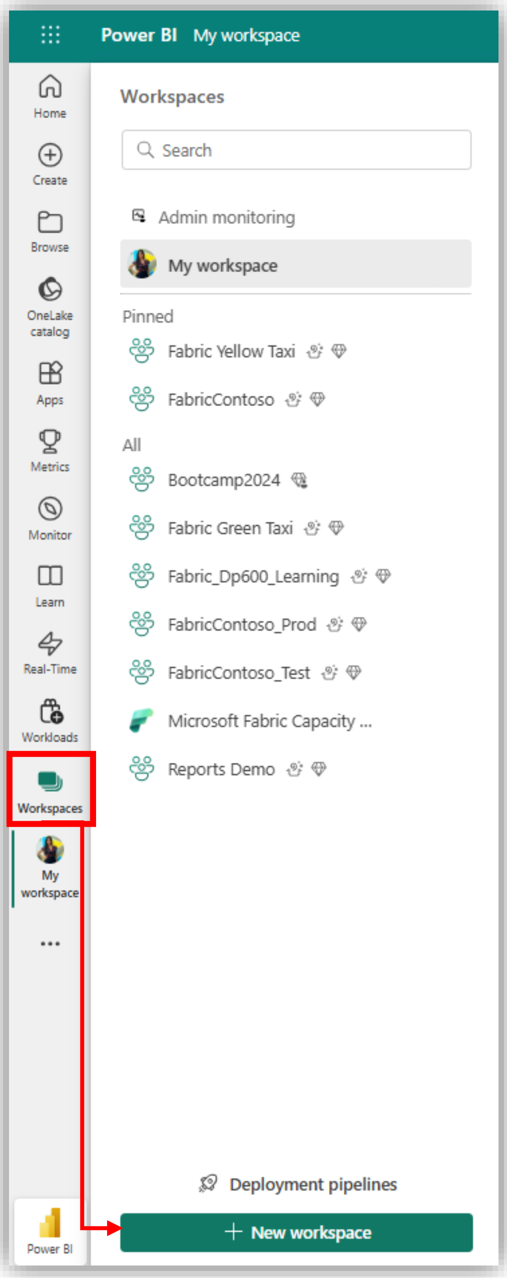
Lab Instructions for the Workshop

Pre-requisites for the Lab

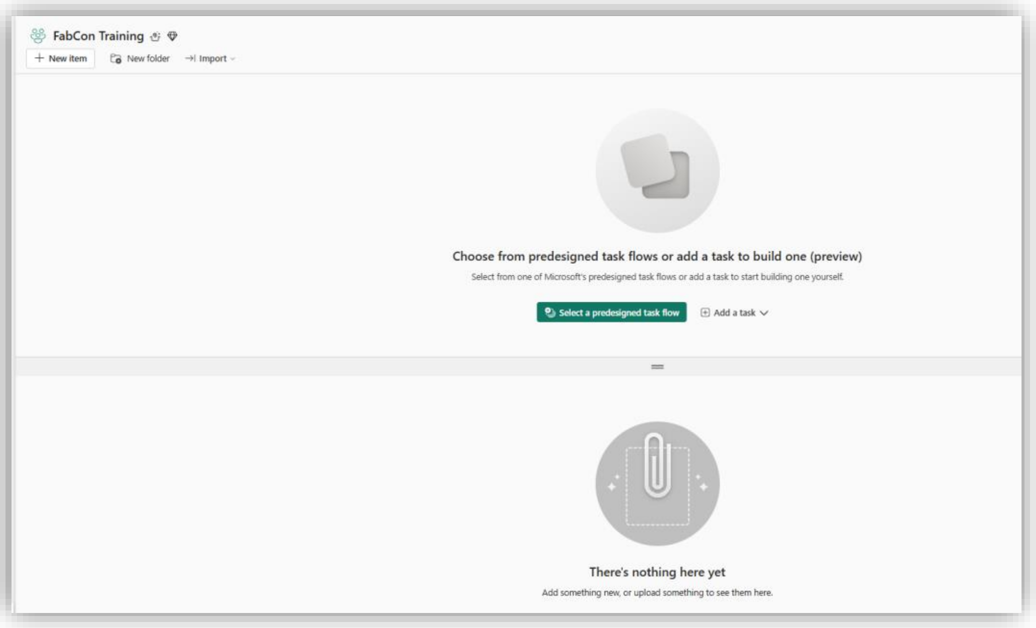
Make sure to sign-up for a Fabric Trial license before executing the instructions for the lab.

Section 1: Creating a Workspace

1. Navigate to **Workspaces** option in the left pane and then select **+ New Workspace**.



2. A new window opens on the right side of the page. Fill in the details:
    a. **Name**: give a unique name to your workspace
    b. Expand **Advanced** option and make sure **Trial** Fabric license is selected
    c. Hit **Apply.** A new workspace is created. In this case it is **Fabric Yellow Taxi**.
    d. Once the workspace is created, an empty workspace opens.



Choose from predesigned task flows or add a task to build one (preview)

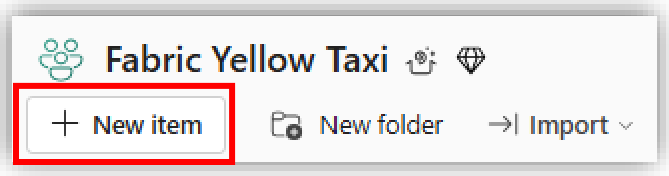
Select from one of Microsoft's predesigned task flows or add a task to start building one yourself.

There's nothing here yet

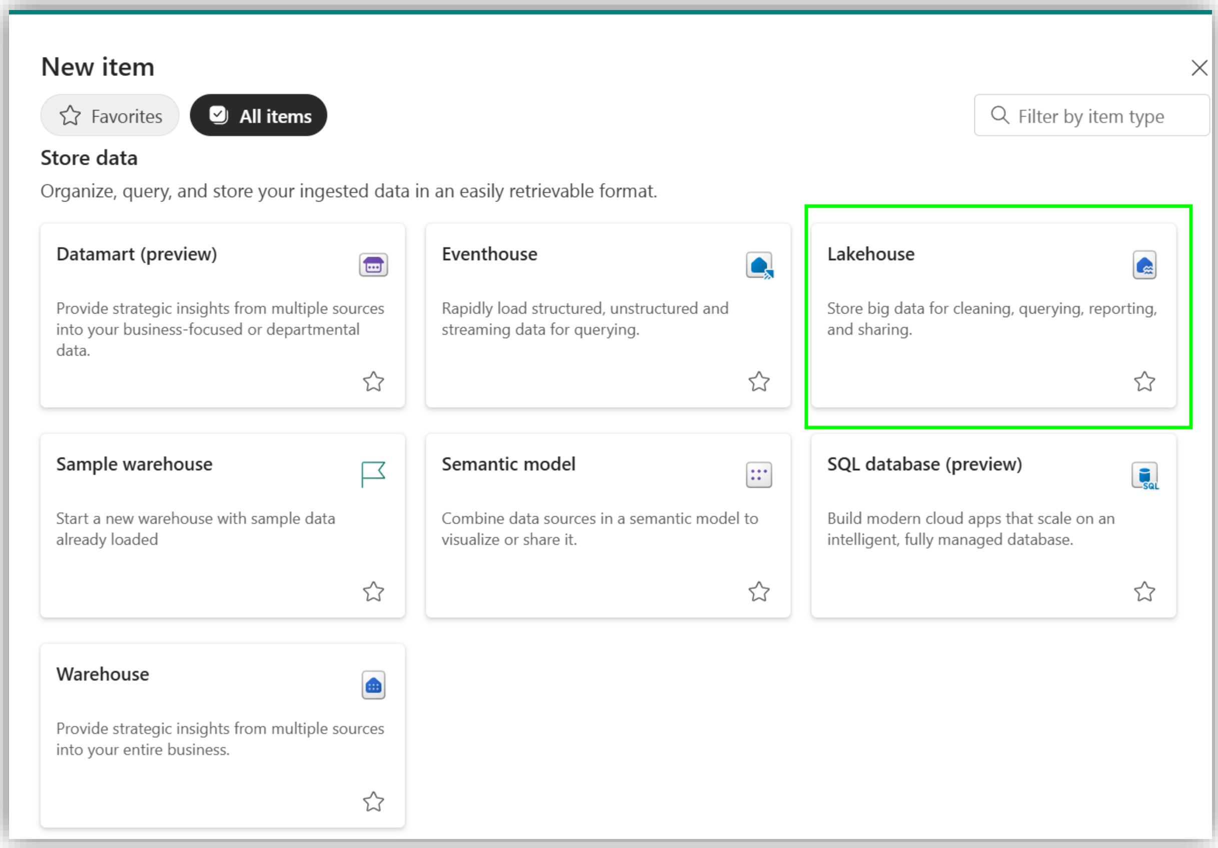
Add something new, or upload something to see them here.

