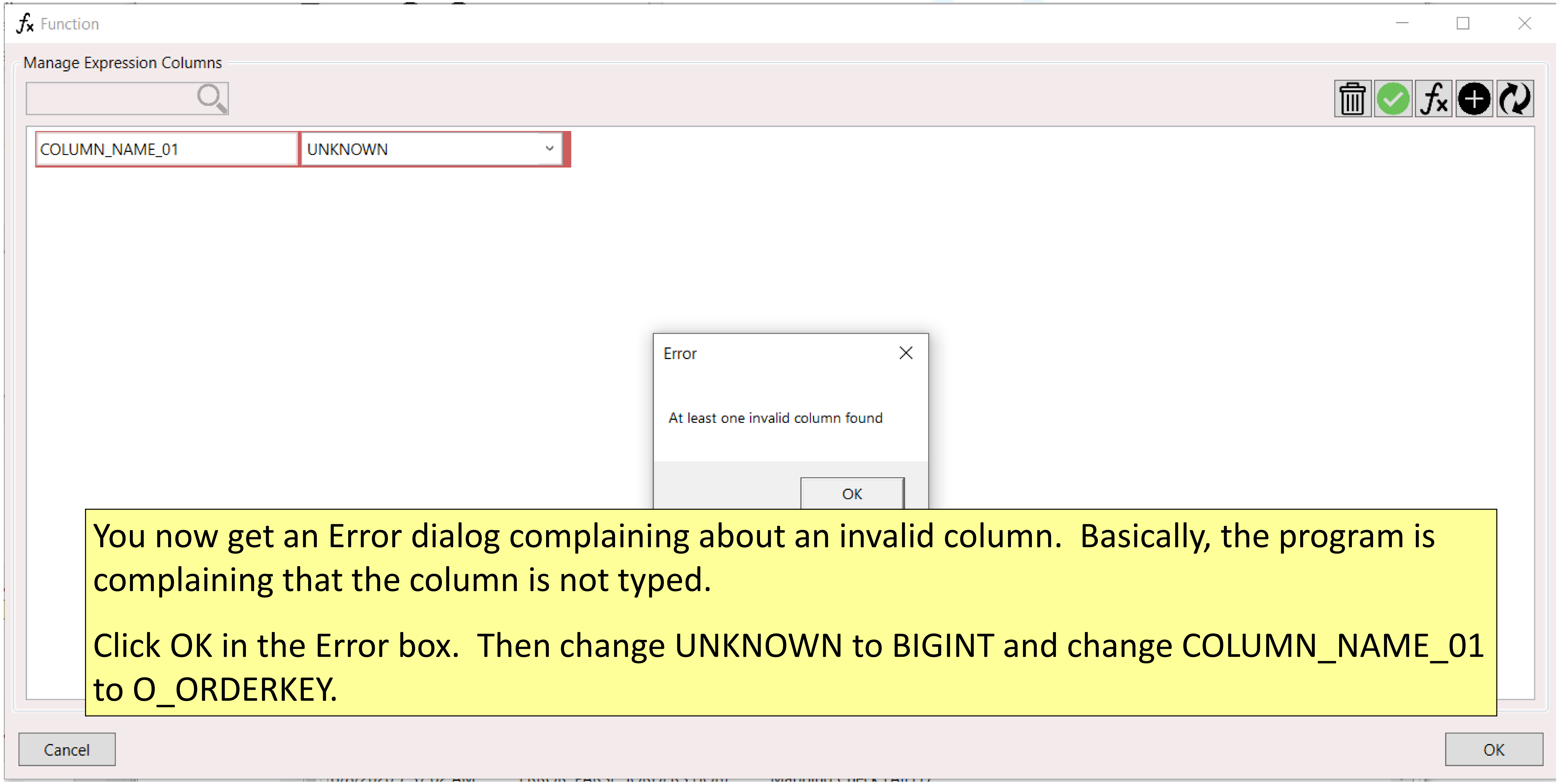
Click on the “Add Column” button (⊕).



```
PARSE_JSON(ORDERS):o_orderkey
```

The “Add Column” button causes the Expression dialog to appear. Add the expression “PARSE_JSON(ORDERS):o_orderkey” as shown. You can simply type the expression in, or you can drag elements of the expression from the pane on the left. ORDERS can be found under “Columns;” PARSE_JSON is under “Functions.”

After entering the expression, click OK.



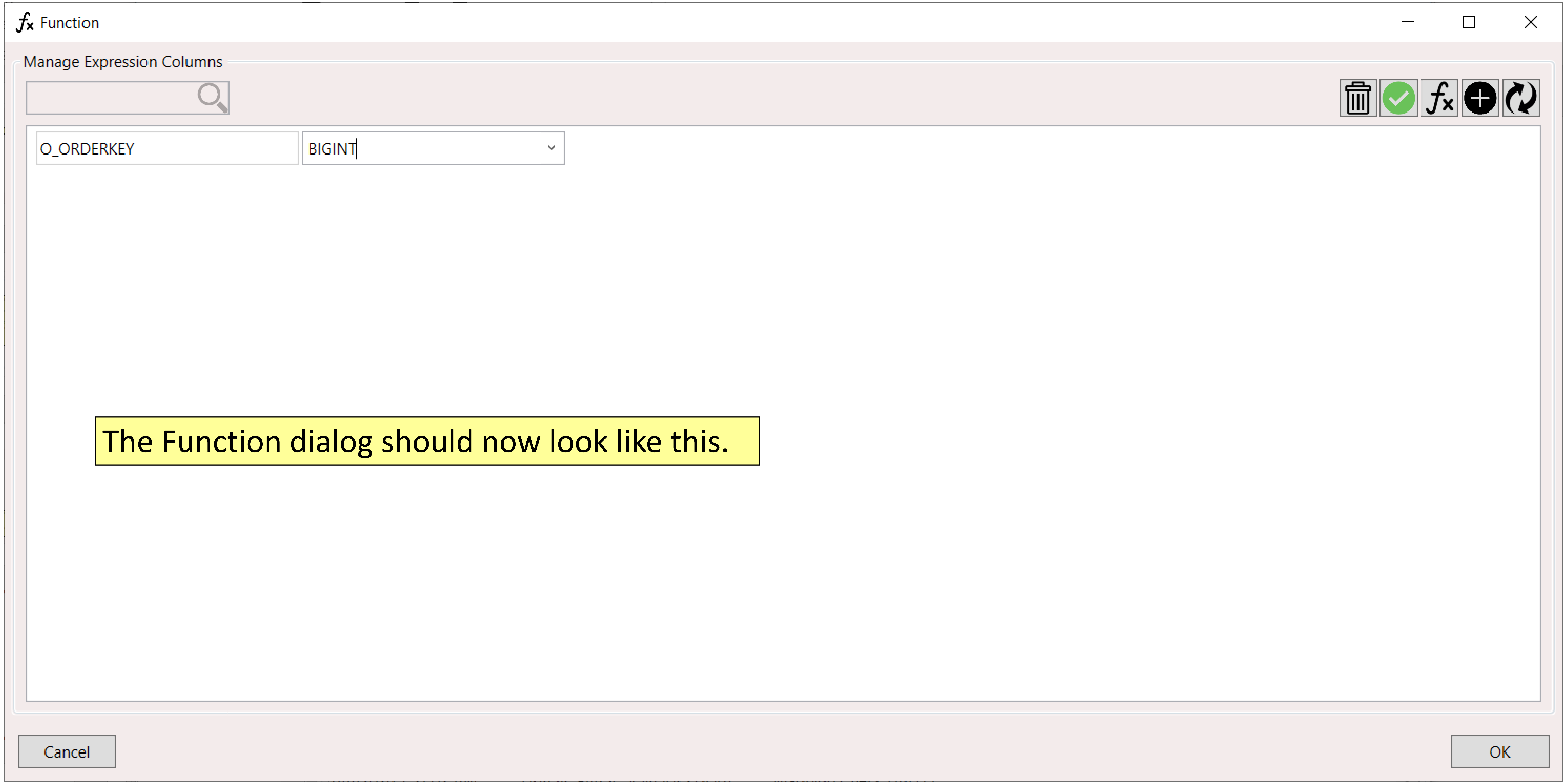
COLUMN_NAME_01 | UNKNOWN

Error

At least one invalid column found

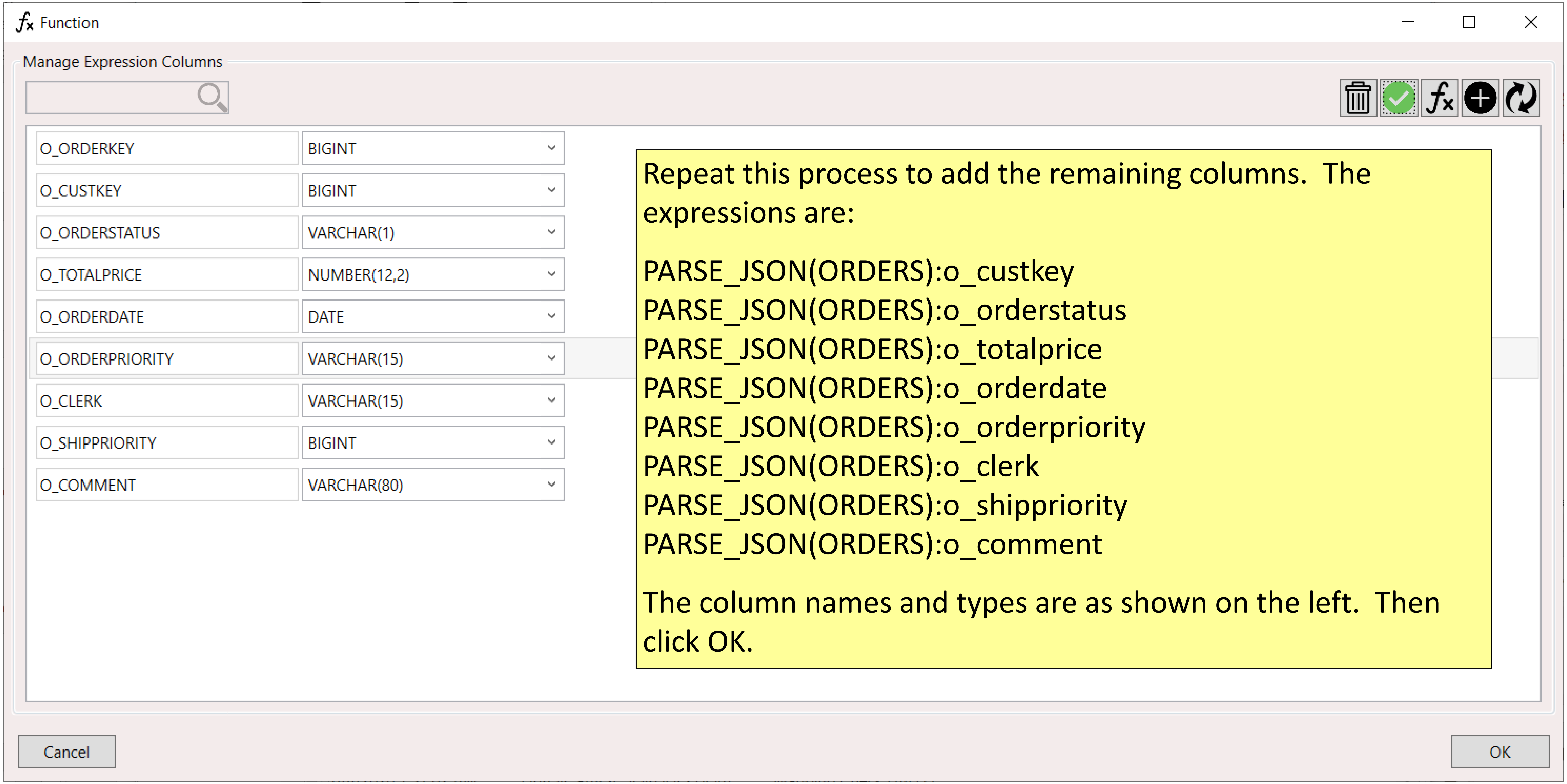
You now get an Error dialog complaining about an invalid column. Basically, the program is complaining that the column is not typed.

Click OK in the Error box. Then change UNKNOWN to BIGINT and change COLUMN_NAME_01 to O_ORDERKEY.



O_ORDERKEY BIGINT

The Function dialog should now look like this.

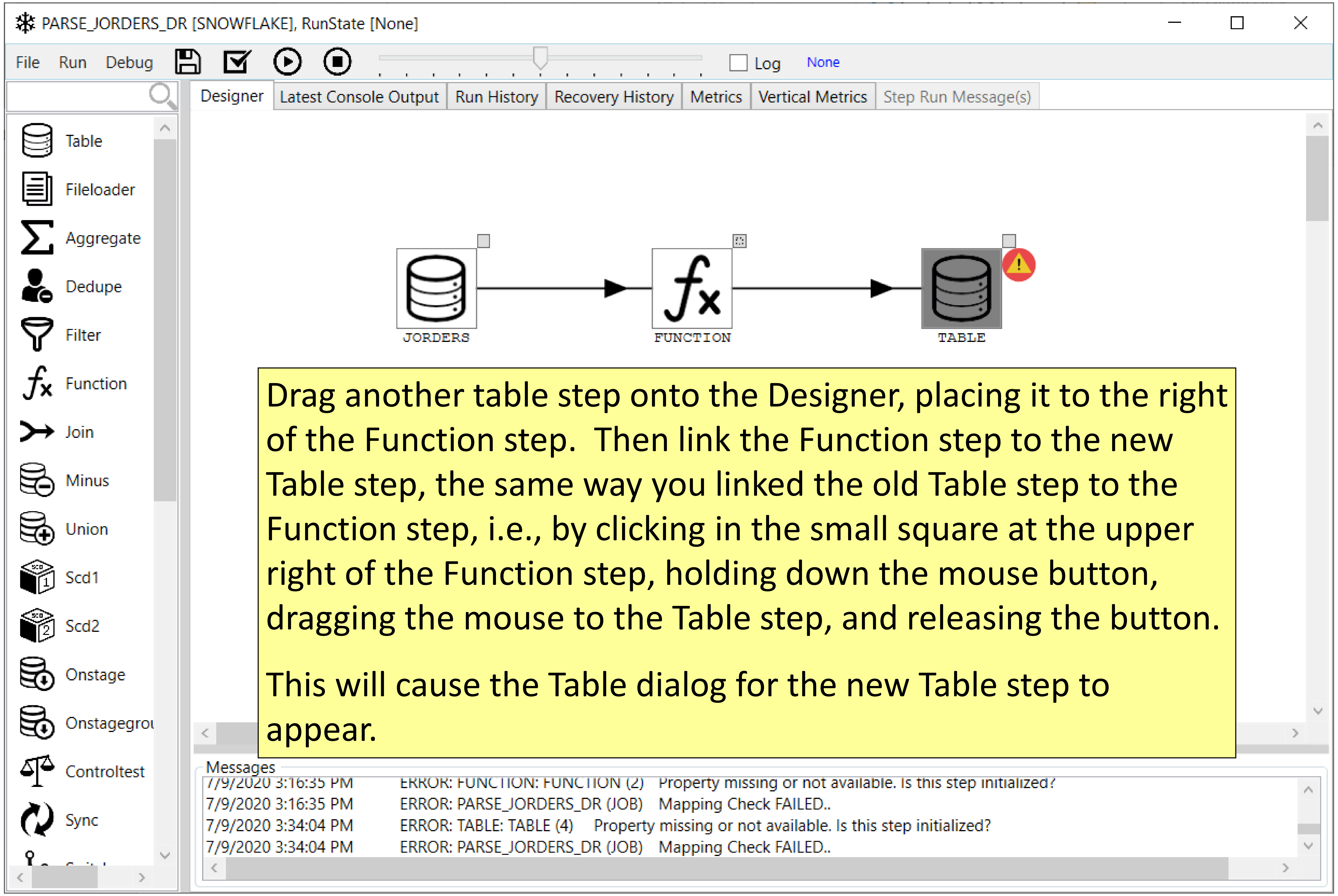


O_ORDERKEY	BIGINT
O_CUSTKEY	BIGINT
O_ORDERSTATUS	VARCHAR(1)
O_TOTALPRICE	NUMBER(12,2)
O_ORDERDATE	DATE
O_ORDERPRIORITY	VARCHAR(15)
O_CLERK	VARCHAR(15)
O_SHIPPRIORITY	BIGINT
O_COMMENT	VARCHAR(80)

