A Data Factory solution for moving and transforming data with dataflows and data pipelines

Overview

This hands-on lab is designed to give you a fast and practical introduction to **Microsoft Fabric's Data Factory**. In roughly one hour, you'll work through a complete data integration example that demonstrates how to move, transform, and automate data workflows. By completing the exercises, you'll gain firsthand experience with Data Factory's main features and understand how it can streamline end-to-end data processes.

Goals

You'll complete the lab through three guided exercises that build upon one another:

1. **Exercise 1:** Design and run a **data pipeline** in Data Factory to **extract raw files** from **Azure Blob Storage** and load them into a **Bronze layer** in your Data Lakehouse.

2. **Exercise 2:** Create a **data transformation flow** that cleans, enriches, and transfers the data from the **Bronze** layer to a refined **Gold** layer.

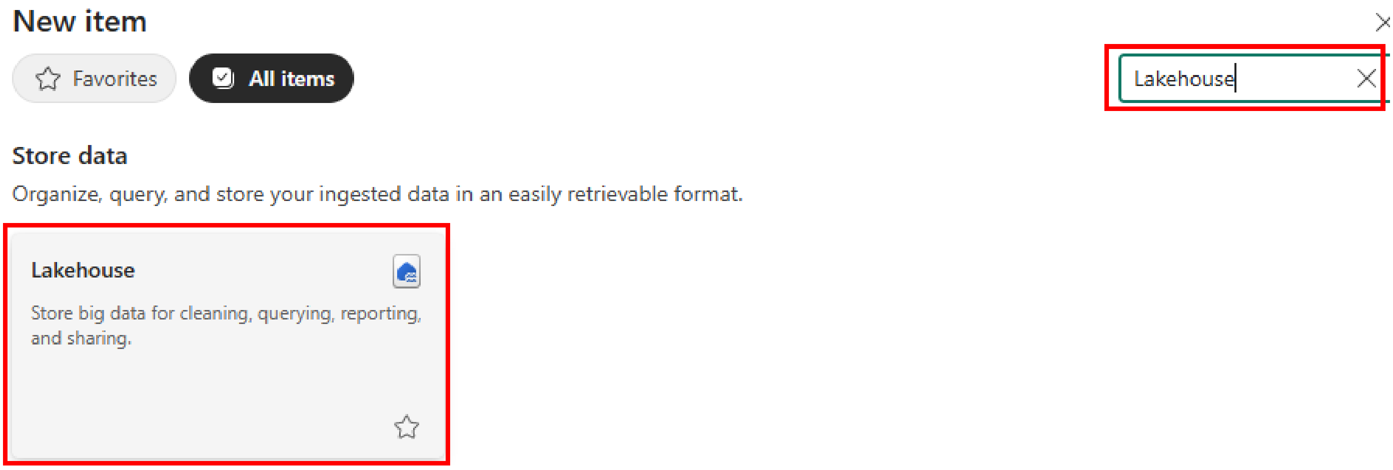
3. **Optional Exercise:** Implement **automation and alerts** by configuring Data Factory to **notify you via email** once all processes finish, and set up a **recurring schedule** for the complete workflow.

By the end of the lab, you'll have a working automated data pipeline that demonstrates the flexibility and efficiency of Data Factory within Microsoft Fabric.

Exercise 1: Create a pipeline with Data Factory

Create a Lakehouse

1. Create a workspace **Data-FactoryXX** if you don't have with trial capacity.
2. In the **Data-FactoryXX** workspace page, navigate and click on **+New** button, then select **Lakehouse.**



New item

Favorites

All items

Lakehouse

Store data

Organize, query, and store your ingested data in an easily retrievable format.

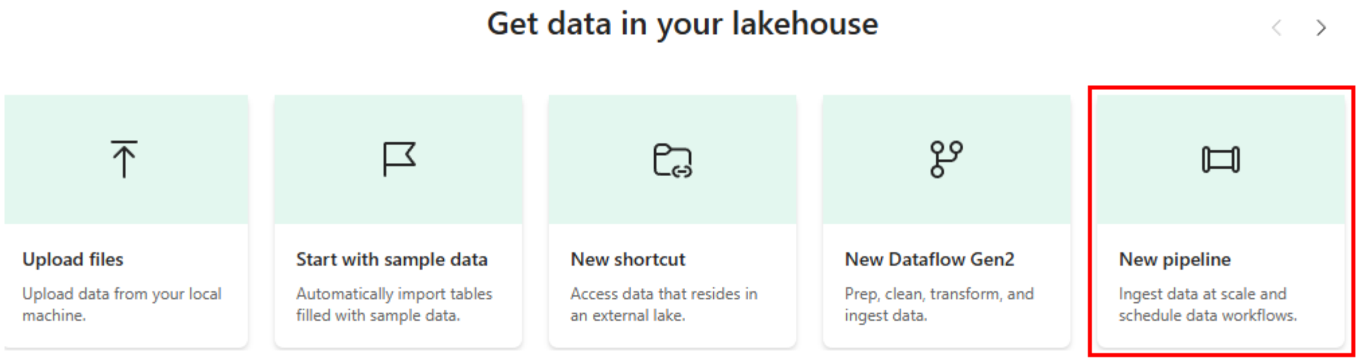
Lakehouse

Store big data for cleaning, querying, reporting, and sharing.

3. In the **New lakehouse** dialog box, enter **lh_df** in the **Name** field, click on the **Create** button and open the new lakehouse.

Create a data pipeline

1. In the **Lakehouse** Home page, click on **New pipeline** as shown in the below image.



Get data in your lakehouse

Upload files
Upload data from your local machine.

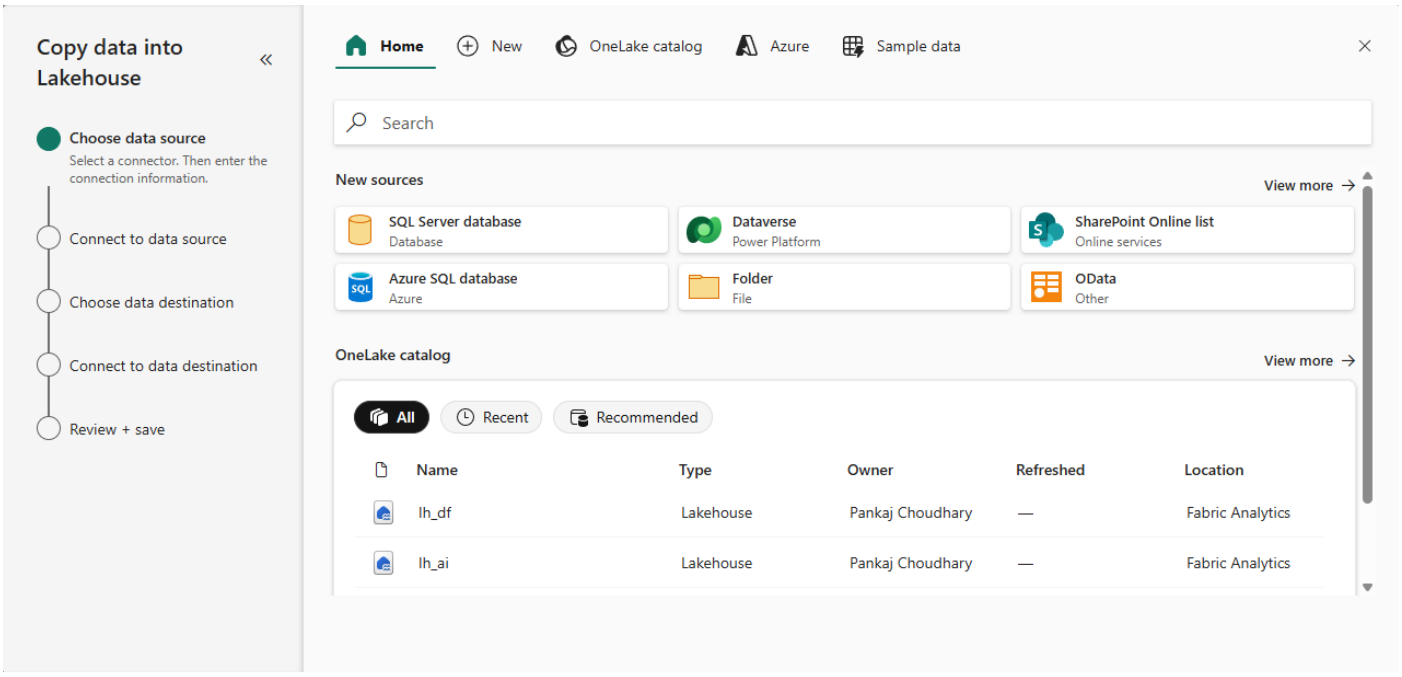
Start with sample data
Automatically import tables filled with sample data.

New shortcut
Access data that resides in an external lake.

New Dataflow Gen2
Prep, clean, transform, and ingest data.

New pipeline
Ingest data at scale and schedule data workflows.

2. In the **New pipeline** dialog box, enter!! **First_Pipeline1** !! in the **Name** field, then click on the **Create** button. A **copy assistant tool** will open.



Copy data into Lakehouse

Choose data source
Select a connector. Then enter the connection information.

Connect to data source

Choose data destination

Connect to data destination

Review + save

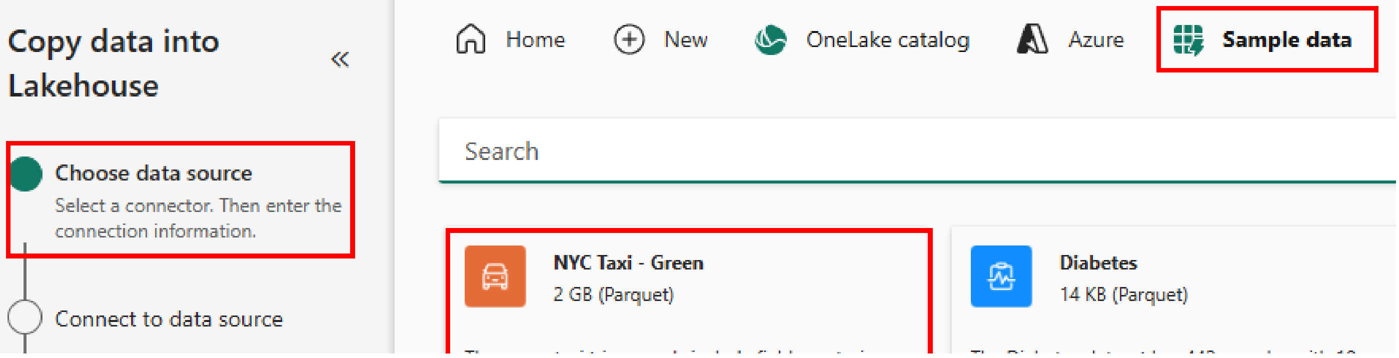
New sources

SQL Server database Database	Dataverse Power Platform	SharePoint Online list Online services
Azure SQL database Azure	Folder File	OData Other

OneLake catalog

Name	Type	Owner	Refreshed	Location
lh_df	Lakehouse	Pankaj Choudhary	—	Fabric Analytics
lh_ai	Lakehouse	Pankaj Choudhary	—	Fabric Analytics

3. The **Copy data** dialog is displayed with the first step, **Choose data source**, highlighted. Select **Sample data** section, and select the **NYC Taxi-Green** data source type. Then select **Next**.



Copy data into Lakehouse

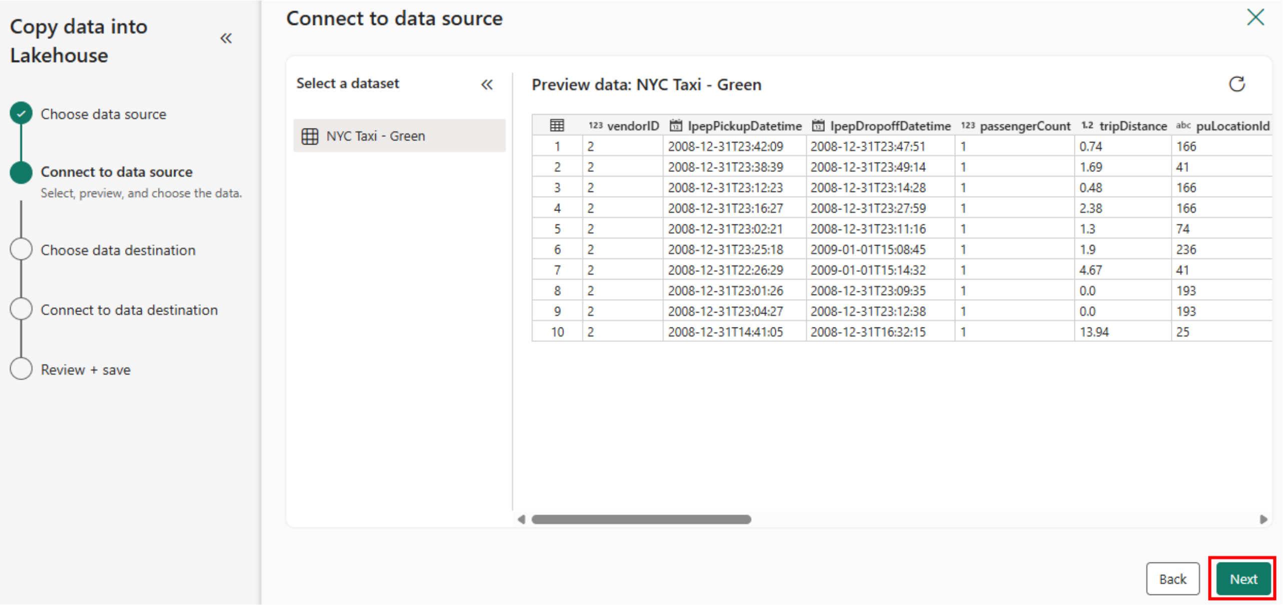
Choose data source
Select a connector. Then enter the connection information.

Connect to data source

NYC Taxi - Green
2 GB (Parquet)

Diabetes
14 KB (Parquet)

4. In the **Connect to data source**, click on **Next** button.



Copy data into Lakehouse

- Choose data source
- Connect to data source
  Select, preview, and choose the data.
- Choose data destination
- Connect to data destination
- Review + save

Connect to data source

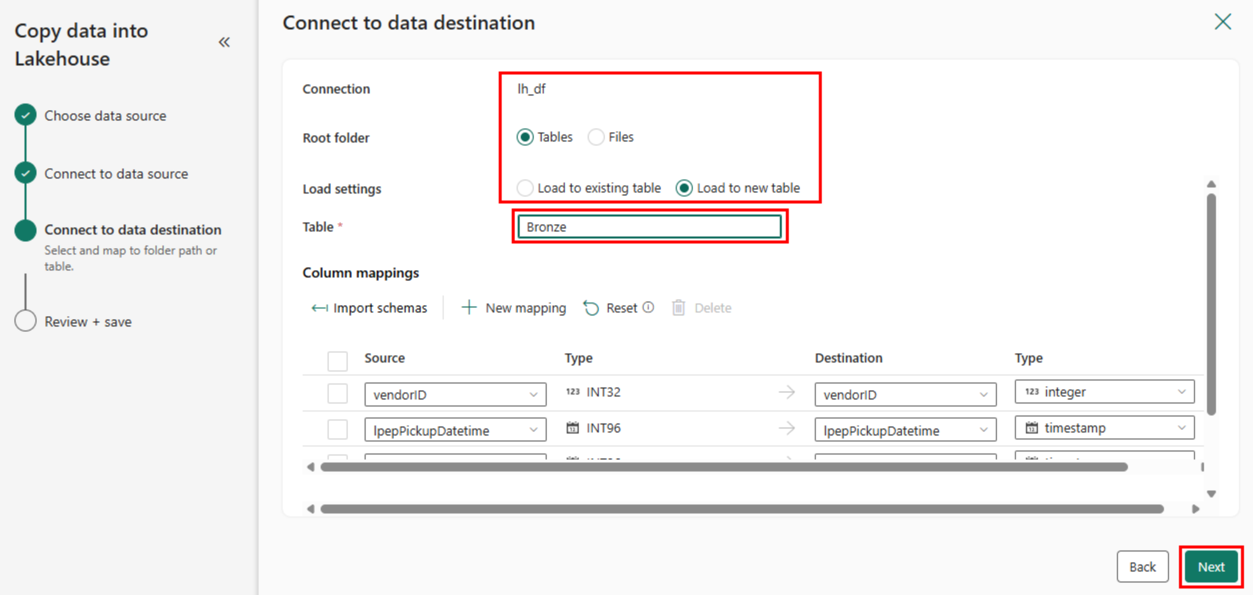
Select a dataset

NYC Taxi - Green

Preview data: NYC Taxi - Green

	vendorID	IpepPickupDatetime	IpepDropoffDatetime	passengerCount	tripDistance	puLocationId
1	2	2008-12-31T23:42:09	2008-12-31T23:47:51	1	0.74	166
2	2	2008-12-31T23:38:39	2008-12-31T23:49:14	1	1.69	41
3	2	2008-12-31T23:12:23	2008-12-31T23:14:28	1	0.48	166
4	2	2008-12-31T23:16:27	2008-12-31T23:27:59	1	2.38	166
5	2	2008-12-31T23:02:21	2008-12-31T23:11:16	1	1.3	74
6	2	2008-12-31T23:25:18	2009-01-01T15:08:45	1	1.9	236
7	2	2008-12-31T22:26:29	2009-01-01T15:14:32	1	4.67	41
8	2	2008-12-31T23:01:26	2008-12-31T23:09:35	1	0.0	193
9	2	2008-12-31T23:04:27	2008-12-31T23:12:38	1	0.0	193
10	2	2008-12-31T14:41:05	2008-12-31T16:32:15	1	13.94	25

5. For the **Choose data destination** step of the copy assistant, you **Lakehouse** selected and then **Next**.
6. Provide a table name **Bronze**, and select the **Next**.