Section 2: Creating a Lakehouse

1. Go to the workspace you created on the previous step
2. Click the button **+ New Item** on the top left corner of the navigation pane



3. Select the item **Lakehouse (Group Store Data)**:



Store data

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

Datamart (preview)

Provide strategic insights from multiple sources into your business-focused or departmental data.

Eventhouse

Rapidly load structured, unstructured and streaming data for querying.

Lakehouse

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

Sample warehouse

Start a new warehouse with sample data already loaded

Semantic model

Combine data sources in a semantic model to visualize or share it.

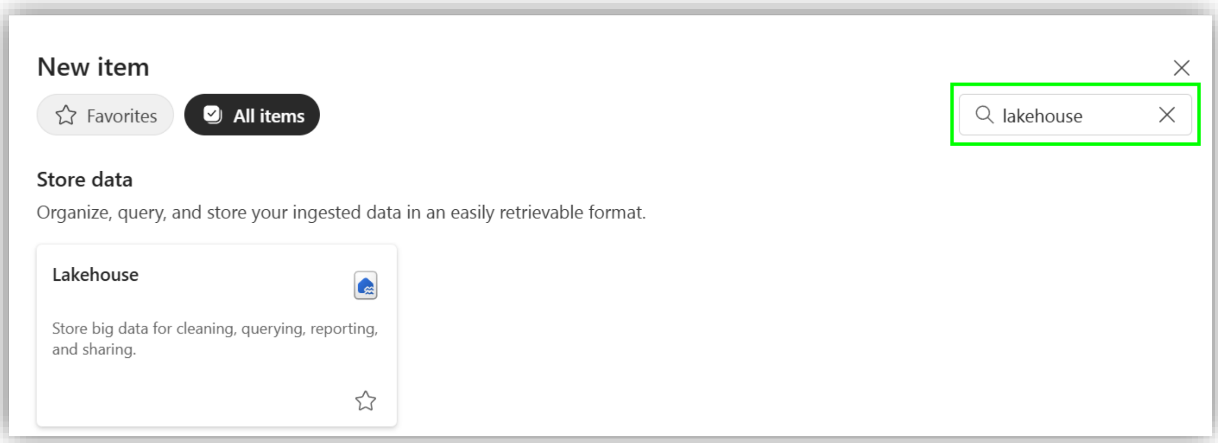
SQL database (preview)

Build modern cloud apps that scale on an intelligent, fully managed database.

Warehouse

Provide strategic insights from multiple sources into your entire business.

You can also search for the **Lakehouse** item in the search bar:



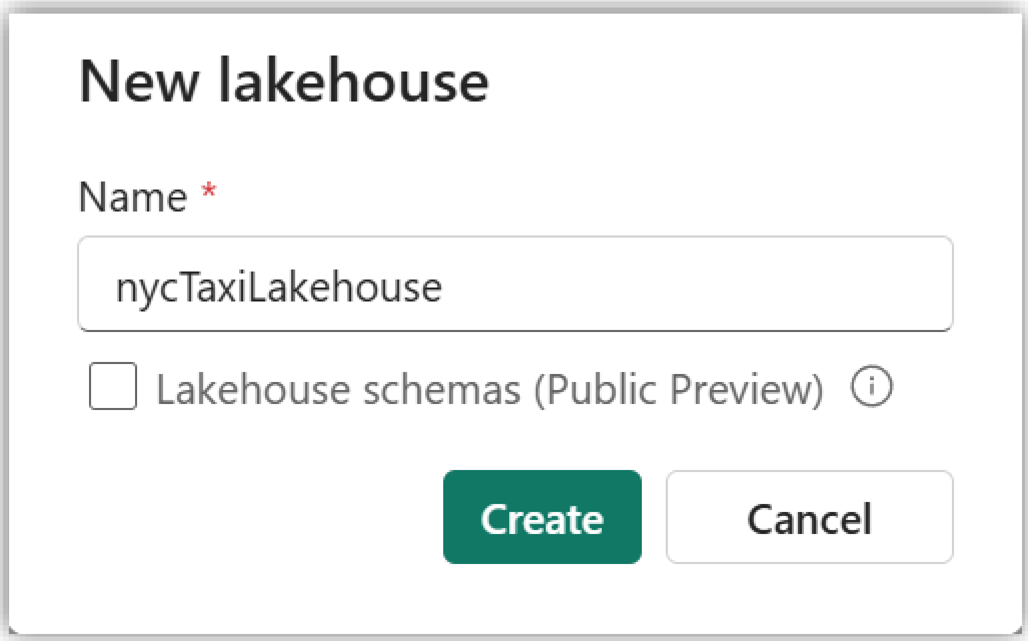
Store data

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

Lakehouse

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

4. In the dialog box name your Lakehouse: **nycTaxiLakehouse**
Information: Lakehouse names can only include numbers, letters and underscore (_)



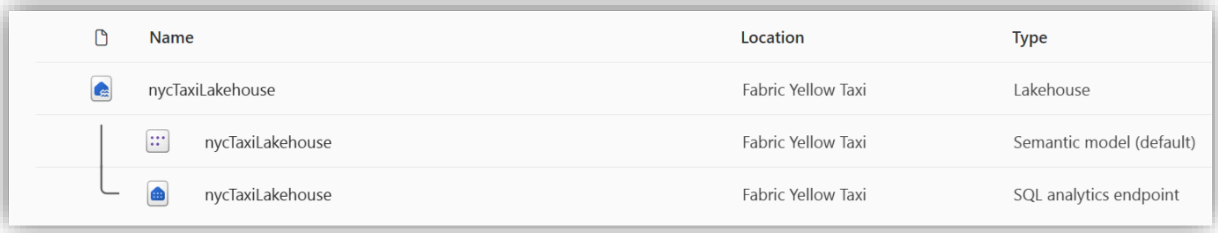
New lakehouse

Name *

nycTaxiLakehouse

□ Lakehouse schemas (Public Preview)

5. Click **Create**
6. The Lakehouse is created alongside with the SQL analytics endpoint and the Default Semantic Model

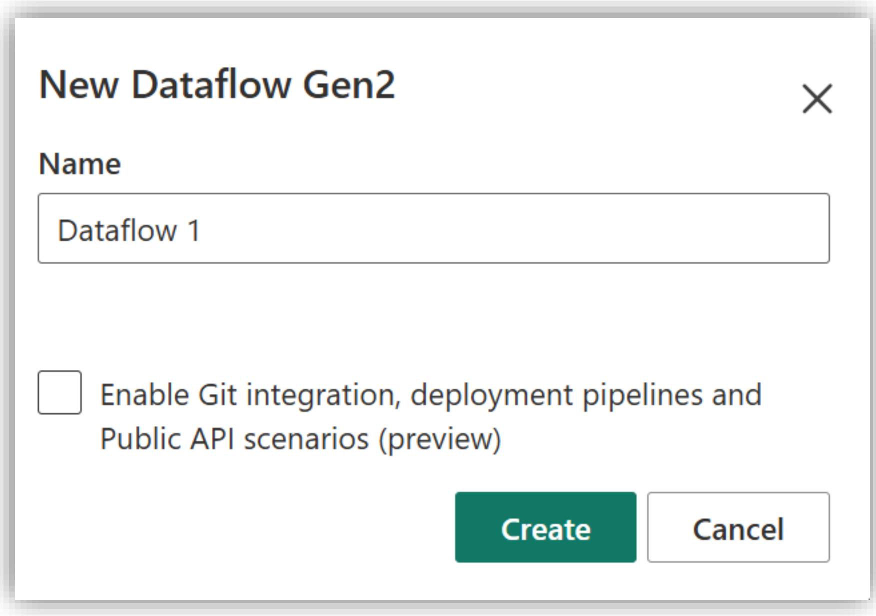


Name	Location	Type
nycTaxiLakehouse	Fabric Yellow Taxi	Lakehouse
nycTaxiLakehouse	Fabric Yellow Taxi	Semantic model (default)
nycTaxiLakehouse	Fabric Yellow Taxi	SQL analytics endpoint

Section 3: Data Ingestion

Dataflow Gen2

Create a Dataflow Gen2 in the workspace.