Repeat this process to add the remaining columns. The expressions are:

```
PARSE_JSON(ORDERS):o_custkey
PARSE_JSON(ORDERS):o_orderstatus
PARSE_JSON(ORDERS):o_totalprice
PARSE_JSON(ORDERS):o_orderdate
PARSE_JSON(ORDERS):o_orderpriority
PARSE_JSON(ORDERS):o_clerk
PARSE_JSON(ORDERS):o_shippriority
PARSE_JSON(ORDERS):o_comment
```

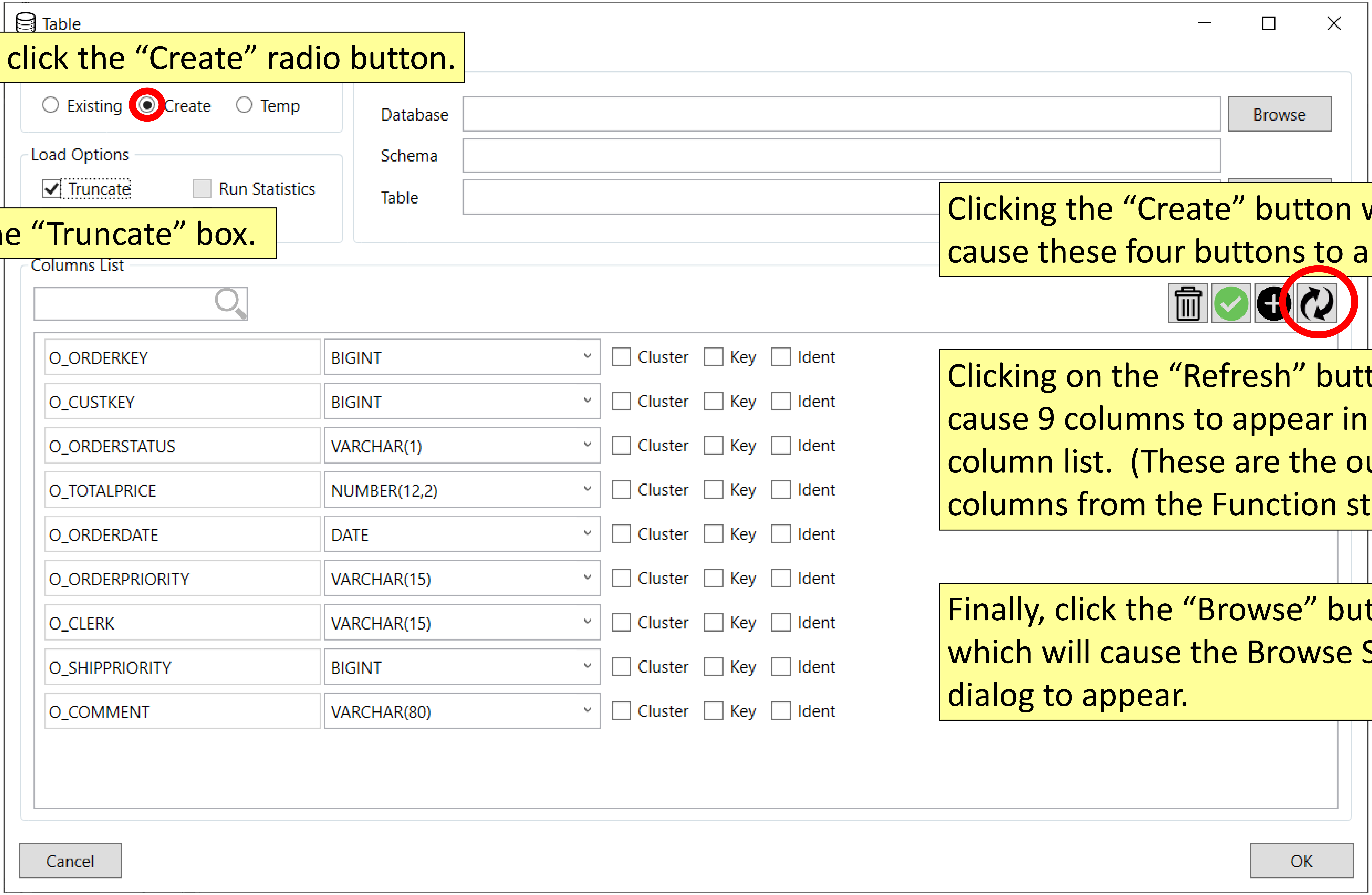
The column names and types are as shown on the left. Then click OK.



```
7/9/2020 3:16:35 PM     ERROR: FUNCTION: FUNCTION (2)    Property missing or not available. Is this step initialized?
7/9/2020 3:16:35 PM     ERROR: PARSE_JORDERS_DR (JOB)    Mapping Check FAILED..
7/9/2020 3:34:04 PM     ERROR: TABLE: TABLE (4)     Property missing or not available. Is this step initialized?
7/9/2020 3:34:04 PM     ERROR: PARSE_JORDERS_DR (JOB)    Mapping Check FAILED..
```

Drag another table step onto the Designer, placing it to the right of the Function step. Then link the Function step to the new Table step, the same way you linked the old Table step to the Function step, i.e., by clicking in the small square at the upper right of the Function step, holding down the mouse button, dragging the mouse to the Table step, and releasing the button.

This will cause the Table dialog for the new Table step to appear.



Table

This time, click the “Create” radio button.

○ Existing ● Create ○ Temp

Database

Schema

Table

Browse

Load Options

☑ Truncate □ Run Statistics

Check the “Truncate” box.

Clicking the “Create” button will cause these four buttons to appear.

Columns List

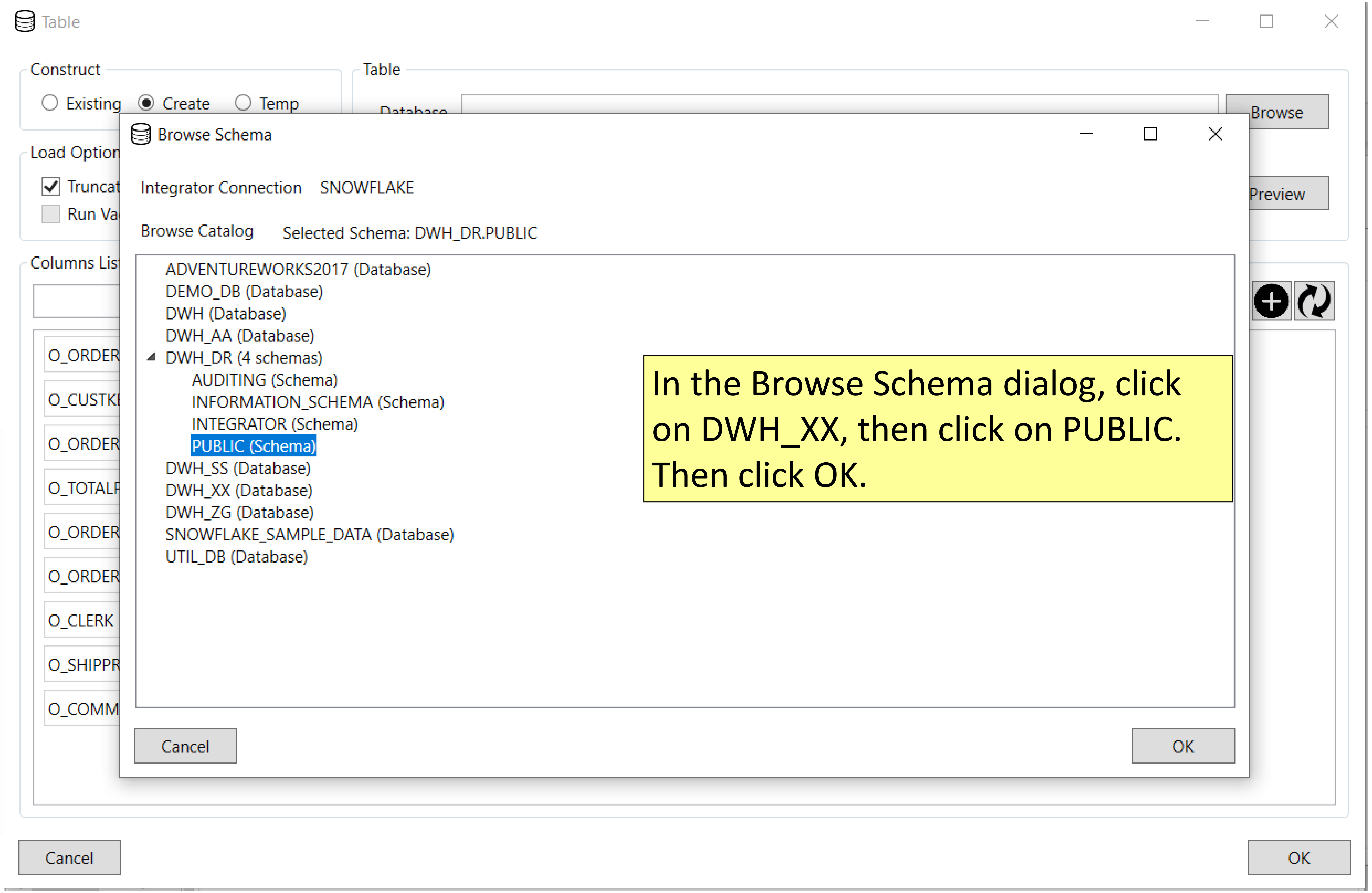
O_ORDERKEY	BIGINT	□ Cluster	□ Key	□ Ident
O_CUSTKEY	BIGINT	□ Cluster	□ Key	□ Ident
O_ORDERSTATUS	VARCHAR(1)	□ Cluster	□ Key	□ Ident
O_TOTALPRICE	NUMBER(12,2)	□ Cluster	□ Key	□ Ident
O_ORDERDATE	DATE	□ Cluster	□ Key	□ Ident
O_ORDERPRIORITY	VARCHAR(15)	□ Cluster	□ Key	□ Ident
O_CLERK	VARCHAR(15)	□ Cluster	□ Key	□ Ident
O_SHIPPRIORITY	BIGINT	□ Cluster	□ Key	□ Ident
O_COMMENT	VARCHAR(80)	□ Cluster	□ Key	□ Ident

Clicking on the “Refresh” button will cause 9 columns to appear in the column list. (These are the output columns from the Function step.)

Finally, click the “Browse” button, which will cause the Browse Schema dialog to appear.

Cancel

OK

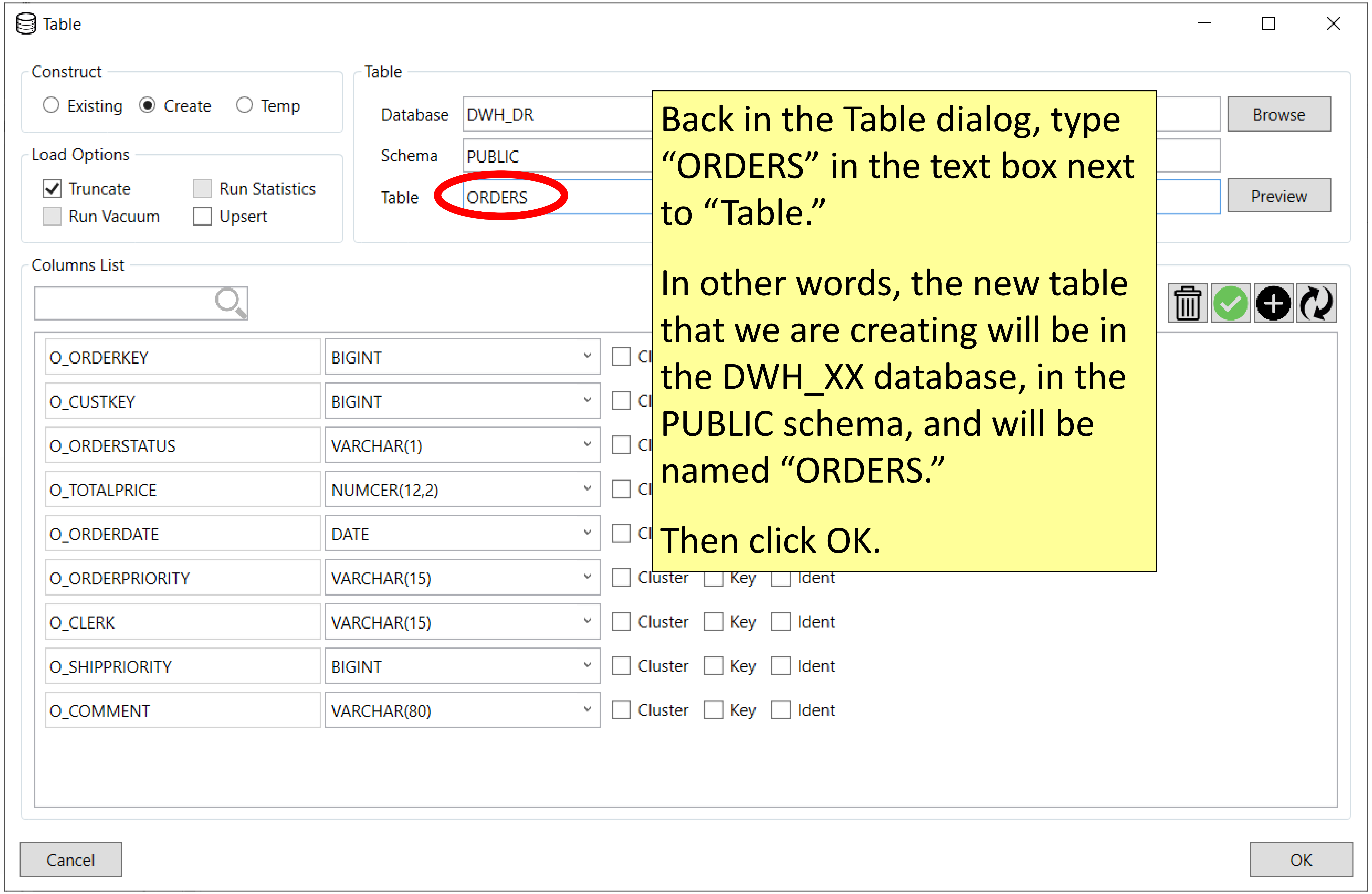


Integrator Connection SNOWFLAKE

Selected Schema: DWH_DR.PUBLIC

ADVENTUREWORKS2017 (Database)
DEMO_DB (Database)
DWH (Database)
DWH_AA (Database)
DWH_DR (4 schemas)
AUDITING (Schema)
INFORMATION_SCHEMA (Schema)
INTEGRATOR (Schema)
PUBLIC (Schema)
DWH_SS (Database)
DWH_XX (Database)
DWH_ZG (Database)
SNOWFLAKE_SAMPLE_DATA (Database)
UTIL_DB (Database)

In the Browse Schema dialog, click on DWH_XX, then click on PUBLIC. Then click OK.



