

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

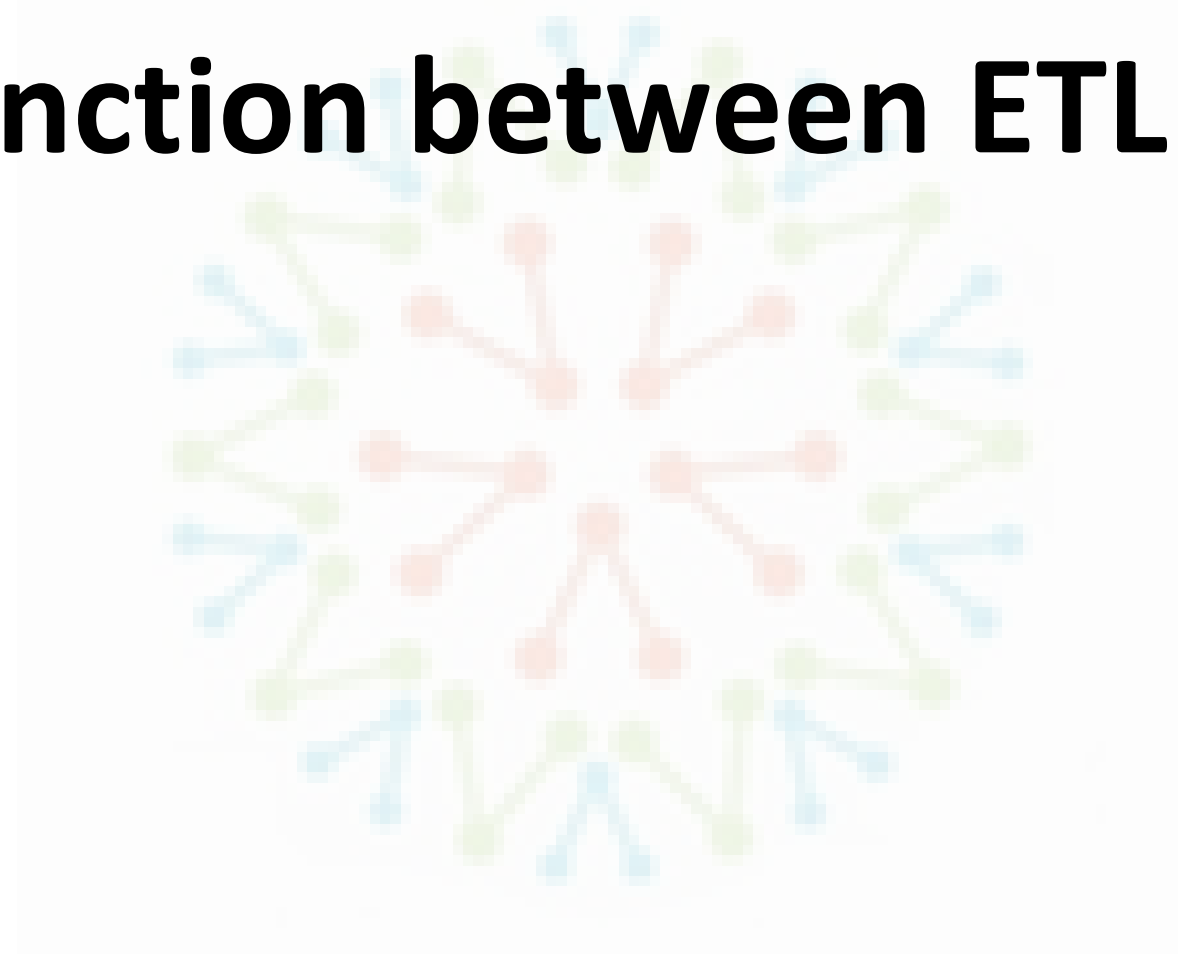
- DWs handle questions of the form “Tell me about all the <noun> having <attributes>.” The *nouns* in question are referred to as *facts*.
- *Sales* and *claims* are examples of facts.
- The *attributes* are referred to as *dimensions*. *Date* and *location* are typical examples of dimensions.
- Facts 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.**

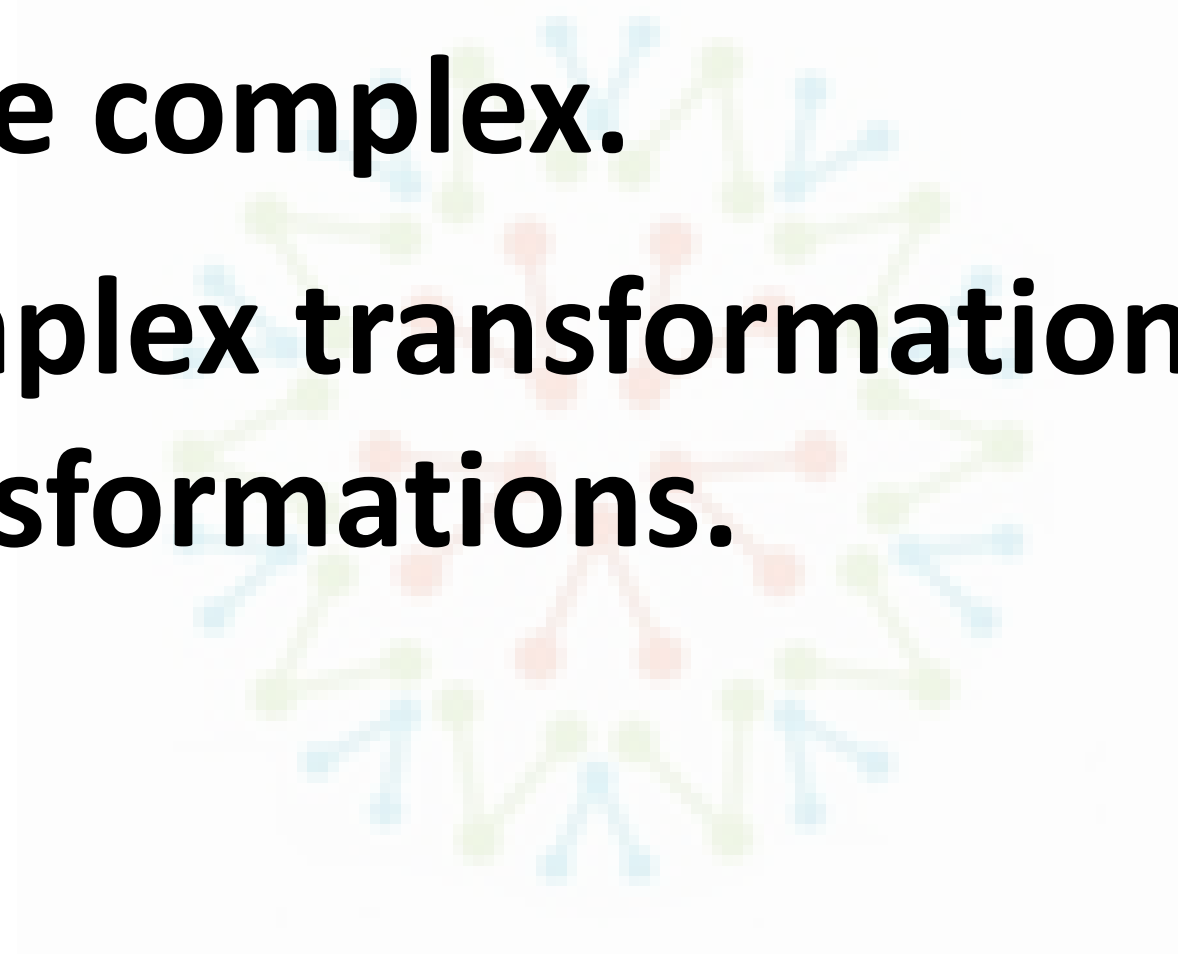
ETL and ELT, cont.

- **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**



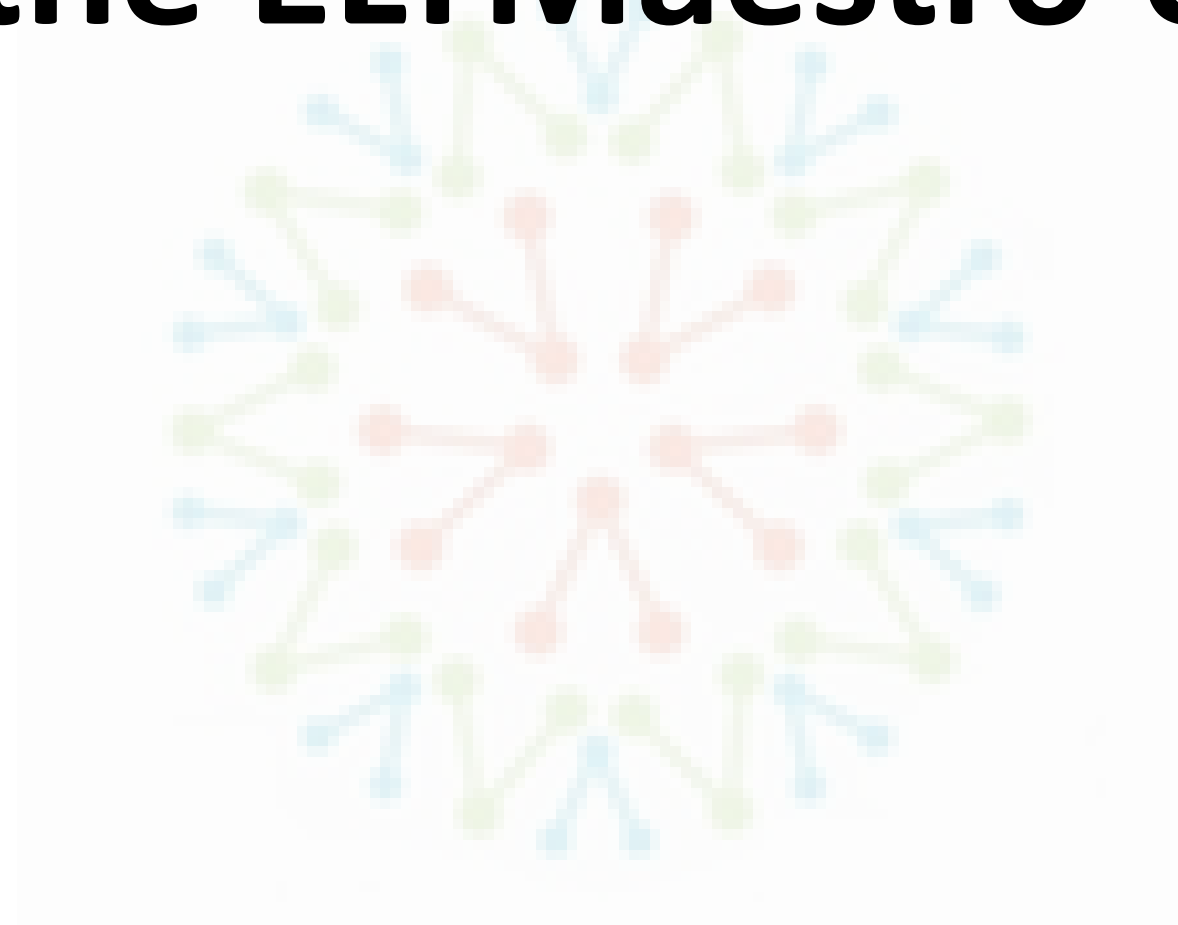
Assumptions

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

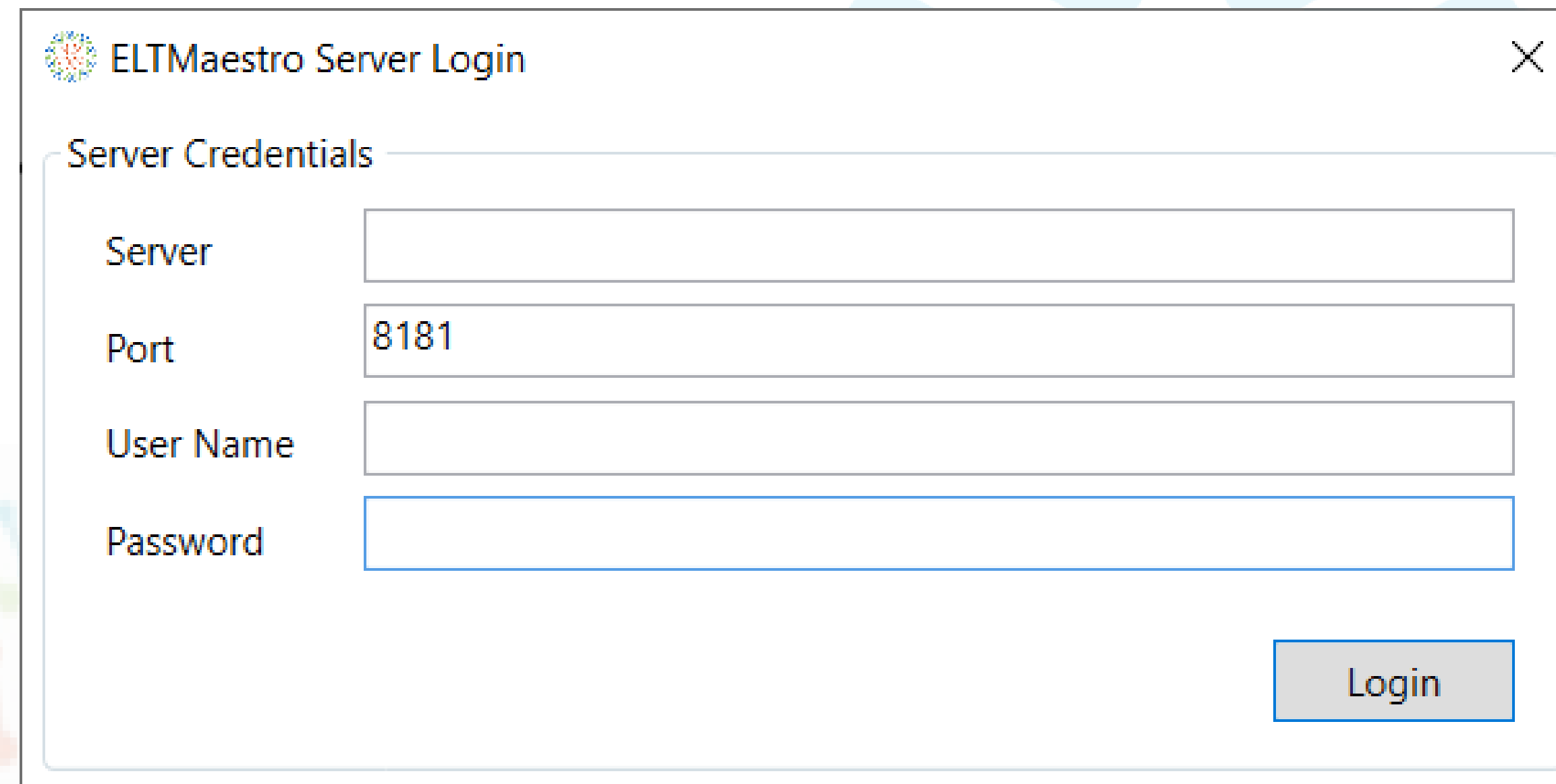


A quick overview of the ELTMaestro UI

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



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



The screenshot shows a window titled "ELTMaestro Server Login" with a close button (X) in the top right corner. Inside the window, there is a section labeled "Server Credentials" containing four input fields: "Server", "Port" (with the value "8181" entered), "User Name", and "Password". A "Login" button is located at the bottom right of the window.