Copy data into Lakehouse

- Choose data source
- Connect to data source
- Connect to data destination
  Select and map to folder path or table.
- Review + save

Connect to data destination

Connection: lh_df

Root folder: ● Tables ○ Files

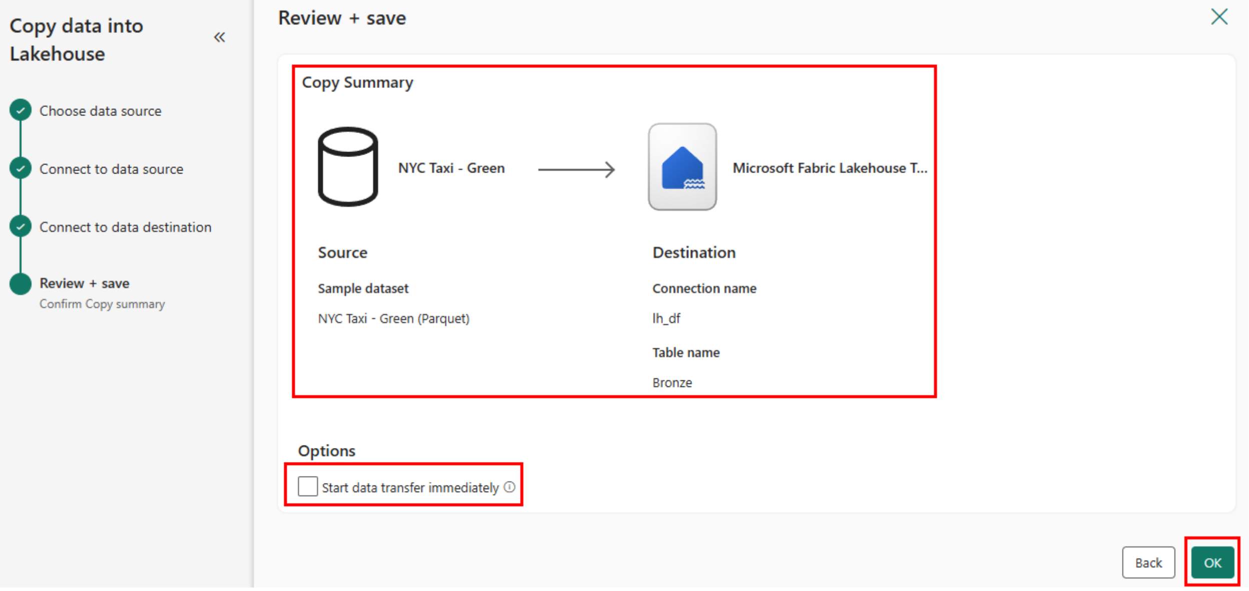
Load settings: ○ Load to existing table ● Load to new table

Table *: Bronze

Column mappings

	Source	Type	Destination	Type
□	vendorID	INT32	vendorID	integer
□	IpepPickupDatetime	INT96	IpepPickupDatetime	timestamp

7. Finally, on the **Review + save** page of the copy data assistant, review the configuration. For this lab, uncheck the **Start data transfer immediately** checkbox, since we run the activity manually in the next step. Then select **OK**.



Copy data into Lakehouse

- Choose data source
- Connect to data source
- Connect to data destination
- Review + save
  Confirm Copy summary

Review + save

Copy Summary

NYC Taxi - Green → Microsoft Fabric Lakehouse T...

Source

Sample dataset

NYC Taxi - Green (Parquet)

Destination

Connection name

lh_df

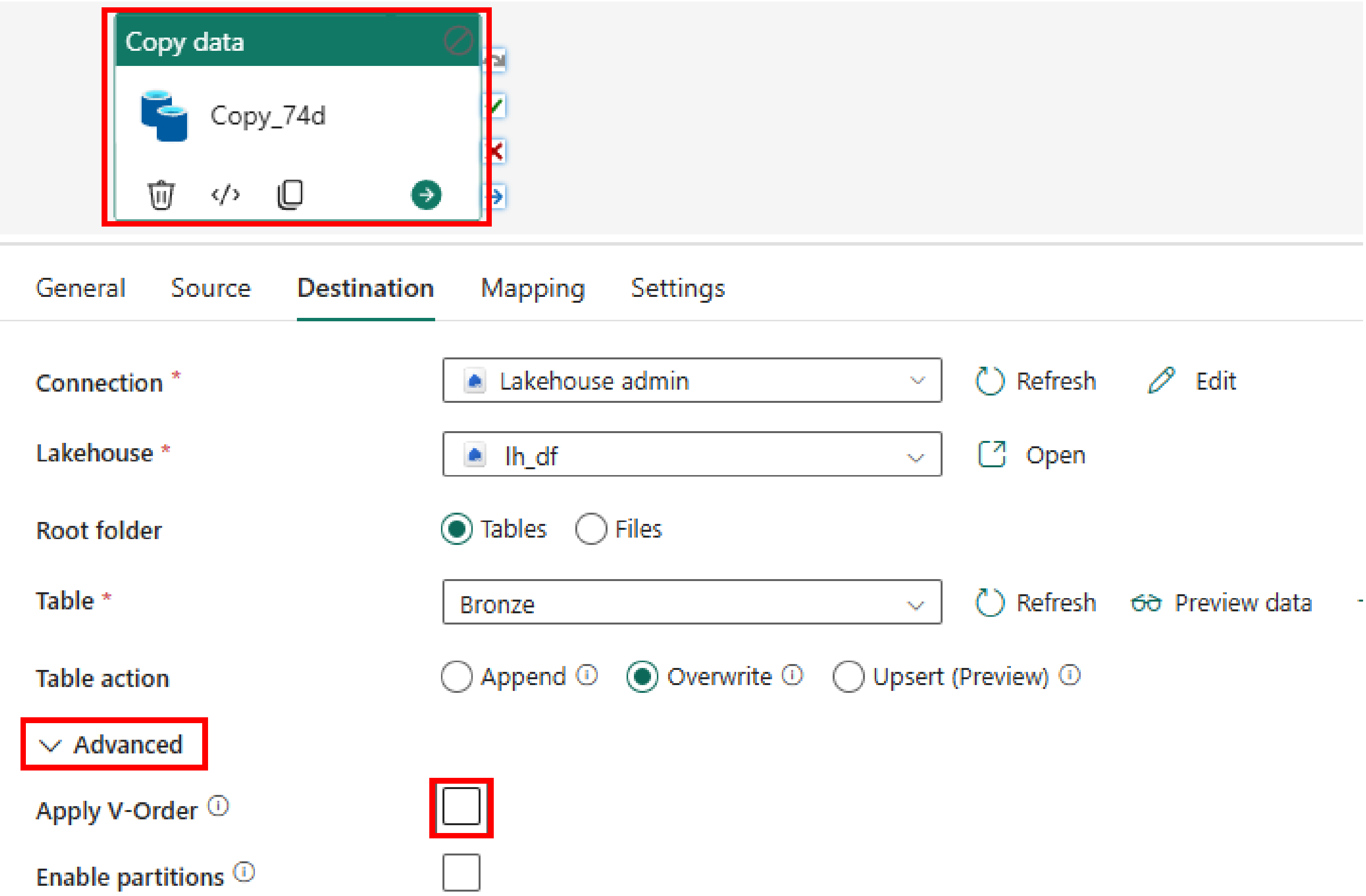
Table name

Bronze

Options

□ Start data transfer immediately

8. Select the **Copy data** activity, in the **Destination** Tab under **Advanced** uncheck **Apply V-Order.**



Connection * Lakehouse admin

Lakehouse * lh_df

Root folder ● Tables ○ Files

Table * Bronze

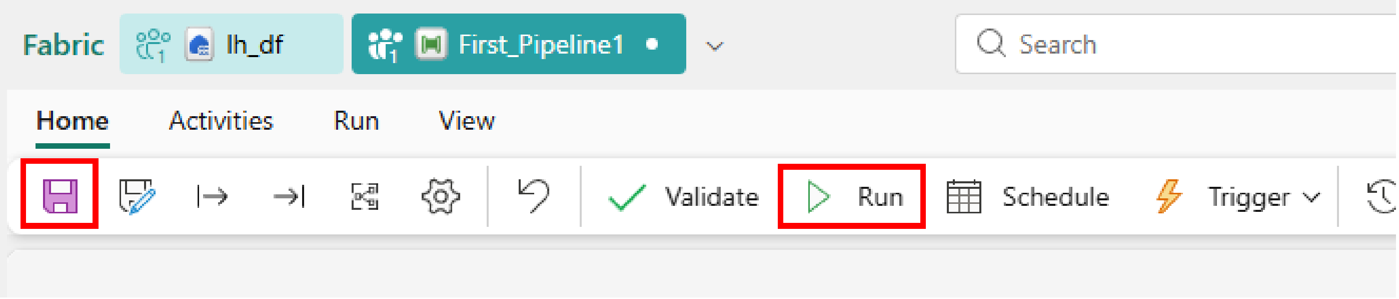
Table action ○ Append ● Overwrite ○ Upsert (Preview)

Advanced

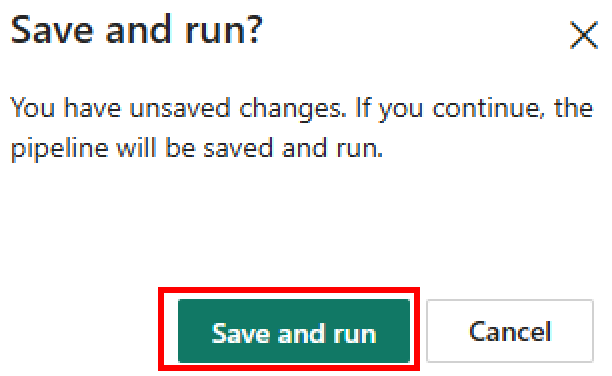
Apply V-Order □

Enable partitions □

9. On the **Home** tab of the pipeline editor window, then select the **Save** and then **Run** button.



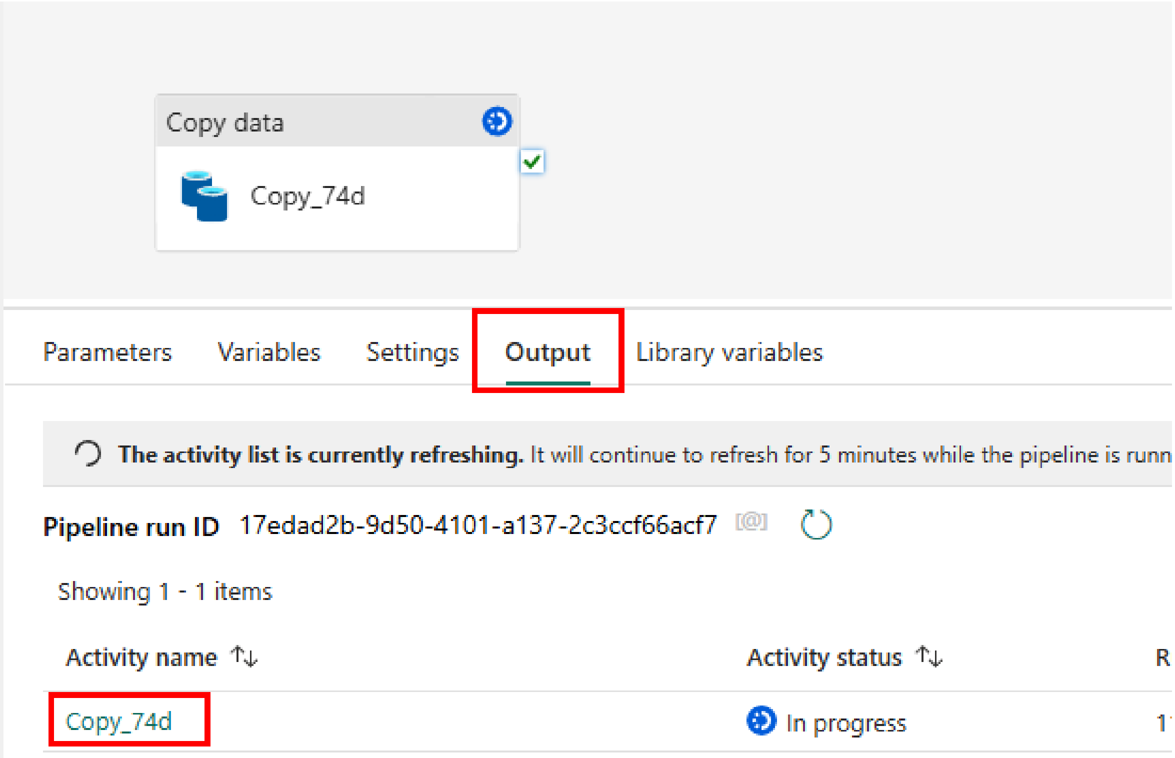
10. In the **Save and run?** dialog box, click on **Save and run** button to execute these activities.



Save and run?

You have unsaved changes. If you continue, the pipeline will be saved and run.

11. You can monitor the run and check the results on the **Output** tab below the pipeline canvas. Select the **activity name** as **Copy_74d** to view the run details.