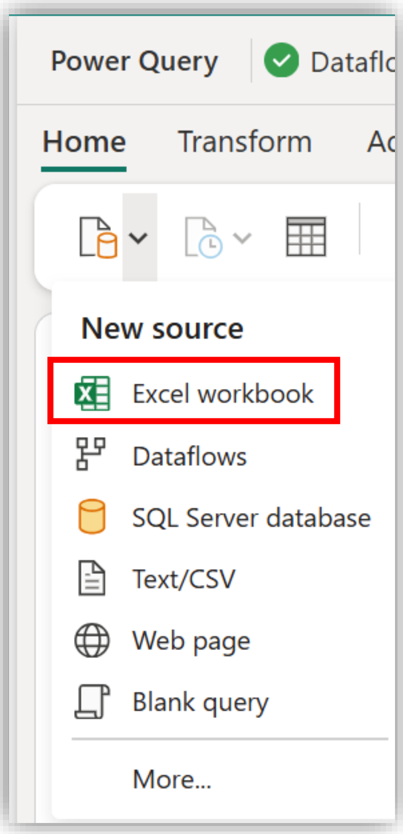
New Dataflow Gen2

Name

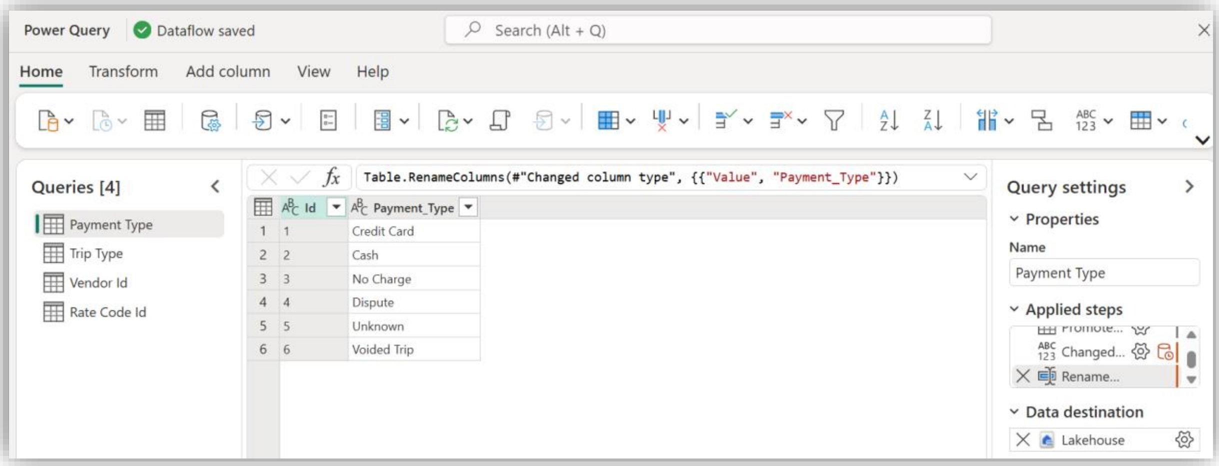
Dataflow 1

□ Enable Git integration, deployment pipelines and Public API scenarios (preview)

Connect to excel file for DIM tables as queries in DF Gen2



Once all the queries are connected, we have this in the DF Gen2



```
Table.RenameColumns(#"Changed column type", {{"Value", "Payment_Type"}})
```

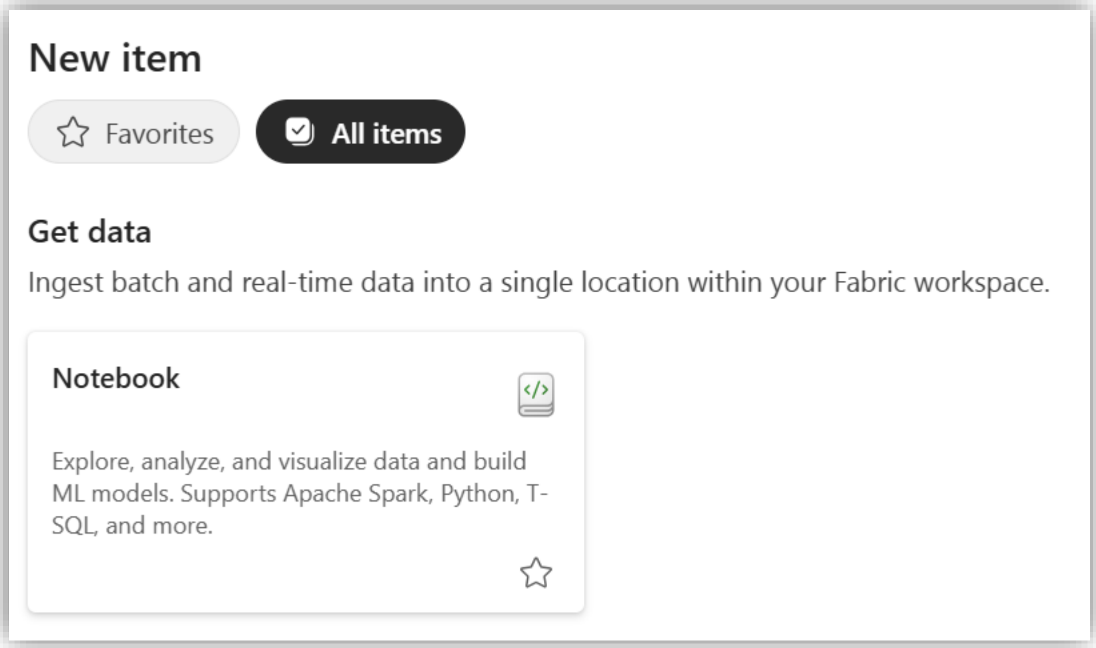
	Id	Payment_Type
1	1	Credit Card
2	2	Cash
3	3	No Charge
4	4	Dispute
5	5	Unknown
6	6	Voided Trip

Name: Payment Type

Dataflow Gen2 it is important to give a destinated where the data will land. In this case, we have chosen a Lakehouse as the destination.

Notebook

Create a new item in the workspace as a Notebook.



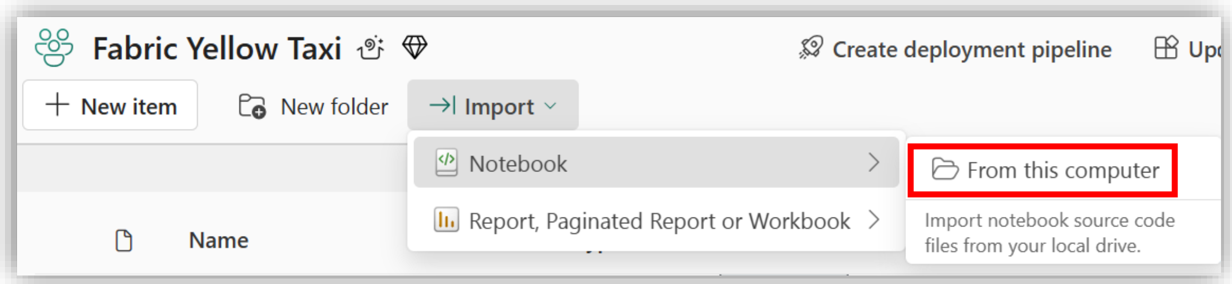
Get data

Ingest batch and real-time data into a single location within your Fabric workspace.