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

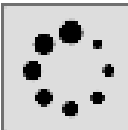

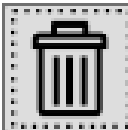


ELT Maestro -Workspace


Workflow(s)






Daily Warehouse Metrics

FileAdministrationRuntime & LogsHelp

Workflow(s)





	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

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

ELT Maestro Server Connection

Server

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

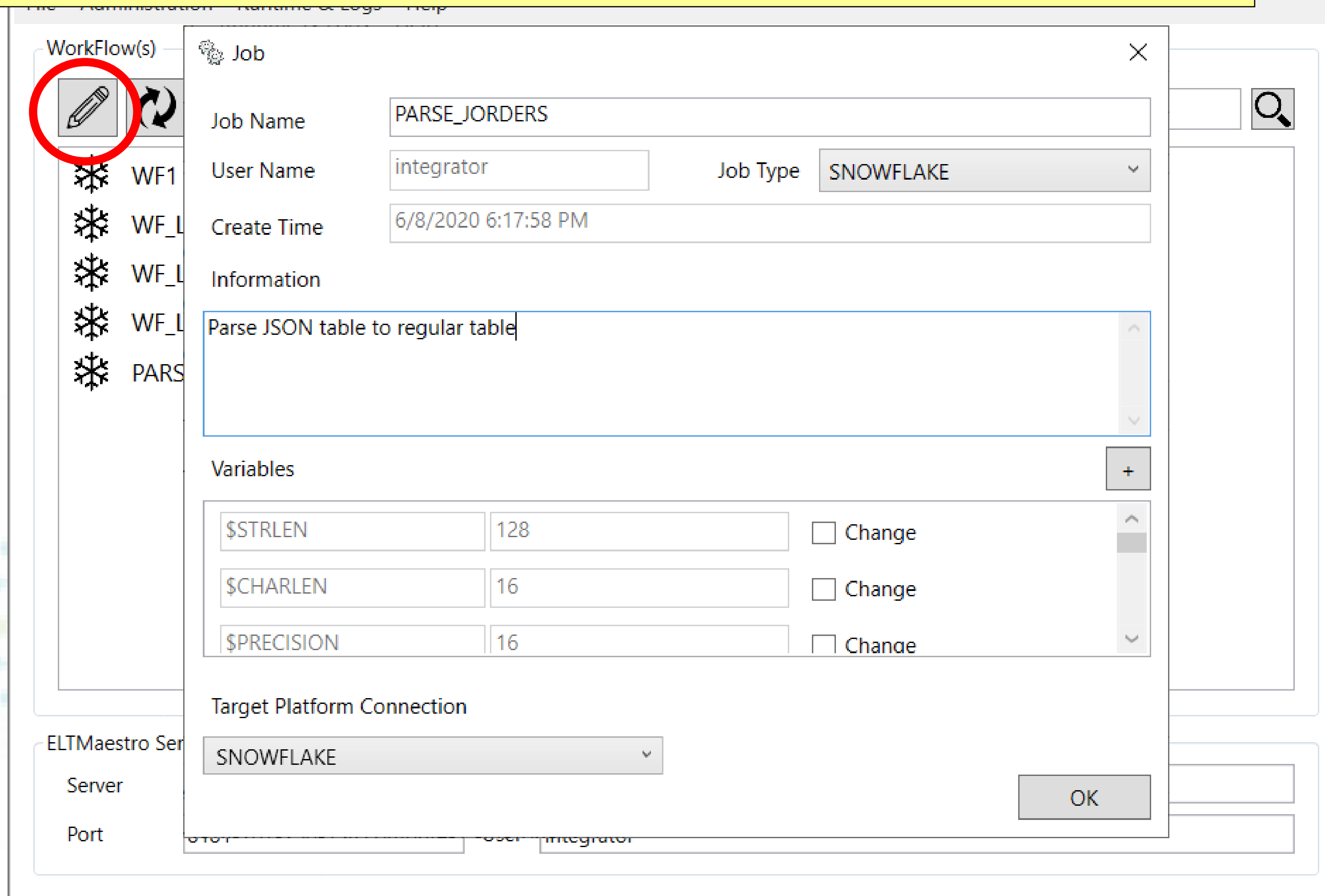
Port

8181

User

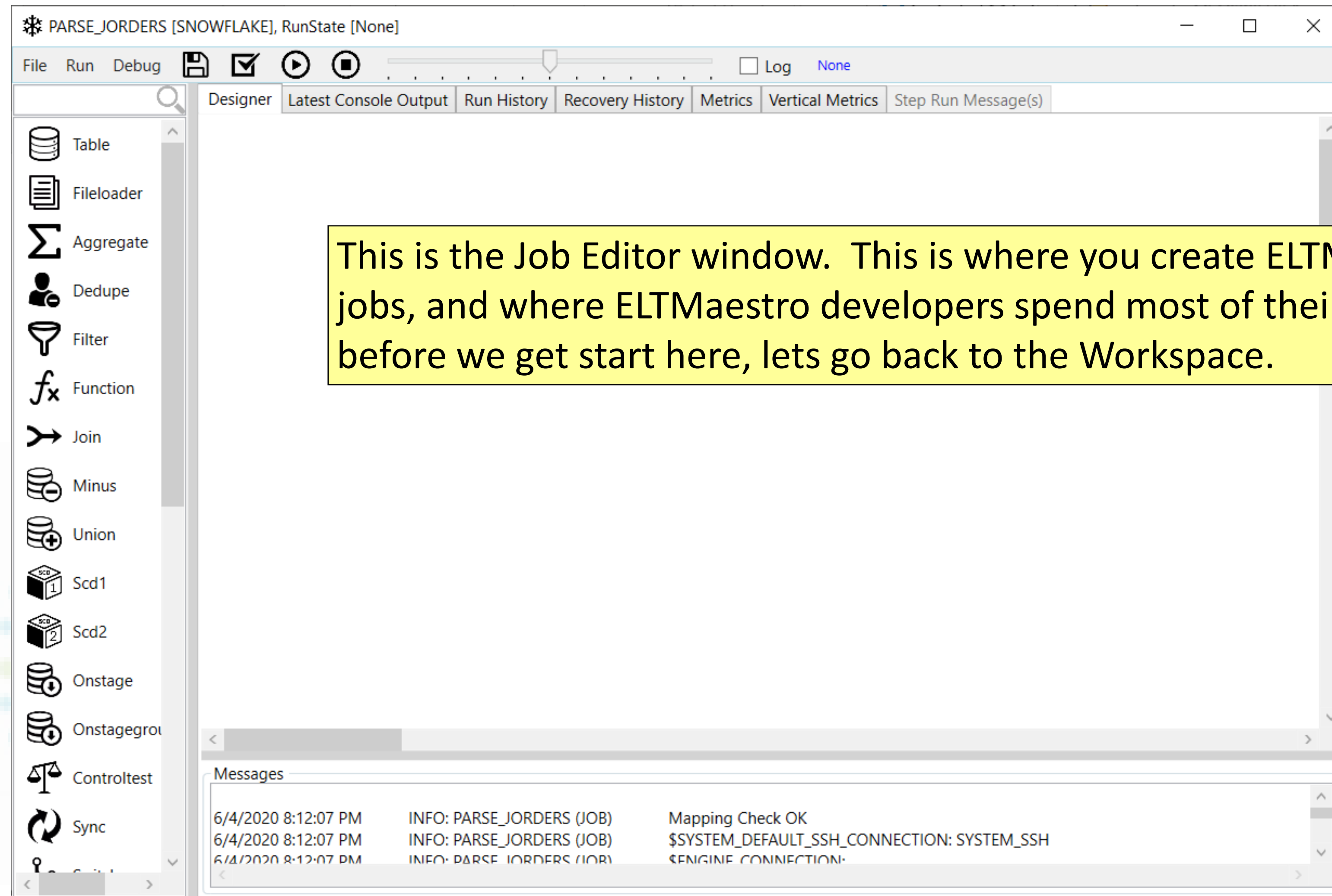
integrator

3. Click on Create New Workflow (). Name your new workflow PARSE_JORDERS, 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.



The screenshot shows the 'Job' configuration dialog in ELT Maestro. The 'Create New Workflow' button (pencil icon) is highlighted with a red circle. The dialog contains the following fields and sections:

- Job Name:** PARSE_JORDERS
- User Name:** integrator
- Job Type:** SNOWFLAKE
- Create Time:** 6/8/2020 6:17:58 PM
- Information:** Parse JSON table to regular table
- Variables:**
 - \$STRLEN: 128 (Change checkbox)
 - \$CHARLEN: 16 (Change checkbox)
 - \$PRECISION: 16 (Change checkbox)
- Target Platform Connection:** SNOWFLAKE
- OK** button



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

Refresh the list of jobs.

The screenshot shows the ELT Maestro -Workspace application window. The title bar reads "ELT Maestro -Workspace". Below the title bar is a menu bar with "File", "Administration", "Runtime & Logs", and "Help". The main content area is titled "Workflow(s)" and contains a list of workflows. Above the list is a toolbar with five icons: a pencil (edit), a circular arrow (refresh), a trash can (delete), a wrench (configure), and a gear (refresh). A red arrow points from the "Refresh the list of jobs." text box to the circular arrow icon. Another red arrow points from the "Delete a job." text box to the trash can icon. A third red arrow points from the "Edit an existing job." text box to the pencil icon. A fourth red arrow points from the "Note that the job we just created is now on the list." text box to the "PARSE_JORDERS" workflow entry. The workflow list includes:

Workflow(s)	Owner
WF1	INTEGRATOR
WF_	IN
WF_LOAD_ADVENTUREWORKS	INTEGRATOR
WF_LOAD_ADVENTUREWORKS_DW	INTEGRATOR
PARSE_JORDERS_SOLUTION	INTEGRATOR
PARSE_JORDERS	INTEGRATOR

The "PARSE_JORDERS" workflow is highlighted with a blue dashed border. At the bottom of the window is a section titled "ELT Maestro Server Connection" with fields for "Server" (ec2-18-207-250-134.compute-1.amazonaws.com), "Port" (8181), and "User" (integrator).

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": "0",  
  "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	0	230578.84	1996-02-15	5-LOW	Clerk#000000385	0	refully special platelets
5242402	98561	0	222665.47	1997-02-17	1-URGENT	Clerk#000000370	0	eodolites wake furiously
5242403	63685	0	187295.18	1996-07-01	2-HIGH	Clerk#000000480	0	ously unusual requests ar
5242404	91651	0	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": "0",  
  "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

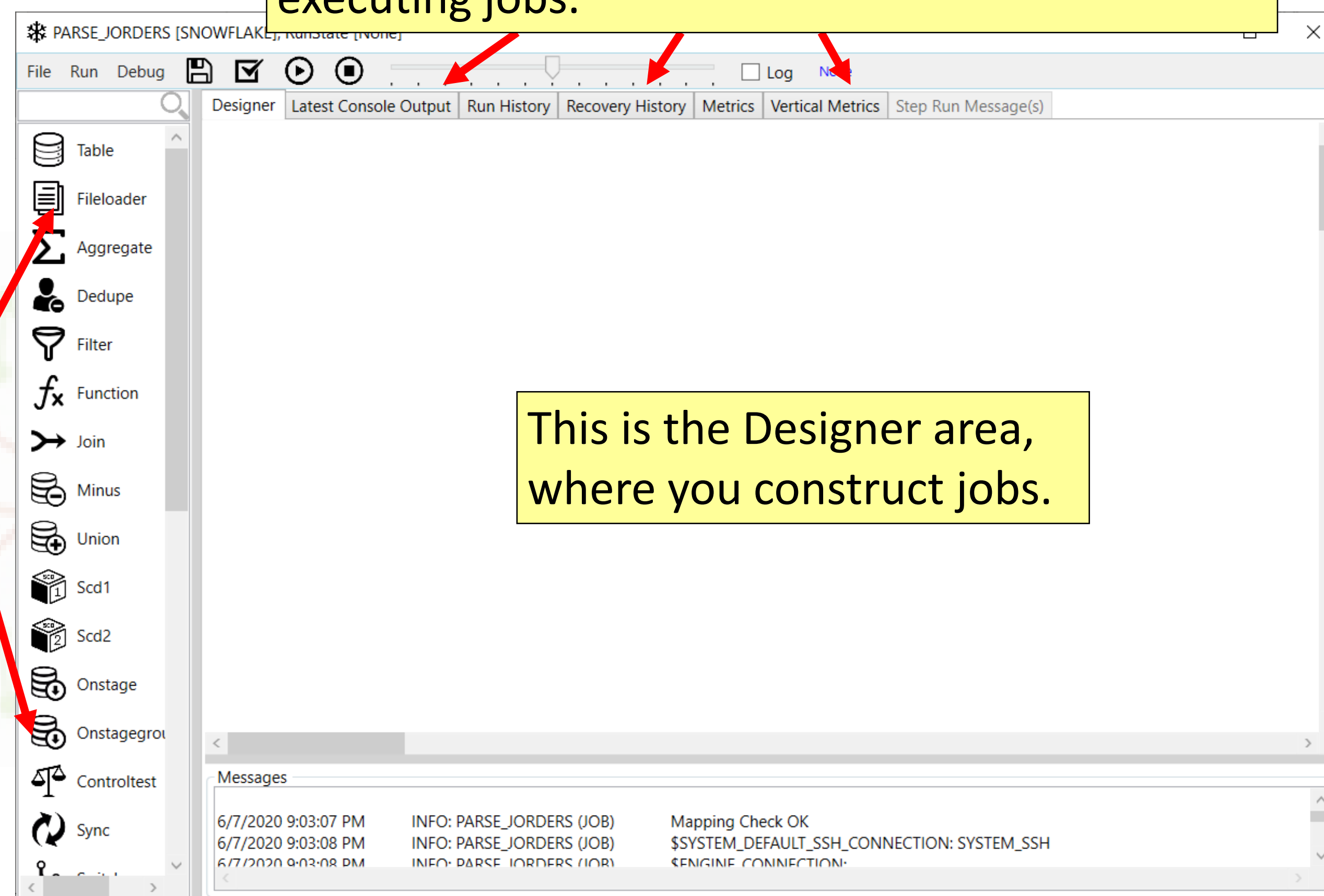
etc.