Copy data

Copy_74d

Parameters | Variables | Settings | **Output** | Library variables

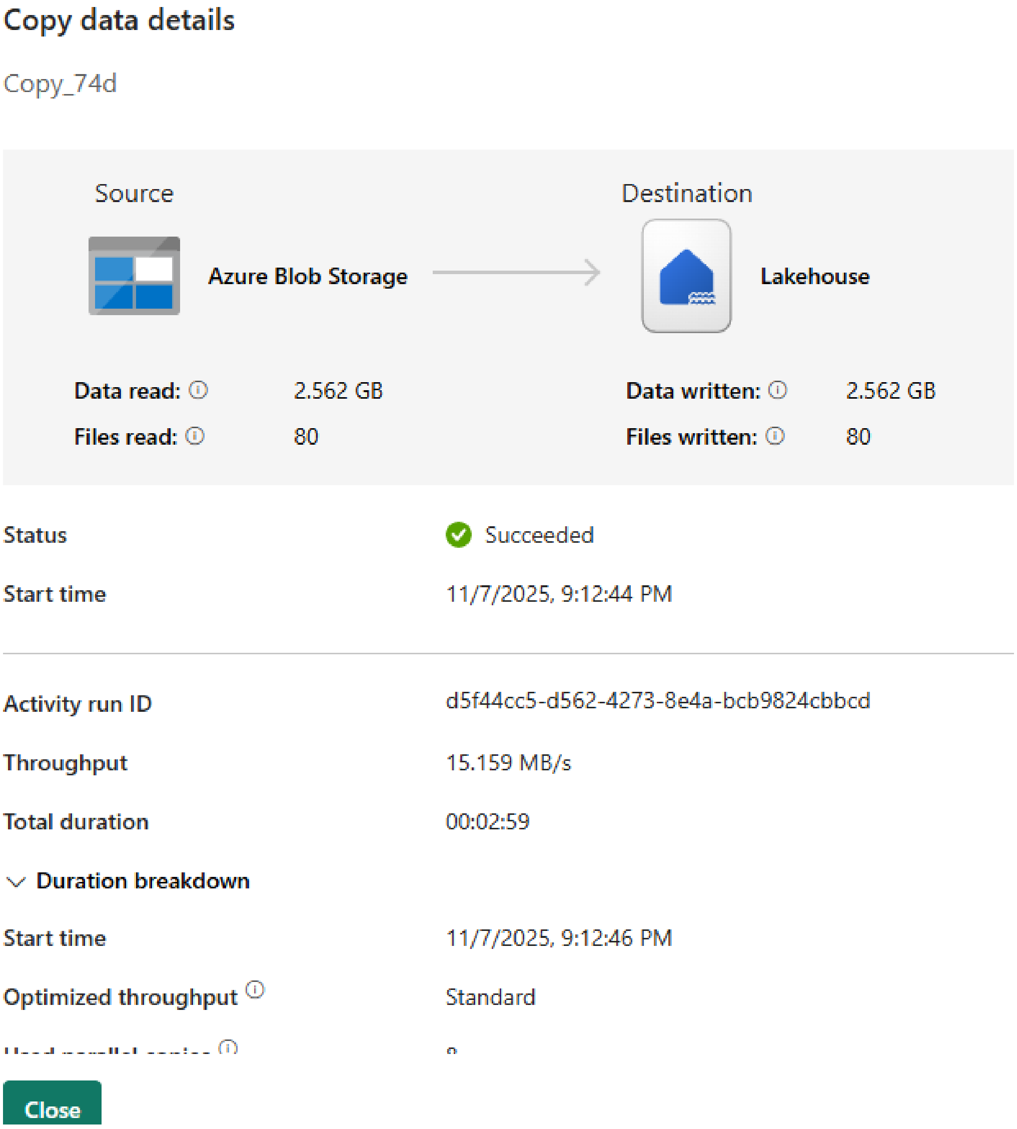
The activity list is currently refreshing. It will continue to refresh for 5 minutes while the pipeline is runn

Pipeline run ID 17edad2b-9d50-4101-a137-2c3ccf66acf7

Showing 1 - 1 items

Activity name ↑↓	Activity status ↑↓	R
Copy_74d	In progress	1

12. The run details show 80 files read and written.



Copy data details

Copy_74d

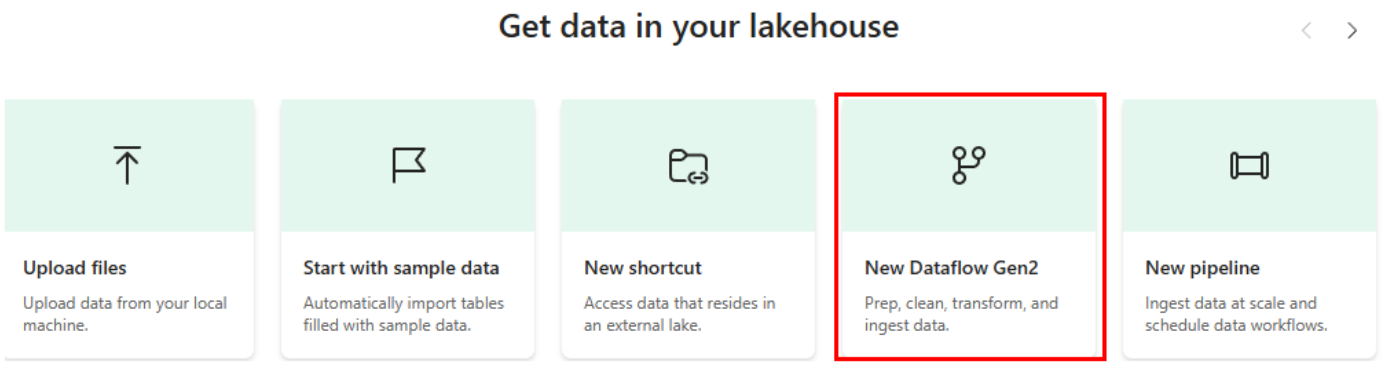
Data read: ⓘ	2.562 GB	**Data written:** ⓘ	2.562 GB
Files read: ⓘ	80	**Files written:** ⓘ	80

Status	✓ Succeeded
Start time	11/7/2025, 9:12:44 PM
Activity run ID	d5f44cc5-d562-4273-8e4a-bcb9824cbbcd
Throughput	15.159 MB/s
Total duration	00:02:59
∨ Duration breakdown	
Start time	11/7/2025, 9:12:46 PM
Optimized throughput ⓘ	Standard
Used parallel copies ⓘ	8

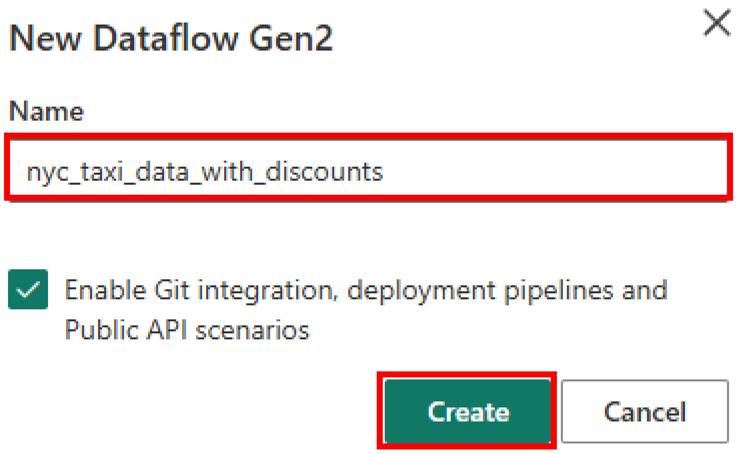
13. Expand the **Duration breakdown** section to see the duration of each stage of the Copy activity. After reviewing the copy details, select **Close**.

Exercise 2: Transform data with a dataflow in Data Factory

1. On the **lh_df** Lakehouse page, select the **New Dataflow Gen2.**


Get data in your lakehouse
Upload files
Upload data from your local machine.
Start with sample data
Automatically import tables filled with sample data.
New shortcut
Access data that resides in an external lake.
New Dataflow Gen2
Prep, clean, transform, and ingest data.
New pipeline
Ingest data at scale and schedule data workflows.

2. On New Dataflow Gen2 Dialog, Enter dataflow name **nyc_taxi_data_with_discounts** and select **Create**.

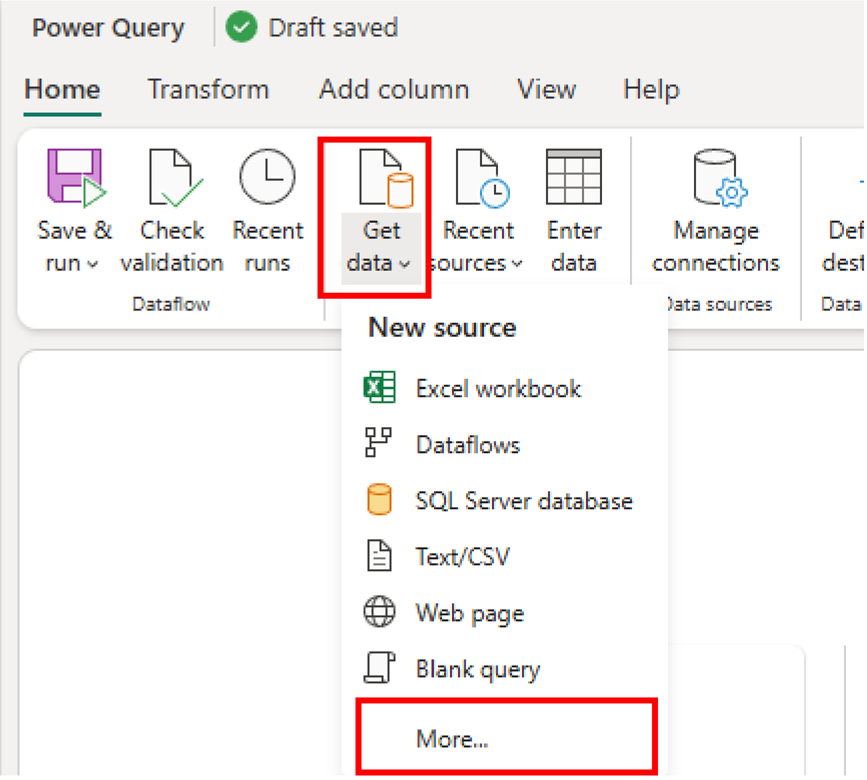

New Dataflow Gen2

Name

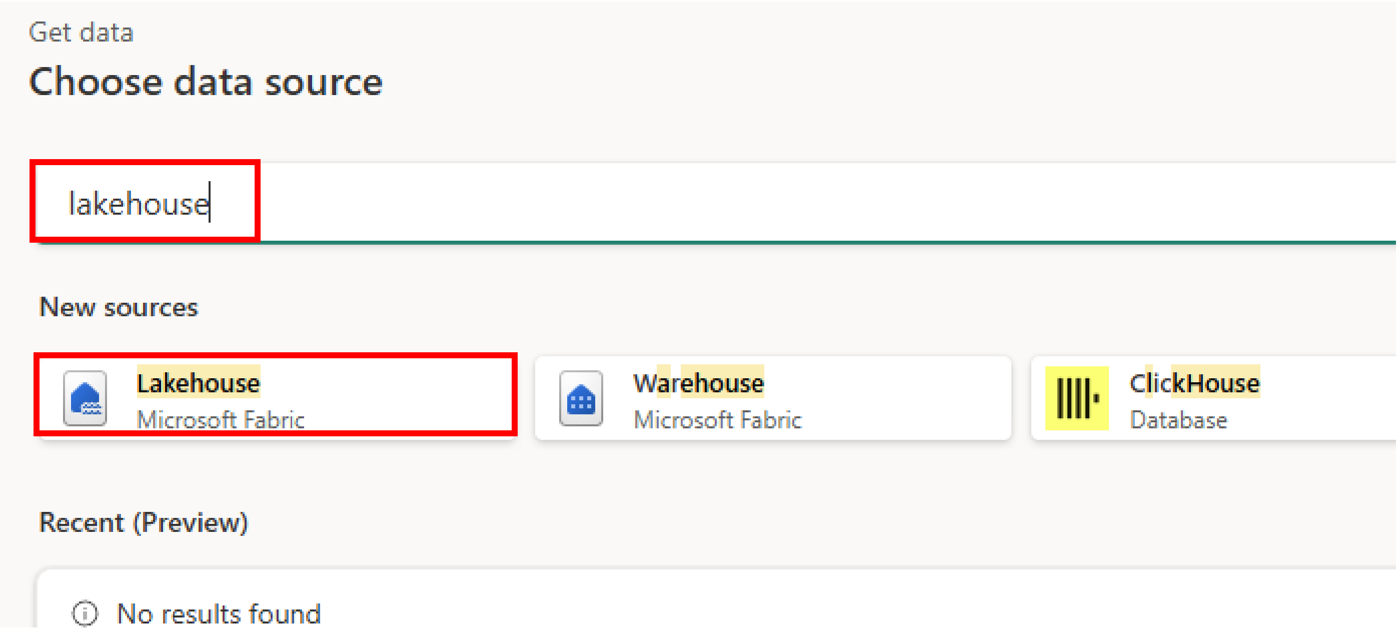
nyc_taxi_data_with_discounts

☑ Enable Git integration, deployment pipelines and Public API scenarios

3. From the dataflow menu, under the **Power Query** pane click on **Get data**, then select **More...**.



4. In the **Choose data source** tab, search box search type!! **Lakehouse!!** and then click on the **Lakehouse** connector.



Get data

Choose data source

lakehouse

New sources

Lakehouse
Microsoft Fabric

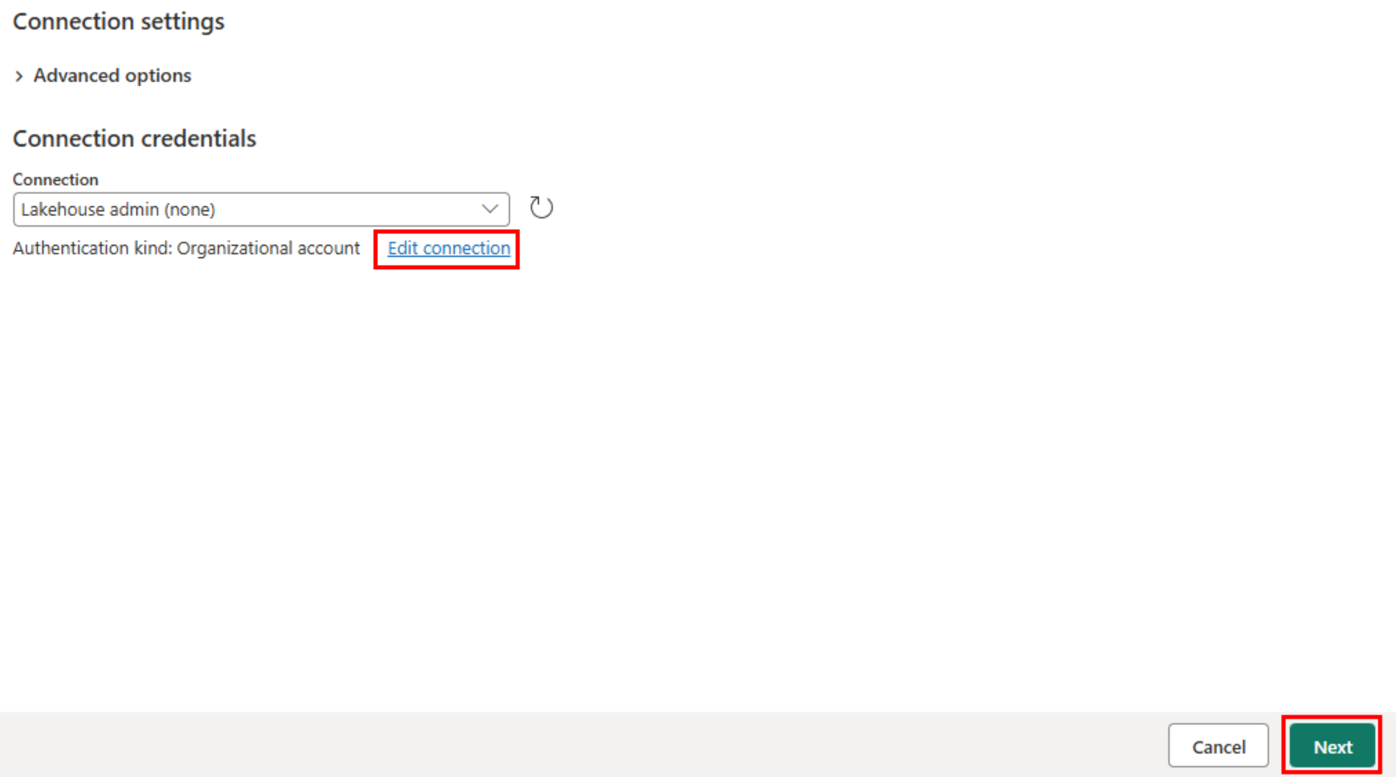
Warehouse
Microsoft Fabric

ClickHouse
Database

Recent (Preview)

ⓘ No results found

5. The **Connect to data source** dialog appears, select **Edit connection.**
6. In the **Connect to data source** dialog box, select **sign in** using your Power BI organizational account to set the identity that the dataflow uses to access the lakehouse.
7. In **Connect to data source** dialog box, select **Next.**



Connection settings

› Advanced options

Connection credentials

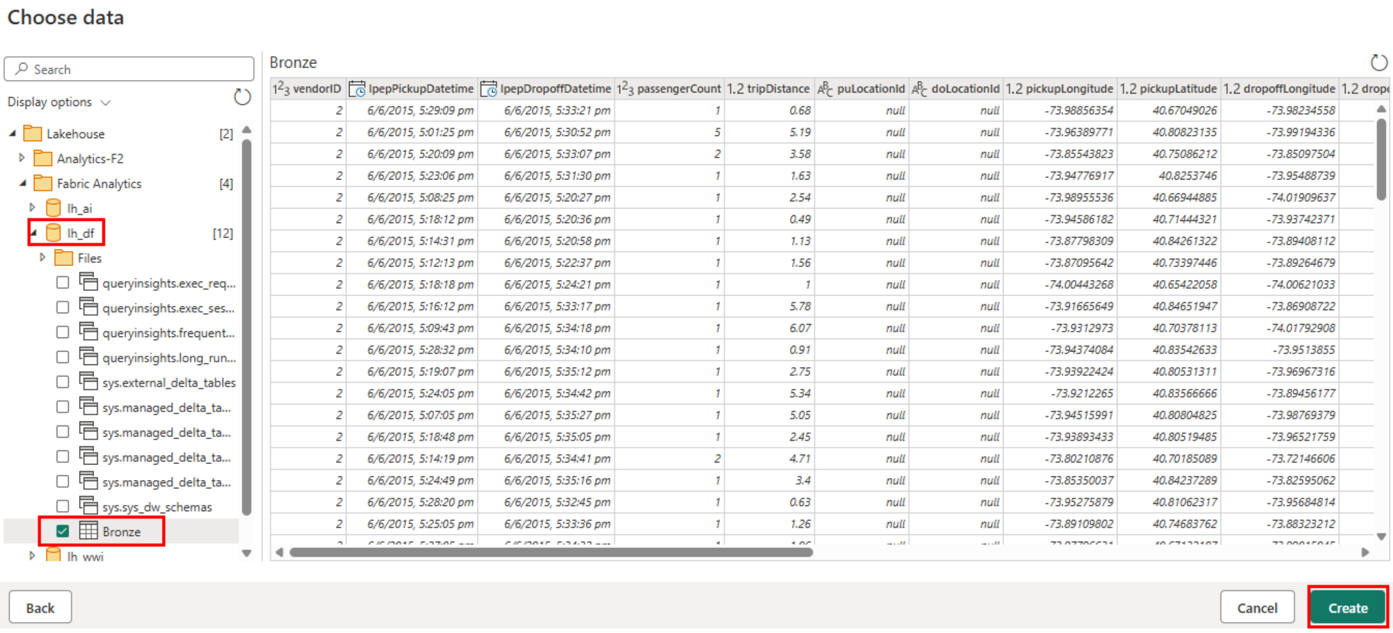
Connection

Lakehouse admin (none)

Authentication kind: Organizational account Edit connection

Cancel Next

8. The **Choose data** dialog is displayed. Use the navigation pane to find the Lakehouse you created for the destination in the prior module, and select the **lh_df** lakehouse and **Bronze** data table then click on **Create** button.