Construct: ○ Existing ● Create ○ Temp

Load Options: ☑ Truncate □ Run Statistics □ Run Vacuum □ Upsert

Table

Database: DWH_DR

Schema: PUBLIC

Table: ORDERS

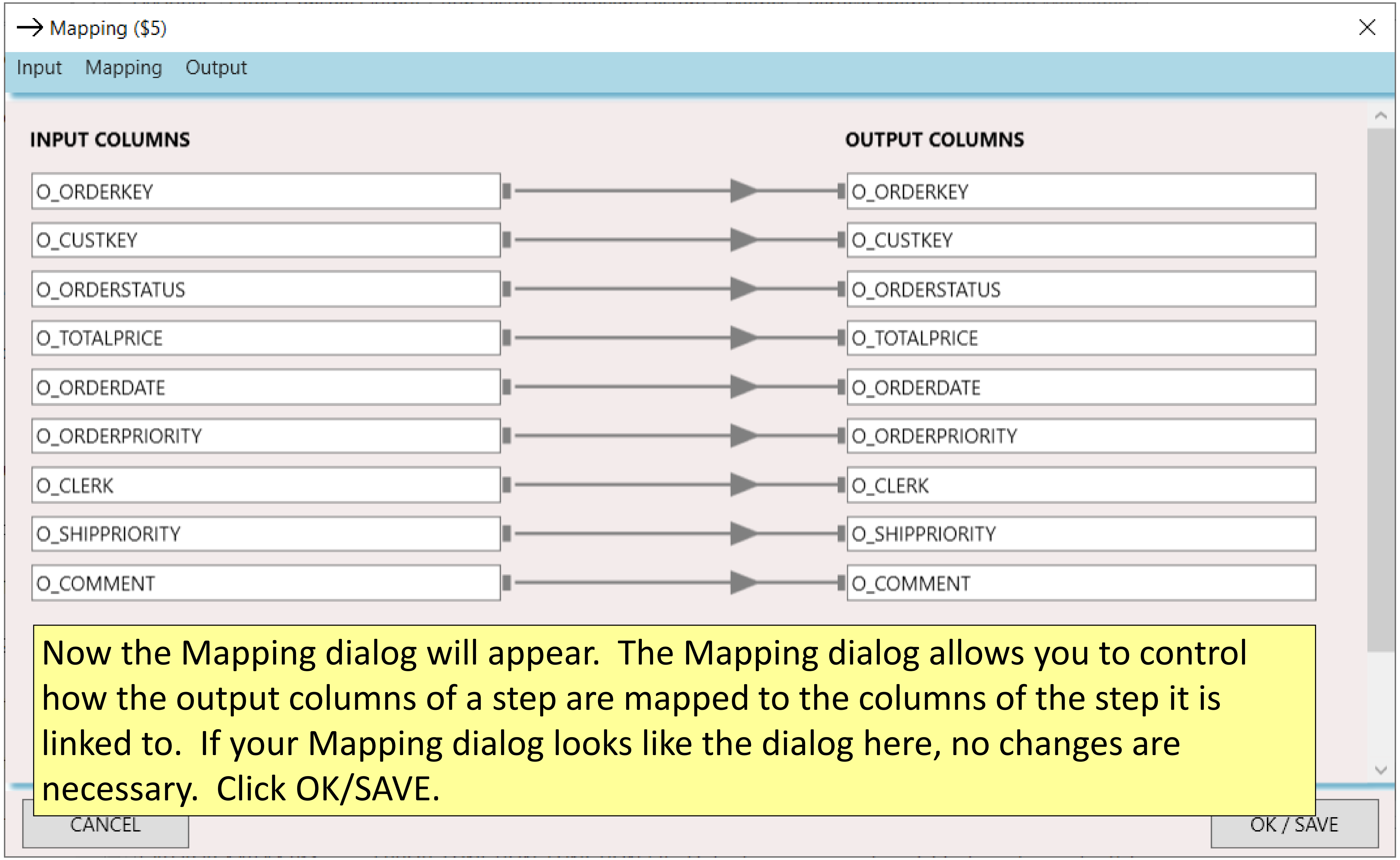
Columns List

O_ORDERKEY	BIGINT	□ Cl		
O_CUSTKEY	BIGINT	□ Cl		
O_ORDERSTATUS	VARCHAR(1)	□ Cl		
O_TOTALPRICE	NUMCER(12,2)	□ Cl		
O_ORDERDATE	DATE	□ Cl		
O_ORDERPRIORITY	VARCHAR(15)	□ Cluster	□ Key	□ Ident
O_CLERK	VARCHAR(15)	□ Cluster	□ Key	□ Ident
O_SHIPPRIORITY	BIGINT	□ Cluster	□ Key	□ Ident
O_COMMENT	VARCHAR(80)	□ Cluster	□ Key	□ Ident

Back in the Table dialog, type “ORDERS” in the text box next to “Table.”

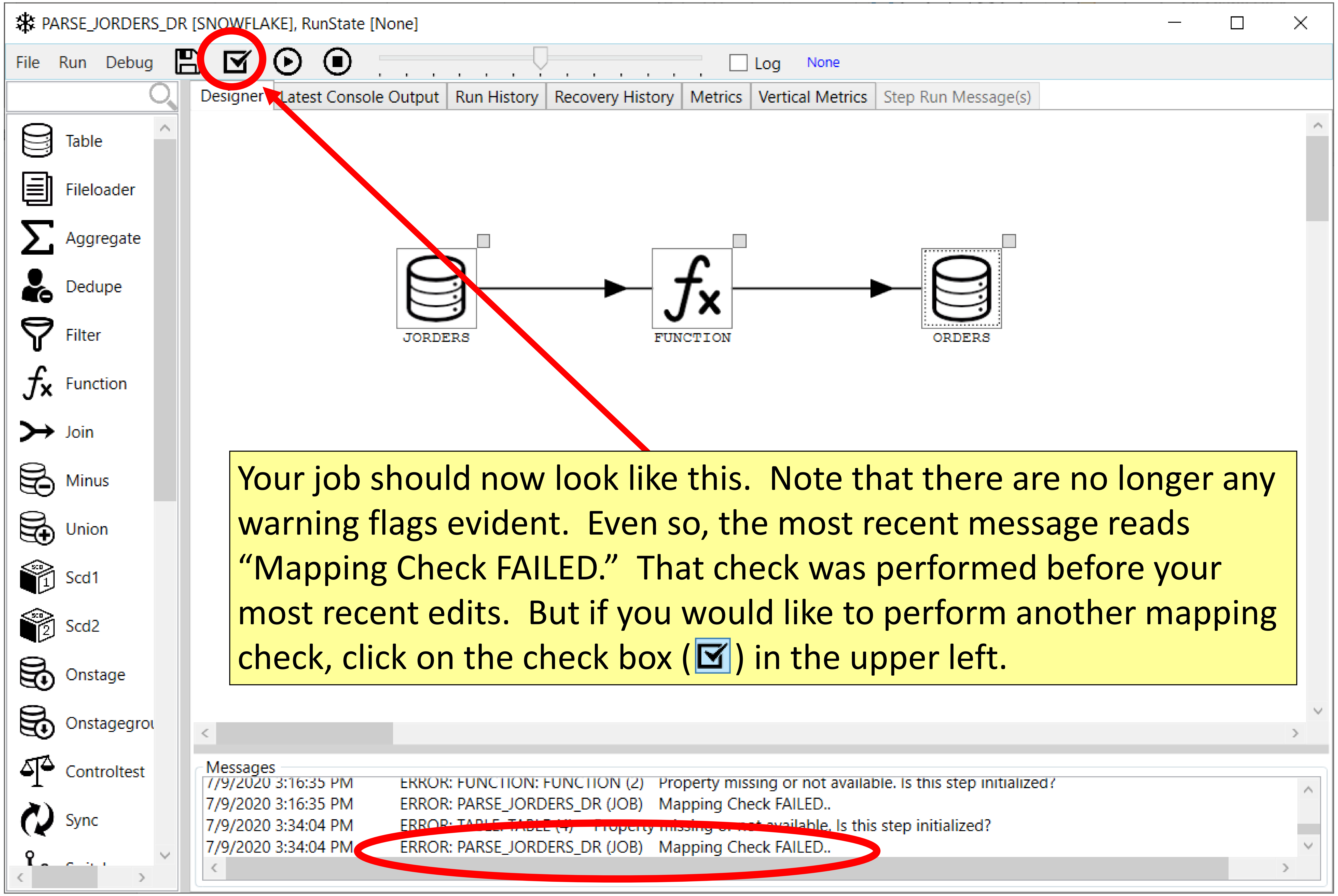
In other words, the new table that we are creating will be in the DWH_XX database, in the PUBLIC schema, and will be named “ORDERS.”

Then click OK.



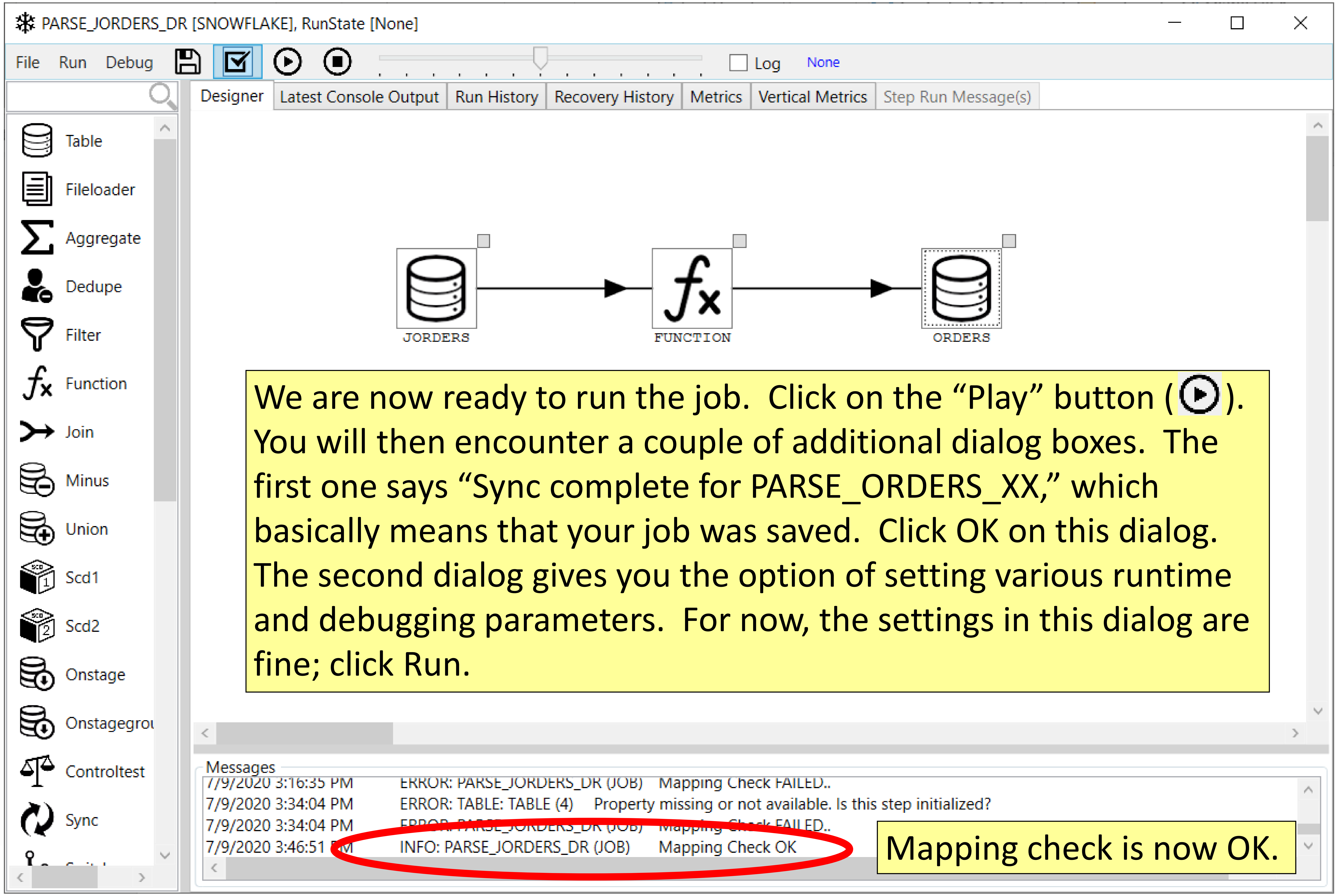
INPUT COLUMNS	OUTPUT COLUMNS
O_ORDERKEY	O_ORDERKEY
O_CUSTKEY	O_CUSTKEY
O_ORDERSTATUS	O_ORDERSTATUS
O_TOTALPRICE	O_TOTALPRICE
O_ORDERDATE	O_ORDERDATE
O_ORDERPRIORITY	O_ORDERPRIORITY
O_CLERK	O_CLERK
O_SHIPPRIORITY	O_SHIPPRIORITY
O_COMMENT	O_COMMENT

Now the Mapping dialog will appear. The Mapping dialog allows you to control how the output columns of a step are mapped to the columns of the step it is linked to. If your Mapping dialog looks like the dialog here, no changes are necessary. Click OK/SAVE.



Your job should now look like this. Note that there are no longer any warning flags evident. Even so, the most recent message reads “Mapping Check FAILED.” That check was performed before your most recent edits. But if you would like to perform another mapping check, click on the check box (☑) in the upper left.

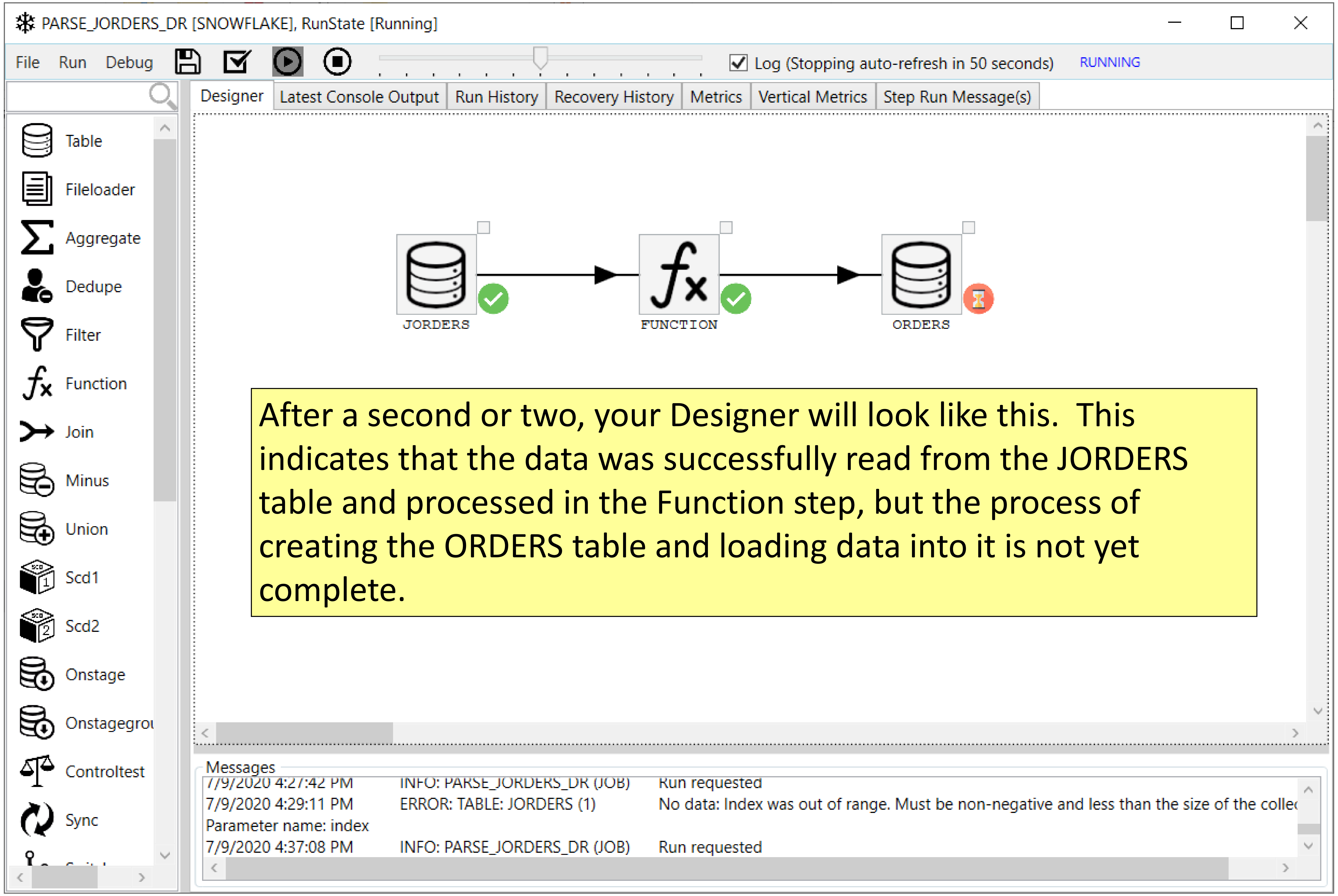
```
7/9/2020 3:16:35 PM     ERROR: FUNCTION: FUNCTION (2)   Property missing or not available. Is this step initialized?
7/9/2020 3:16:35 PM     ERROR: PARSE_JORDERS_DR (JOB)   Mapping Check FAILED..
7/9/2020 3:34:04 PM     ERROR: TABLE: TABLE (4)     Property missing or not available. Is this step initialized?
7/9/2020 3:34:04 PM     ERROR: PARSE_JORDERS_DR (JOB)   Mapping Check FAILED..
```



We are now ready to run the job. Click on the "Play" button (▶). You will then encounter a couple of additional dialog boxes. The first one says "Sync complete for PARSE_ORDERS_XX," which basically means that your job was saved. Click OK on this dialog. The second dialog gives you the option of setting various runtime and debugging parameters. For now, the settings in this dialog are fine; click Run.