Notebook

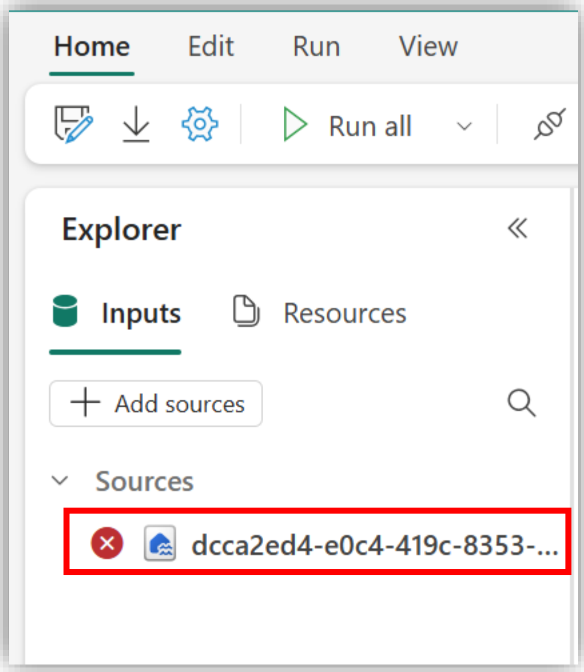
Explore, analyze, and visualize data and build ML models. Supports Apache Spark, Python, T-SQL, and more.

But today, we will import a pre-created notebook to save time.

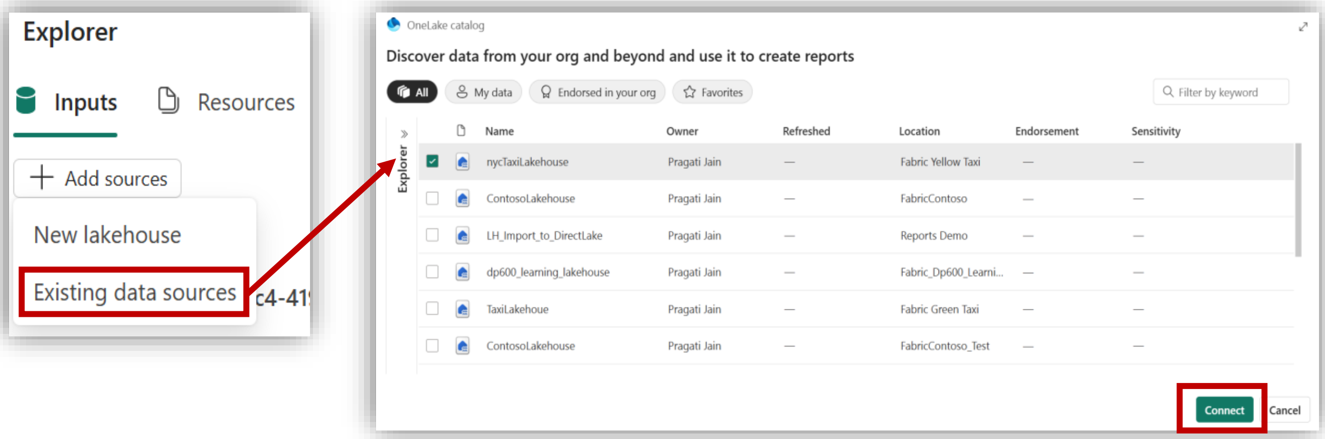

Fabric Yellow Taxi
Create deployment pipeline
Upd
New item
New folder
Import
Notebook
Report, Paginated Report or Workbook
From this computer
Import notebook source code files from your local drive.
Name

Once the notebook is imported, it will show on your workspace.

On opening the notebook, you will get an error on the left against the Lakehouse.



This error is due to the default Lakehouse to which this notebook was connected earlier. To make this notebook work, we need to attach it to the Lakehouse in our workspace.

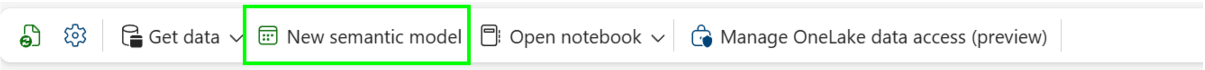


	Name	Owner	Refreshed	Location	Endorsement	Sensitivity
☑	nycTaxiLakehouse	Pragati Jain	—	Fabric Yellow Taxi	—	—
□	ContosoLakehouse	Pragati Jain	—	FabricContoso	—	—
□	LH_Import_to_DirectLake	Pragati Jain	—	Reports Demo	—	—
□	dp600_learning_lakehouse	Pragati Jain	—	Fabric_Dp600_Learni...	—	—
□	TaxiLakehoue	Pragati Jain	—	Fabric Green Taxi	—	—
□	ContosoLakehouse	Pragati Jain	—	FabricContoso_Test	—	—

Section 4: Semantic Modelling

Semantic Model

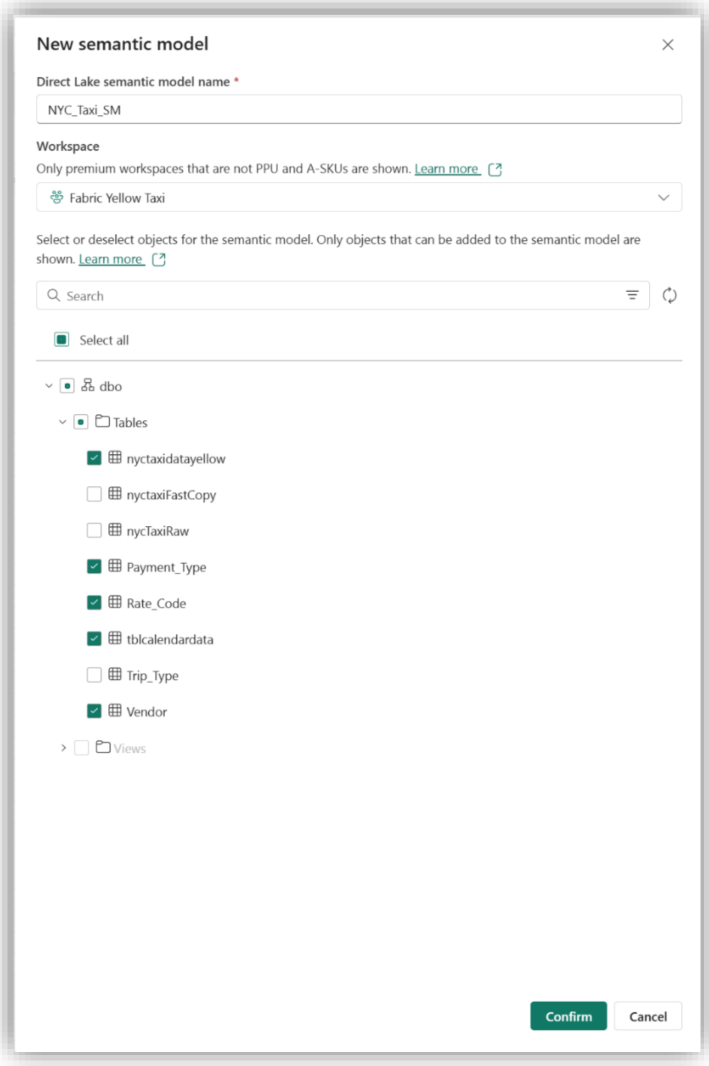
1. Open the lakehouse: **nycTaxiLakehouse**
2. In the menu bar select the button new semantic model



3. Name your Semantic Model: **NYC_Taxi_SM**

Information: semantic models can be store in a different Workspace than the one were the lakehouse is located in.