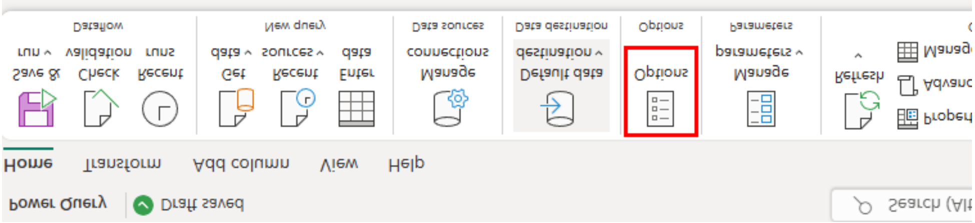
Choose data

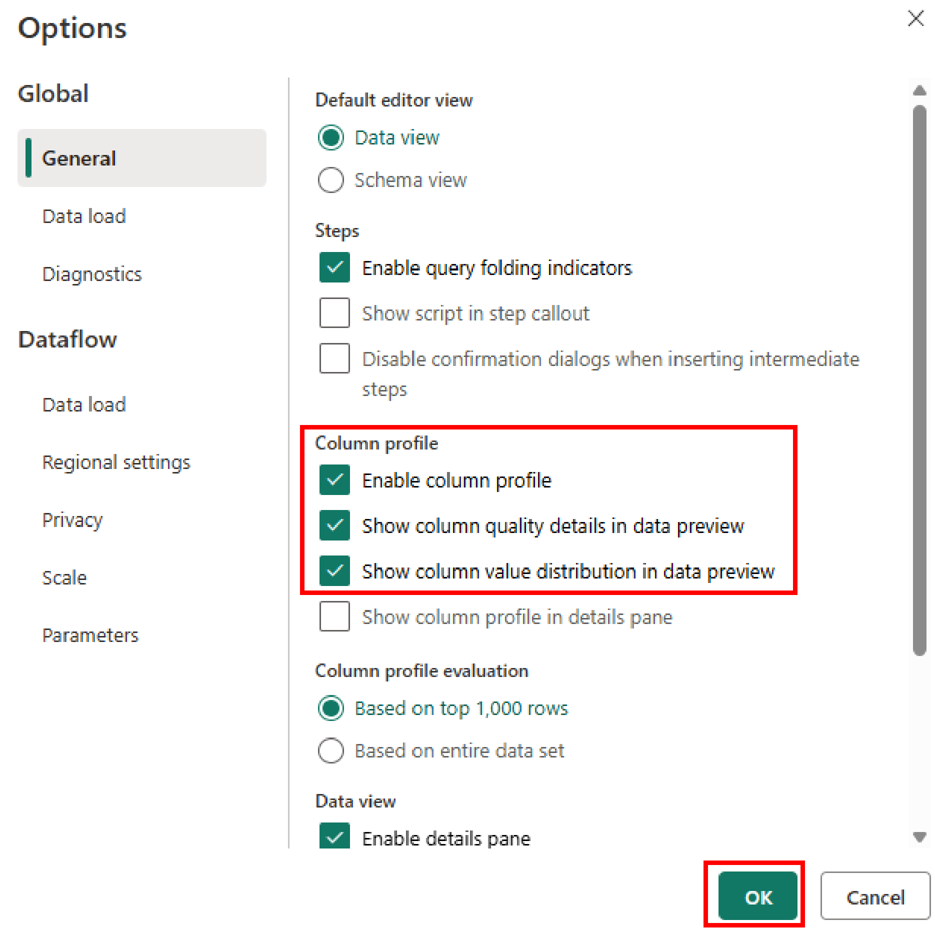
- Lakehouse [2]
  - Analytics-F2
  - Fabric Analytics [4]
    - lh_ai
    - lh_df [12]
      - Files
      - □ queryinsights.exec_req...
      - □ queryinsights.exec_ses...
      - □ queryinsights.frequent...
      - □ queryinsights.long_run...
      - □ sys.external_delta_tables
      - □ sys.managed_delta_ta...
      - □ sys.managed_delta_ta...
      - □ sys.managed_delta_ta...
      - □ sys.managed_delta_ta...
      - □ sys.sys_dw_schemas
      - ☑ Bronze
    - lh wwi

Bronze

vendorID	lpepPickupDatetime	lpepDropoffDatetime	passengerCount	tripDistance	puLocationId	doLocationId	pickupLongitude	pickupLatitude	dropoffLongitude	1.2 drop
2	6/6/2015, 5:29:09 pm	6/6/2015, 5:33:21 pm	1	0.68	null	null	-73.98856354	40.67049026	-73.98234558	
2	6/6/2015, 5:01:25 pm	6/6/2015, 5:30:52 pm	5	5.19	null	null	-73.96389771	40.80823135	-73.99194336	
2	6/6/2015, 5:20:09 pm	6/6/2015, 5:33:07 pm	2	3.58	null	null	-73.85543823	40.75086212	-73.85097504	
2	6/6/2015, 5:23:06 pm	6/6/2015, 5:31:30 pm	1	1.63	null	null	-73.94776917	40.8253746	-73.95488739	
2	6/6/2015, 5:08:25 pm	6/6/2015, 5:20:27 pm	1	2.54	null	null	-73.98955536	40.66944885	-74.01909637	
2	6/6/2015, 5:18:12 pm	6/6/2015, 5:20:36 pm	1	0.49	null	null	-73.94586182	40.71444321	-73.93742371	
2	6/6/2015, 5:14:31 pm	6/6/2015, 5:20:58 pm	1	1.13	null	null	-73.87798309	40.84261322	-73.89408112	
2	6/6/2015, 5:12:13 pm	6/6/2015, 5:22:37 pm	1	1.56	null	null	-73.87095642	40.73397446	-73.89264679	
2	6/6/2015, 5:18:18 pm	6/6/2015, 5:24:21 pm	1	1	null	null	-74.00443268	40.65422058	-74.00621033	
2	6/6/2015, 5:16:12 pm	6/6/2015, 5:33:17 pm	1	5.78	null	null	-73.91665649	40.84651947	-73.86908722	
2	6/6/2015, 5:09:43 pm	6/6/2015, 5:34:18 pm	1	6.07	null	null	-73.9312973	40.70378113	-74.01792908	
2	6/6/2015, 5:28:32 pm	6/6/2015, 5:34:10 pm	1	0.91	null	null	-73.94374084	40.83542633	-73.9513855	
2	6/6/2015, 5:19:07 pm	6/6/2015, 5:35:12 pm	1	2.75	null	null	-73.93922424	40.80531311	-73.96967316	
2	6/6/2015, 5:24:05 pm	6/6/2015, 5:34:42 pm	1	5.34	null	null	-73.9212265	40.83566666	-73.89456177	
2	6/6/2015, 5:07:05 pm	6/6/2015, 5:35:27 pm	1	5.05	null	null	-73.94515991	40.80804825	-73.98769379	
2	6/6/2015, 5:18:48 pm	6/6/2015, 5:35:05 pm	1	2.45	null	null	-73.93893433	40.80519485	-73.96521759	
2	6/6/2015, 5:14:19 pm	6/6/2015, 5:34:41 pm	2	4.71	null	null	-73.80210876	40.70185089	-73.72146606	
2	6/6/2015, 5:24:49 pm	6/6/2015, 5:35:16 pm	1	3.4	null	null	-73.85350037	40.84237289	-73.82595062	
2	6/6/2015, 5:28:20 pm	6/6/2015, 5:32:45 pm	1	0.63	null	null	-73.95275879	40.81062317	-73.95684814	
2	6/6/2015, 5:25:05 pm	6/6/2015, 5:33:36 pm	1	1.26	null	null	-73.89109802	40.74683762	-73.88323212	

9. Once your canvas is populated with the data, you can set **column profile** information, as this is useful for data profiling. You can apply the right transformation and target the right data values based on it.
10. To do this, select **Options** from the ribbon pane, then select the first three options under **Column profile**, and then select **OK**.





Options

Global

- General
- Data load
- Diagnostics

Dataflow

- Data load
- Regional settings
- Privacy
- Scale
- Parameters

Default editor view

● Data view

○ Schema view

Steps

☑ Enable query folding indicators

□ Show script in step callout

□ Disable confirmation dialogs when inserting intermediate steps

Column profile

☑ Enable column profile

☑ Show column quality details in data preview

☑ Show column value distribution in data preview

□ Show column profile in details pane

Column profile evaluation

● Based on top 1,000 rows

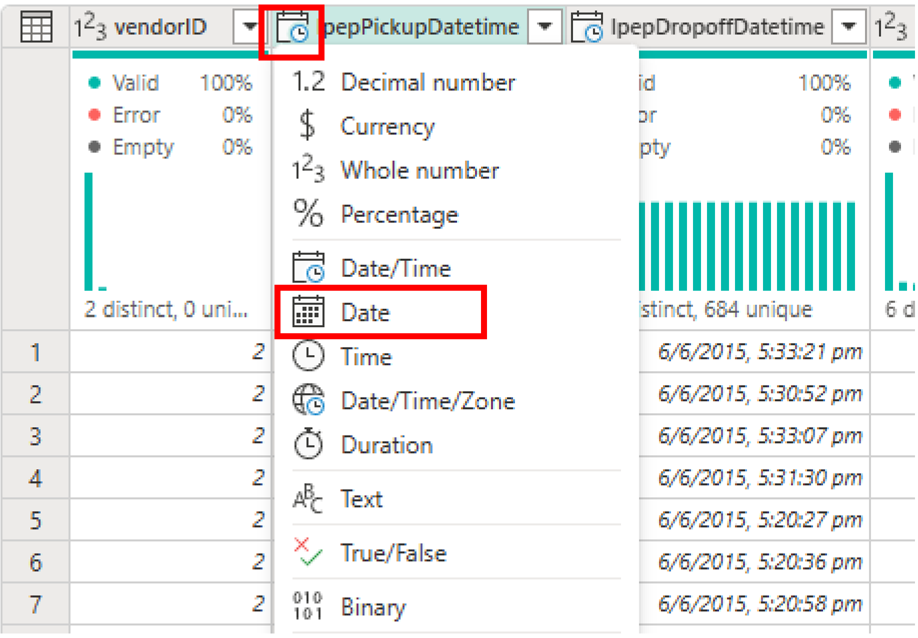
○ Based on entire data set

Data view

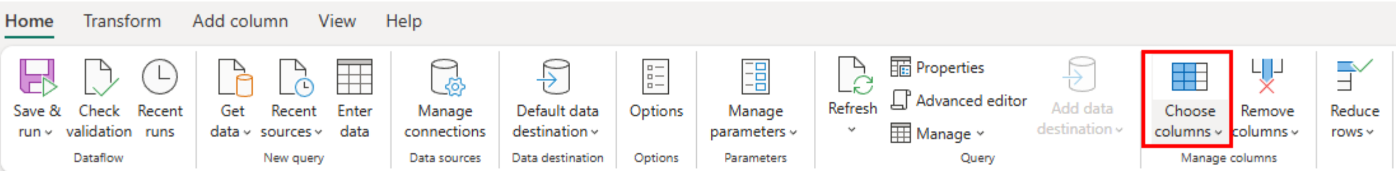
☑ Enable details pane

Transform the data imported from the Lakehouse

1. Select the data type icon in the column header of the second column, **IpepPickupDatetime**, convert the column from the **Date/Time** to **Date** type.

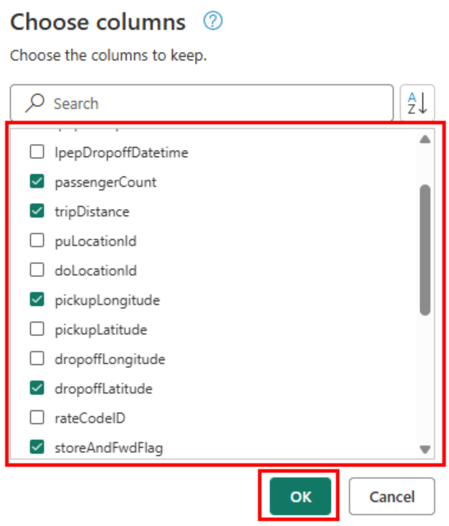


2. On the **Home** tab of the ribbon, select the **Choose columns** option from the **Manage columns** group.



On the **Choose columns** dialog, **deselect** some columns listed here, then select **OK**.

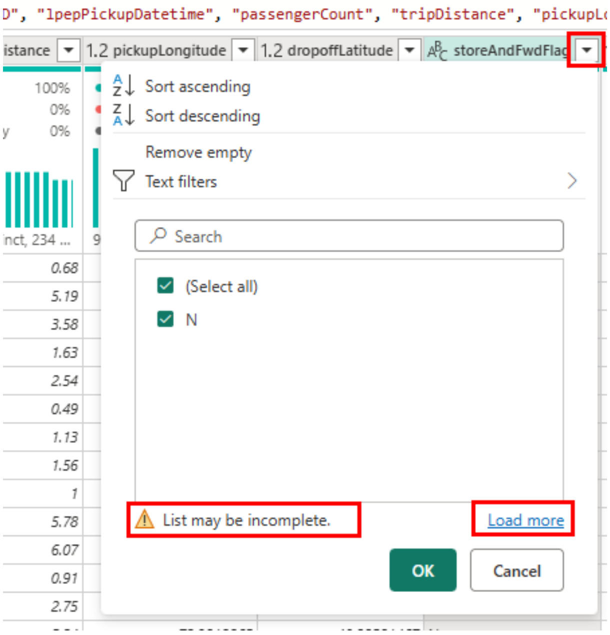
IpepDropoffDatetime
puLocationId
doLocationId
pickupLatitude
dropoffLongitude
rateCodeID


Choose columns

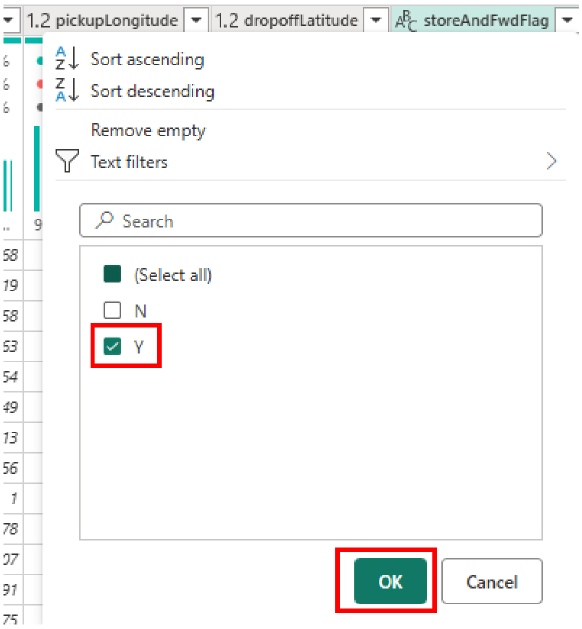
Choose the columns to keep.