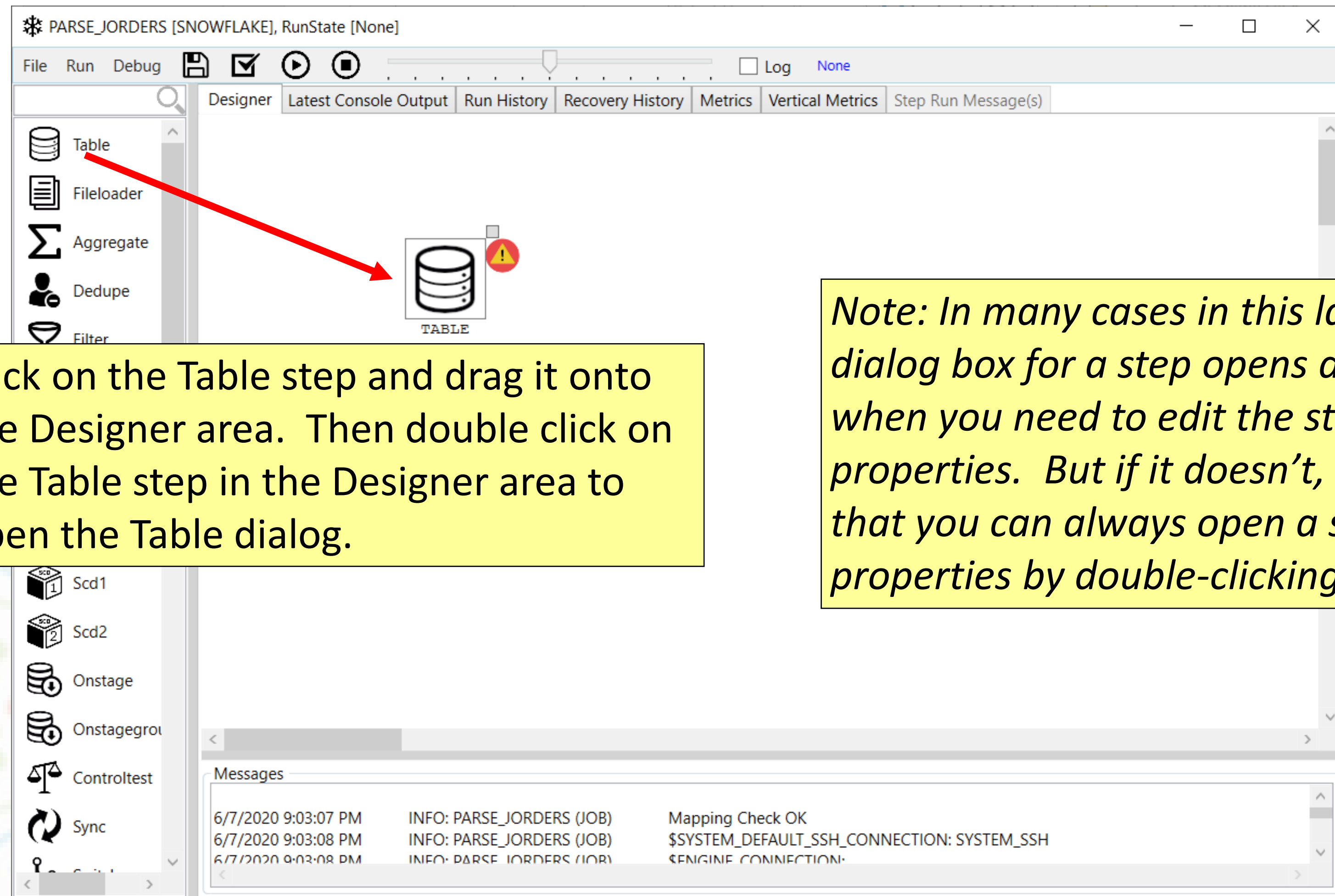
Go back to the Job Editor window for the PARSE_ORDERS job. If it's not still open, open it from the Workspace by double clicking on its name, or by selecting it and then clicking the  button.

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.





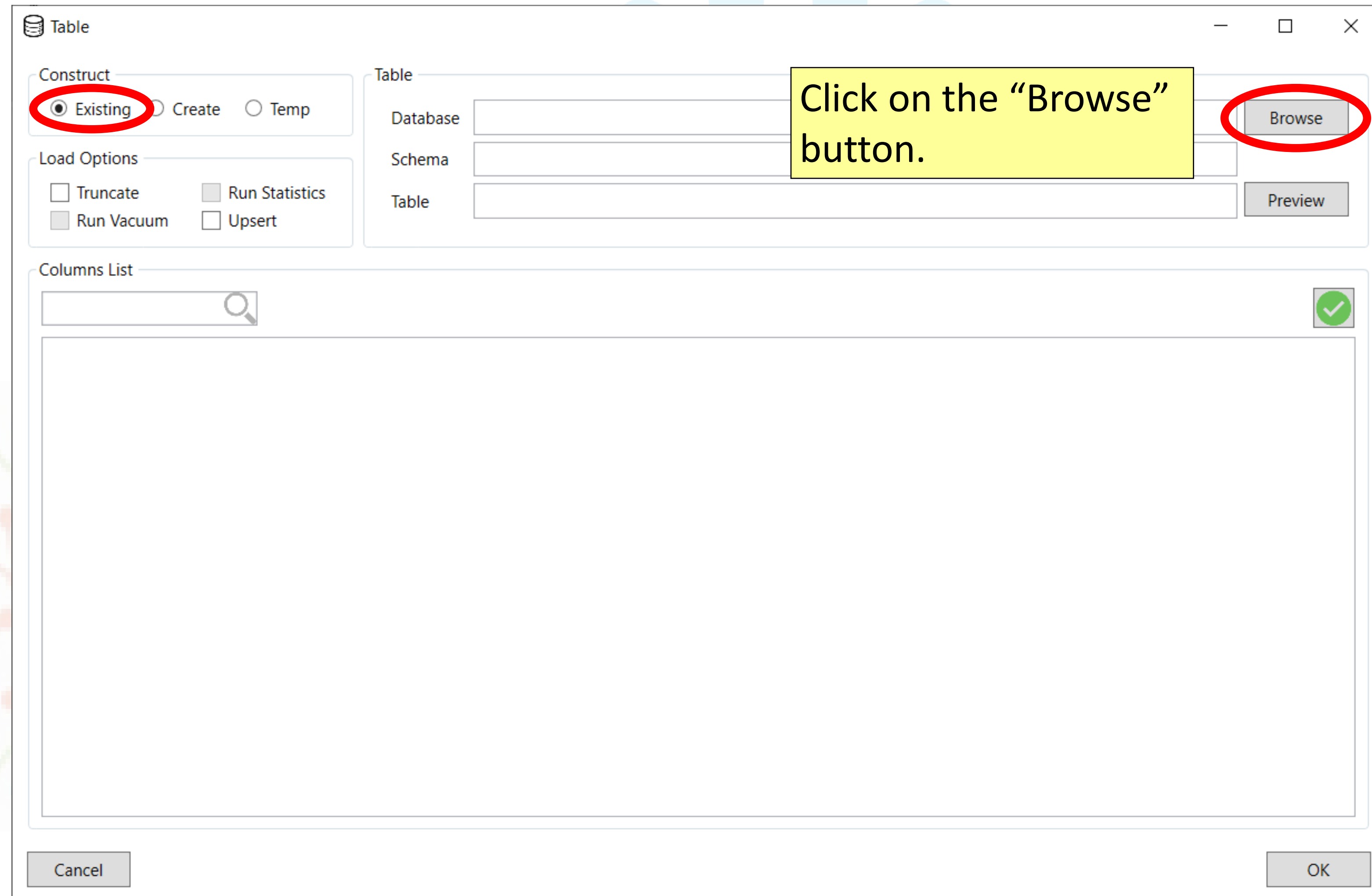
Click on the Table step and drag it onto the Designer area. Then double click on the Table step in the Designer area to open the Table dialog.

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.

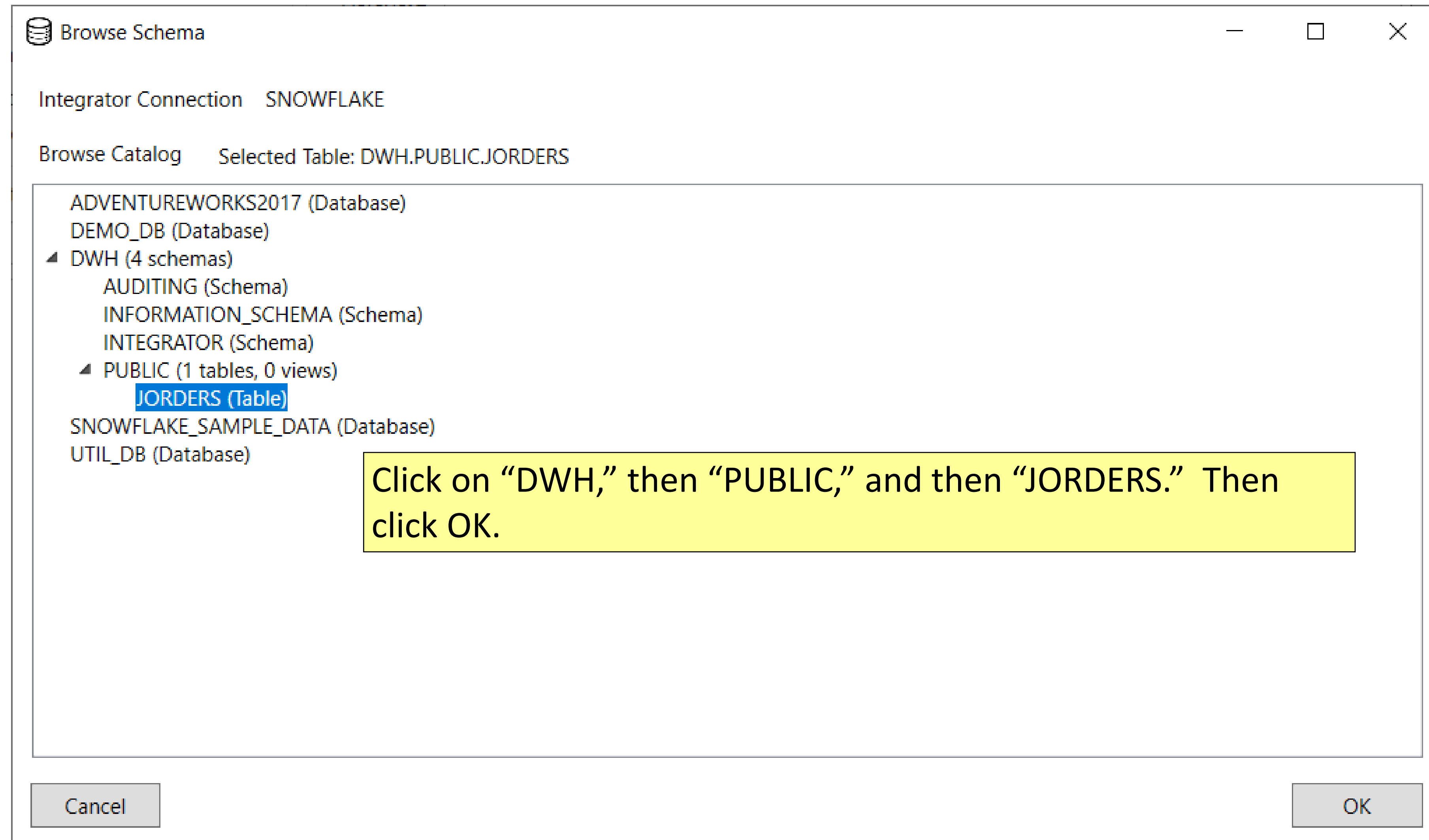
The “Existing” radio button should be selected, meaning we are reading from an existing table.

Click on the “Browse” button.



The screenshot shows a 'Table' dialog box with the following sections:

- Construct:** Three radio buttons: 'Existing' (selected and circled in red), 'Create', and 'Temp'.
- Load Options:** Four checkboxes: 'Truncate' (unchecked), 'Run Statistics' (unchecked), 'Run Vacuum' (checked), and 'Upsert' (unchecked).
- Table:** Three text input fields for 'Database', 'Schema', and 'Table'. The 'Browse' button next to the 'Table' field is circled in red.
- Columns List:** A search bar with a magnifying glass icon and a large empty list area. A green checkmark icon is in the top right corner of this section.
- Buttons:** 'Cancel' and 'OK' buttons at the bottom.



Table

Construct

☒ Existing

☐ Create

☐ Temp

Load Options

☐ Truncate

☐ Run Statistics

☐ Run Vacuum

☐ Upsert

Table

Database

DWH

Browse

Schema

PUBLIC

Table

JORDERS

Preview

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.