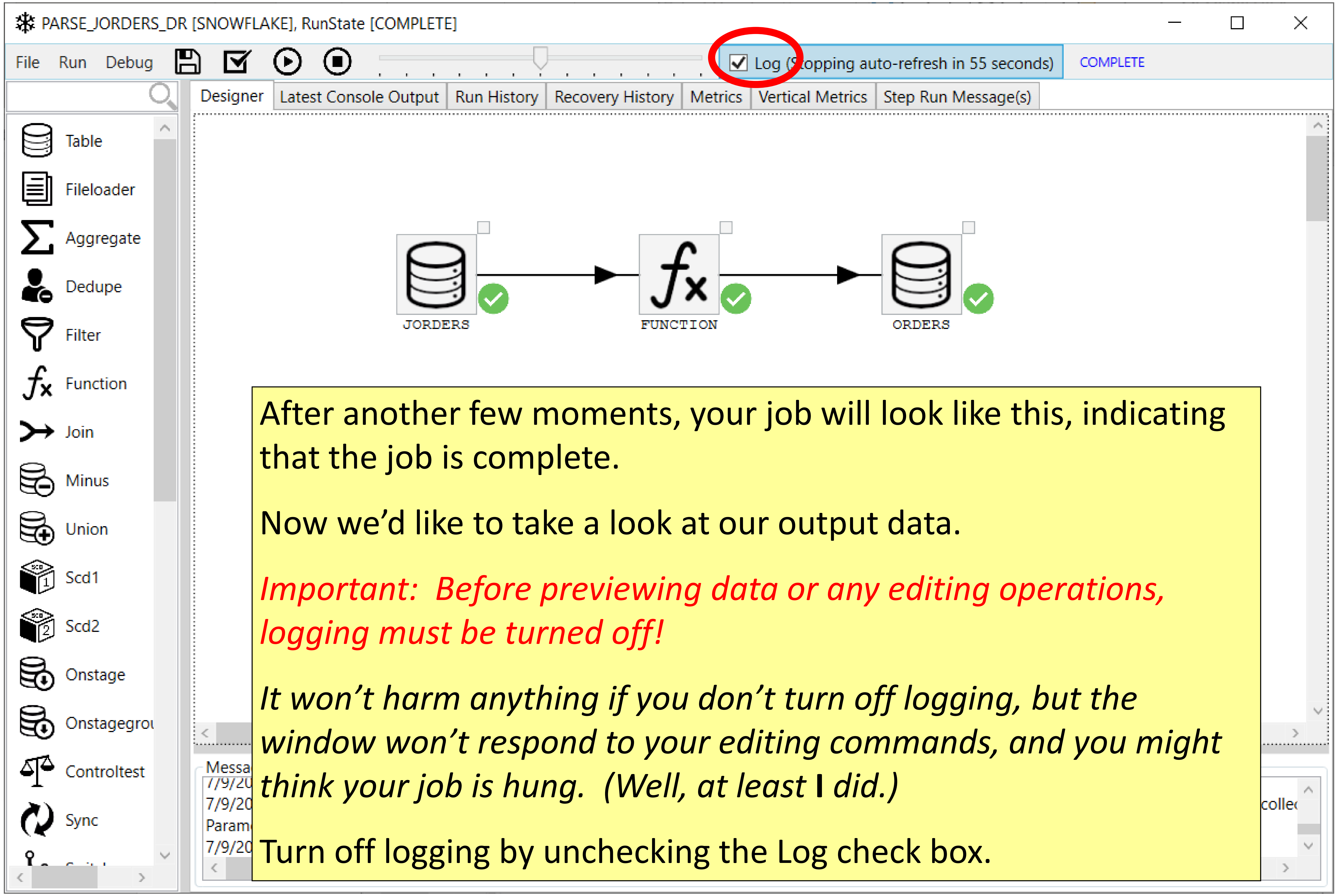
```
7/9/2020 3:16:35 PM     ERROR: PARSE_JORDERS_DR (JOB)    Mapping Check FAILED..
7/9/2020 3:34:04 PM     ERROR: TABLE: TABLE (4)    Property missing or not available. Is this step initialized?
7/9/2020 3:34:04 PM     ERROR: PARSE_JORDERS_DR (JOB)    Mapping Check FAILED..
7/9/2020 3:46:51 PM     INFO: PARSE_JORDERS_DR (JOB)    Mapping Check OK
```

Mapping check is now OK.



After a second or two, your Designer will look like this. This indicates that the data was successfully read from the JORDERS table and processed in the Function step, but the process of creating the ORDERS table and loading data into it is not yet complete.

```
7/9/2020 4:27:42 PM     INFO: PARSE_JORDERS_DR (JOB)     Run requested
7/9/2020 4:29:11 PM     ERROR: TABLE: JORDERS (1)        No data: Index was out of range. Must be non-negative and less than the size of the collec
Parameter name: index
7/9/2020 4:37:08 PM     INFO: PARSE_JORDERS_DR (JOB)     Run requested
```



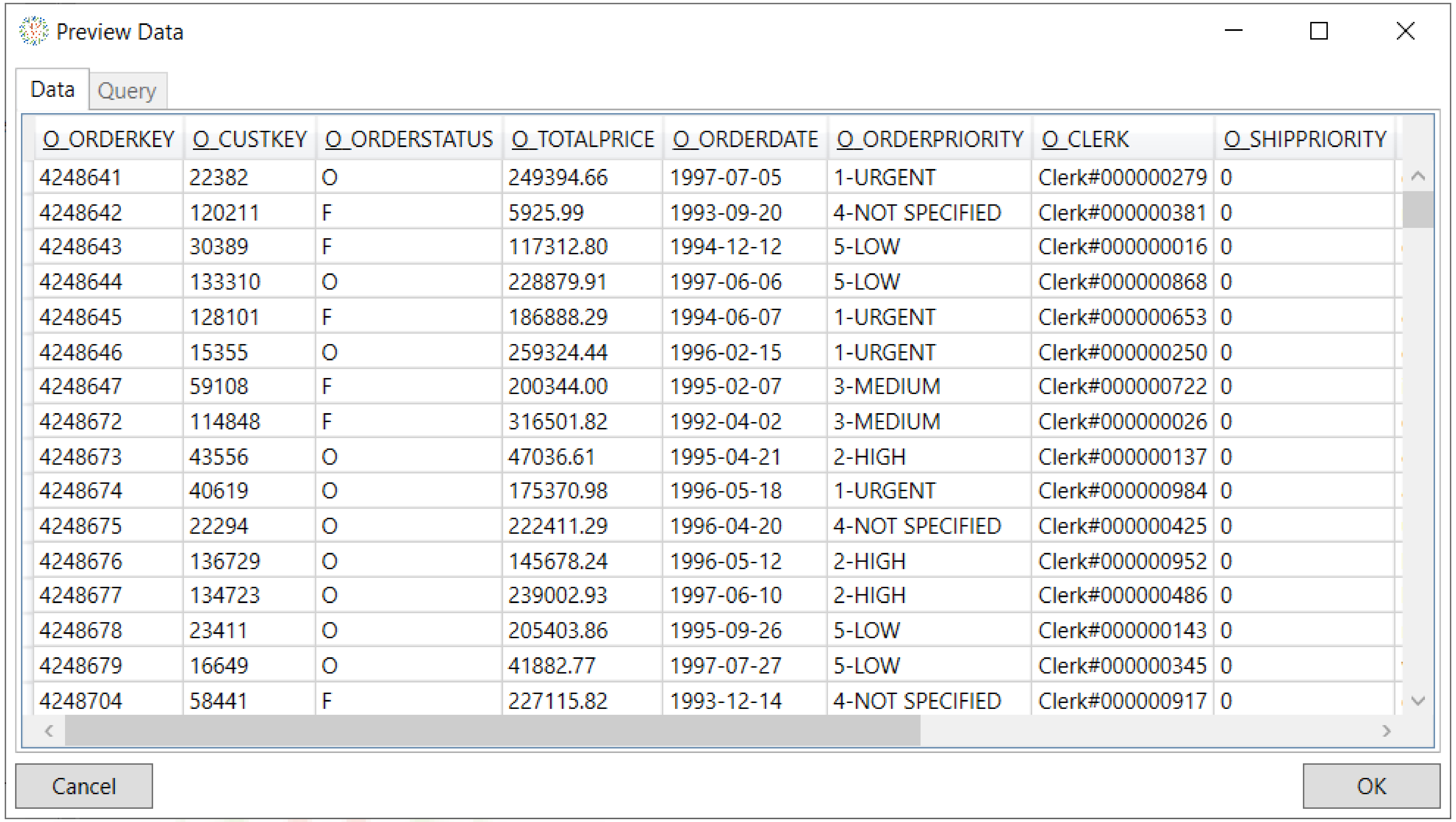
After another few moments, your job will look like this, indicating that the job is complete.

Now we'd like to take a look at our output data.

Important: Before previewing data or any editing operations, logging must be turned off!

*It won't harm anything if you don't turn off logging, but the window won't respond to your editing commands, and you might think your job is hung. (Well, at least **I** did.)*

Turn off logging by unchecking the Log check box.



O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERDATE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY
4248641	22382	O	249394.66	1997-07-05	1-URGENT	Clerk#000000279	0
4248642	120211	F	5925.99	1993-09-20	4-NOT SPECIFIED	Clerk#000000381	0
4248643	30389	F	117312.80	1994-12-12	5-LOW	Clerk#000000016	0
4248644	133310	O	228879.91	1997-06-06	5-LOW	Clerk#000000868	0
4248645	128101	F	186888.29	1994-06-07	1-URGENT	Clerk#000000653	0
4248646	15355	O	259324.44	1996-02-15	1-URGENT	Clerk#000000250	0
4248647	59108	F	200344.00	1995-02-07	3-MEDIUM	Clerk#000000722	0
4248672	114848	F	316501.82	1992-04-02	3-MEDIUM	Clerk#000000026	0
4248673	43556	O	47036.61	1995-04-21	2-HIGH	Clerk#000000137	0
4248674	40619	O	175370.98	1996-05-18	1-URGENT	Clerk#000000984	0
4248675	22294	O	222411.29	1996-04-20	4-NOT SPECIFIED	Clerk#000000425	0
4248676	136729	O	145678.24	1996-05-12	2-HIGH	Clerk#000000952	0
4248677	134723	O	239002.93	1997-06-10	2-HIGH	Clerk#000000486	0
4248678	23411	O	205403.86	1995-09-26	5-LOW	Clerk#000000143	0
4248679	16649	O	41882.77	1997-07-27	5-LOW	Clerk#000000345	0
4248704	58441	F	227115.82	1993-12-14	4-NOT SPECIFIED	Clerk#000000917	0

Your data should look something like this.

Adding a date dimension

- We’ve just constructed the ORDERS table, which looks like a good candidate for a fact table.
- Next, we want to make some changes so that we can make interesting date-related queries about orders.

Date-related queries

- We can already make some DW-like date-related queries on the ORDERS table. ORDERS contains the column O_ORDERDATE, so we can write the query

```
select * from ORDERS where O_ORDERDATE between TO_DATE('1998-07-01') AND
TO_DATE('1998-08-31’);
```

which will retrieve all the orders in 1998 between July and August.

- But suppose we want to retrieve all the orders for Wednesdays in 1998? Or suppose we wanted to know which day of the week had the most orders? What if we wanted to exclude holidays?

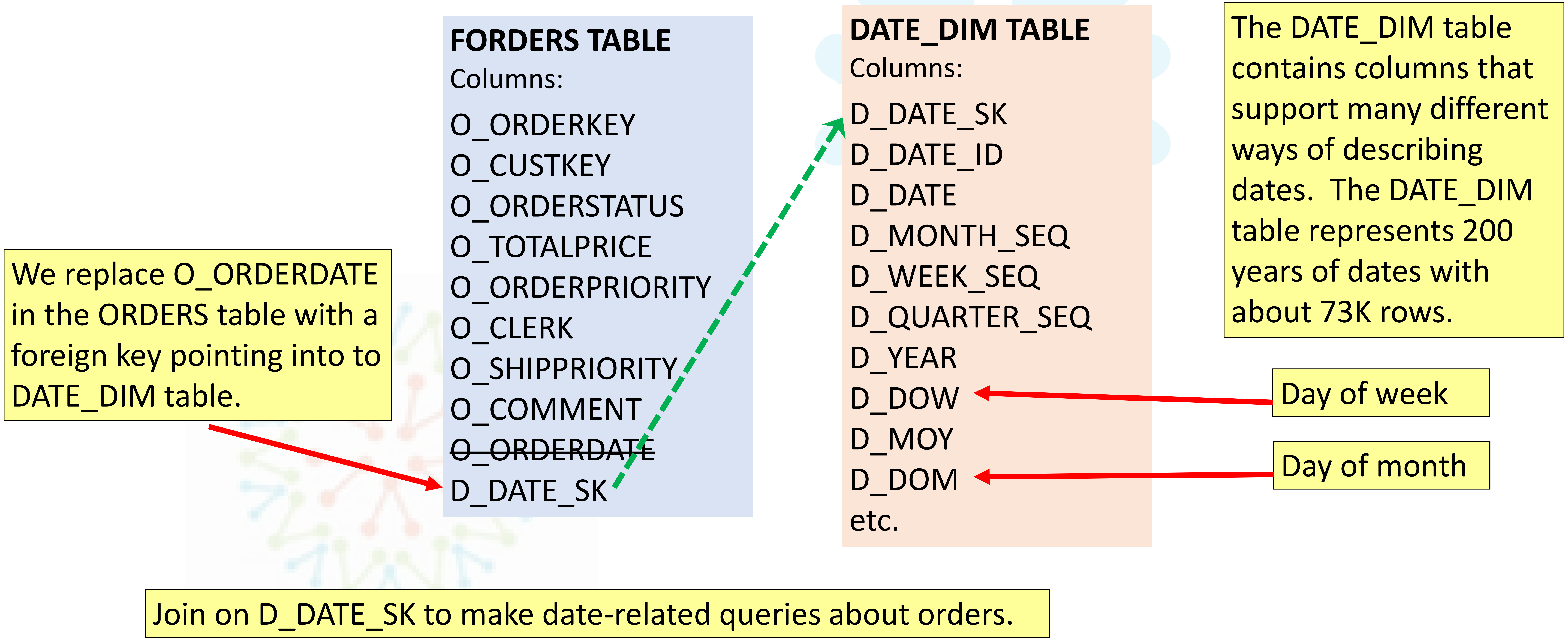
Date-related queries, cont.

- One way of handling these kinds of queries is to add a set of new columns to the ORDER table, with new ways of representing dates.
- New columns might include DAY_OF_WEEK, DAY_OF_MONTH, MONTH, YEAR, HOLIDAY, WEEKEND, QUARTER, and so on.
- Is this an ideal solution? Well, it has the following drawbacks:
  - Other fact tables will probably require the same changes, and the new date-related columns will need to be maintained in multiple places.
  - Fact tables tend to be long (e.g., billions of rows). Adding new columns to a long table is costly in terms of storage.
  - (That was a No.)

The Date Dimension Table

- **A better solution is to gather all the date representations in a single table, called the *date dimension* table.**
- **This table is then *joined* to the ORDERS table and other fact tables when we want to make date-related queries.**
- **The computational cost of the join, even for large tables, is small. Platforms like Snowflake perform well on such operations.**

The Date Dimension Table



We replace O_ORDERDATE in the ORDERS table with a foreign key pointing into to DATE_DIM table.

FORDERS TABLE

Columns:

O_ORDERKEY
O_CUSTKEY
O_ORDERSTATUS
O_TOTALPRICE
O_ORDERPRIORITY
O_CLERK
O_SHIPPRIORITY
O_COMMENT
~~O_ORDERDATE~~
D_DATE_SK

DATE_DIM TABLE

Columns:

D_DATE_SK
D_DATE_ID
D_DATE
D_MONTH_SEQ
D_WEEK_SEQ
D_QUARTER_SEQ
D_YEAR
D_DOW
D_MOY
D_DOM
etc.