- □ IpepDropoffDatetime
- ☑ passengerCount
- ☑ tripDistance
- □ puLocationId
- □ doLocationId
- ☑ pickupLongitude
- □ pickupLatitude
- □ dropoffLongitude
- ☑ dropoffLatitude
- □ rateCodeID
- ☑ storeAndFwdFlag

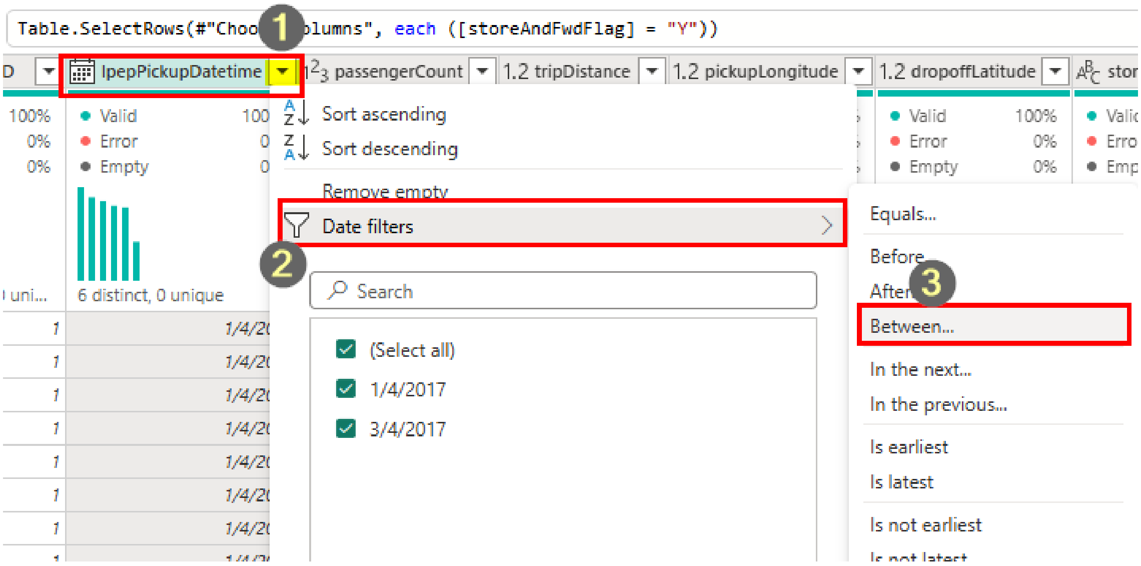
3. Select the **storeAndFwdFlag** column's filter and sort dropdown menu. (If you see a warning **List may be incomplete**, select **Load more** to see all the data.)
4. Select Y to show only rows where a discount was applied, and then select OK.



```
D", "lpepPickupDatetime", "passengerCount", "tripDistance", "pickupL
```

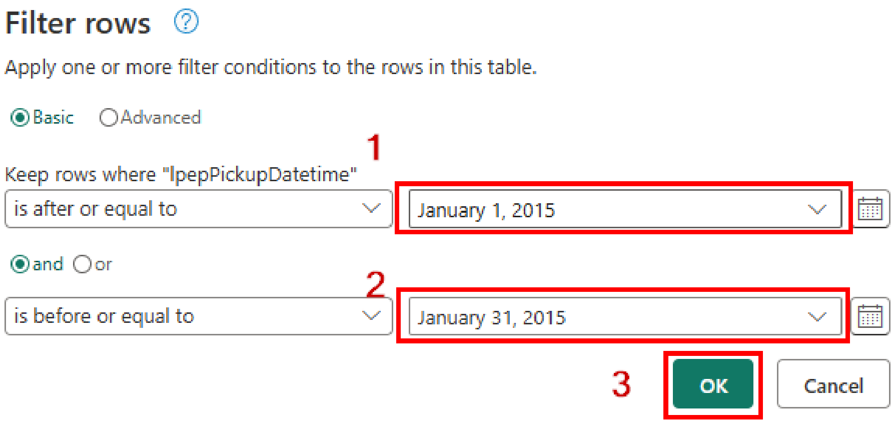


5. Select the **Ipep_Pickup_Datetime** column sort and filter dropdown menu, then select **Date filters**, and choose the **Between...** filter provided for Date and Date/Time types.



```
Table.SelectRows(#"Choo[1]olumns", each ([storeAndFwdFlag] = "Y"))
```

6. In the **Filter rows** dialog, select dates between **January 1, 2015**, and **January 31, 2015**, then select **OK**.



Filter rows

Apply one or more filter conditions to the rows in this table.

● Basic ○ Advanced

Keep rows where "lpepPickupDatetime"

is after or equal to | January 1, 2015

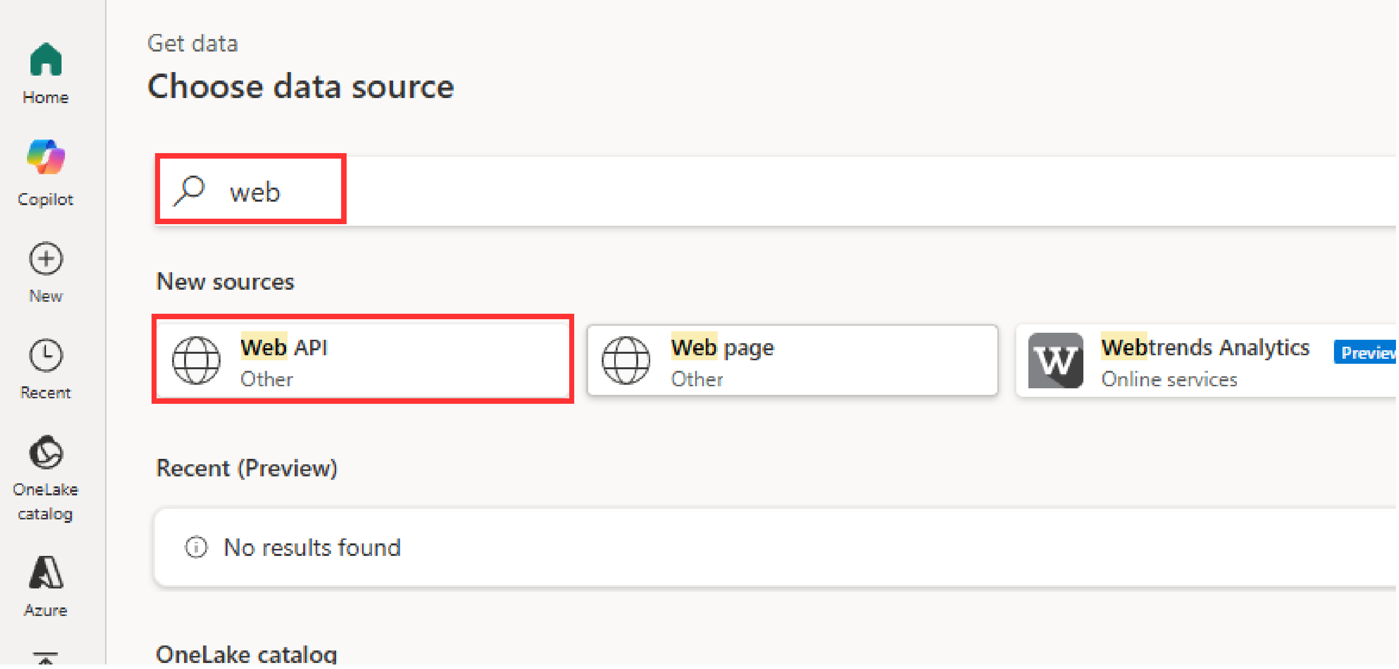
● and ○ or

is before or equal to | January 31, 2015

Connect to a CSV file containing discount data

Now, with the data from the trips in place, we want to load the data that contains the respective discounts for each day and VendorID, and prepare the data before combining it with the trips data.

1. From the **Home** tab in the dataflow editor menu, select the **Get data** option, and then choose **more…**. On **Choose data source** search **Web API,** select **Web API**.

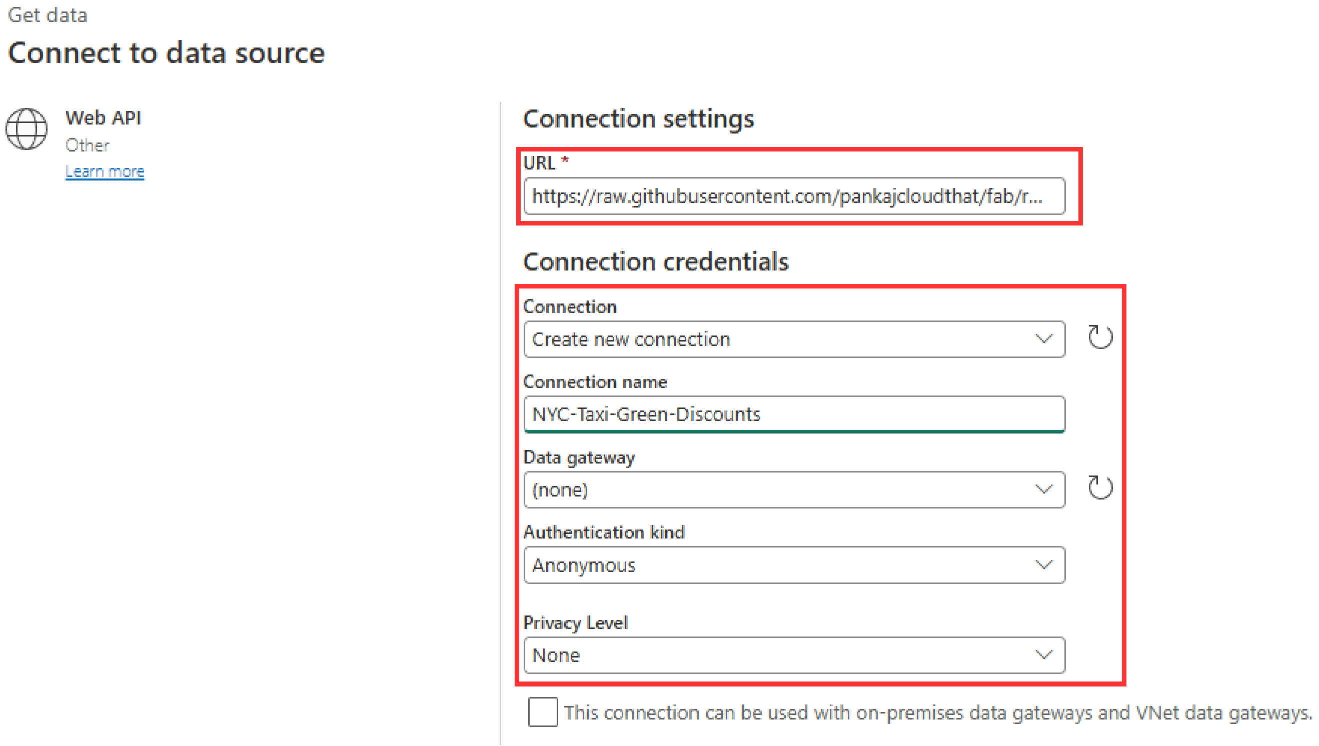


Get data

Choose data source

web

2. In the **Connect to data source** pane, under **Connection settings**, enter **URL:** *https://raw.githubusercontent.com/pankajcloudthat/fab/refs/heads/main/NYC-Taxi-Green-Discounts.csv*
then click on **Next** button.



Get data

Connect to data source

Connection settings

URL *
https://raw.githubusercontent.com/pankajcloudthat/fab/r...

Connection credentials

Connection
Create new connection

Connection name
NYC-Taxi-Green-Discounts

Data gateway
(none)

Authentication kind
Anonymous

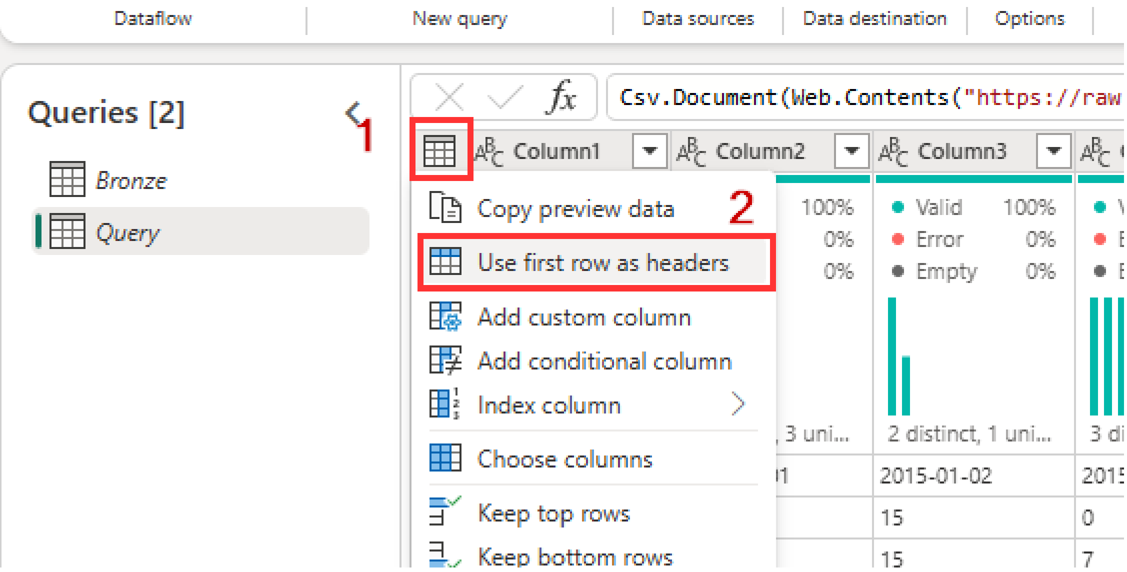
Privacy Level
None

☐ This connection can be used with on-premises data gateways and VNet data gateways.

3. On the **Preview file data** dialog, select **Create**.

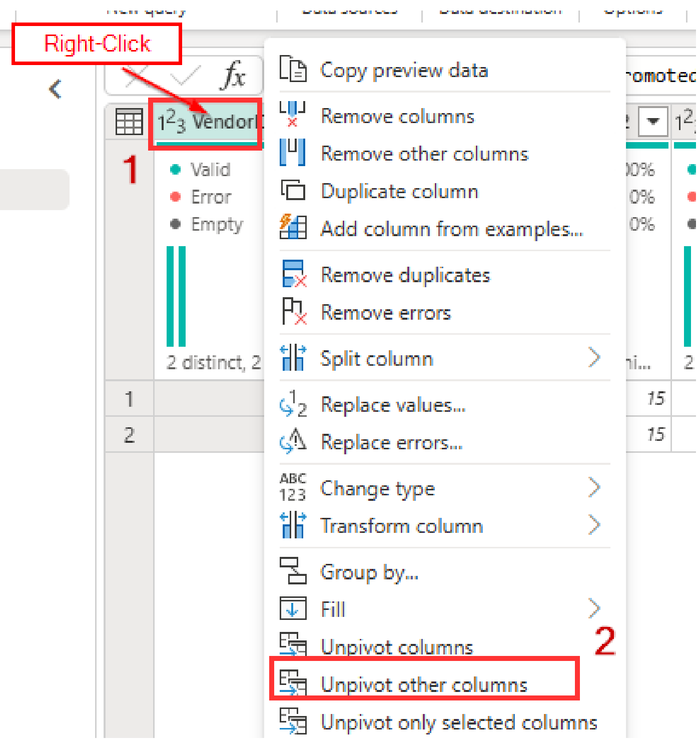
Transform the discount data

1. Reviewing the data, we see the headers appear to be in the first row. Promote them to headers by selecting the table\'s context menu at the top left of the preview grid area to select **Use first row as headers**.

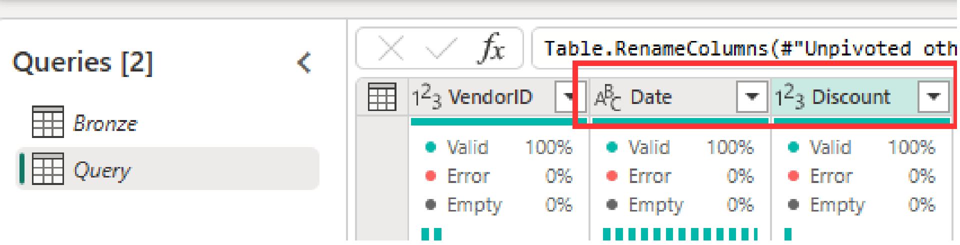


```
Csv.Document(Web.Contents("https://raw
```

2. Right-click the **VendorID** column, and from the context menu displayed, select the option **Unpivot other columns**. This allows you to transform columns into attribute-value pairs, where columns become rows.