Cancel

OK

PARSE_JORDERS [SNOWFLAKE], RunState [None]

File Run Debug [Icons] Log None

Designer Latest Console Output Run History Recovery History Metrics Vertical Metrics Step Run Message(s)

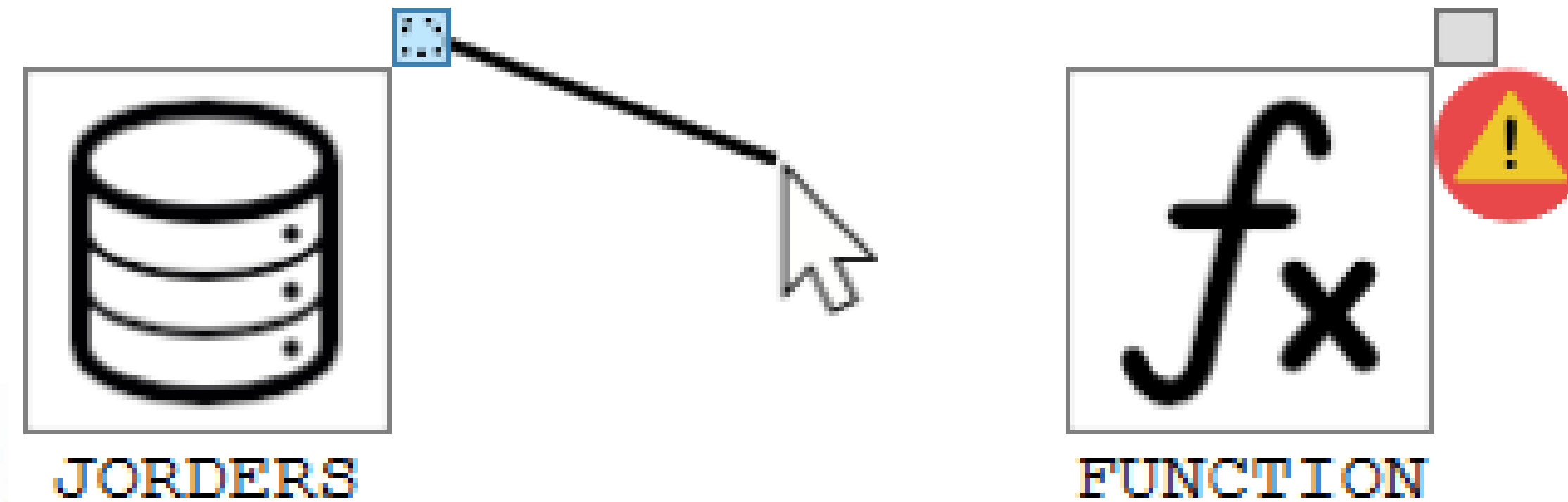
Note that the label for the Table step has now changed to "JORDERS."

Next, drag a Function step onto the Designer.

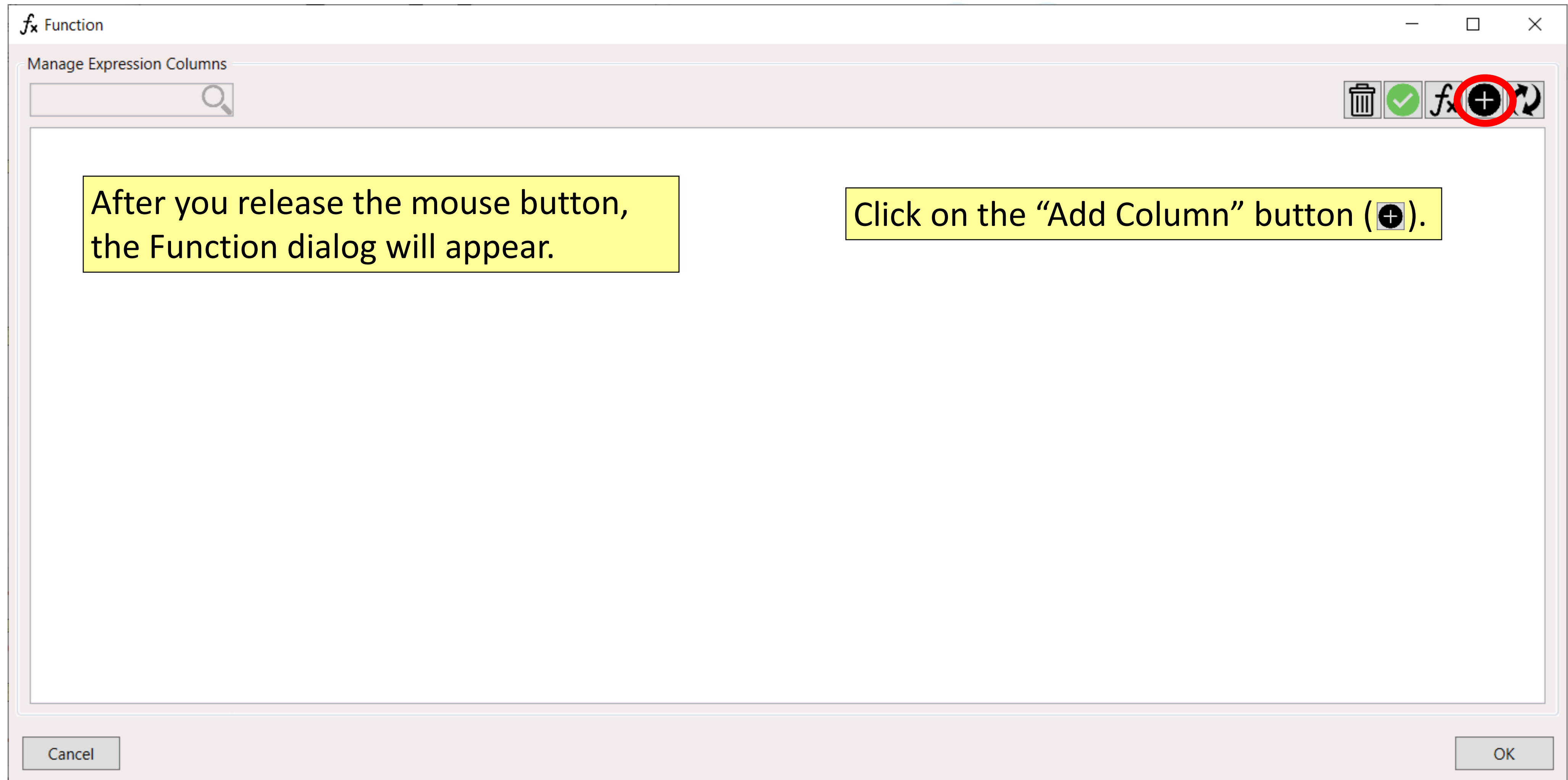
Messages

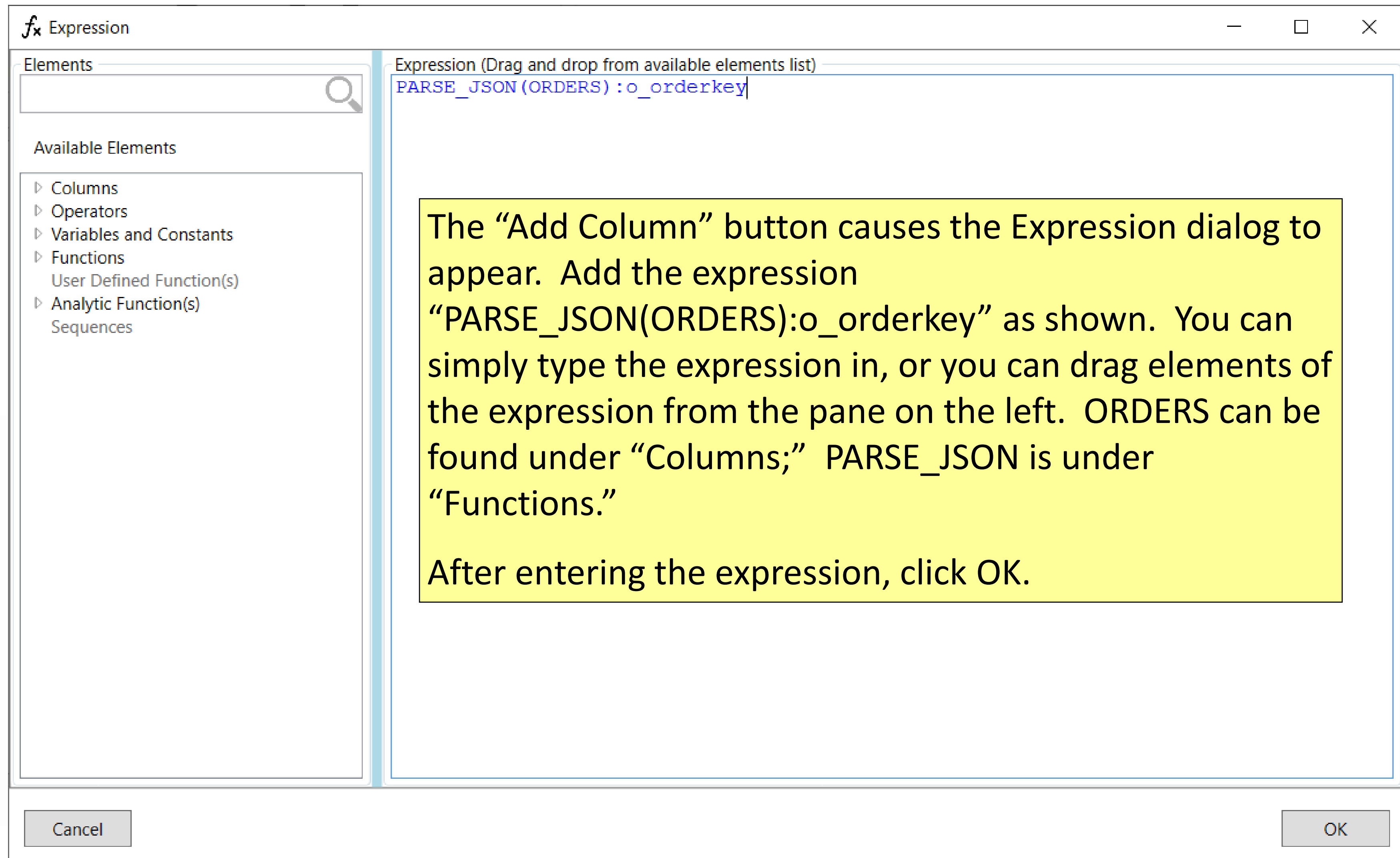
6/7/2020 9:03:07 PM	INFO: PARSE_JORDERS (JOB)	Mapping Check OK
6/7/2020 9:03:08 PM	INFO: PARSE_JORDERS (JOB)	\$SYSTEM_DEFAULT_SSH_CONNECTION: SYSTEM_SSH
6/7/2020 9:03:08 PM	INFO: PARSE_JORDERS (JOB)	\$ENGINE_CONNECTION:

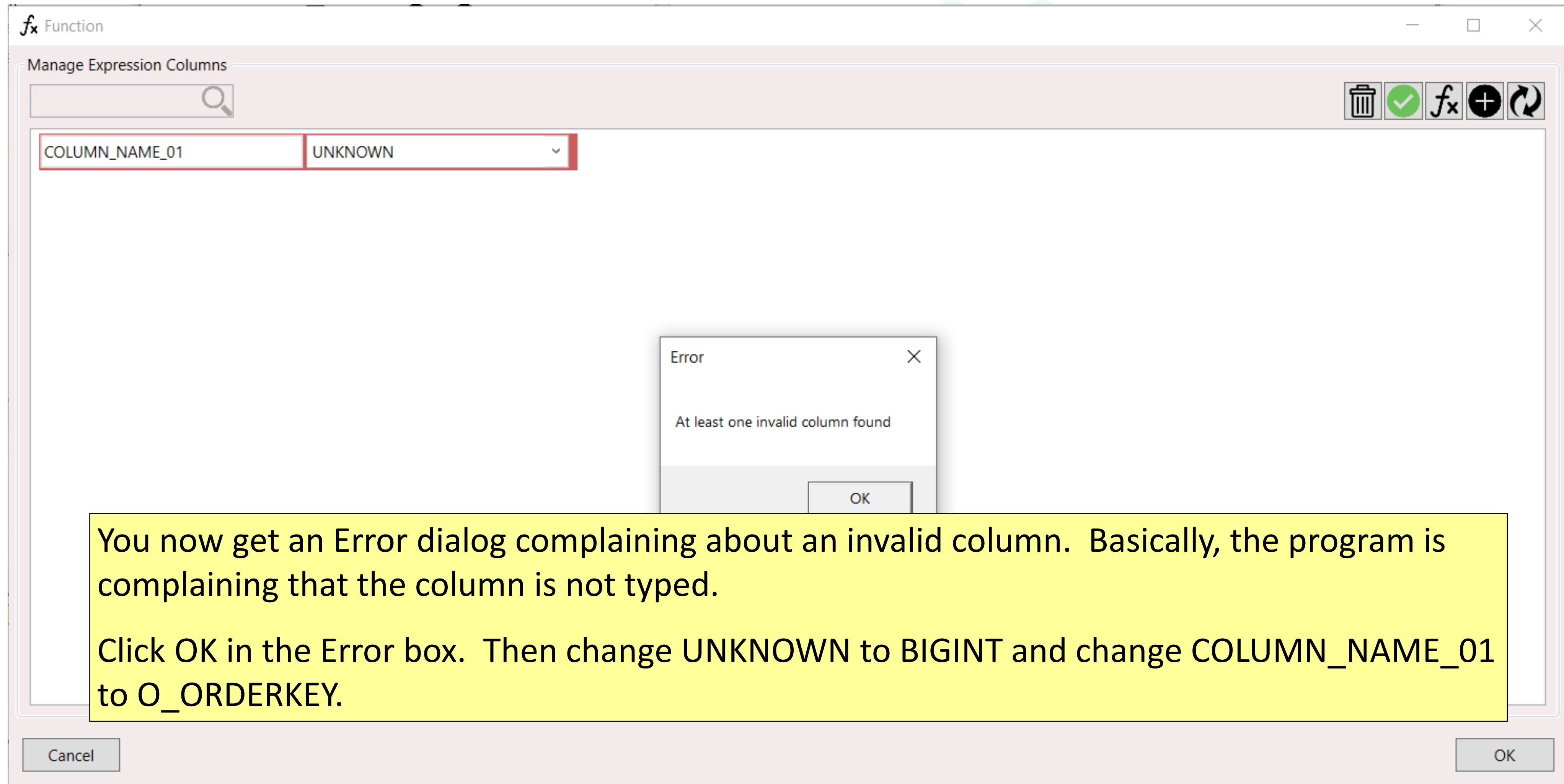
Click in the small box in the upper right corner of the Table step, and, holding down the mouse button, drag the mouse to the Function step.