The DATE_DIM table contains columns that support many different ways of describing dates. The DATE_DIM table represents 200 years of dates with about 73K rows.

Day of week

Day of month

Join on D_DATE_SK to make date-related queries about orders.

Querying a joined fact and dimension table

To find all the orders that occurred on Wednesdays in 1998:

```
SELECT * from FORDERS F, DATE_DIM D where F.D_DATE_SK = D.D_DATE_SK and D.D_YEAR
= 1998 and D.D_DOW = 3;
```

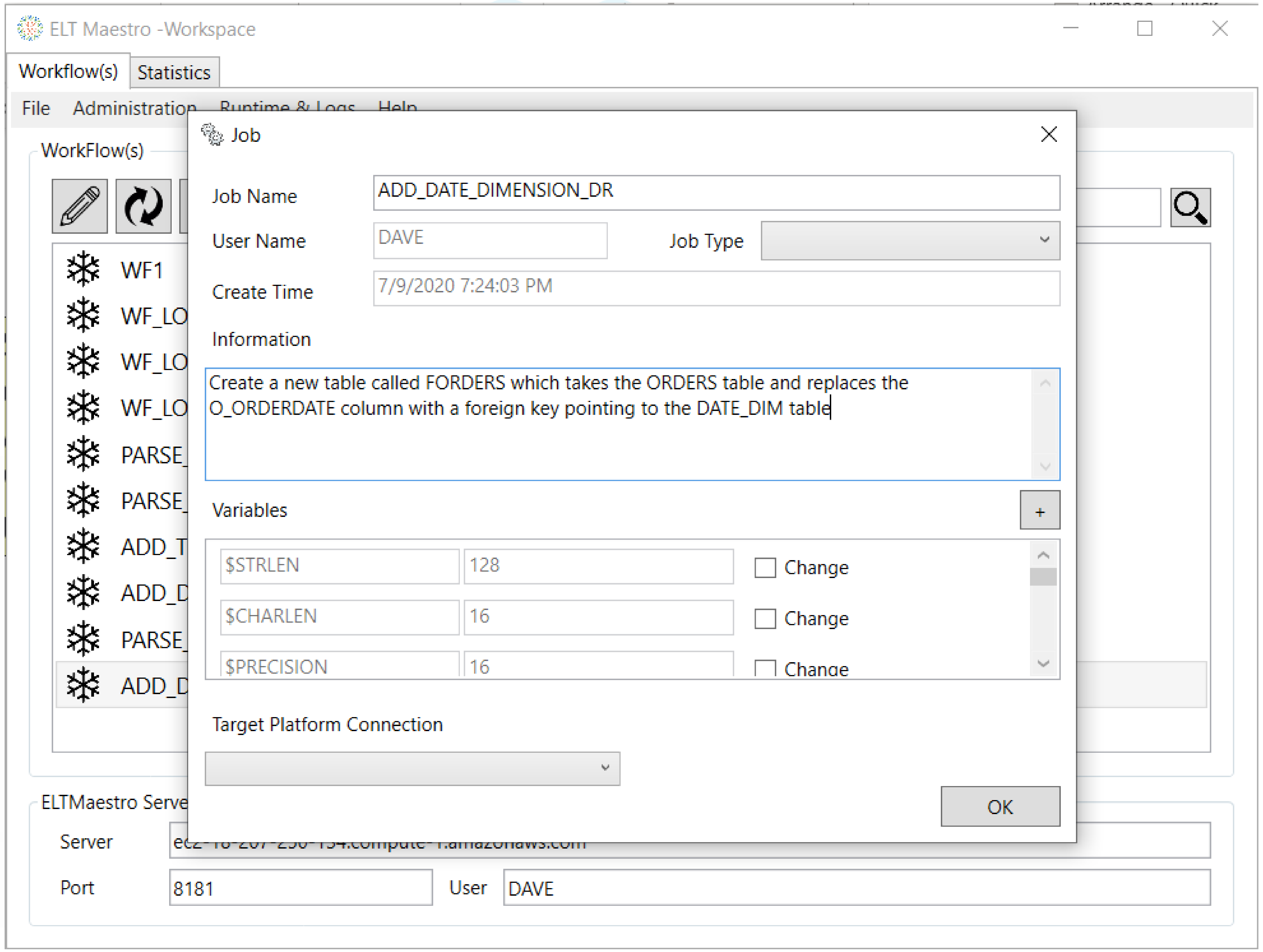
Which year between 1995 and 2000 had the most successful 4th quarter (as measured by order count)?

```
SELECT count(*) C, D.D_YEAR Y FROM FORDERS F, DATE_DIM D where F.D_DATE_SK =
D.D_DATE_SK and D.D_YEAR BETWEEN 1995 AND 2000 AND D.D_QOY = 4 GROUP BY Y
ORDER BY C; -- Gives results for all quarters in order; use max to select best.
```

Which year between 1995 and 2000 had the most successful 4th quarter (as measured by sum of total price)?

```
SELECT sum(F.O_TOTALPRICE ) S, D.D_YEAR Y FROM FORDERS F, DATE_DIM D where
F.D_DATE_SK = D.D_DATE_SK and D.D_YEAR BETWEEN 1995 AND 2000 AND D.D_QOY = 4
GROUP BY Y ORDER BY S;
```

Starting in the Workspace, create a new job called ADD_DATE_DIMENSION_XX. When the UI asks if you want to edit it, click Yes.



Job Name: ADD_DATE_DIMENSION_DR

User Name: DAVE

Job Type:

Create Time: 7/9/2020 7:24:03 PM

Information

Create a new table called FORDERS which takes the ORDERS table and replaces the O_ORDERDATE column with a foreign key pointing to the DATE_DIM table

Variables +

$STRLEN	128	□ Change
$CHARLEN	16	□ Change
$PRECISION	16	□ Change

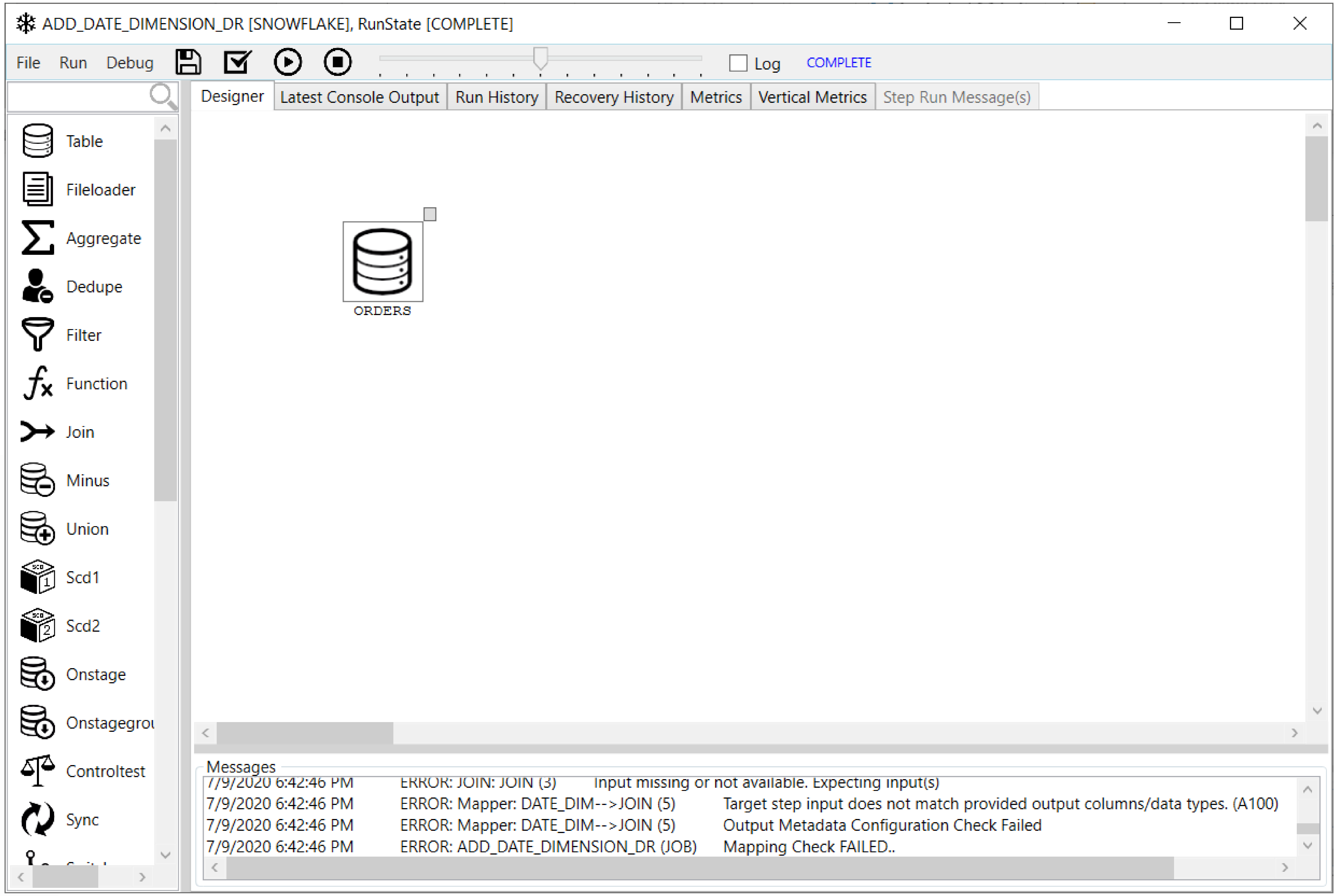
Target Platform Connection

OK

Server: ec2-18-207-250-134.compute-1.amazonaws.com

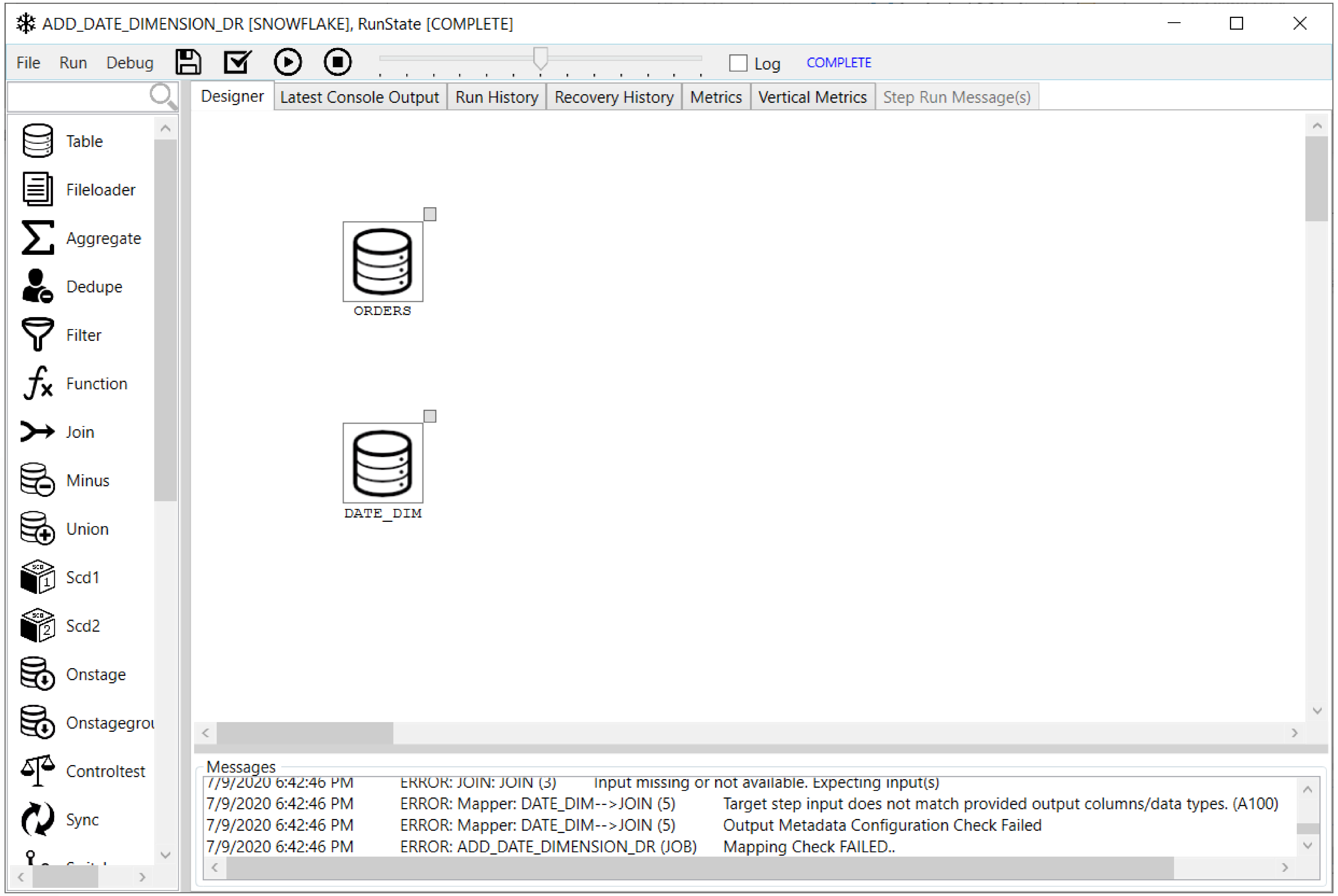
Port: 8181 User: DAVE

Drag a table step onto the Designer area of your new job. Make sure the “Existing” radio button is selected and click the “Browse” button. Select DWH_XX.PUBLIC.ORDERS. Click OK to exit the Browse dialog, and click OK again to exit the Table dialog.



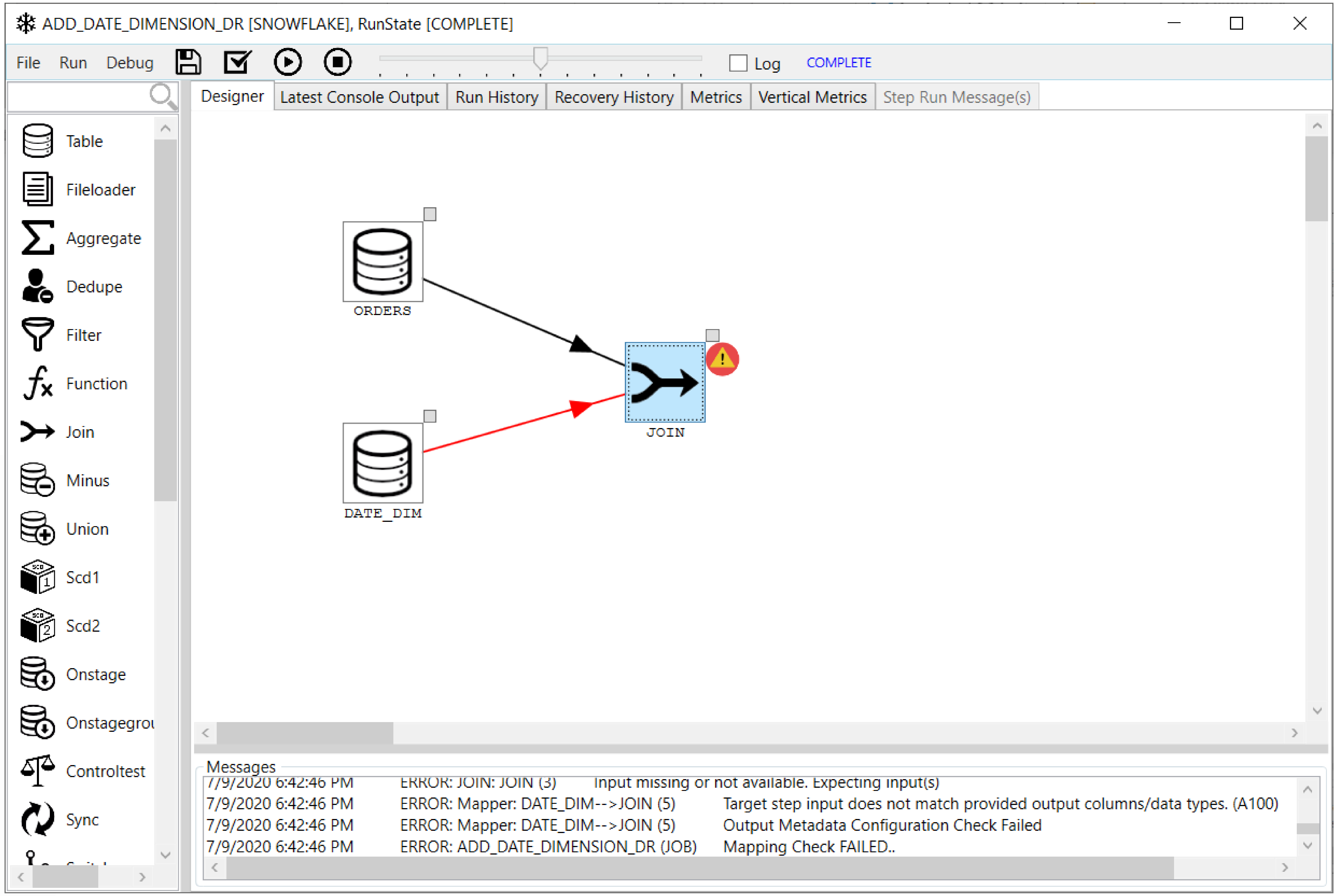
```
7/9/2020 6:42:46 PM    ERROR: JOIN: JOIN (3)    Input missing or not available. Expecting input(s)
7/9/2020 6:42:46 PM    ERROR: Mapper: DATE_DIM-->JOIN (5)    Target step input does not match provided output columns/data types. (A100)
7/9/2020 6:42:46 PM    ERROR: Mapper: DATE_DIM-->JOIN (5)    Output Metadata Configuration Check Failed
7/9/2020 6:42:46 PM    ERROR: ADD_DATE_DIMENSION_DR (JOB)    Mapping Check FAILED..
```

Repeat the steps from the last slide, but this time select the DATE_DIM table.