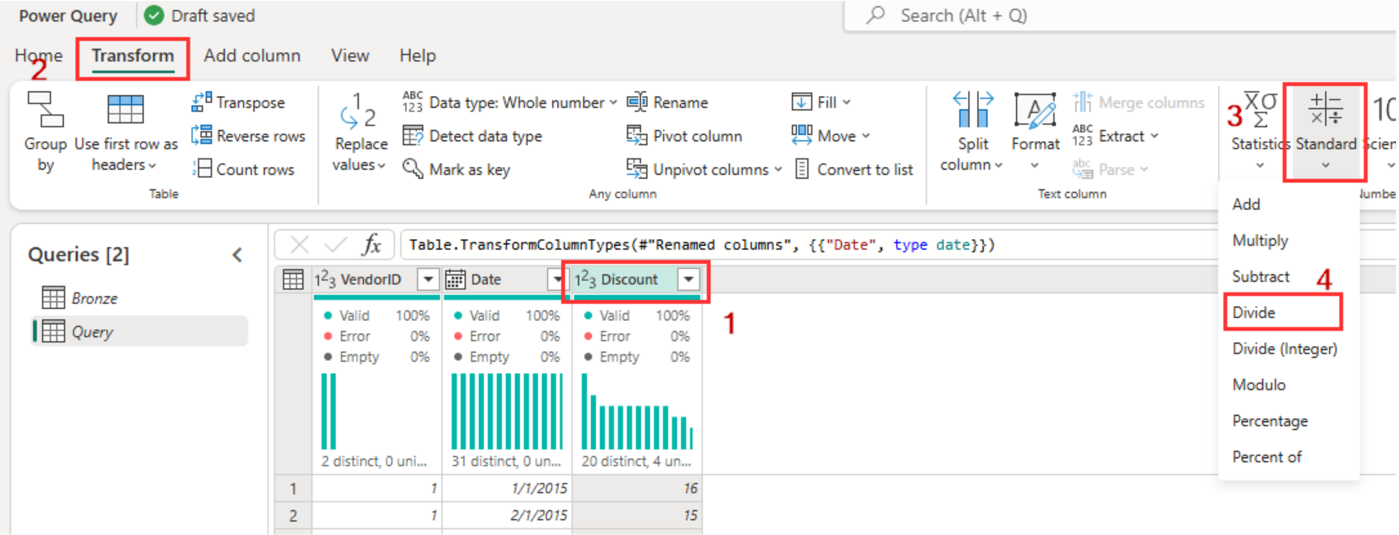


3. With the table unpivoted, rename the **Attribute** and **Value** columns by double-clicking them and changing **Attribute** to **Date** and **Value** to **Discount**.



```
Table.RenameColumns(#"Unpivoted oth
```

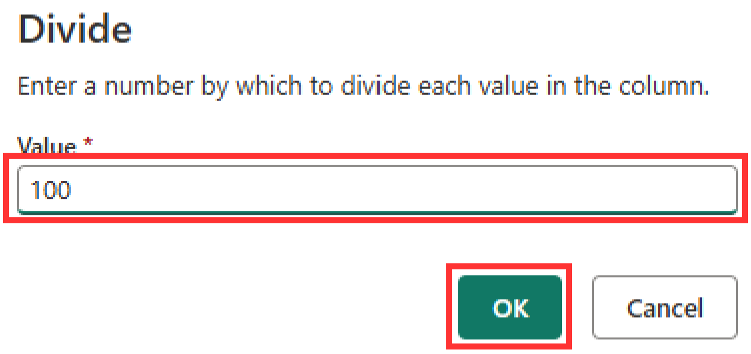
4. Change the data type of the Date column by selecting the data type menu to the left of the column name and choosing **Date**.
5. Select the **Discount** column and then select the **Transform** tab on the menu. Select **Number column**, and then select **Standard** numeric transformations from the submenu, and choose **Divide**.



```
Table.TransformColumnTypes(#"Renamed columns", {{"Date", type date}})
```

	VendorID	Date	Discount
1	1	1/1/2015	16
2	1	2/1/2015	15

6. On the **Divide** dialog, enter the value 100, then click on **OK** button.



Divide

Enter a number by which to divide each value in the column.

Value *

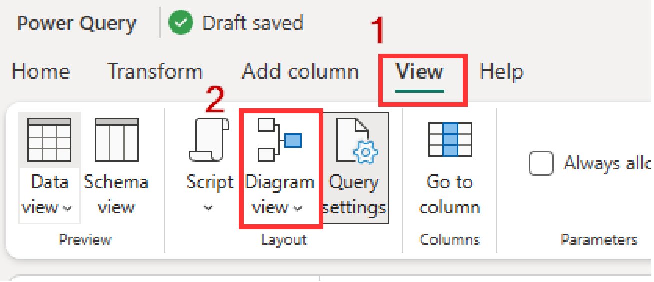
100

OK   Cancel

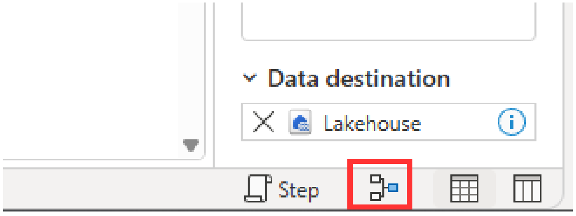
Combine trips and discounts data

The next step is to combine both tables into a single table that has the discount that should be applied to the trip, and the adjusted total.

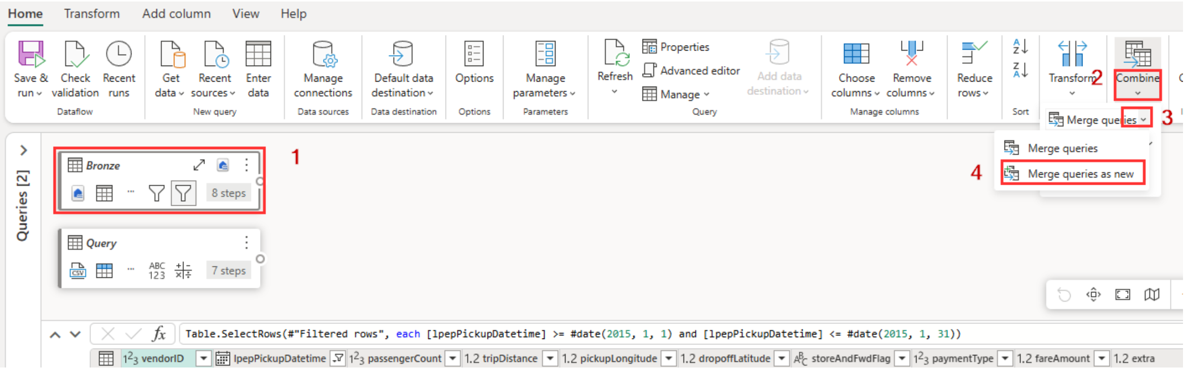
1. First, toggle the Diagram view button so you can see both of your queries.



or

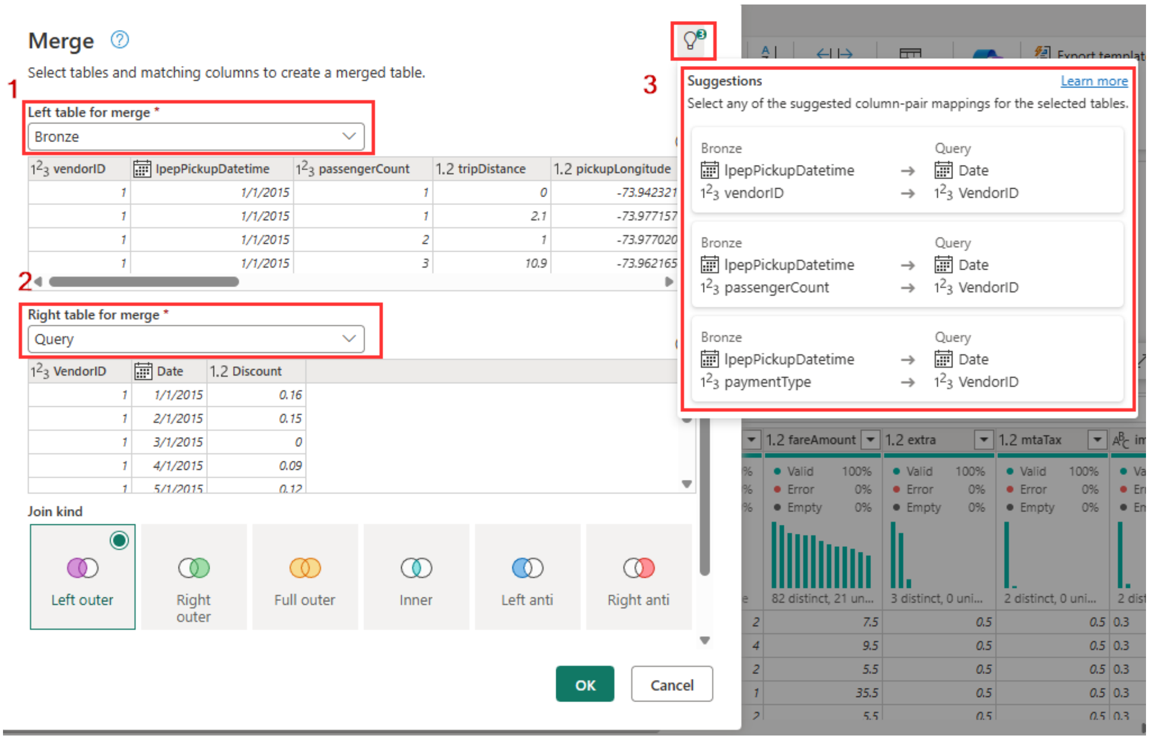


2. Select the **Bronze** query, and on the **Home** tab, Select the **Combine** menu and choose **Merge queries**, then **Merge queries as new**.



```
Table.SelectRows(#"Filtered rows", each [lpepPickupDatetime] >= #date(2015, 1, 1) and [lpepPickupDatetime] <= #date(2015, 1, 31))
```

3. On the **Merge** dialog, select **Query** from the **Right table for merge** drop down, and then select the \"**light bulb**\" icon on the top right of the dialog to see the suggested mapping of columns between the three tables.



Select tables and matching columns to create a merged table.

1²3 vendorID	lpepPickupDatetime	1²3 passengerCount	1.2 tripDistance	1.2 pickupLongitude
1	1/1/2015	1	0	-73.942321
1	1/1/2015	1	2.1	-73.977157
1	1/1/2015	2	1	-73.977020
1	1/1/2015	3	10.9	-73.962165

1²3 VendorID	Date	1.2 Discount
1	1/1/2015	0.16
1	2/1/2015	0.15
1	3/1/2015	0
1	4/1/2015	0.09
1	5/1/2015	0.12

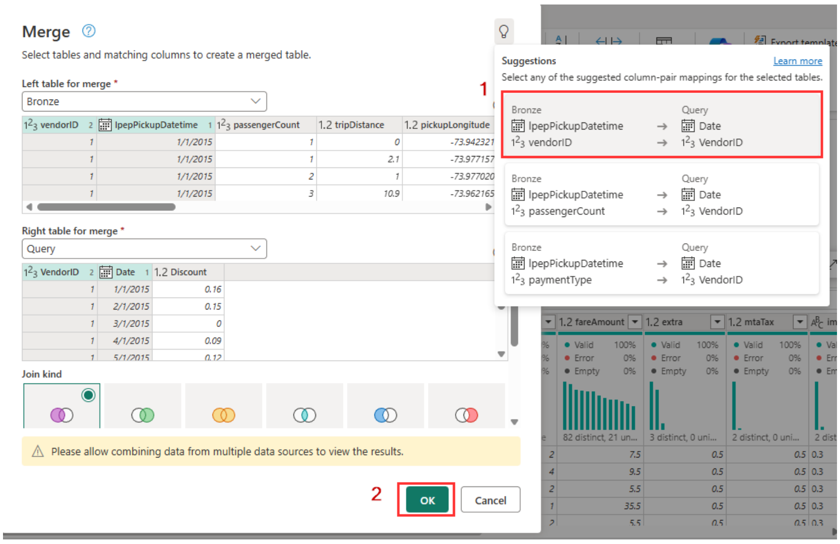
Suggestions
Learn more
Select any of the suggested column-pair mappings for the selected tables.

Bronze		Query
lpepPickupDatetime	→	Date
vendorID	→	VendorID

Bronze		Query
lpepPickupDatetime	→	Date
passengerCount	→	VendorID

Bronze		Query
lpepPickupDatetime	→	Date
paymentType	→	VendorID

4. Choose each of the two suggested column mappings, one at a time, mapping the VendorID and date columns from both tables. When both mappings are added, the matched column headers are highlighted in each table.
5. A message is shown asking you to allow combining data from multiple data sources to view the results. Select OK

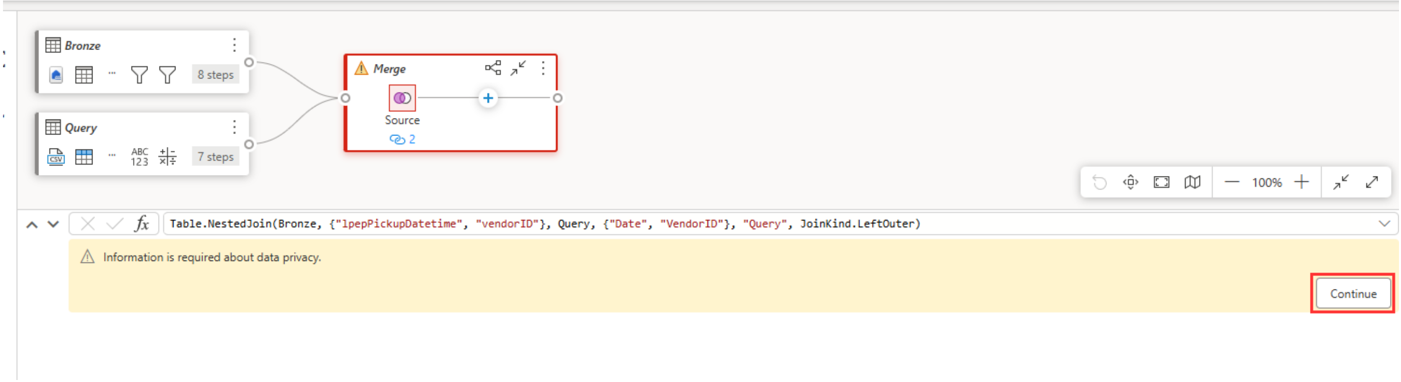


Select tables and matching columns to create a merged table.

Suggestions
Learn more
Select any of the suggested column-pair mappings for the selected tables.

⚠ Please allow combining data from multiple data sources to view the results.

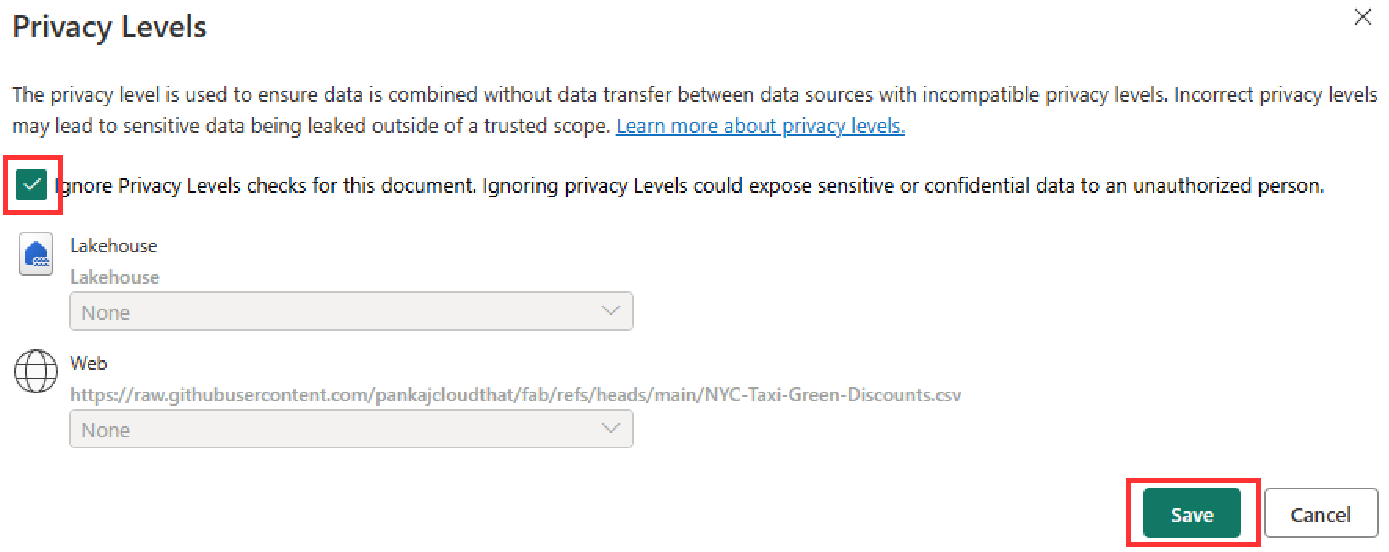
6. In the table area, you'll initially see a warning that "The evaluation was canceled because combining data from multiple sources may reveal data from one source to another. Select continue if the possibility of revealing data is okay." Select **Continue** to display the combined data.



```
Table.NestedJoin(Bronze, {"lpepPickupDatetime", "vendorID"}, Query, {"Date", "VendorID"}, "Query", JoinKind.LeftOuter)
```

⚠ Information is required about data privacy.