4. Select the tables to add to your data model:
   i. nyctaxidatayellow
   ii. Payment_Type
   iii. Rate_Code
   iv. Vendor
   v. Tblcalendardata
5. Press Confirm



New semantic model

Direct Lake semantic model name *

NYC_Taxi_SM

Workspace

Only premium workspaces that are not PPU and A-SKUs are shown. Learn more

Fabric Yellow Taxi

Select or deselect objects for the semantic model. Only objects that can be added to the semantic model are shown. Learn more

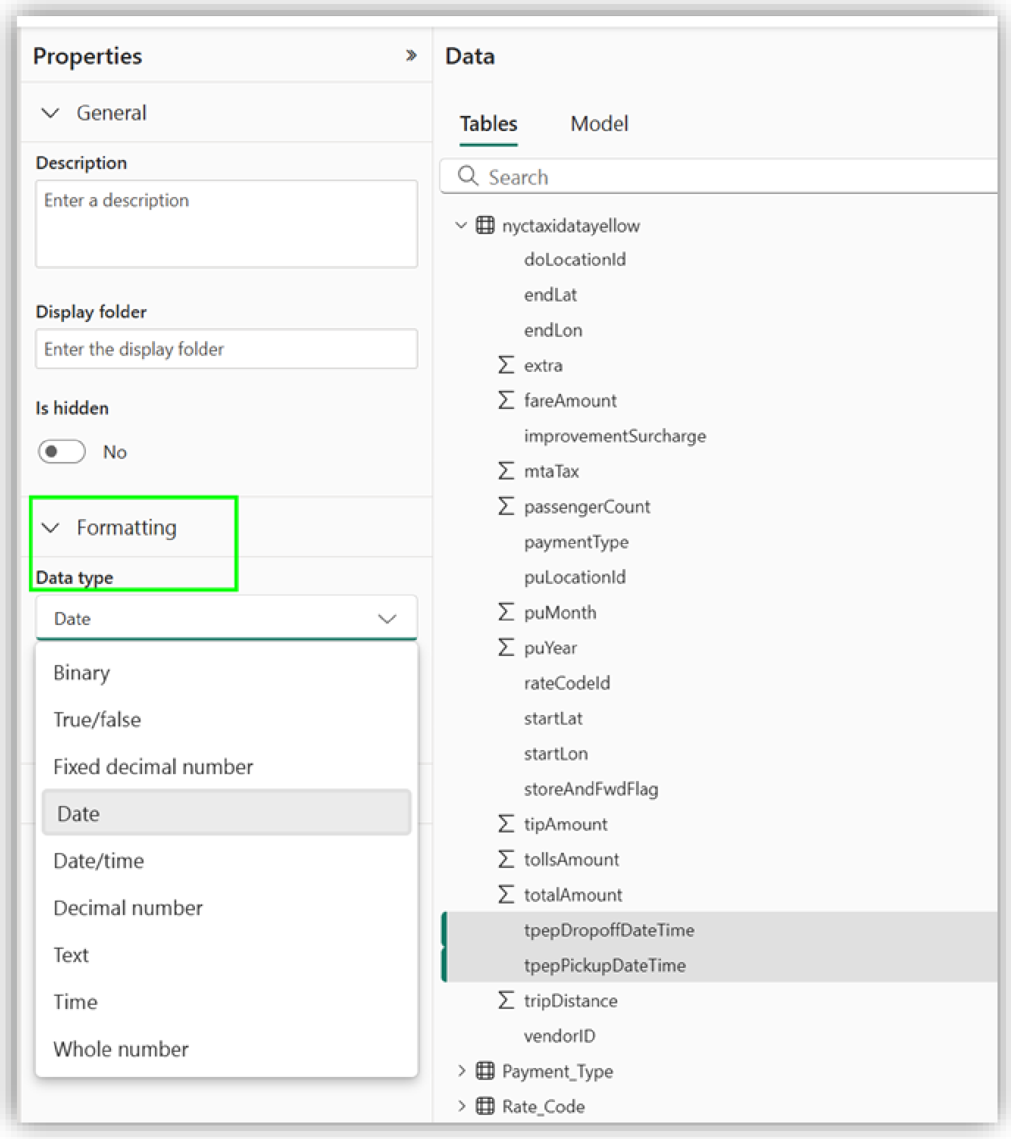
Search

■ Select all

- ■ dbo
  - ■ Tables
    - ☑ nyctaxidatayellow
    - □ nyctaxiFastCopy
    - □ nycTaxiRaw
    - ☑ Payment_Type
    - ☑ Rate_Code
    - ☑ tblcalendardata
    - □ Trip_Type
    - ☑ Vendor
  - □ Views

Format Fields:

1. Select the **nyctaxidatayellow** table
2. On **Data pane** located in the right side select the **tpepDropoffDateTime** and **tpepPickupDateTime** (use CTRL +click to select both of the columns)
3. On the formatting option select Data type Date



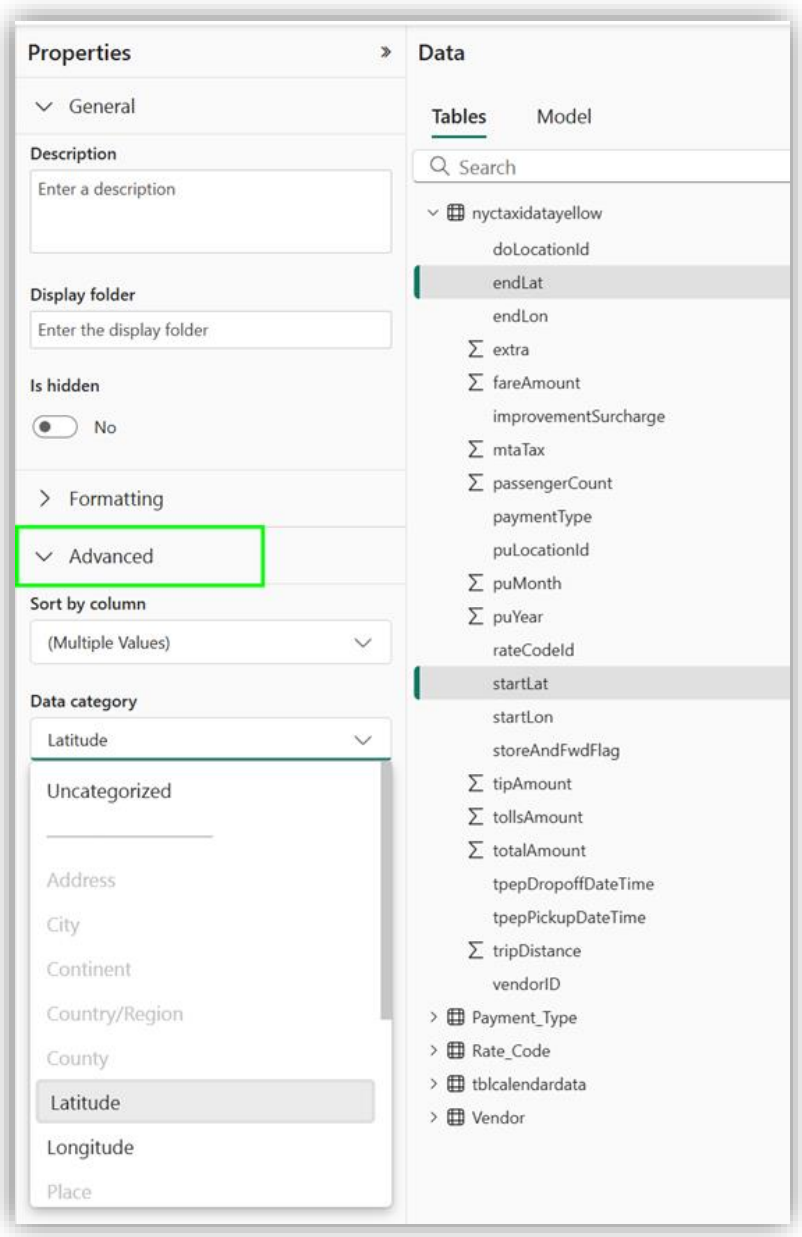
Description: Enter a description

Display folder: Enter the display folder

Is hidden: No

Data type: Date

4. Select the columns **endLat** and **startLat**
5. On the formatting option open the **advance pane** and define **data category** as latitude



Properties

Description: Enter a description

Display folder: Enter the display folder

Is hidden: No

Sort by column: (Multiple Values)