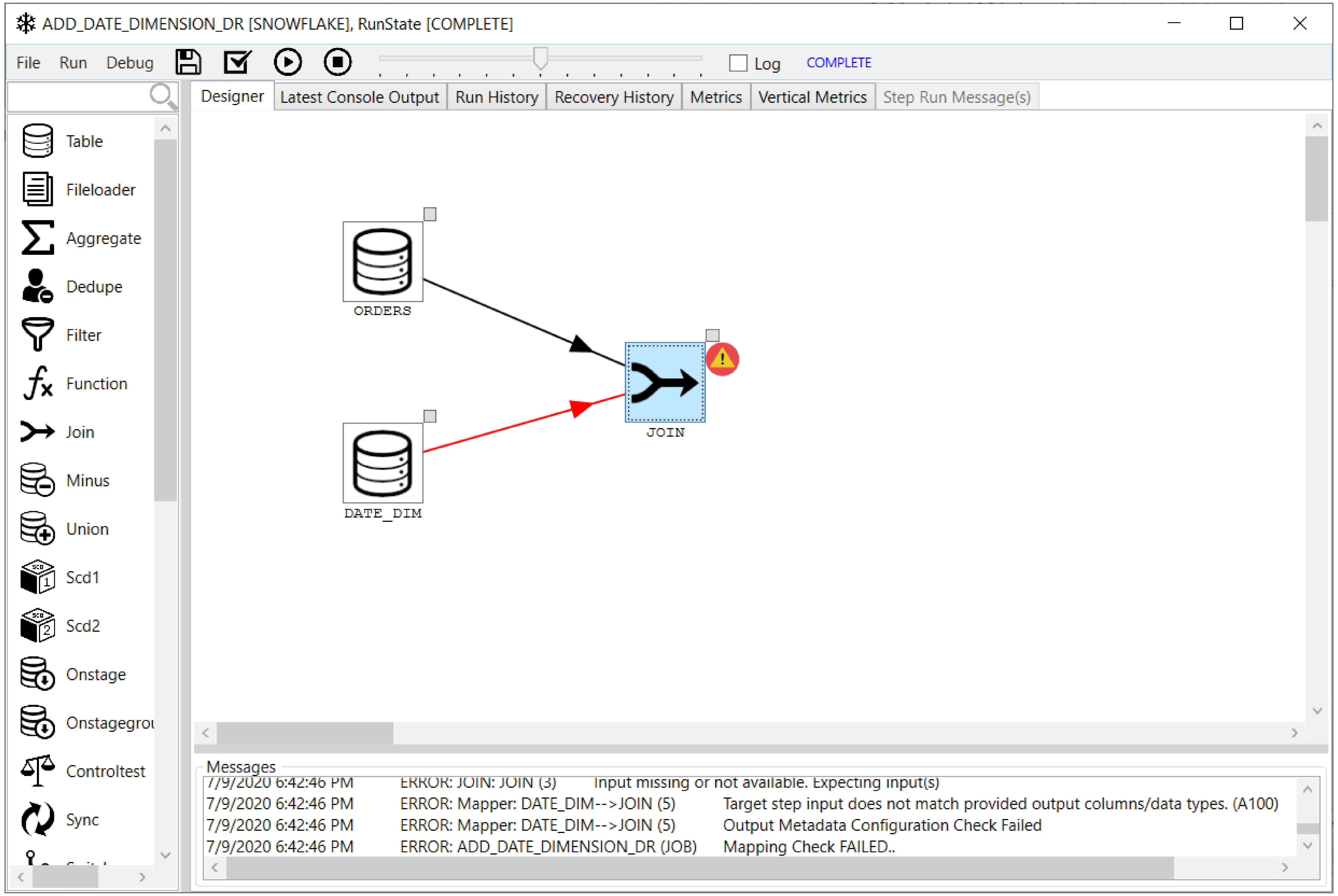
```
7/9/2020 6:42:46 PM    ERROR: JOIN: JOIN (3)       Input missing or not available. Expecting input(s)
7/9/2020 6:42:46 PM    ERROR: Mapper: DATE_DIM-->JOIN (5)      Target step input does not match provided output columns/data types. (A100)
7/9/2020 6:42:46 PM    ERROR: Mapper: DATE_DIM-->JOIN (5)      Output Metadata Configuration Check Failed
7/9/2020 6:42:46 PM    ERROR: ADD_DATE_DIMENSION_DR (JOB)      Mapping Check FAILED..
```

Drag a Join step onto the Designer and Join the two table steps to it as shown.



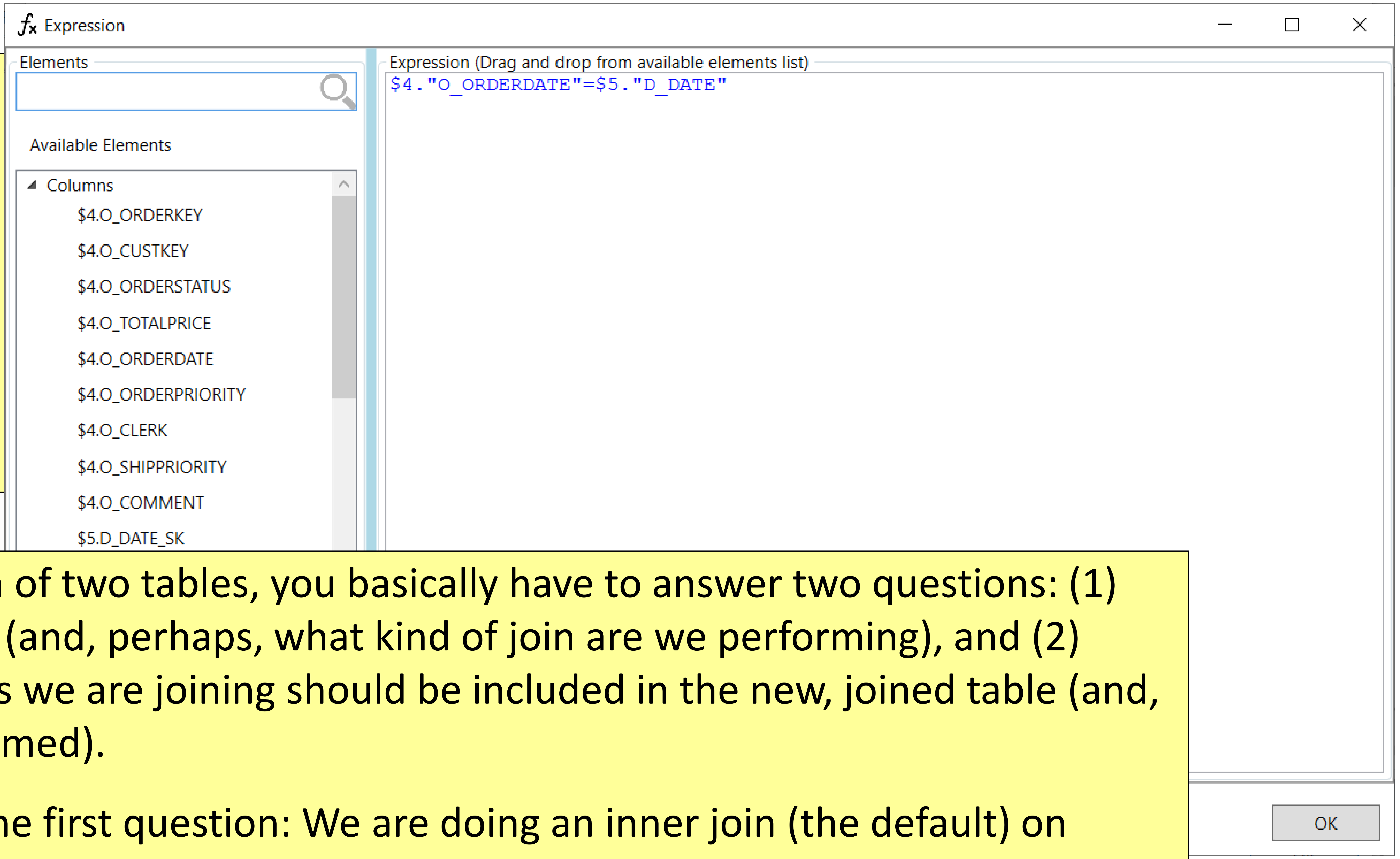
```
7/9/2020 6:42:46 PM     ERROR: JOIN: JOIN (3)      Input missing or not available. Expecting input(s)
7/9/2020 6:42:46 PM     ERROR: Mapper: DATE_DIM-->JOIN (5)      Target step input does not match provided output columns/data types. (A100)
7/9/2020 6:42:46 PM     ERROR: Mapper: DATE_DIM-->JOIN (5)      Output Metadata Configuration Check Failed
7/9/2020 6:42:46 PM     ERROR: ADD_DATE_DIMENSION_DR (JOB)      Mapping Check FAILED..
```

Double-click on the Join step to edit its properties. In the drop-down box labeled “First Join Source,” choose the number corresponding to “ORDER” (probably $4). Then click the button that says “Add.” Leave the Type as Inner Join, and in the drop-down box corresponding to “Join With,” choose the number corresponding to “DATE_DIM” (probably $5). Then click the button that says Expr.



```
7/9/2020 6:42:46 PM    ERROR: JOIN: JOIN (3)      Input missing or not available. Expecting input(s)
7/9/2020 6:42:46 PM    ERROR: Mapper: DATE_DIM-->JOIN (5)      Target step input does not match provided output columns/data types. (A100)
7/9/2020 6:42:46 PM    ERROR: Mapper: DATE_DIM-->JOIN (5)      Output Metadata Configuration Check Failed
7/9/2020 6:42:46 PM    ERROR: ADD_DATE_DIMENSION_DR (JOB)      Mapping Check FAILED..
```

The Expression dialog will appear. Expand the Columns section in the Available Elements list on the left, and drag the columns “4."O_ORDERDATE“ and $5."D_DATE“ onto the Expression area, setting them equal as shown here. Then click OK.



Columns
- $4.O_ORDERKEY
- $4.O_CUSTKEY
- $4.O_ORDERSTATUS
- $4.O_TOTALPRICE
- $4.O_ORDERDATE
- $4.O_ORDERPRIORITY
- $4.O_CLERK
- $4.O_SHIPPRIORITY
- $4.O_COMMENT
- $5.D_DATE_SK

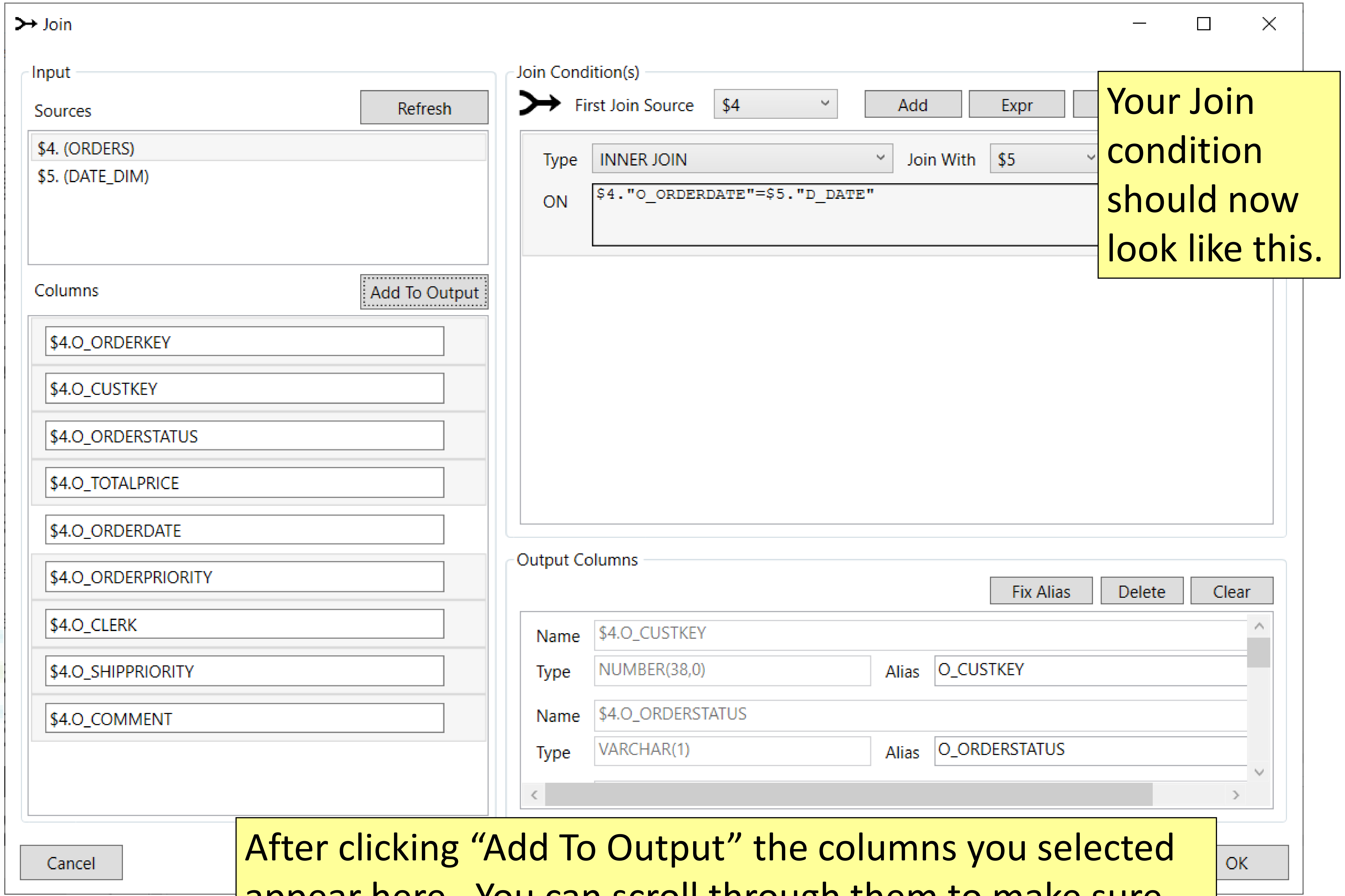
```
$4."O_ORDERDATE"=$5."D_DATE"
```

What’s going on? To specify a join of two tables, you basically have to answer two questions: (1) What column(s) are we joining on (and, perhaps, what kind of join are we performing), and (2) What columns from the two tables we are joining should be included in the new, joined table (and, perhaps, how should they be renamed).