7. In Privacy Levels dialog box, select the **check box :Ignore Privacy Levels checks for this document. Ignoring privacy Levels could expose sensitive or confidential data to an unauthorized person** and click on the **Save** button.


Privacy Levels

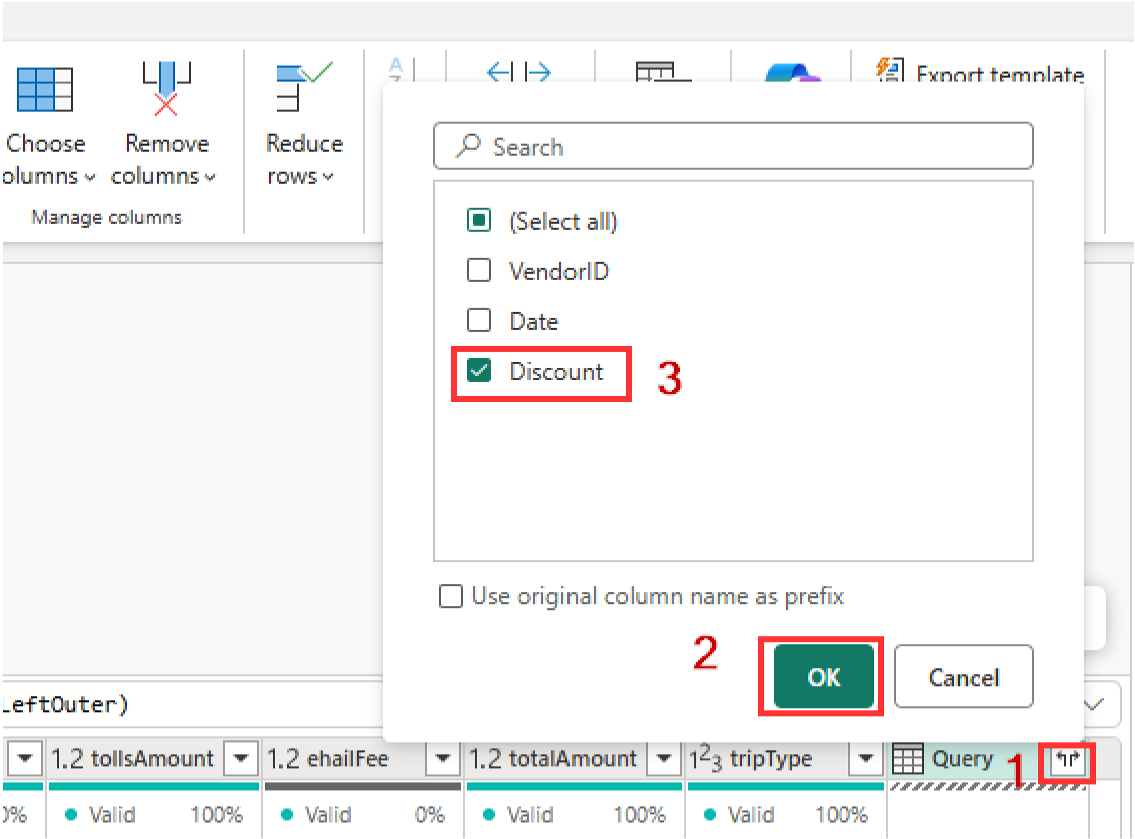
The privacy level is used to ensure data is combined without data transfer between data sources with incompatible privacy levels. Incorrect privacy levels may lead to sensitive data being leaked outside of a trusted scope. Learn more about privacy levels.

☑ Ignore Privacy Levels checks for this document. Ignoring privacy Levels could expose sensitive or confidential data to an unauthorized person.

Lakehouse
Lakehouse
None

Web
https://raw.githubusercontent.com/pankajcloudthat/fab/refs/heads/main/NYC-Taxi-Green-Discounts.csv
None

8. Notice how a new query was created in Diagram view showing the relationship of the new Merge query with the two queries you previously created. Looking at the table pane of the editor, scroll to the right of the Merge query column list to see a new column with table values is present. This is the "Generated NYC Taxi-Green-Discounts" column, and its type is **[Table]**. In the column header there\'s an icon with two arrows going in opposite directions, allowing you to select columns from the table. Deselect all of the columns except **Discount**, and then select **OK**.



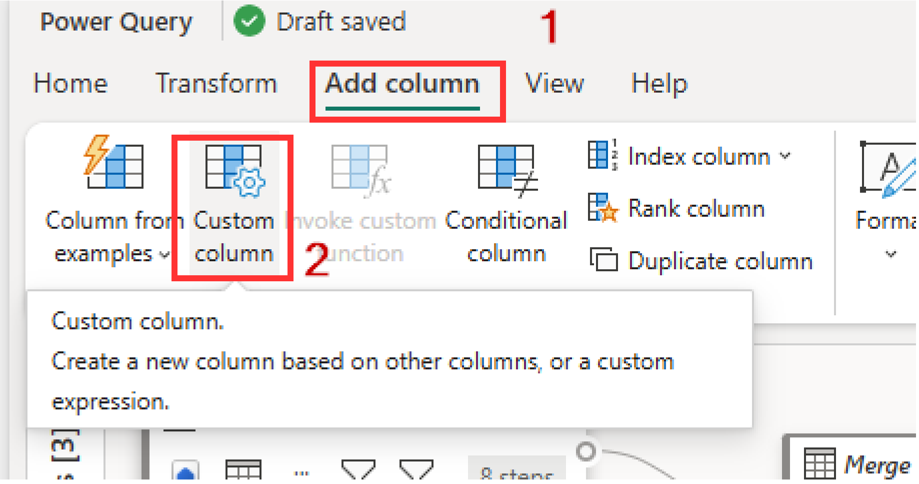
Search

☒ (Select all)
□ VendorID
□ Date
☑ Discount

□ Use original column name as prefix

OK Cancel

9. With the discount value now at the row level, we can create a new column to calculate the total amount after discount. To do so, select the **Add column** tab at the top of the editor, and choose **Custom column** from the **General** group.

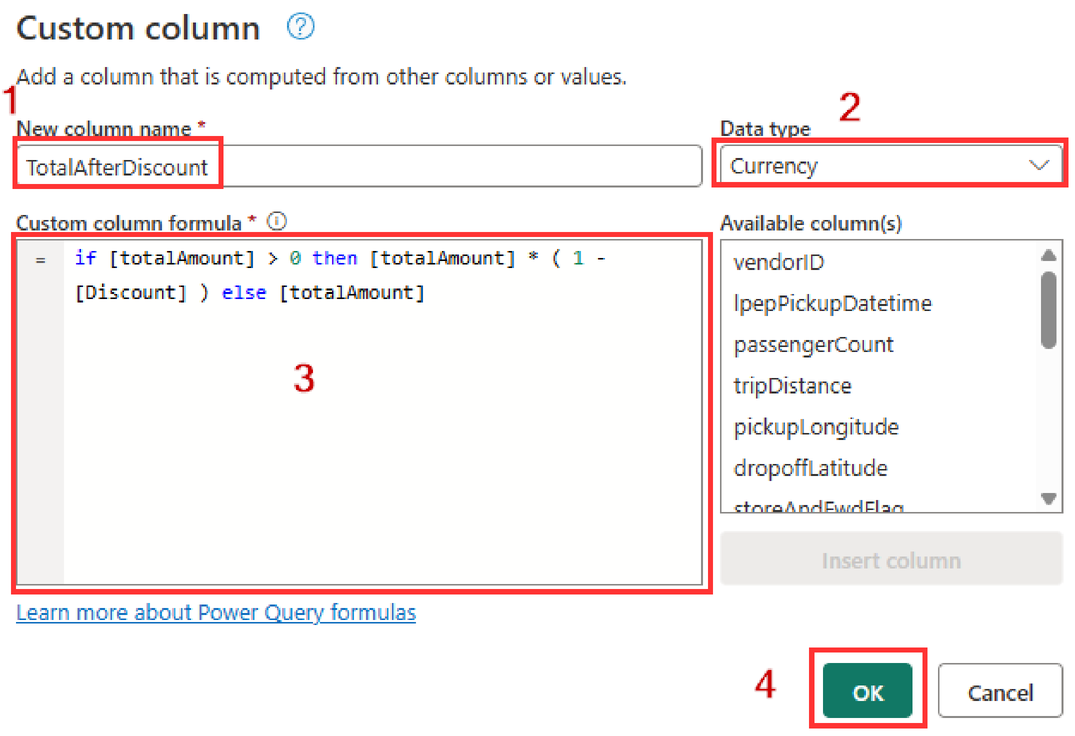


Custom column.
Create a new column based on other columns, or a custom expression.

10. On the **Custom column** dialog, you can use the [Power Query formula language (also known as M) to define how your new column should be calculated. Enter **TotalAfterDiscount** for the **New column name**, select **Currency** for the **Data type**, and provide the following M expression for the **Custom column formula**:

if [totalAmount] > 0 then [totalAmount] * (1 -[Discount]) else [totalAmount]

Then select OK.



Custom column

Add a column that is computed from other columns or values.

New column name *

TotalAfterDiscount

Data type

Currency

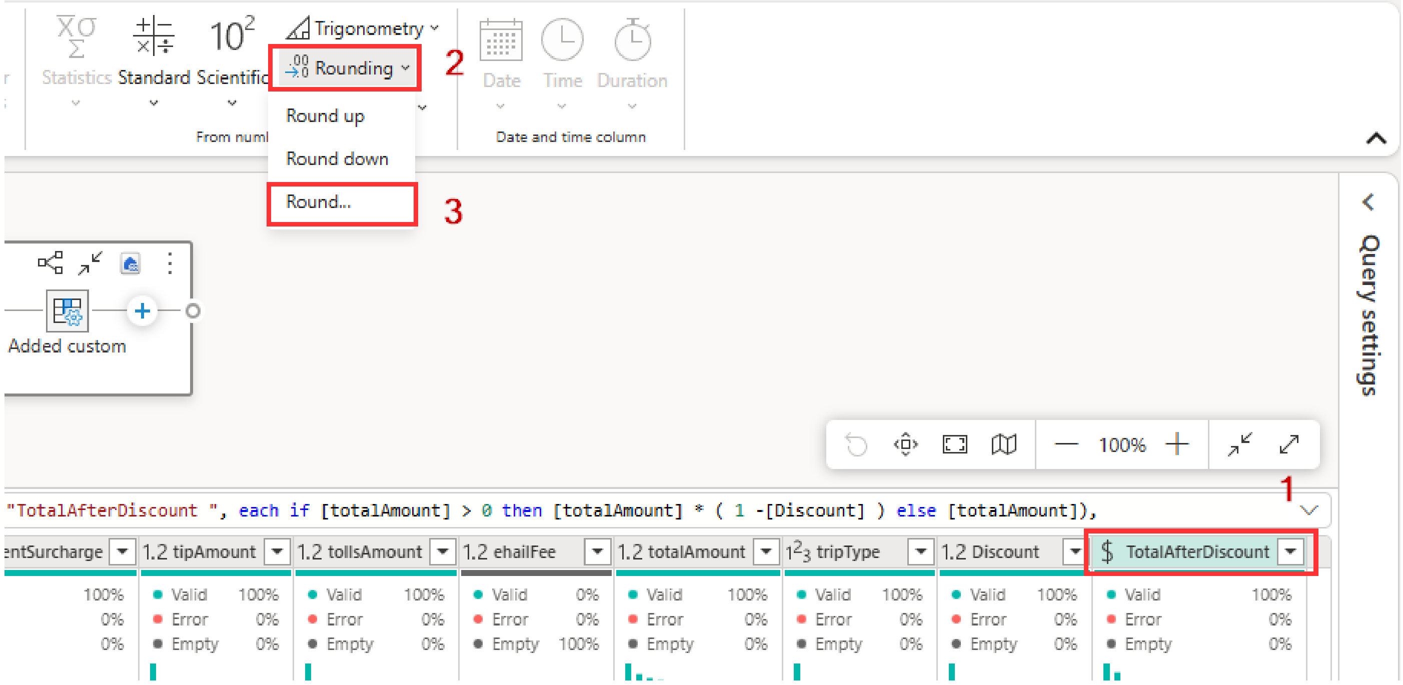
Custom column formula *

```
= if [totalAmount] > 0 then [totalAmount] * ( 1 -
  [Discount] ) else [totalAmount]
```

Available column(s)

vendorID
lpepPickupDatetime
passengerCount
tripDistance
pickupLongitude
dropoffLatitude
storeAndFwdFlag

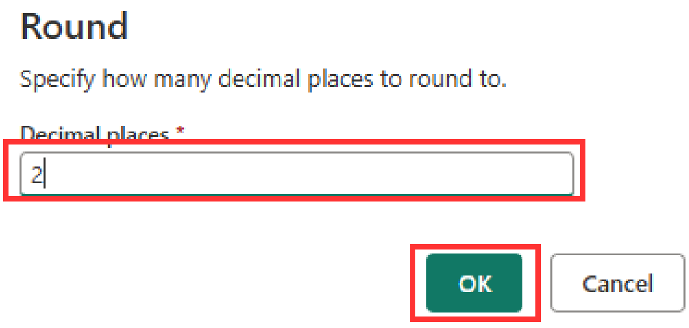
11. Select the newly create **TotalAfterDiscount** column and then select the **Transform** tab at the top of the editor window. On the **Number column** group, select the **Rounding** drop down and then choose **Round....**



```
"TotalAfterDiscount ", each if [totalAmount] > 0 then [totalAmount] * ( 1 -[Discount] ) else [totalAmount]),
```

entSurcharge	1.2 tipAmount	1.2 tollsAmount	1.2 ehailFee	1.2 totalAmount	1²3 tripType	1.2 Discount	$ TotalAfterDiscount
100%	● Valid 100%	● Valid 100%	● Valid 0%	● Valid 100%	● Valid 100%	● Valid 100%	● Valid 100%
0%	● Error 0%	● Error 0%	● Error 0%	● Error 0%	● Error 0%	● Error 0%	● Error 0%
0%	● Empty 0%	● Empty 0%	● Empty 100%	● Empty 0%	● Empty 0%	● Empty 0%	● Empty 0%

12. On the **Round** dialog, enter **2** for the number of decimal places and then select **OK**.



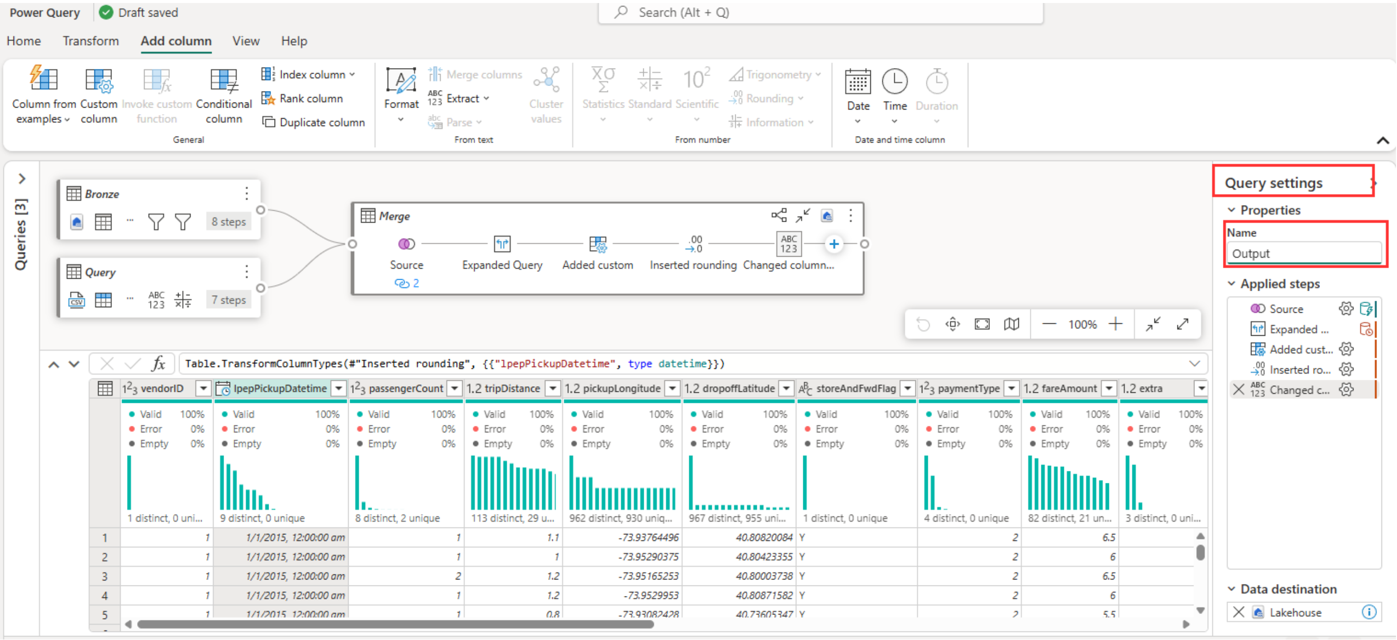
Round

Specify how many decimal places to round to.