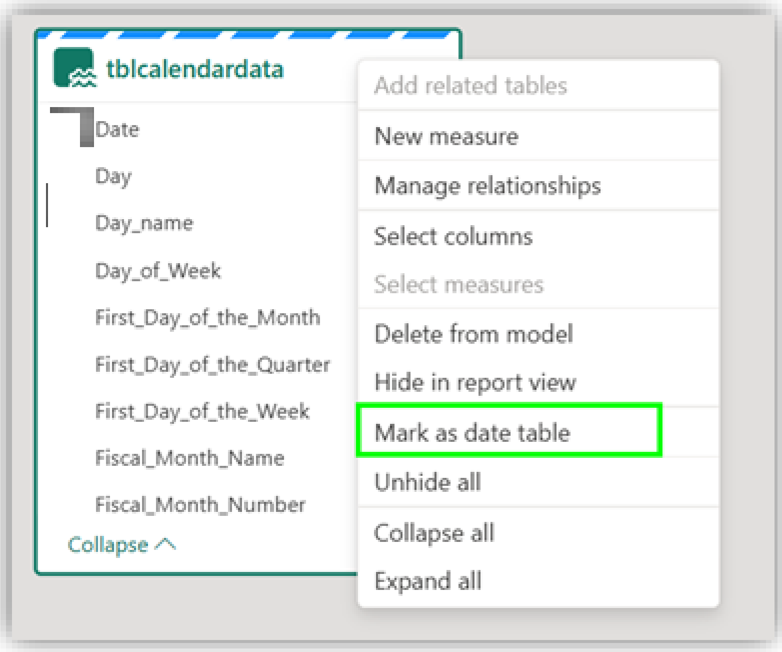
Data category: Latitude

6. Select the columns **endLon** and **startLon**
7. On the formatting option open the **advance pane** and define **data category** as Longitude

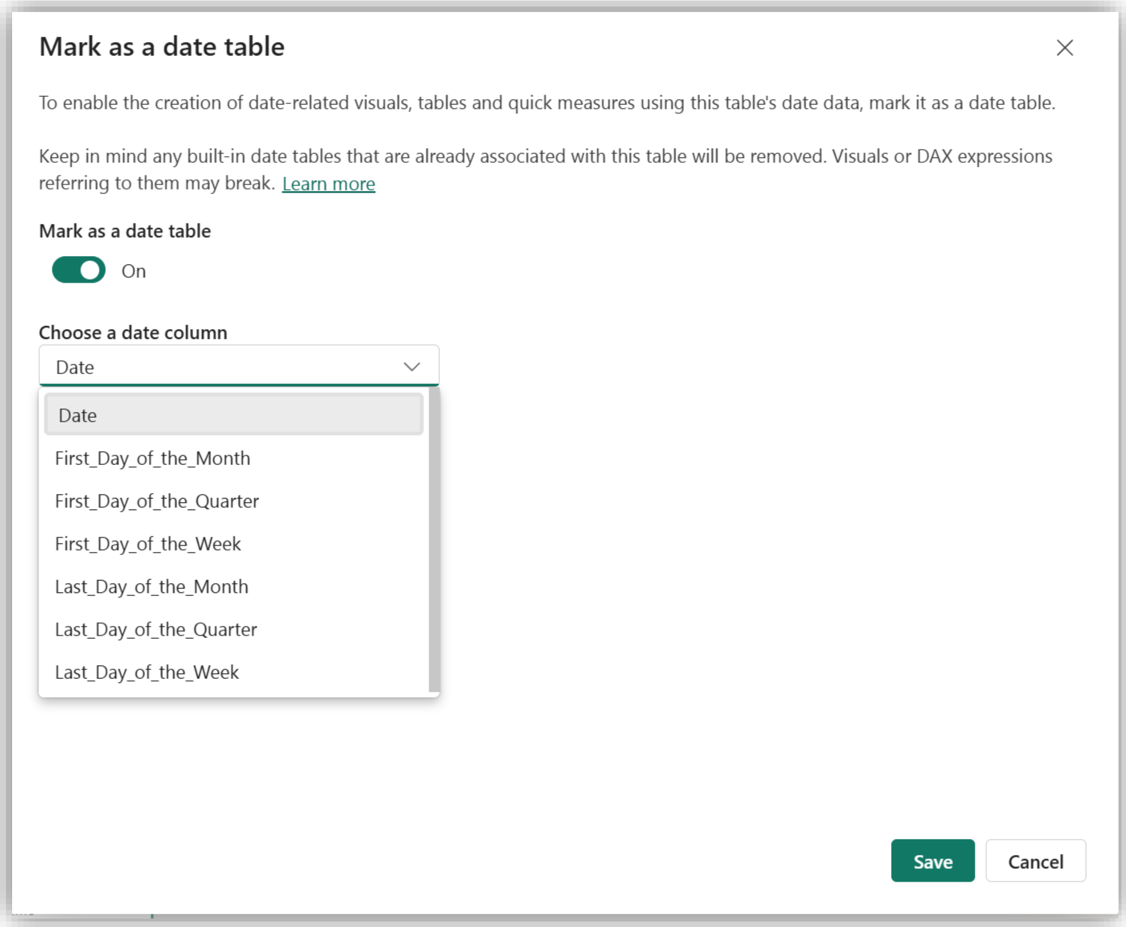
8. Select the **tblcalendardata** table
9. Format the **date** column as date

Define tblcalendardata as date table:

1. Right Click the **tblcalendardata** table
2. Select the Mark as date table on the pop-up menu



3. On the **Mark as date table** dialog box turn on the Mark as date tabl
4. Select the **Date** column
5. Click **save**



Mark as a date table

To enable the creation of date-related visuals, tables and quick measures using this table's date data, mark it as a date table.

Keep in mind any built-in date tables that are already associated with this table will be removed. Visuals or DAX expressions referring to them may break. Learn more

Mark as a date table

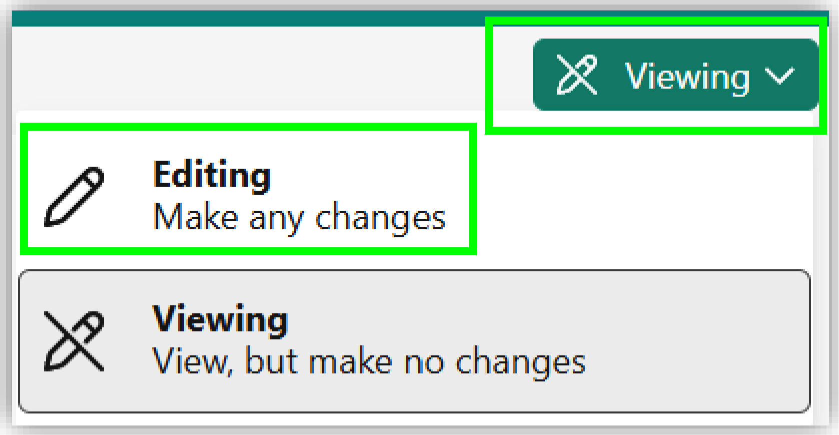
● On

Choose a date column

Date

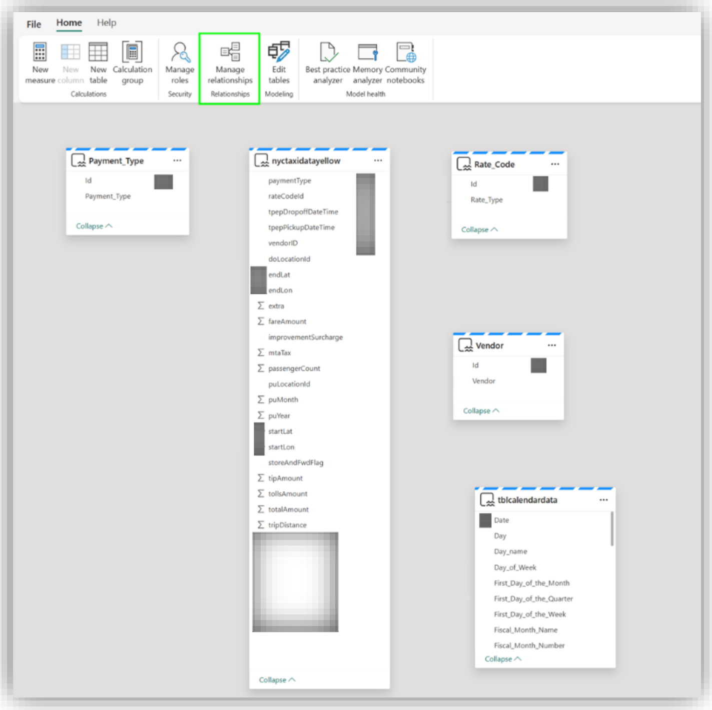
Create Relationships:

1. Open the semantic model NYC_Taxi_SM
2. Be sure that the model is on Editing mode (top right corner)

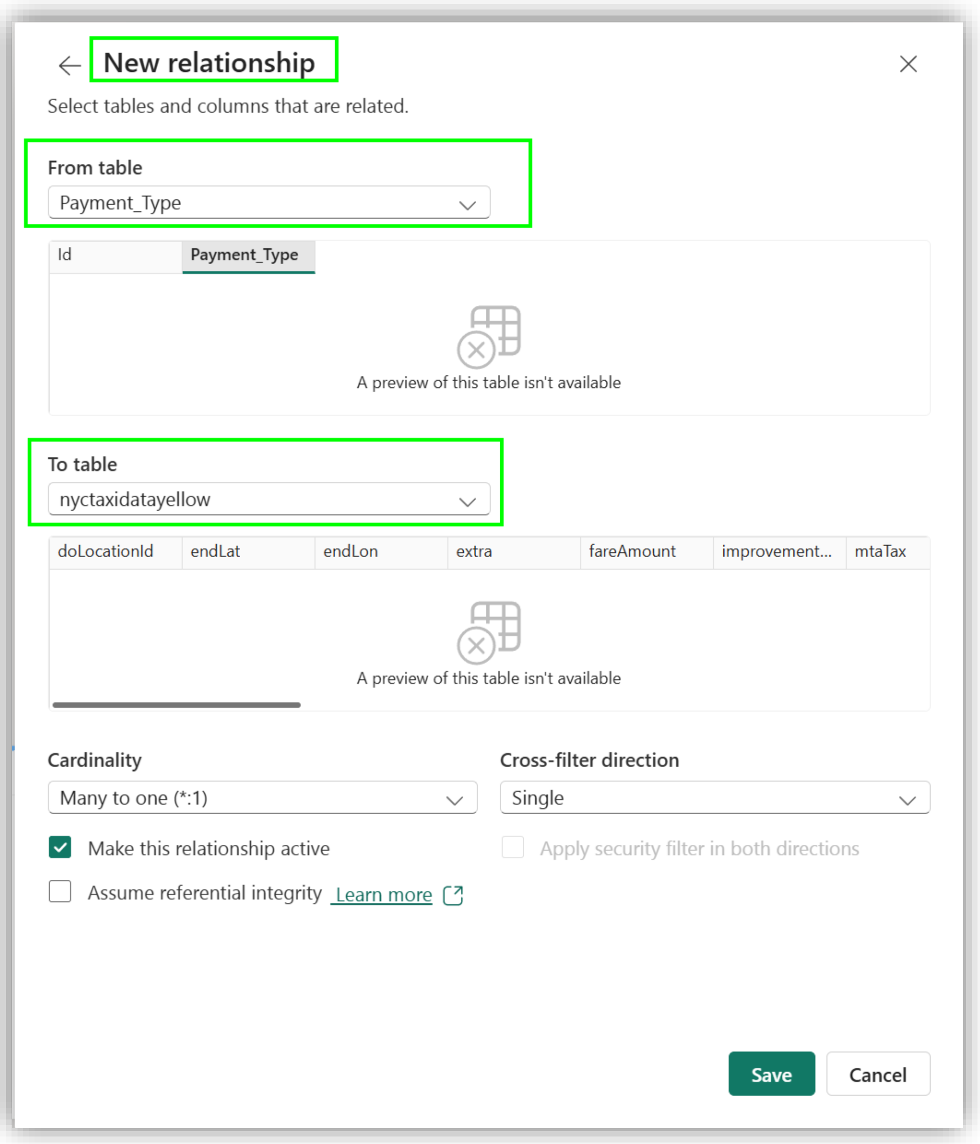


Relationship between Payment Type and nyctaxidatayellow

1. Click the button **Manage Relationship** on the navigation bar



2. In the dialog box select the **Payment_Type** table on the From Table and the **nyctaxidatayellow** table on the To table



New relationship

Select tables and columns that are related.

From table

Payment_Type

Id	Payment_Type

A preview of this table isn't available

To table

nyctaxidatayellow

doLocationId	endLat	endLon	extra	fareAmount	improvement...	mtaTax

A preview of this table isn't available

Cardinality

Many to one (*:1)

Cross-filter direction

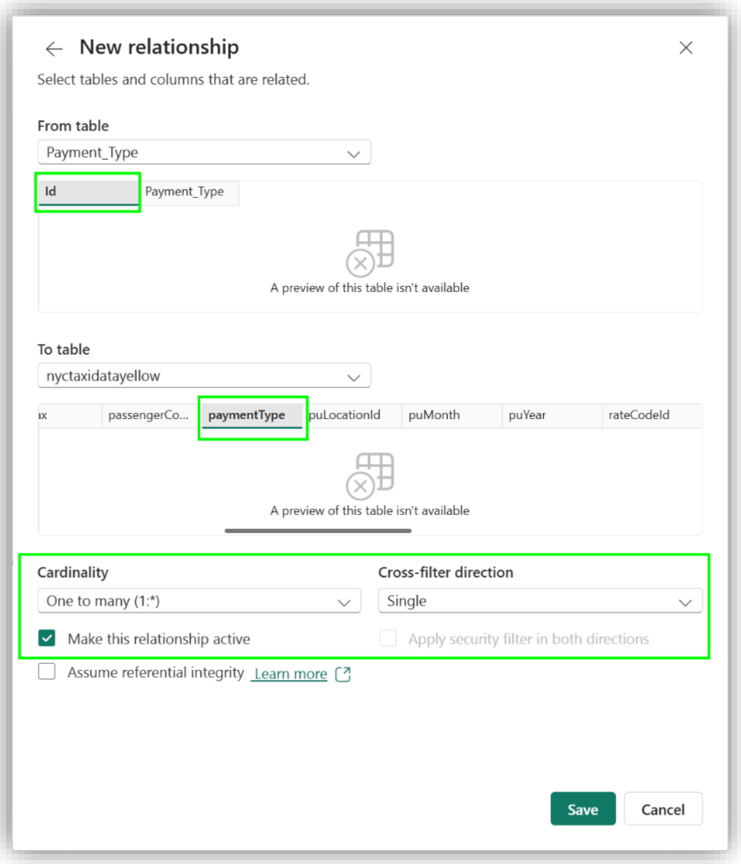
Single

☑ Make this relationship active

□ Apply security filter in both directions

□ Assume referential integrity Learn more

3. Select the Id column on the Payment_Type table and the paymenttype on the nyctaxidatayellow
4. Confirm if the Cardinality is **One to many (1:*),** Cross-filter direction is single and the relationship is active



← New relationship ×

Select tables and columns that are related.

From table

Payment_Type

Id	Payment_Type

A preview of this table isn't available

To table

nyctaxidatayellow

ıx	passengerCo...	paymentType	puLocationId	puMonth	puYear	rateCodeId

A preview of this table isn't available

Cardinality: One to many (1:*)

Cross-filter direction: Single

☑ Make this relationship active

□ Apply security filter in both directions

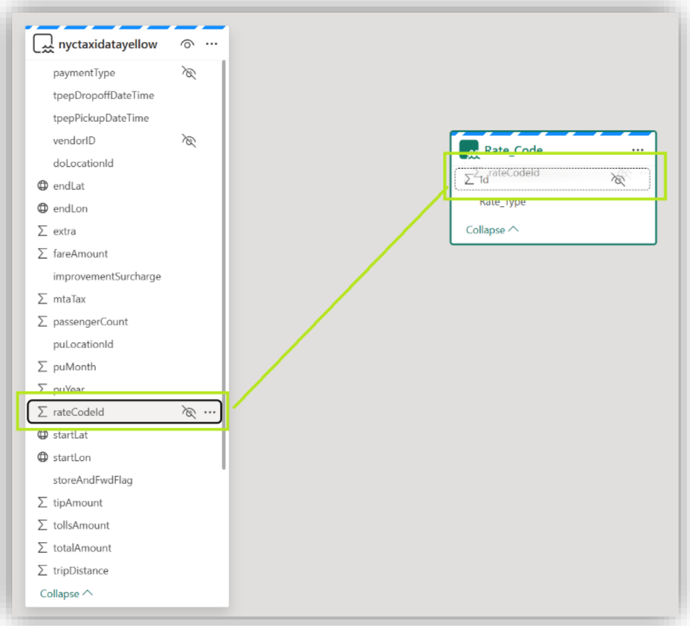
□ Assume referential integrity Learn more