At this point, we have answered the first question: We are doing an inner join (the default) on O_ORDERDATE from the ORDERS table and D_DATE from the DATE_DIM table.

The rest of this section is devoted the second question: Specifying which columns from ORDERS and DATE_DIM will be included in the new FORDERS table.

Now select “$4.(ORDERS)” under Sources. This will cause the columns from the ORDERS table to appear in the Columns box below. Select all of the columns *except* O_ORDERDATE. (Hold down the control key while clicking to make multiple selections.) Then click on the “Add To Output” button.



Sources

- $4. (ORDERS)
- $5. (DATE_DIM)

Columns

- $4.O_ORDERKEY
- $4.O_CUSTKEY
- $4.O_ORDERSTATUS
- $4.O_TOTALPRICE
- $4.O_ORDERDATE
- $4.O_ORDERPRIORITY
- $4.O_CLERK
- $4.O_SHIPPRIORITY
- $4.O_COMMENT

First Join Source: $4

Type: INNER JOIN   Join With: $5

ON:

```
$4."O_ORDERDATE"=$5."D_DATE"
```

Name: $4.O_CUSTKEY
Type: NUMBER(38,0)   Alias: O_CUSTKEY

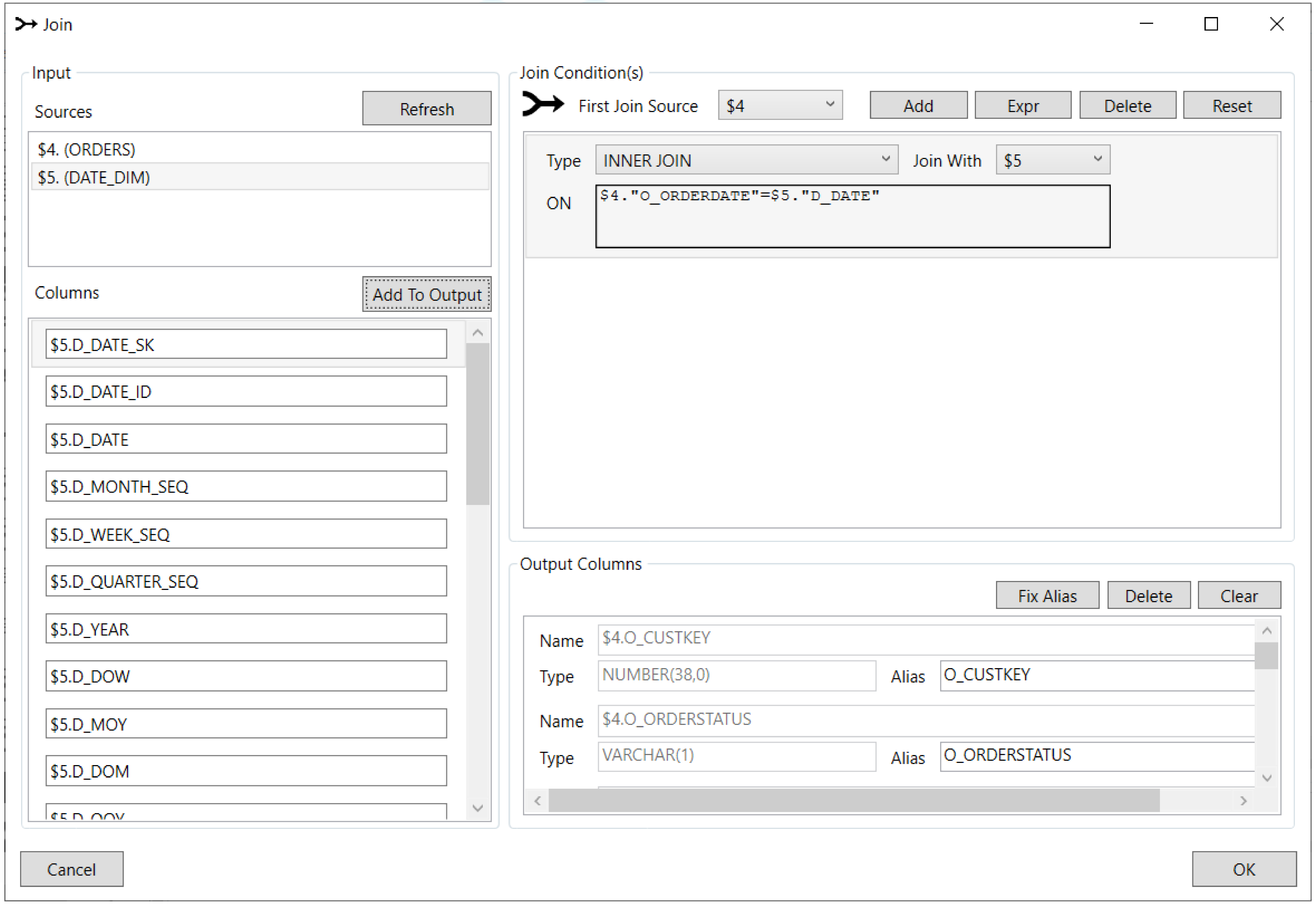
Name: $4.O_ORDERSTATUS
Type: VARCHAR(1)   Alias: O_ORDERSTATUS

Your Join condition should now look like this.

After clicking “Add To Output” the columns you selected appear here. You can scroll through them to make sure you got all the ORDERS columns except O_ORDERDATE.

Next, select “$5.(DATE_DIM)” under Sources. From the list of columns below, just select D_DATE_SK. Then click “Add To Output.”

Verify that the output columns consist of the ORDER table’s columns, with O_DATEDATE replaced by D_DATE_SK. Then click OK.



Sources:
- $4. (ORDERS)
- $5. (DATE_DIM)

Columns:
- $5.D_DATE_SK
- $5.D_DATE_ID
- $5.D_DATE
- $5.D_MONTH_SEQ
- $5.D_WEEK_SEQ
- $5.D_QUARTER_SEQ
- $5.D_YEAR
- $5.D_DOW
- $5.D_MOY
- $5.D_DOM
- $5.D_QOY

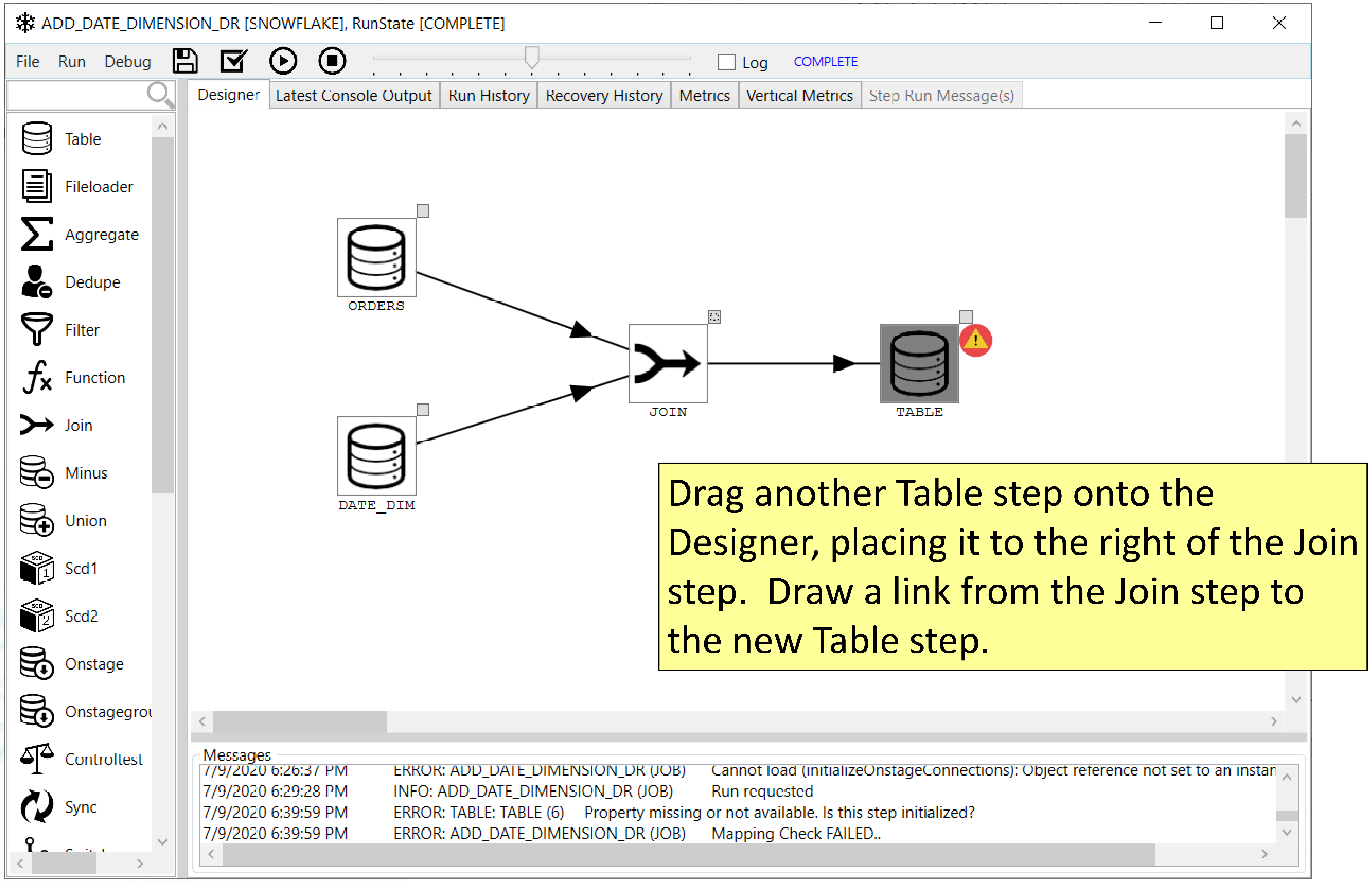
First Join Source: $4

Type: INNER JOIN Join With: $5

ON:
```
$4."O_ORDERDATE"=$5."D_DATE"
```

Name: $4.O_CUSTKEY
Type: NUMBER(38,0) Alias: O_CUSTKEY

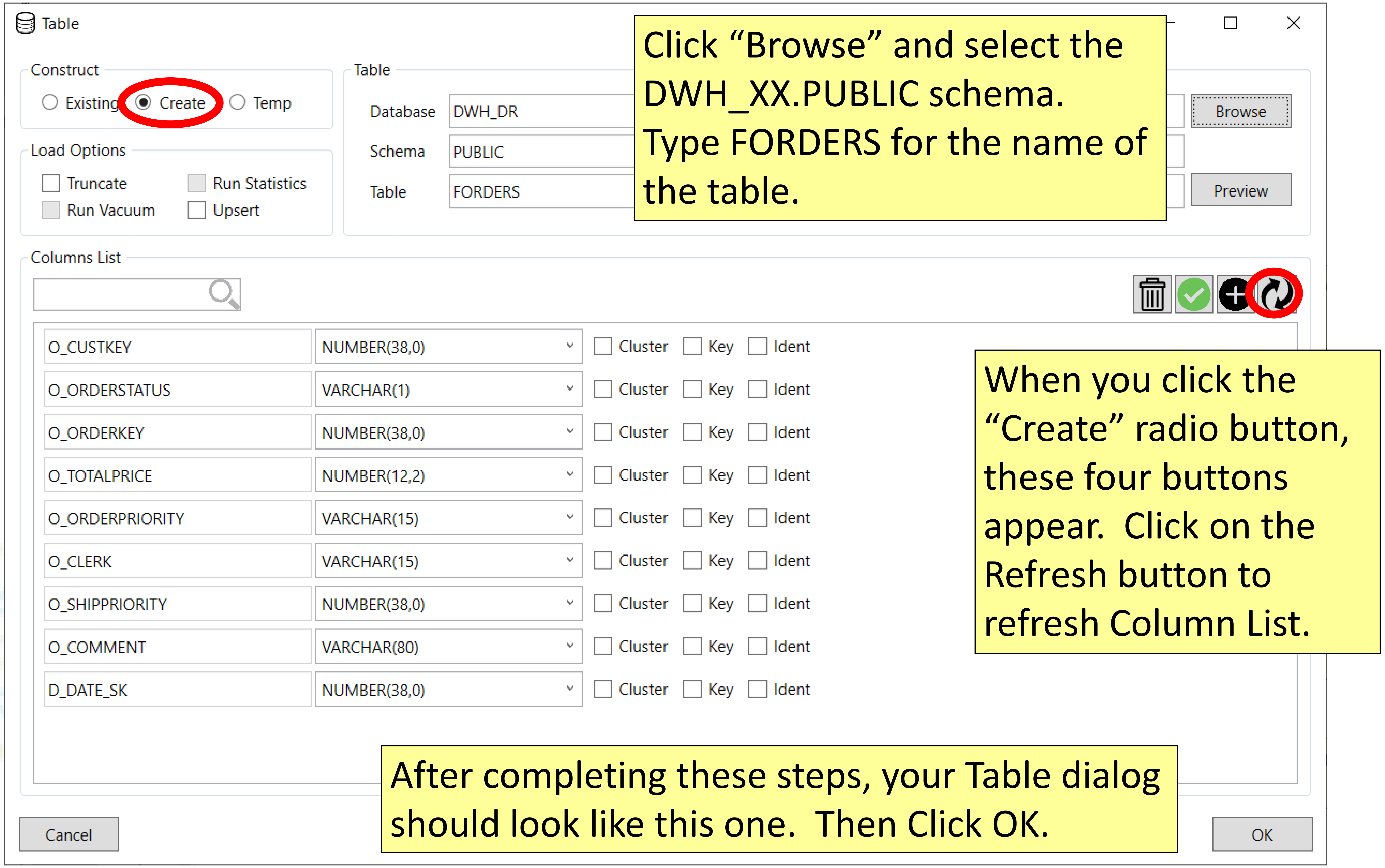
Name: $4.O_ORDERSTATUS
Type: VARCHAR(1) Alias: O_ORDERSTATUS



```
7/9/2020 6:26:37 PM     ERROR: ADD_DATE_DIMENSION_DR (JOB)     Cannot load (initializeOnstageConnections): Object reference not set to an instan
7/9/2020 6:29:28 PM     INFO: ADD_DATE_DIMENSION_DR (JOB)      Run requested
7/9/2020 6:39:59 PM     ERROR: TABLE: TABLE (6)     Property missing or not available. Is this step initialized?
7/9/2020 6:39:59 PM     ERROR: ADD_DATE_DIMENSION_DR (JOB)     Mapping Check FAILED..
```

Drag another Table step onto the Designer, placing it to the right of the Join step. Draw a link from the Join step to the new Table step.

As soon as you connect the link to the new Table step, the Table dialog will appear. Change the “Construct” radio button from “Existing” to “Create.”