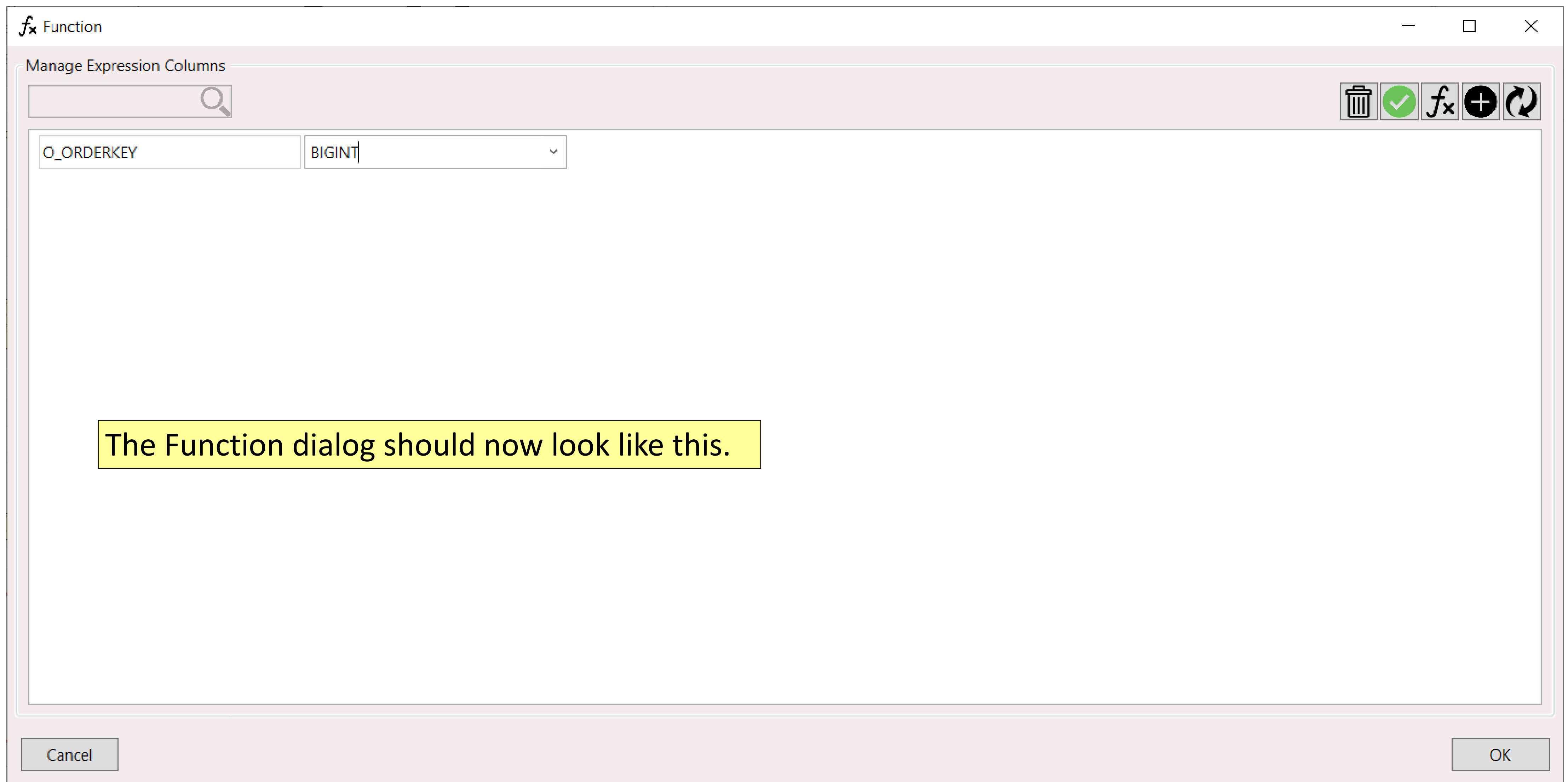


When the mouse is inside the function step (so that the line extending from the JORDERS Table step touches the Function step), release the mouse button.









PARSE_JORDERS [SNOWFLAKE], RunState [None]

File Run Debug Log None

Designer Latest Console Output Run History Recovery History Metrics Vertical Metrics Step Run Message(s)

Table
Fileloader
Aggregate
Dedupe
Filter
Function
Join
Minus
Union
Scd1
Scd2
Onstage
Onstagegro
Controltest
Sync

JORDERS → FUNCTION → TABLE

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.

Messages

6/8/2020 4:19:14 PM	ERROR: FUNCTION: FUNCTION (2)	Output is not terminated with input.(OAB)
6/8/2020 4:19:14 PM	ERROR: FUNCTION: FUNCTION (2)	Output missing or not available. Expecting output(s)
6/9/2020 12:15:12 AM	ERROR: TABLE: TABLE (4)	Property missing or not available. Is this step initialized?
6/9/2020 12:15:12 AM	ERROR: PARSE_JORDERS (JOB)	Mapping Check FAILED..

This time, click the “Create” radio button.

Check the “Truncate” box.

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

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.

Table

☐ Existing ☒ Create ☐ Temp

Load Options

☒ Truncate ☐ Run Statistics

Database Browse

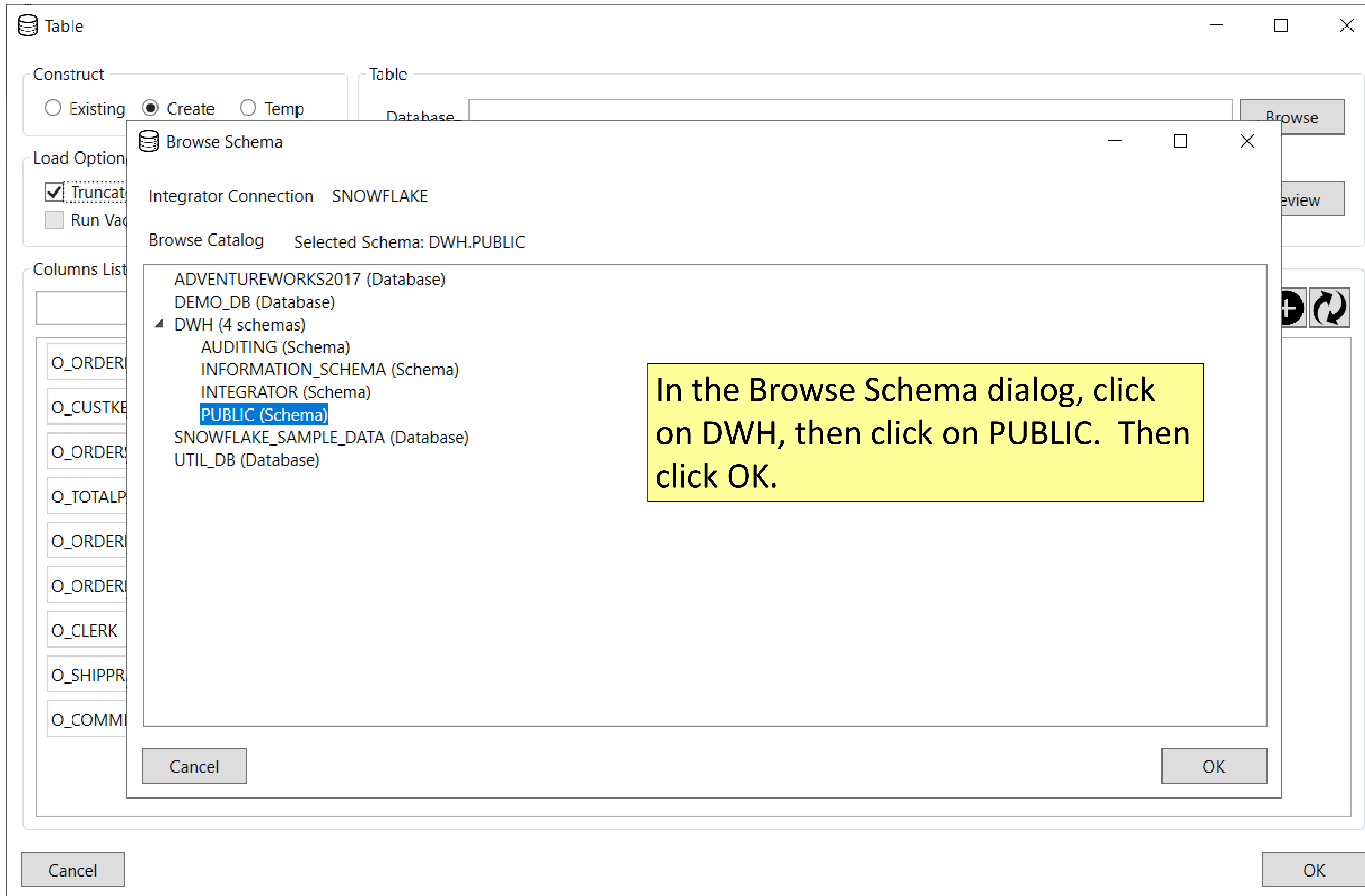
Schema

Table

Columns List

O_ORDERKEY	BIGINT	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_CUSTKEY	BIGINT	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_ORDERSTATUS	VARCHAR(1)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_TOTALPRICE	NUMBER(12,2)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_ORDERDATE	DATE	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_ORDERPRIORITY	VARCHAR(15)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_CLERK	VARCHAR(15)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_SHIPPRIORITY	BIGINT	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_COMMENT	VARCHAR(80)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident

Cancel OK



Table

Construct

☐ Existing

☒ Create

☐ Temp

Load Options

☒ Truncate

☐ Run Statistics

☐ Run Vacuum

☐ Upsert

Columns List

O_ORDERKEY	BIGINT	<input type="checkbox"/>
O_CUSTKEY	BIGINT	<input type="checkbox"/>
O_ORDERSTATUS	VARCHAR(1)	<input type="checkbox"/>
O_TOTALPRICE	NUMBER(12,2)	<input type="checkbox"/>
O_ORDERDATE	DATE	<input type="checkbox"/>
O_ORDERPRIORITY	VARCHAR(15)	<input type="checkbox"/> Cluster <input type="checkbox"/> Key <input type="checkbox"/> Ident
O_CLERK	VARCHAR(15)	<input type="checkbox"/> Cluster <input type="checkbox"/> Key <input type="checkbox"/> Ident
O_SHIPPRIORITY	BIGINT	<input type="checkbox"/> Cluster <input type="checkbox"/> Key <input type="checkbox"/> Ident
O_COMMENT	VARCHAR(80)	<input type="checkbox"/> Cluster <input type="checkbox"/> Key <input type="checkbox"/> Ident

Table

Database

DWH

Schema

PUBLIC

Table

ORDERS

Browse

Preview

Cancel

OK

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 database, in the PUBLIC schema, and will be named “ORDERS.”

Then click OK.

→ Mapping (\$5) X




Input Mapping Output

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.

CANCELOK / SAVE


PARSE_JORDERS [SNOWFLAKE], RunState [None]

File Run Debug    Log None

Designer Latest Console Output Run History Recovery History Metrics Vertical Metrics Step Run Message(s)

Table
Fileloader
Aggregate
Dedupe
Filter
Function
Join
Minus
Union
Scd1
Scd2
Onstage
Onstagegroup
Controltest
Sync

JORDERS → FUNCTION → ORDERS

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 right.

Messages

6/8/2020 4:19:14 PM	ERROR: FUNCTION: FUNCTION (2) Output is not terminated with input.(OAB)
6/8/2020 4:19:14 PM	ERROR: FUNCTION: FUNCTION (2) Output missing or not available. Expecting output(s)
6/9/2020 12:15:12 AM	ERROR: TABLE: TABLE (4) Property missing or not available. Is this step initialized?
6/9/2020 12:15:12 AM	ERROR: PARSE_JORDERS (JOB) Mapping Check FAILED..

PARSE_JORDERS [SNOWFLAKE], RunState [None]

File Run Debug Log None

Designer Latest Console Output Run History Recovery History Metrics Vertical Metrics Step Run Message(s)

Table
Fileloader
Aggregate
Dedupe
Filter
Function
Join
Minus
Union
Scd1
Scd2
Onstage
Onstagegroup
Controltest
Sync

JORDERS → FUNCTION → ORDERS

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,” 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.

Messages

6/8/2020 4:19:14 PM	ERROR: FUNCTION: FUNCTION (2)	Output missing or not available. Expecting output(s)
6/9/2020 12:15:12 AM	ERROR: TABLE: TABLE (4)	Property missing or not available. Is this step initialized?
6/9/2020 12:15:12 AM	ERROR: PARSE_JORDERS (JOB)	Mapping Check FAILED..
6/9/2020 10:44:55 AM	INFO: PARSE_JORDERS (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.

PARSE_JORDERS [SNOWFLAKE], RunState [COMPLETE]

File Run Debug ☒ Log (Stopped) COMPLETE

Designer Latest Console Output Run History Recovery History Metrics Vertical Metrics Step Run Message(s)

Table
Fileloader
Aggregate
Dedupe
Filter
Function
Join
Minus
Union
Scd1
Scd2
Onstage
Onstagegro
Controltest
Sync

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.)