Decimal places *

2

13. Change the data type of the **IpepPickupDatetime** from **Date** to **Date/Time**.
14. Finally, expand the **Query settings** pane from the right side of the editor if it isn't already expanded, and rename the query from **Merge** to **Output**.



```
Table.TransformColumnTypes(#"Inserted rounding", {{"lpepPickupDatetime", type datetime}})
```

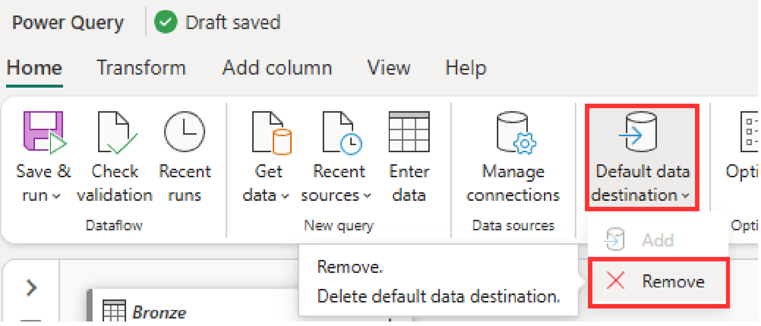
	vendorID	IpepPickupDatetime	passengerCount	tripDistance	pickupLongitude	dropoffLatitude	storeAndFwdFlag	paymentType	fareAmount	extra
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
	1 distinct, 0 uni...	9 distinct, 0 unique	8 distinct, 2 unique	113 distinct, 29 u...	962 distinct, 930 uniq...	967 distinct, 955 uni...	1 distinct, 0 unique	4 distinct, 0 unique	82 distinct, 21 un...	3 distinct, 0 uni...
1	1	1/1/2015, 12:00:00 am	1	1.1	-73.93764496	40.80820084	Y	2	6.5	
2	1	1/1/2015, 12:00:00 am	1	1	-73.95290375	40.80423355	Y	2	6	
3	1	1/1/2015, 12:00:00 am	2	1.2	-73.95165253	40.80003738	Y	2	6.5	
4	1	1/1/2015, 12:00:00 am	1	1.2	-73.9529953	40.80871582	Y	2	6	
5	1	1/1/2015, 12:00:00 am	1	0.8	-73.93082428	40.73605347	Y	2	5.5	

Load the output query to a table in the Lakehouse

With the output query now fully prepared and with data ready to output, we can define the output destination for the query.

1. Select the **Output** merge query created previously. Then select the **Home** tab in the editor, and **Add data destination** from the **Query** grouping, to select a **Lakehouse** destination.

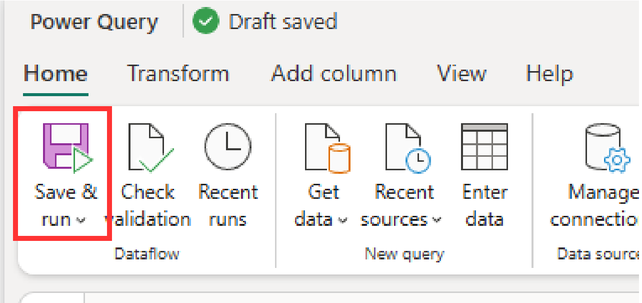
   Note: if created Dataflow Gen2 from lakehouse a default destination is already set to the lakehouse. **If you want to change the destination, then delete the existing and add a new destination.**



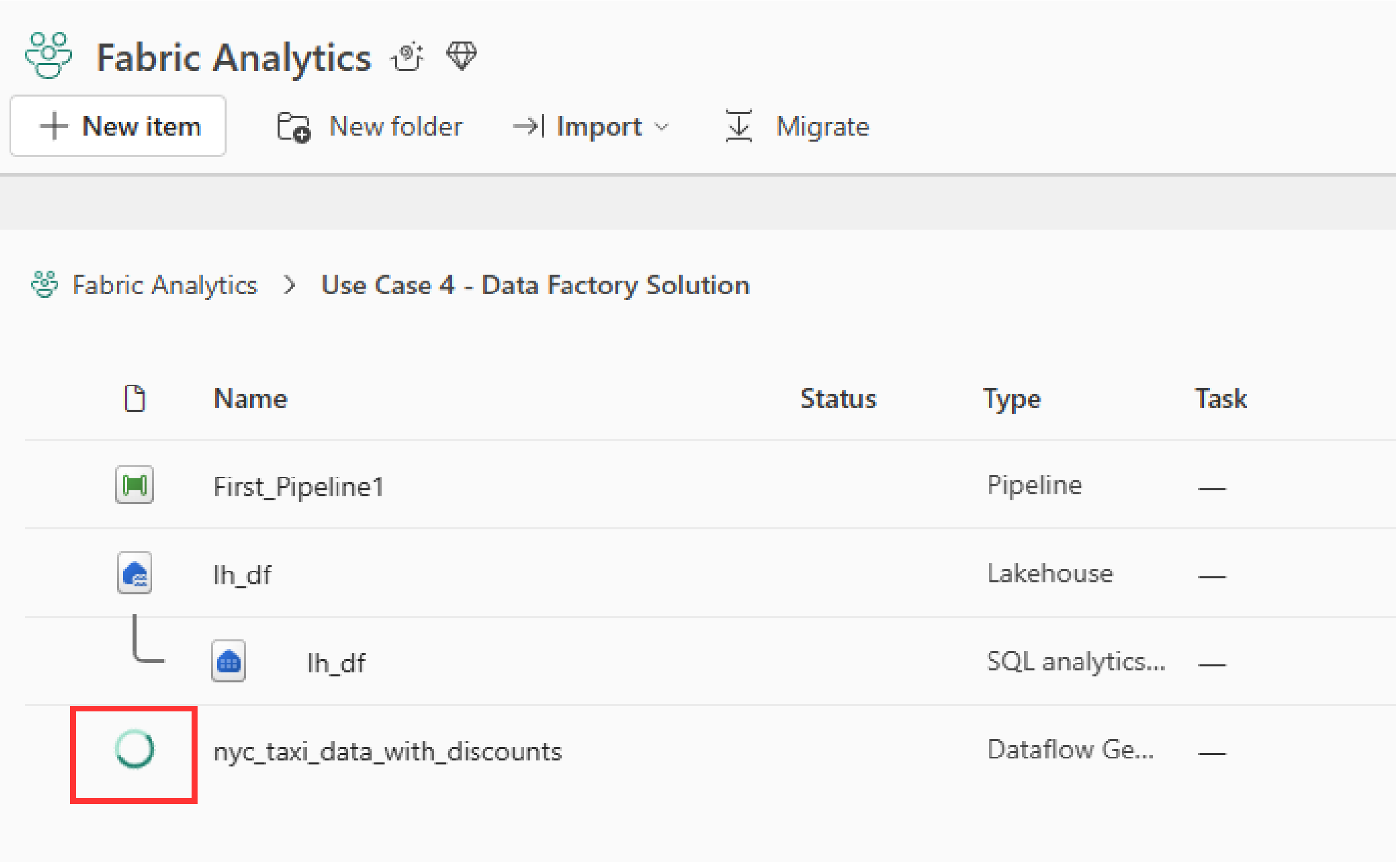
Remove.
Delete default data destination.

   No need to change the destination.

2. Select Save and Run in the Home tab.



3. On the workspace page, you will see the Dataflow Gen2 refreshing. When complete, you should see your new Lakehouse table created.



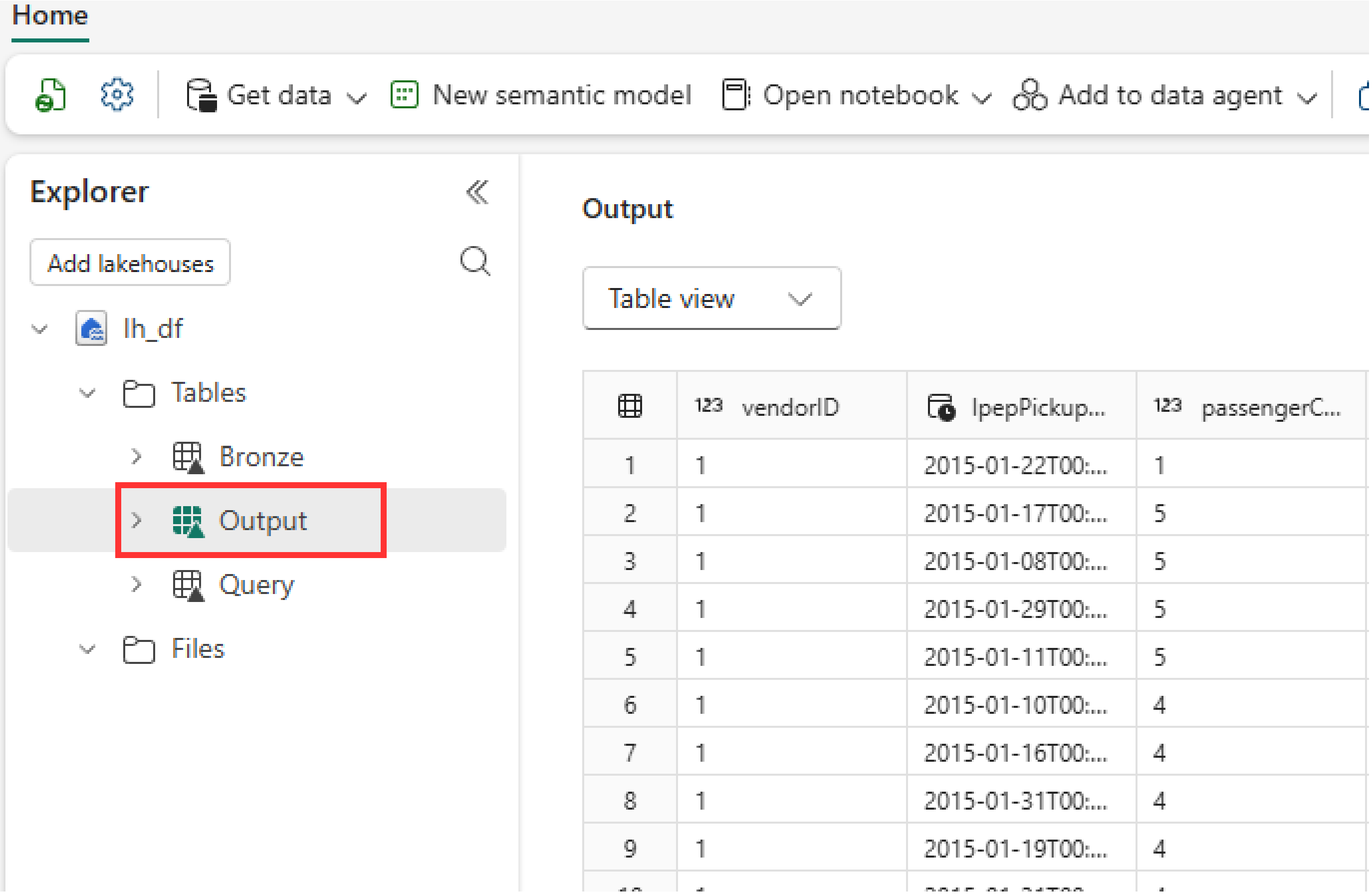
Fabric Analytics

＋ New item | New folder | Import | Migrate

Fabric Analytics > Use Case 4 - Data Factory Solution

	Name	Status	Type	Task
	First_Pipeline1		Pipeline	—
	lh_df		Lakehouse	—
	lh_df		SQL analytics...	—
	nyc_taxi_data_with_discounts		Dataflow Ge...	—

4. In the **Data_FactoryXX** pane, select **lh_df** to view the new table loaded there.



Output

Table view

	123 vendorID	lpepPickup...	123 passengerC...
1	1	2015-01-22T00:...	1
2	1	2015-01-17T00:...	5
3	1	2015-01-08T00:...	5
4	1	2015-01-29T00:...	5
5	1	2015-01-11T00:...	5
6	1	2015-01-10T00:...	4
7	1	2015-01-16T00:...	4
8	1	2015-01-31T00:...	4
9	1	2015-01-19T00:...	4

(Optional Exercise) Automate and send notifications with Data Factory

Add an Office 365 Outlook activity to your pipeline

1. From **Tutorial_Lakehouse** page, navigate and click on **Data_FactoryXX** Workspace on the left-sided navigation menu.
2. In the **Data_FactoryXX** view, select the **First_Pipeline1**.
3. Select the **Activities** tab in the pipeline editor and find the **Office Outlook** activity.
4. Select and drag the **On success** path (a green checkbox on the top right side of the activity in the pipeline canvas) from your **Copy activity** to your new **Office 365 Outlook** activity.
5. Select the Office 365 Outlook activity from the pipeline canvas, then select the **Settings** tab of the property area below the canvas to configure the email. Click on **Sing in** button.
6. Select your Power BI organizational account and then select **Allow access** to confirm.
   Note: The service doesn't currently support personal email. You must use an enterprise email address.
7. Select the Office 365 Outlook activity from the pipeline canvas, on the **Settings** tab of the property area below the canvas to configure the email. Enter your email address in the **To** section. If you want to use several addresses, use **;** to separate them.
8. For the **Subject**, select the field so that the **Add dynamic content** option appears, and then select it to display the pipeline expression builder canvas.
9. The **Pipeline expression builder** dialog appears. Enter the following expression, then select **OK**:

   @concat('DI in an Hour Pipeline Succeeded with Pipeline Run Id', pipeline().RunId)

10. For the **Body**, select the field again and choose the **View in expression builder** option when it appears below the text area. Add the following expression again in the **Pipeline expression builder** dialog that appears, then select **OK**:

    @concat('RunID = ', pipeline().RunId, ' ; ', 'Copied rows ', activity('Copy data1').output.rowsCopied, ' ; ','Throughput ', activity('Copy data1').output.throughput)

    Note: Replace **Copy data1** with the name of your own pipeline copy activity.

11. Finally select the **Home** tab at the top of the pipeline editor, and choose **Run**. Then select **Save and run** again on the confirmation dialog to execute these activities.
12. After the pipeline runs successfully, check your email to find the confirmation email sent from the pipeline.

Schedule pipeline execution

Once you finish developing and testing your pipeline, you can schedule it to execute automatically.

1. On the Home tab of the pipeline editor window, select Schedule.
2. Configure the schedule as required. The example here schedules the pipeline to execute daily at 8:00 PM until the end of the year.

Add a Dataflow activity to the pipeline

1. Hover over the green line connecting the Copy activity and the Office 365 Outlook activity on your pipeline canvas, and select the + button to insert a new activity.

2. Choose **Dataflow** from the menu that appears.
3. The newly created Dataflow activity is inserted between the Copy activity and the Office 365 Outlook activity, and selected automatically, showing its properties in the area below the canvas. Select the **Settings** tab on the properties area, and then select your dataflow created in **Exercise 2: Transform data with a dataflow in Data Factory**.
4. Select the **Home** tab at the top of the pipeline editor, and choose **Run**. Then select **Save and run** again on the confirmation dialog to execute these activities.