Table

Construct: ○ Existing ● Create ○ Temp

Load Options: □ Truncate □ Run Statistics □ Run Vacuum □ Upsert

Table

Database: DWH_DR

Schema: PUBLIC

Table: FORDERS

Columns List

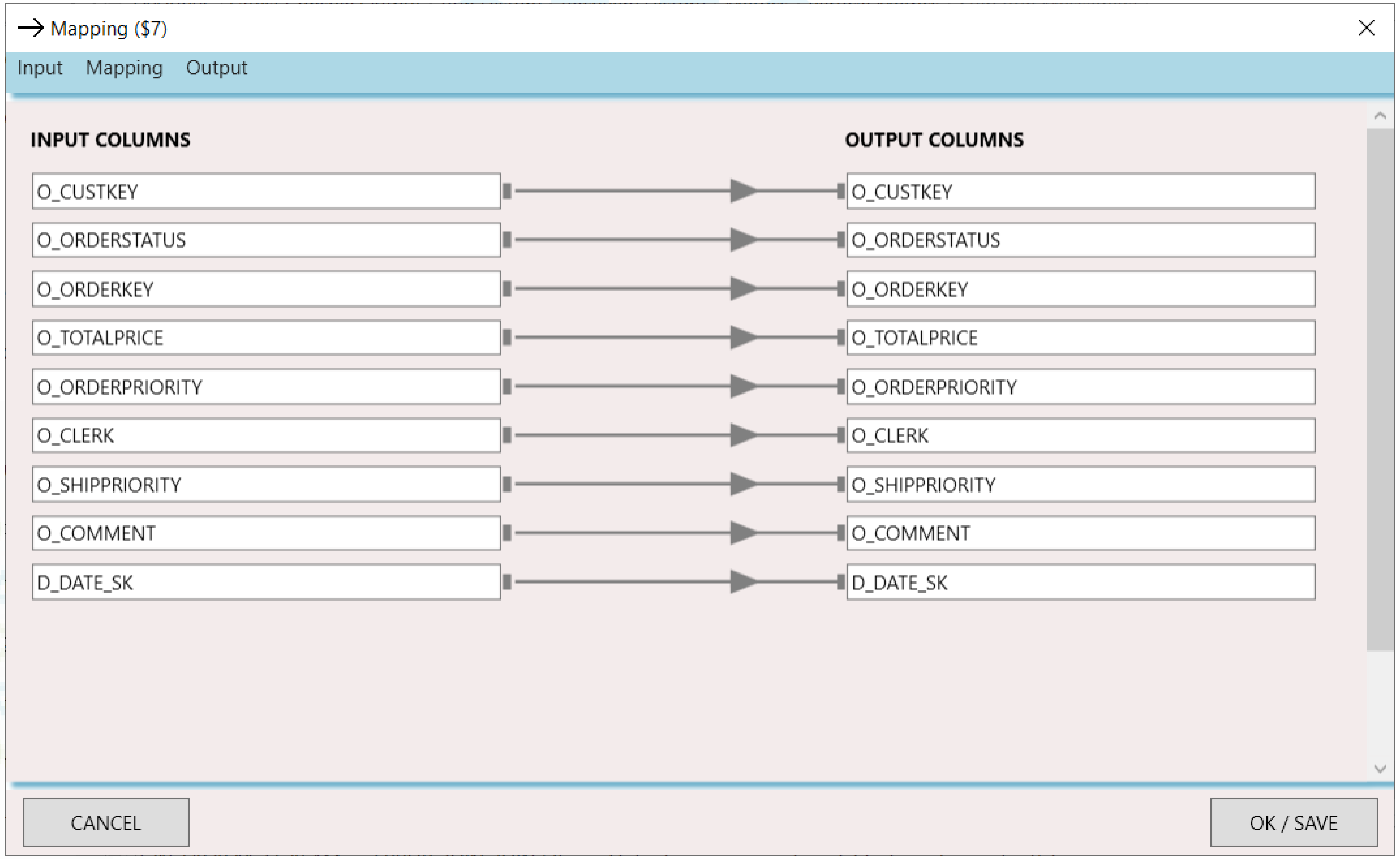
O_CUSTKEY	NUMBER(38,0)	□ Cluster	□ Key	□ Ident
O_ORDERSTATUS	VARCHAR(1)	□ Cluster	□ Key	□ Ident
O_ORDERKEY	NUMBER(38,0)	□ Cluster	□ Key	□ Ident
O_TOTALPRICE	NUMBER(12,2)	□ Cluster	□ Key	□ Ident
O_ORDERPRIORITY	VARCHAR(15)	□ Cluster	□ Key	□ Ident
O_CLERK	VARCHAR(15)	□ Cluster	□ Key	□ Ident
O_SHIPPRIORITY	NUMBER(38,0)	□ Cluster	□ Key	□ Ident
O_COMMENT	VARCHAR(80)	□ Cluster	□ Key	□ Ident
D_DATE_SK	NUMBER(38,0)	□ Cluster	□ Key	□ Ident

Click “Browse” and select the DWH_XX.PUBLIC schema. Type FORDERS for the name of the table.

When you click the “Create” radio button, these four buttons appear. Click on the Refresh button to refresh Column List.

After completing these steps, your Table dialog should look like this one. Then Click OK.

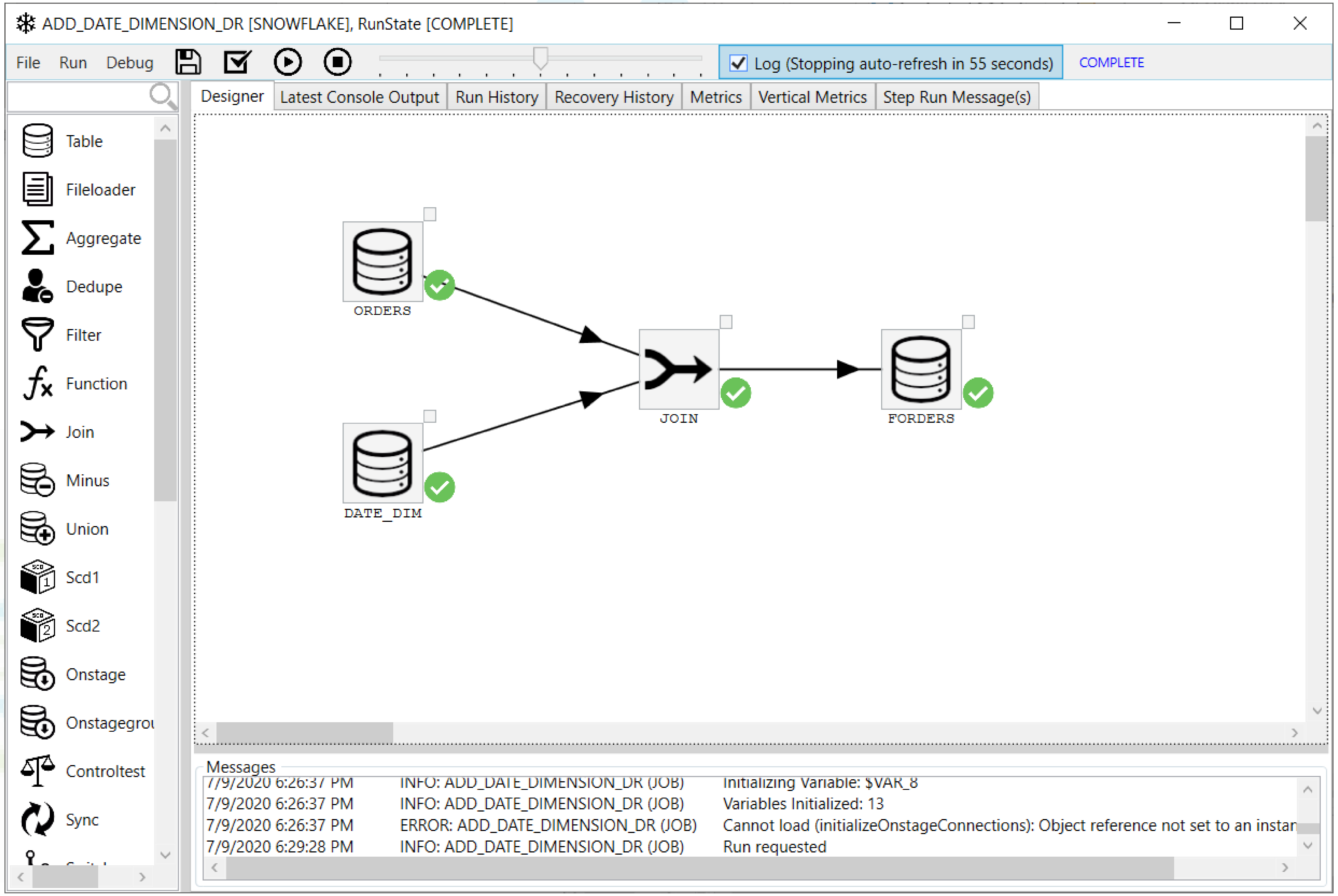
The Mapping dialog will appear. Click OK/SAVE.



INPUT COLUMNS	OUTPUT COLUMNS
O_CUSTKEY	O_CUSTKEY
O_ORDERSTATUS	O_ORDERSTATUS
O_ORDERKEY	O_ORDERKEY
O_TOTALPRICE	O_TOTALPRICE
O_ORDERPRIORITY	O_ORDERPRIORITY
O_CLERK	O_CLERK
O_SHIPPRIORITY	O_SHIPPRIORITY
O_COMMENT	O_COMMENT
D_DATE_SK	D_DATE_SK

Run the job: Click the Play button (▶), then click OK in the following two dialog boxes.

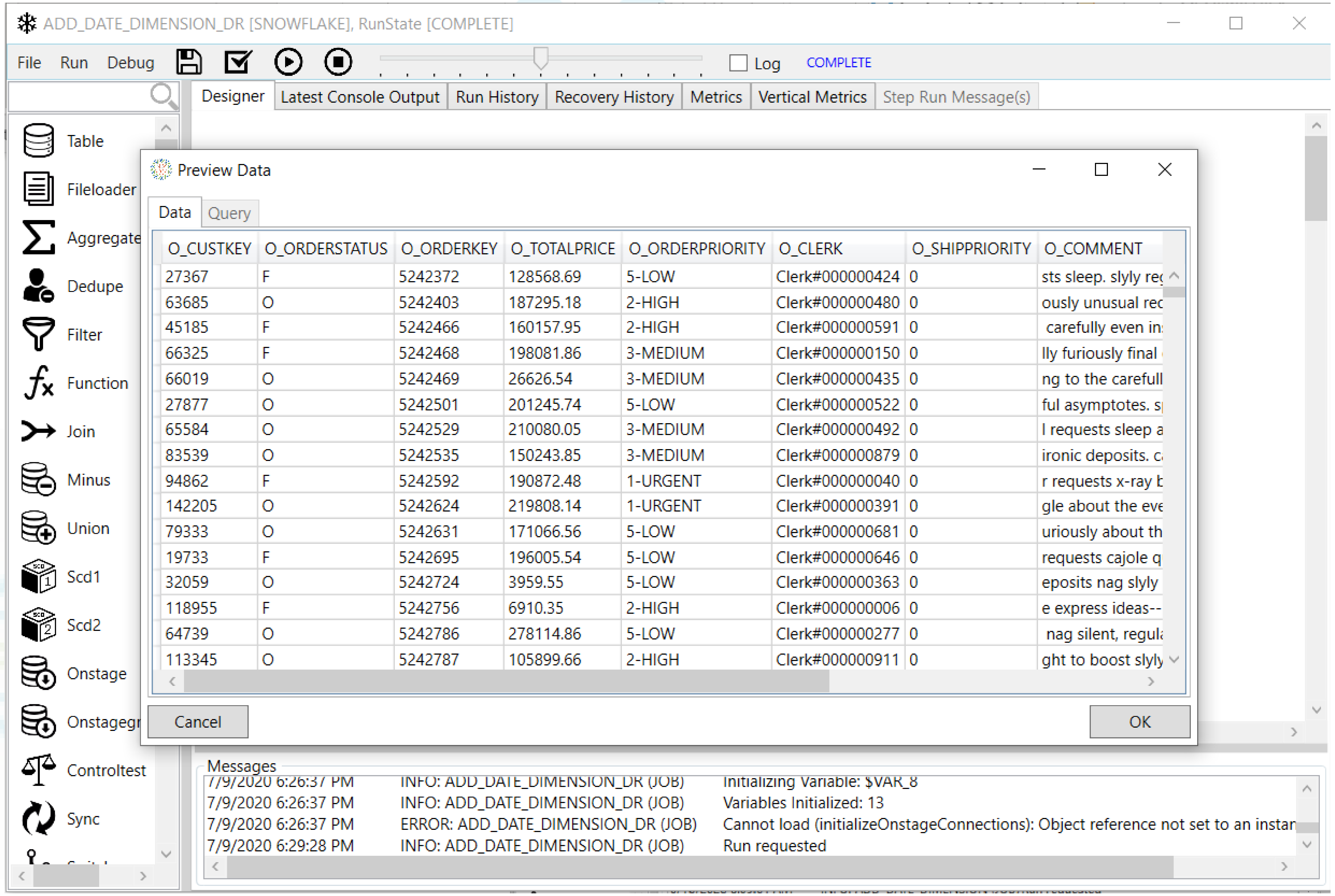
After the run completes, uncheck the Log check box.



☑ Log (Stopping auto-refresh in 55 seconds)

```
7/9/2020 6:26:37 PM    INFO: ADD_DATE_DIMENSION_DR (JOB)     Initializing Variable: $VAR_8
7/9/2020 6:26:37 PM    INFO: ADD_DATE_DIMENSION_DR (JOB)     Variables Initialized: 13
7/9/2020 6:26:37 PM    ERROR: ADD_DATE_DIMENSION_DR (JOB)    Cannot load (initializeOnstageConnections): Object reference not set to an instar
7/9/2020 6:29:28 PM    INFO: ADD_DATE_DIMENSION_DR (JOB)     Run requested
```

Right-click on the FORDERS table step and choose “Preview” from the context Menu to check your results.



O_CUSTKEY	O_ORDERSTATUS	O_ORDERKEY	O_TOTALPRICE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY	O_COMMENT
27367	F	5242372	128568.69	5-LOW	Clerk#000000424	0	sts sleep. slyly reg
63685	O	5242403	187295.18	2-HIGH	Clerk#000000480	0	ously unusual rec
45185	F	5242466	160157.95	2-HIGH	Clerk#000000591	0	carefully even in:
66325	F	5242468	198081.86	3-MEDIUM	Clerk#000000150	0	lly furiously final
66019	O	5242469	26626.54	3-MEDIUM	Clerk#000000435	0	ng to the carefull
27877	O	5242501	201245.74	5-LOW	Clerk#000000522	0	ful asymptotes. s
65584	O	5242529	210080.05	3-MEDIUM	Clerk#000000492	0	l requests sleep a
83539	O	5242535	150243.85	3-MEDIUM	Clerk#000000879	0	ironic deposits. c
94862	F	5242592	190872.48	1-URGENT	Clerk#000000040	0	r requests x-ray b
142205	O	5242624	219808.14	1-URGENT	Clerk#000000391	0	gle about the eve
79333	O	5242631	171066.56	5-LOW	Clerk#000000681	0	uriously about th
19733	F	5242695	196005.54	5-LOW	Clerk#000000646	0	requests cajole q
32059	O	5242724	3959.55	5-LOW	Clerk#000000363	0	eposits nag slyly
118955	F	5242756	6910.35	2-HIGH	Clerk#000000006	0	e express ideas--
64739	O	5242786	278114.86	5-LOW	Clerk#000000277	0	nag silent, regula
113345	O	5242787	105899.66	2-HIGH	Clerk#000000911	0	ght to boost slyly

```
7/9/2020 6:26:37 PM    INFO: ADD_DATE_DIMENSION_DR (JOB)     Initializing Variable: $VAR_8
7/9/2020 6:26:37 PM    INFO: ADD_DATE_DIMENSION_DR (JOB)     Variables Initialized: 13
7/9/2020 6:26:37 PM    ERROR: ADD_DATE_DIMENSION_DR (JOB)    Cannot load (initializeOnstageConnections): Object reference not set to an instan
7/9/2020 6:29:28 PM    INFO: ADD_DATE_DIMENSION_DR (JOB)     Run requested
```

Run the following queries:

To find all the orders that occurred on Wednesdays in 1998:

```
SELECT * from FORDERS F, DATE_DIM D where F.D_DATE_SK = D.D_DATE_SK and D.D_YEAR
= 1998 and D.D_DOW = 3;
```

Which year between 1995 and 2000 had the most successful 4th quarter (as measured by order count)?

```
SELECT count(*) C, D.D_YEAR Y FROM FORDERS F, DATE_DIM D where F.D_DATE_SK =
D.D_DATE_SK and D.D_YEAR BETWEEN 1995 AND 2000 AND D.D_QOY = 4 GROUP BY Y
ORDER BY C; -- Gives results for all quarters in order; use max to select best.
```

Which year between 1995 and 2000 had the most successful 4th quarter (as measured by sum of total price)?

```
SELECT sum(F.O_TOTALPRICE ) S, D.D_YEAR Y FROM FORDERS F, DATE_DIM D where
F.D_DATE_SK = D.D_DATE_SK and D.D_YEAR BETWEEN 1995 AND 2000 AND D.D_QOY = 4
GROUP BY Y ORDER BY S;
```

What we've learned

- Basic concepts: Data warehouse, data mart, fact, dimension, ETL, ELT.
- How organization into facts and dimensions enables analysts to make powerful queries against their business data.
- Hands-on experience with ETL/ELT – the process of transforming operational data into data suitable for a DW.
- A first look at a date dimension table – the most important dimension table in most DWs.
- Hands-on experience with ELTMaestro, and use of three critical steps: Function, Join, and Table.