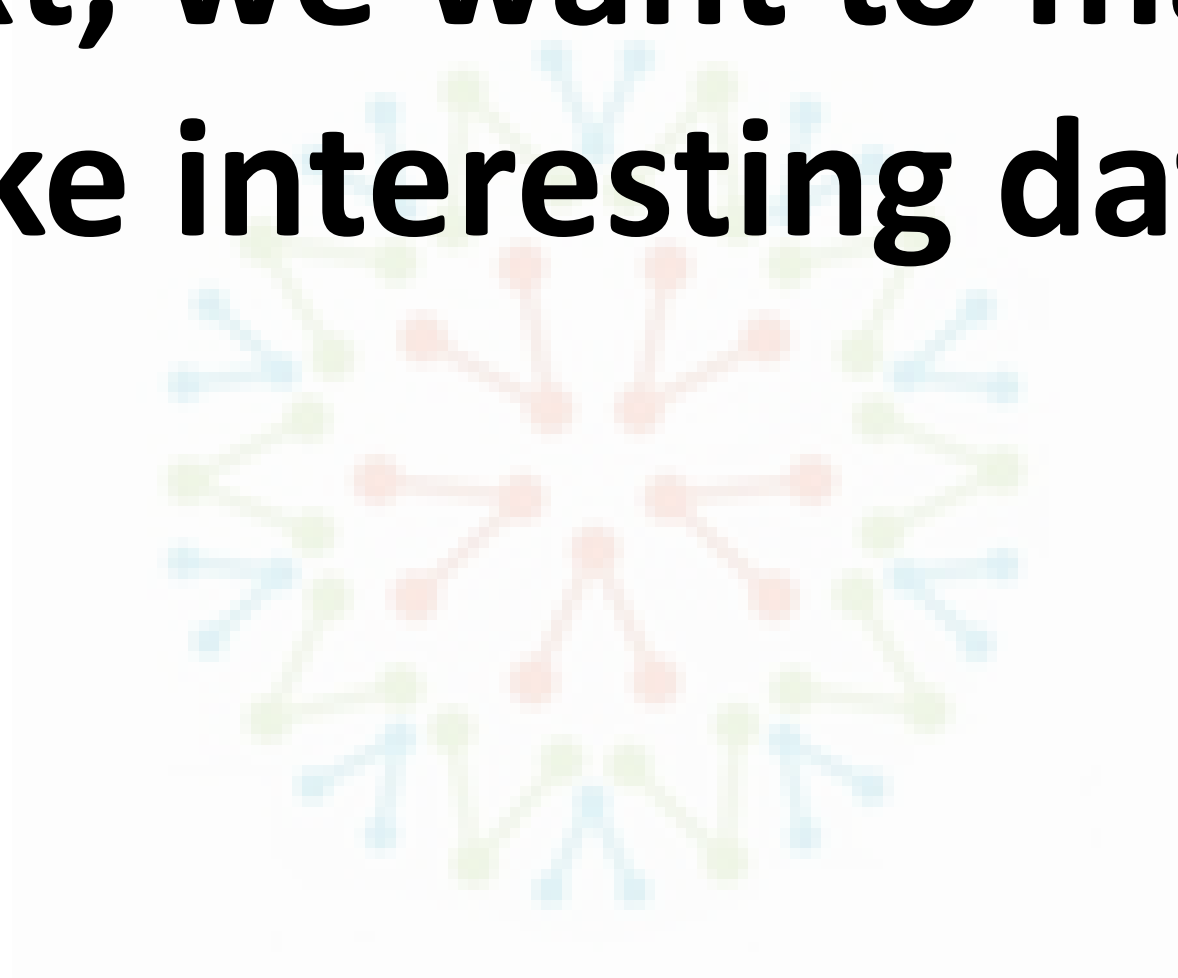
Turn off logging by unchecking the Log check box.

Preview Data							
Data							
<u>Q_ORDERKEY</u>	<u>Q_CUSTKEY</u>	<u>Q_ORDERSTATUS</u>	<u>Q_TOTALPRICE</u>	<u>Q_ORDERDATE</u>	<u>Q_ORDERPRIORITY</u>	<u>Q_CLERK</u>	<u>Q_SHIPPRIORITY</u>
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

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.

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

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. When the UI asks if you want to edit it, click Yes.



ELT Maestro -Workspace

Workflow(s) Daily Warehouse Metrics

File Administration Job

WorkFlow(s)

Job Name ADD_DATE_DIMENSION

User Name integrator Job Type SNOWFLAKE

Create Time 6/14/2020 1:46:17 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	<input type="checkbox"/> Change
\$CHARLEN	16	<input type="checkbox"/> Change
\$PRECISION	16	<input type="checkbox"/> Change

Target Platform Connection

SNOWFLAKE

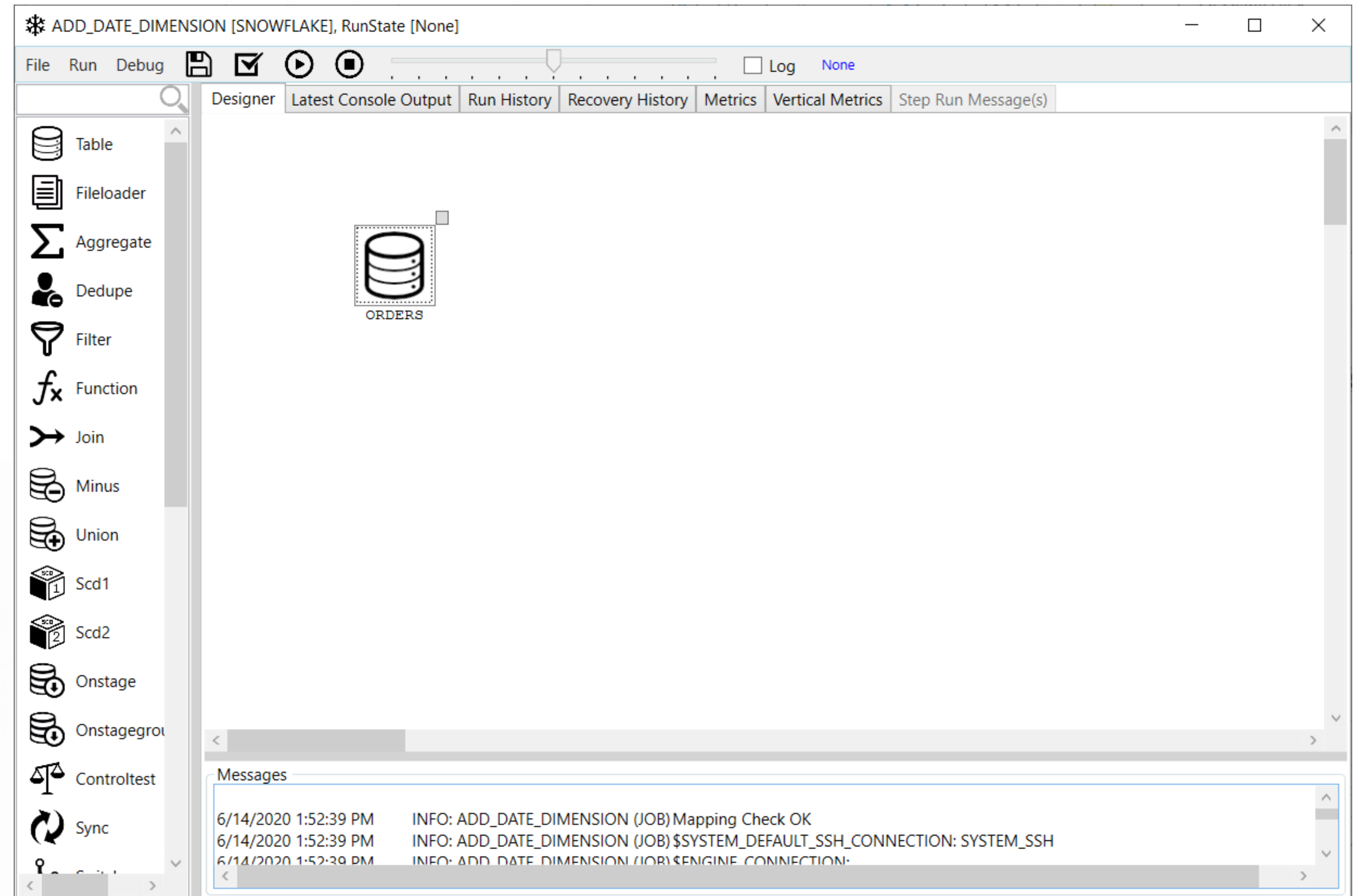
OK

ELT Maestro S

Server

Port 8181 User integrator

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.PUBLIC.ORDERS. Click OK to exit the Browse dialog, and click OK again to exit the Table dialog.



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



ADD_DATE_DIMENSION [SNOWFLAKE], RunState [None]

File Run Debug [Save] [Checkmark] [Play] [Stop] [Log] None

Designer Latest Console Output Run History Recovery History Metrics Vertical Metrics Step Run Message(s)

Table
Fileloader
Aggregate
Dedupe
Filter
Function
Join
Minus
Union
Scd1
Scd2
Onstage
Onstagegroup
Controltest
Sync

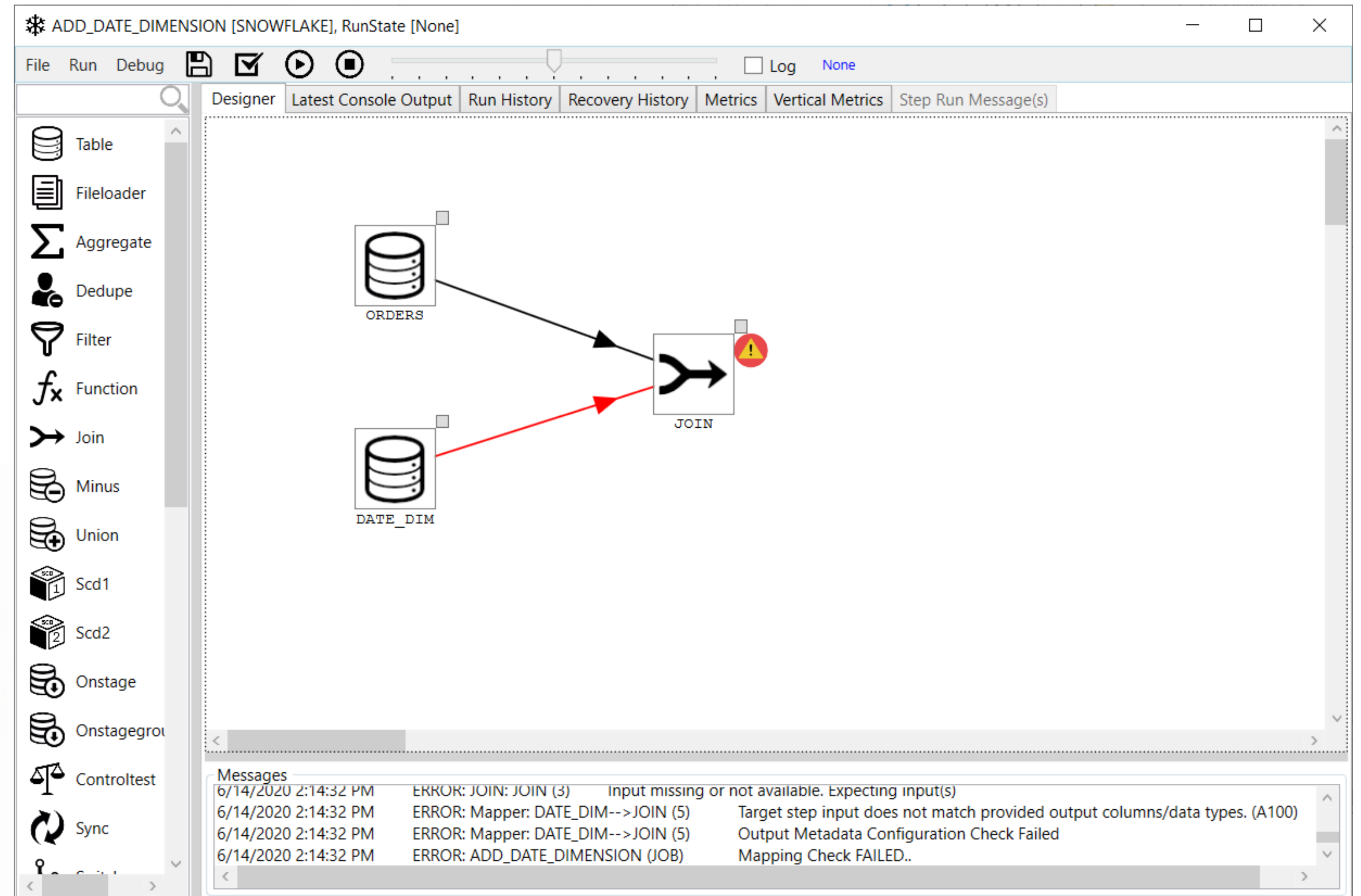
ORDERS

DATE_DIM

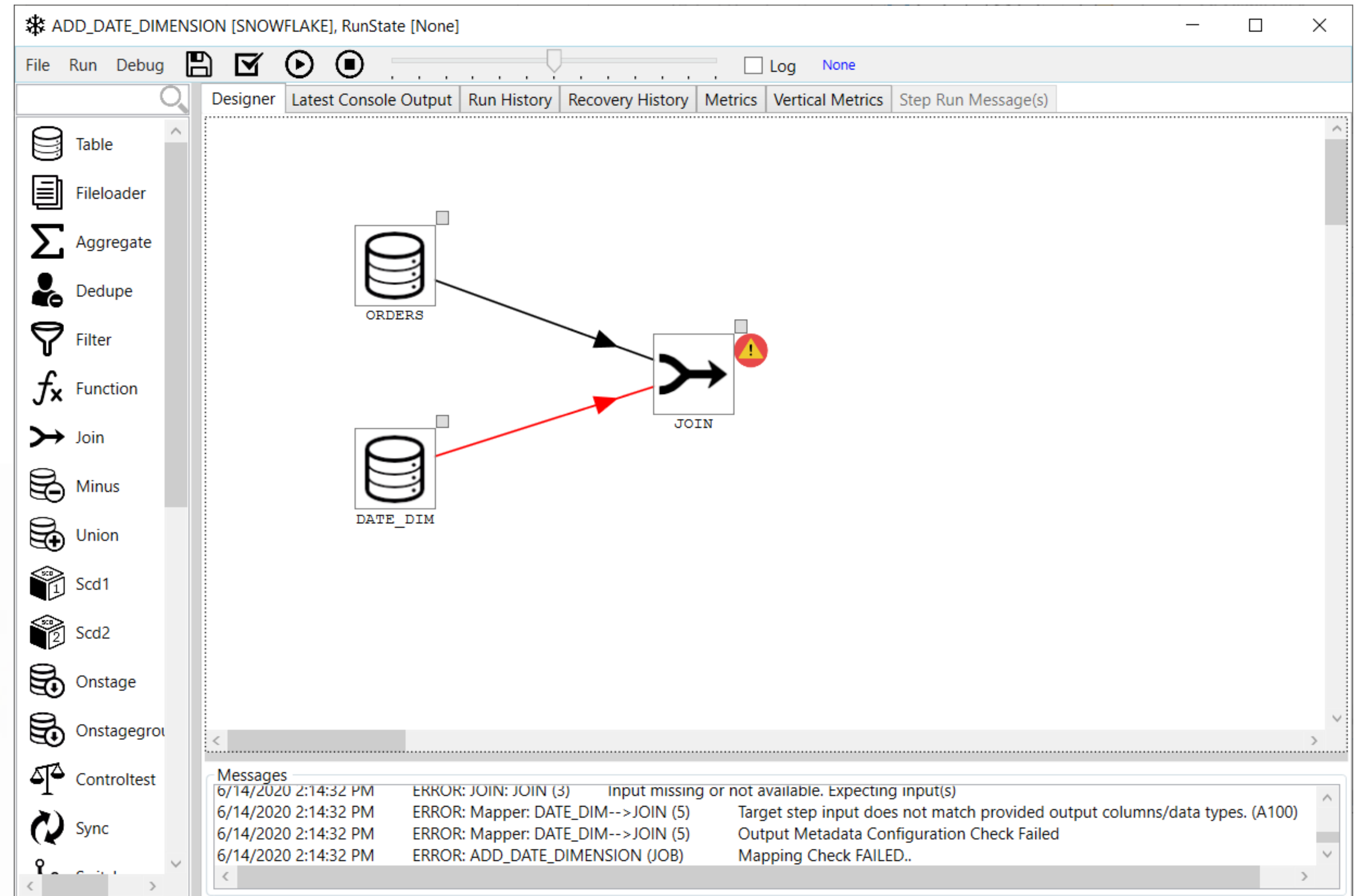
Messages

6/14/2020 1:52:39 PM INFO: ADD_DATE_DIMENSION (JOB) Mapping Check OK
6/14/2020 1:52:39 PM INFO: ADD_DATE_DIMENSION (JOB) \$SYSTEM_DEFAULT_SSH_CONNECTION: SYSTEM_SSH
6/14/2020 1:52:39 PM INFO: ADD_DATE_DIMENSION (JOB) \$ENGINE_CONNECTION:

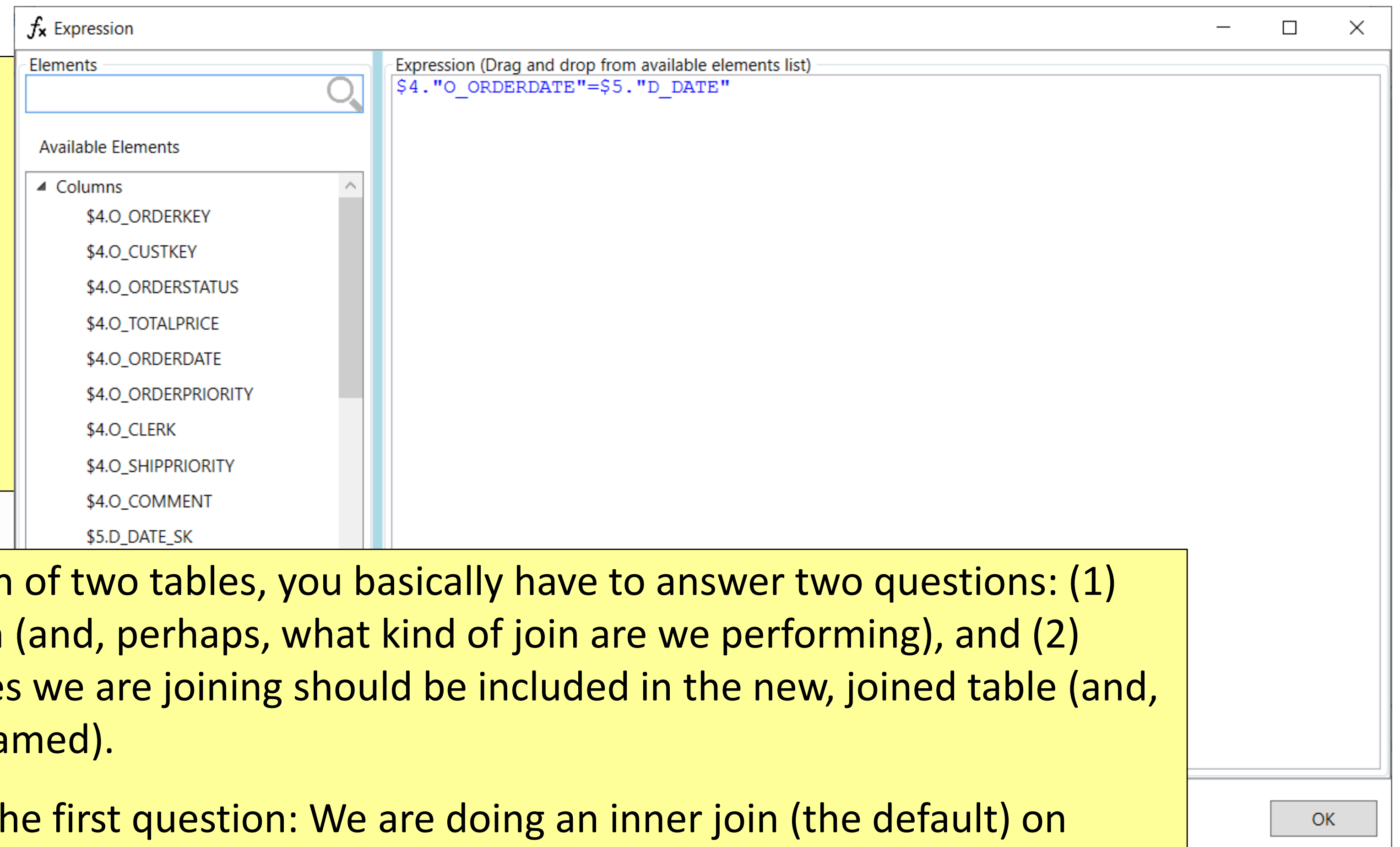
Drag a Join step onto the Designer and Join the two table steps to it as shown.



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.



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.

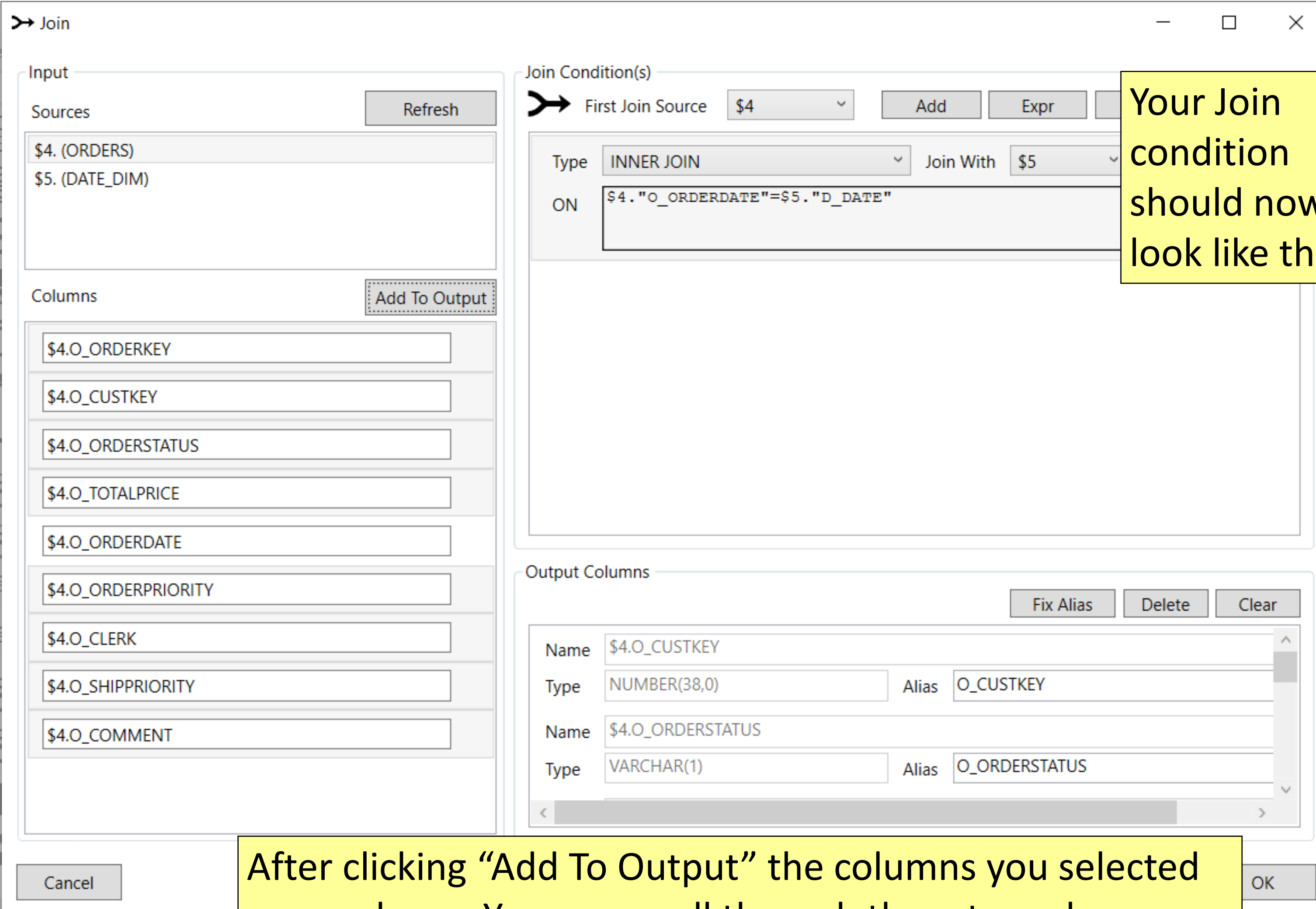


What’s going on? To specify a join of two tables, you basically have to answer two questions: (1) What column(s) are we joining on (and, perhaps, what kind of join are we performing), and (2) What columns from the two tables we are joining should be included in the new, joined table (and, perhaps, how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



The screenshot shows the 'Join' dialog box with the following configuration:

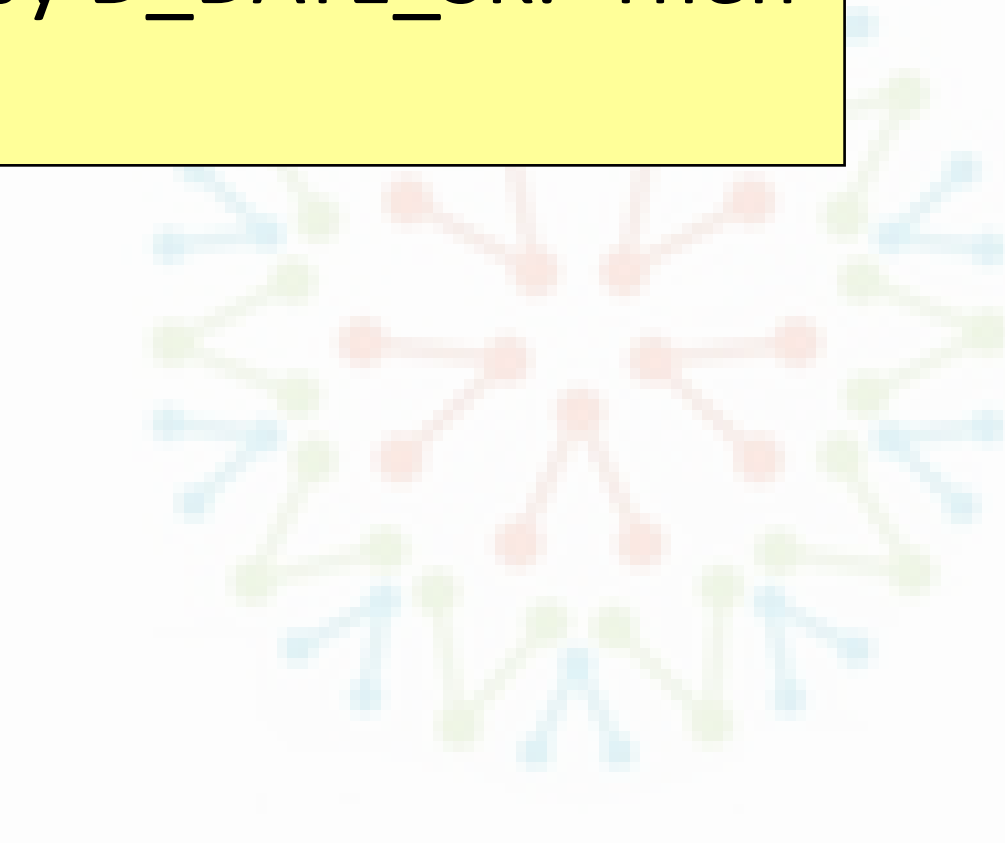
- Input:**
 - Sources:** \$4. (ORDERS), \$5. (DATE_DIM)
 - Columns:** \$4.O_ORDERKEY, \$4.O_CUSTKEY, \$4.O_ORDERSTATUS, \$4.O_TOTALPRICE, \$4.O_ORDERDATE, \$4.O_ORDERPRIORITY, \$4.O_CLERK, \$4.O_SHIPPRIORITY, \$4.O_COMMENT. The column \$4.O_ORDERDATE is selected.
- Join Condition(s):**
 - Type:** INNER JOIN
 - Join With:** \$5
 - ON:** \$4. "O_ORDERDATE"=\$5. "D_DATE"
- Output Columns:**
 - 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.



Join

Input

Sources Refresh

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

Columns Add To Output

- \$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_DOW

Join Condition(s)

First Join Source \$4 Add Expr Delete Reset

Type INNER JOIN Join With \$5

ON \$4."O_ORDERDATE"=\$5."D_DATE"

Output Columns Fix Alias Delete Clear

Name	\$4.O_CUSTKEY	Type	NUMBER(38,0)	Alias	O_CUSTKEY
Name	\$4.O_ORDERSTATUS	Type	VARCHAR(1)	Alias	O_ORDERSTATUS

Cancel OK

❄️ ADD_DATE_DIMENSION [SNOWFLAKE], RunState [None]

File Run Debug [Save] [Check] [Play] [Stop] [Log] None

Designer Latest Console Output Run History Recovery History Metrics Vertical Metrics Step Run Message(s)

Table
Fileloader
Aggregate
Dedupe
Filter
Function
Join
Minus
Union
Scd1
Scd2
Onstage
Onstagegro
Controltest
Sync

ORDERS
DATE_DIM
JOIN
TABLE

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.

Messages

6/15/2020 11:27:29 AM	ERROR: JOIN: JOIN (3)	Output is not terminated with input.(OAB)
6/15/2020 11:27:29 AM	ERROR: JOIN: JOIN (3)	Output missing or not available. Expecting output(s)
6/15/2020 11:28:10 AM	ERROR: TABLE: TABLE (6)	Property missing or not available. Is this step initialized?
6/15/2020 11:28:10 AM	ERROR: ADD_DATE_DIMENSION (JOB)	Mapping Check FAILED..

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
Schema: PUBLIC
Table: FORDERS

Columns List

O_CUSTKEY	NUMBER(38,0)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_ORDERSTATUS	VARCHAR(1)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_ORDERKEY	NUMBER(38,0)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_TOTALPRICE	NUMBER(12,2)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_ORDERPRIORITY	VARCHAR(15)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_CLERK	VARCHAR(15)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_SHIPPRIORITY	NUMBER(38,0)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
O_COMMENT	VARCHAR(80)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident
D_DATE_SK	NUMBER(38,0)	<input type="checkbox"/> Cluster	<input type="checkbox"/> Key	<input type="checkbox"/> Ident

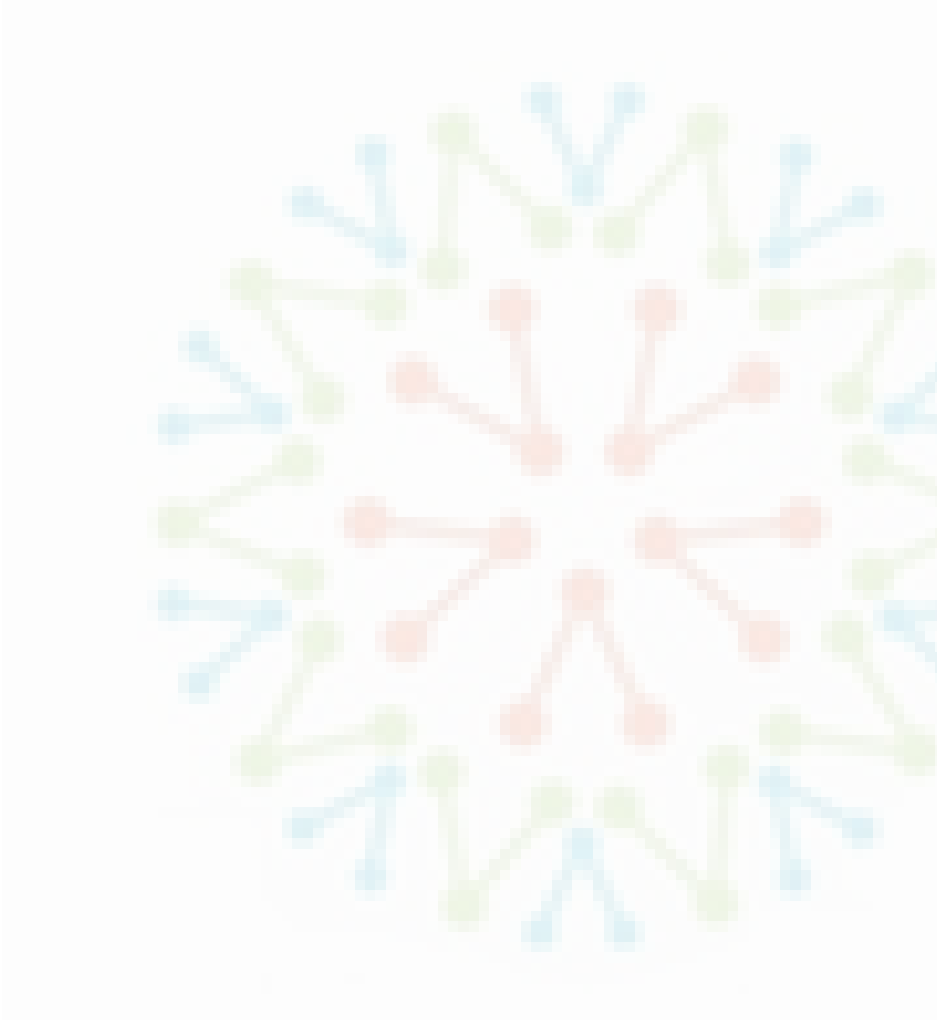
Buttons: Browse, Preview, Refresh (circled in red), Cancel, OK

Click “Browse” and select the DWH.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.



→ Mapping (\$7) [X]

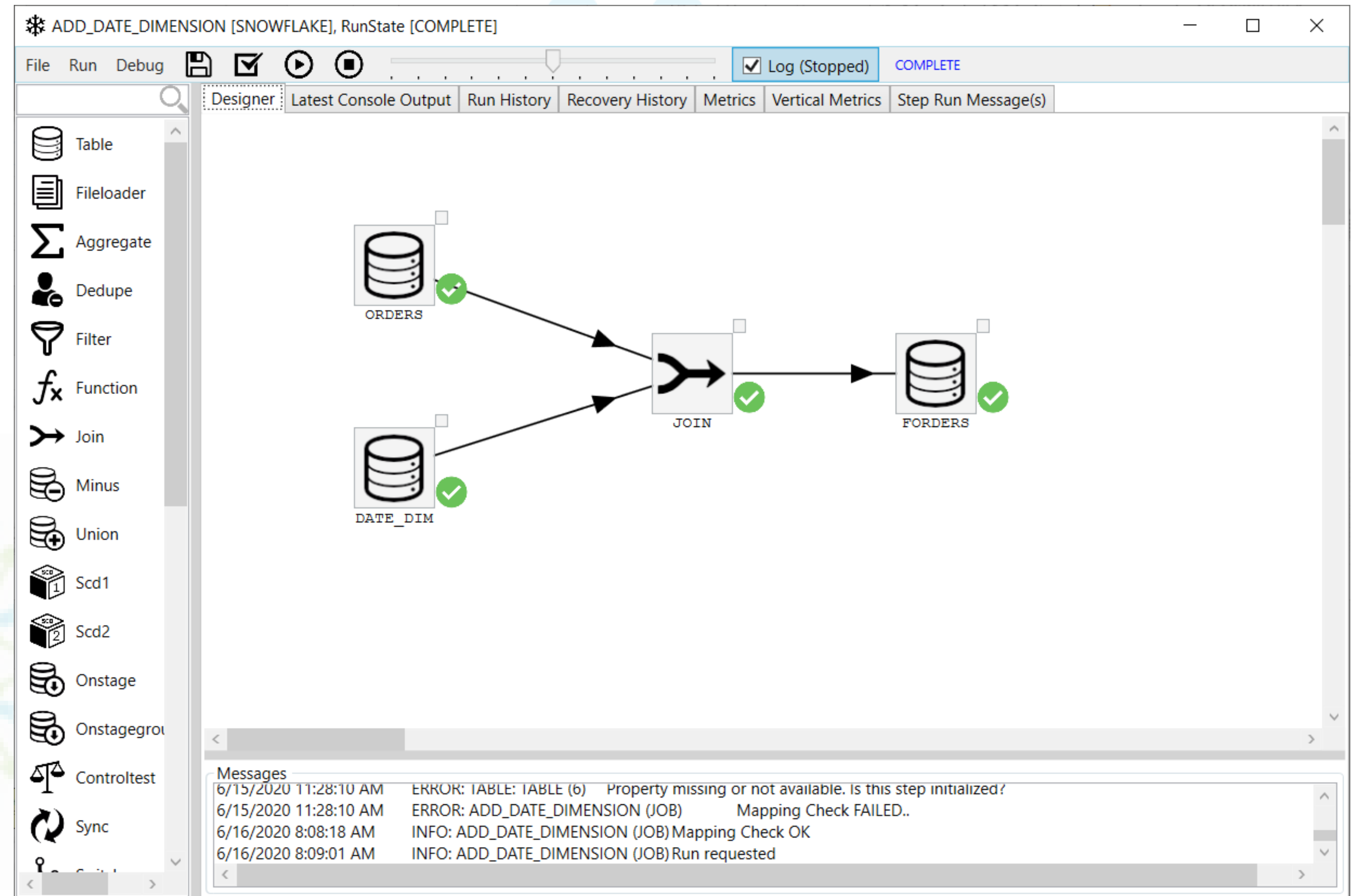
Input Mapping Output

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

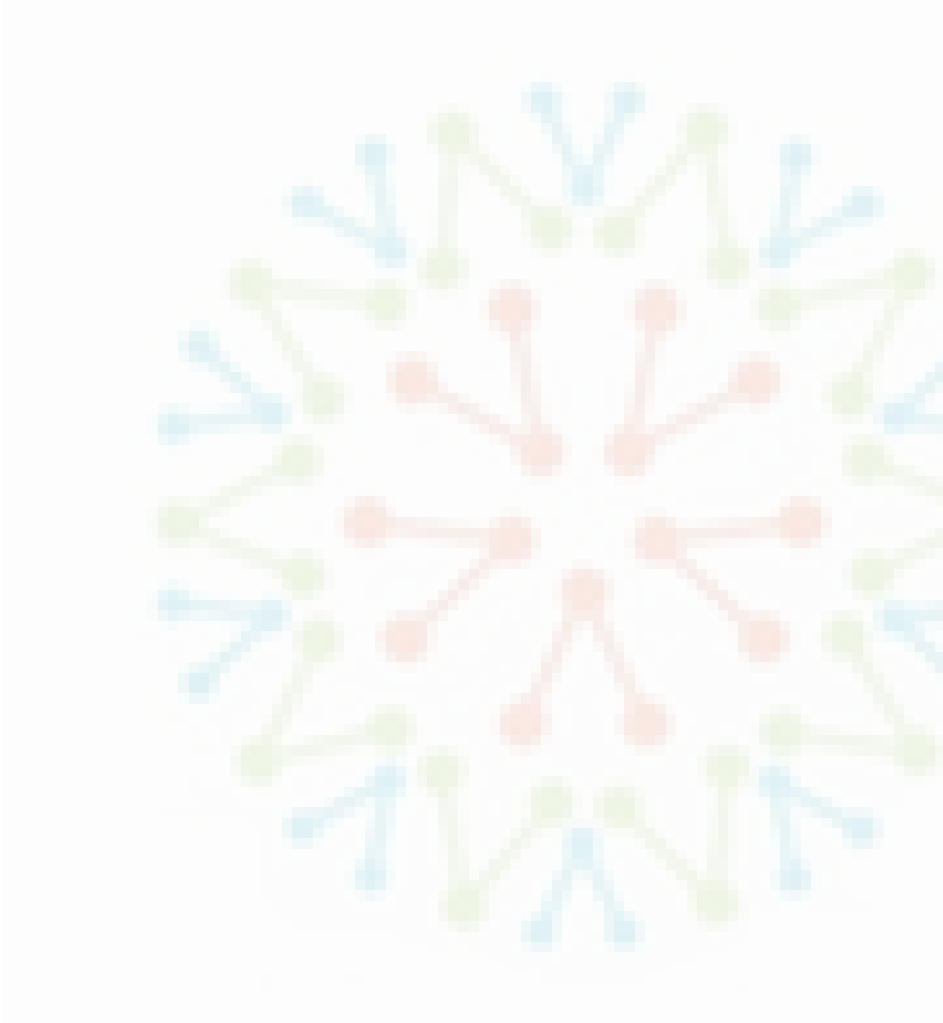
CANCEL OK / SAVE

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.



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



ADD_DATE_DIMENSION [SNOWFLAKE], RunState [COMPLETE]

File Run Debug Log COMPLETE

Designer Latest Console Output Run History Recovery History Metrics Vertical Metrics Step Run Message(s)

Table
Fileloader
Aggregate
Dedupe
Filter
Function
Join
Minus
Union
Scd1
Scd2
Onstage
Onstagegr
Controltest
Sync

Preview Data

Data Query

O_ORDERKEY	O_CUSTKEY	O_ORDERSTATUS	O_TOTALPRICE	O_ORDERPRIORITY	O_CLERK	O_SHIPPRIORITY	O_COMMENT
3282017	116749	F	179665.22	5-LOW	Clerk#000000844	0	beans. final, final
3282051	147890	P	188668.16	5-LOW	Clerk#000000752	0	s after the silent t
3282053	23311	O	52827.26	1-URGENT	Clerk#000000401	0	cuses. carefully ir
3282145	32722	F	92003.86	3-MEDIUM	Clerk#000000008	0	quiet attainments
3282176	17935	O	156633.86	2-HIGH	Clerk#000000619	0	o beans haggle c
3282179	144284	F	276104.37	4-NOT SPECIFIED	Clerk#000000509	0	arefully final pack
3282213	141334	F	74151.88	2-HIGH	Clerk#000000872	0	ideas dazzle blith
3282277	102350	O	94576.80	2-HIGH	Clerk#000000735	0	eve fluffily along
3282278	131905	F	121662.29	5-LOW	Clerk#000000915	0	yly regular ideas.
3282306	126695	O	99602.65	2-HIGH	Clerk#000000764	0	ironic dependenc
3282374	97531	F	52216.23	1-URGENT	Clerk#000000141	0	blithely. carefully
3282401	131536	O	187972.84	3-MEDIUM	Clerk#000000950	0	y regular package
3282436	147403	F	164797.31	2-HIGH	Clerk#000000428	0	otes! instructions
3282464	94417	F	82208.81	1-URGENT	Clerk#000000195	0	ideas. blithely irc
3282531	98983	F	141748.67	5-LOW	Clerk#000000812	0	unusual package:
3282533	27155	F	125129.12	3-MEDIUM	Clerk#000000408	0	packages cajole s

Cancel OK

Messages

6/15/2020 11:28:10 AM ERROR: IABLE: IABLE (6) Property missing or not available. Is this step initialized?
6/15/2020 11:28:10 AM ERROR: ADD_DATE_DIMENSION (JOB) Mapping Check FAILED..
6/16/2020 8:08:18 AM INFO: ADD_DATE_DIMENSION (JOB) Mapping Check OK
6/16/2020 8:09:01 AM INFO: ADD_DATE_DIMENSION (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.**