Save Cancel

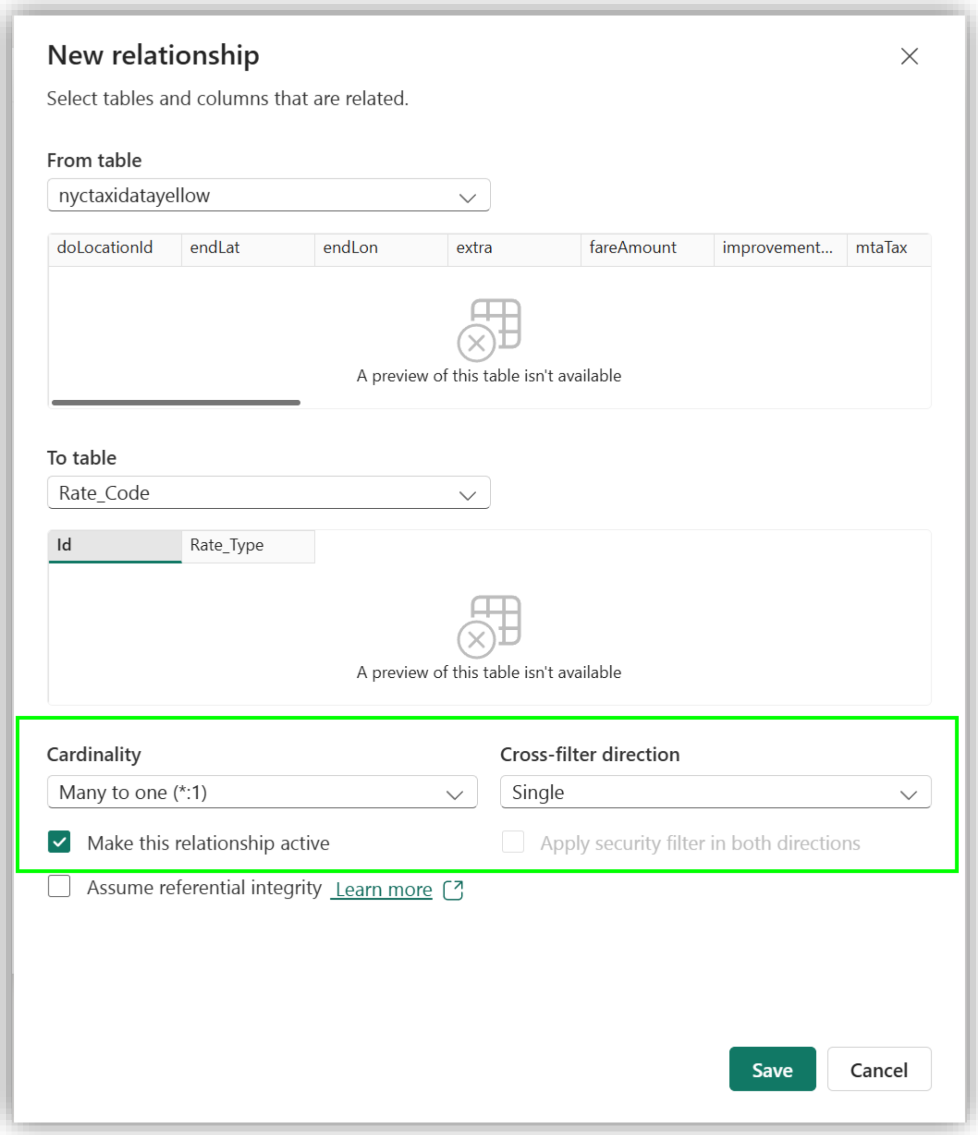
5. Click **Save**

6. **Relationship between Rate_Code and nyctaxidatayellow**

Click on the **ratecodeid** in the **nyctaxidatayellow** table and drag it over to the **Id** in the **Rate_Code** table



7. On the New Relationship dialog box check if cardinality is **many to one (*:1),** Cross-filter direction is single and the relationship is active



New relationship

Select tables and columns that are related.

From table

nyctaxidatayellow

doLocationId	endLat	endLon	extra	fareAmount	improvement...	mtaTax

A preview of this table isn't available

To table

Rate_Code

Id	Rate_Type

A preview of this table isn't available

Cardinality: Many to one (*:1)

Cross-filter direction: Single

☑ Make this relationship active

□ Apply security filter in both directions

□ Assume referential integrity Learn more

8. Click **Save**
9. Create the following relationships using one of the methods above:
   a. **Vendor** and **nyctaxidatayellow**
      i. ID from vendor table
      ii. Vendorid from Nyctaxidatayellow
      iii. One to many (1:*)
      iv. Cross-filter direction: Single

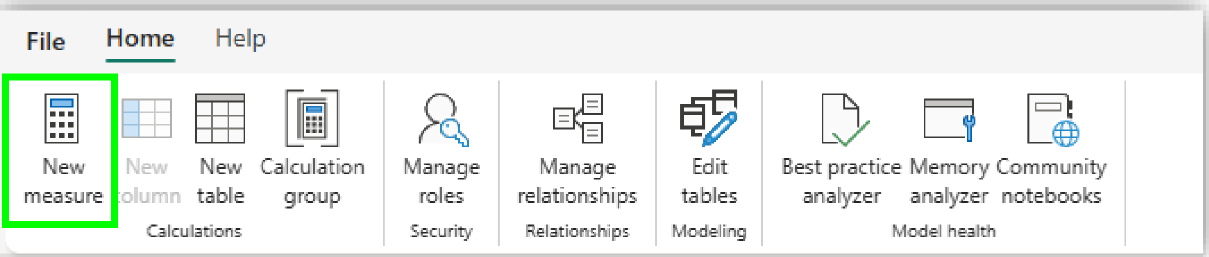
   b. **tblcalendardata** and **Nyctaxidatayellow**
      i. date from tblcalendardata
      ii. tpepPickupDateTime orom Nyctaxidatayellow
      iii. One to many (1:*)
      iv. Cross-filter direction: Single

   c. **tblcalendardata** and **Nyctaxidatayellow**
      i. date from tblcalendardata
      ii. tpepDropoffDateTime orom Nyctaxidatayellow

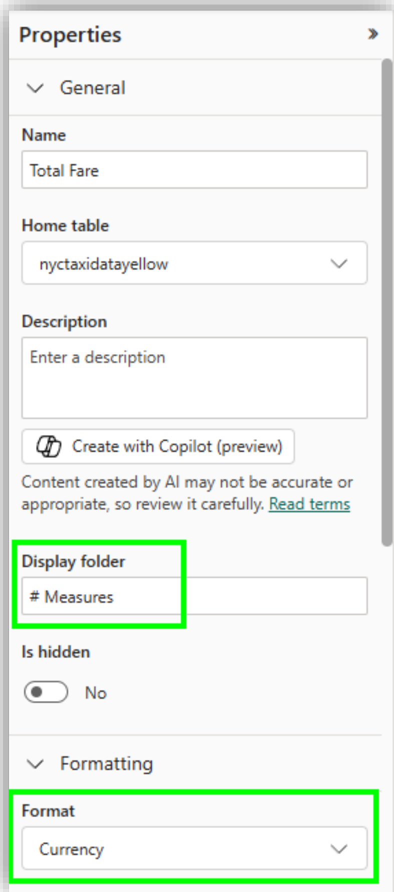
iii. One to many (1:*)
iv. Cross-filter direction: Single
v. Inactive

Create DAX Measures:

1. Select the table n**yctaxidatayellow**
2. On the **Home** in the menu bar select the option **New measure**



3. On the dax formula bar create the following measure:
   Total Fare = SUM(nyctaxidatayellow[fareAmount])
4. On the formatting pane format as currency and add the **# Measures** as display folder



Name
Total Fare

Home table
nyctaxidatayellow

Description
Enter a description

Content created by AI may not be accurate or appropriate, so review it carefully. Read terms

Display folder
Measures

Is hidden
No

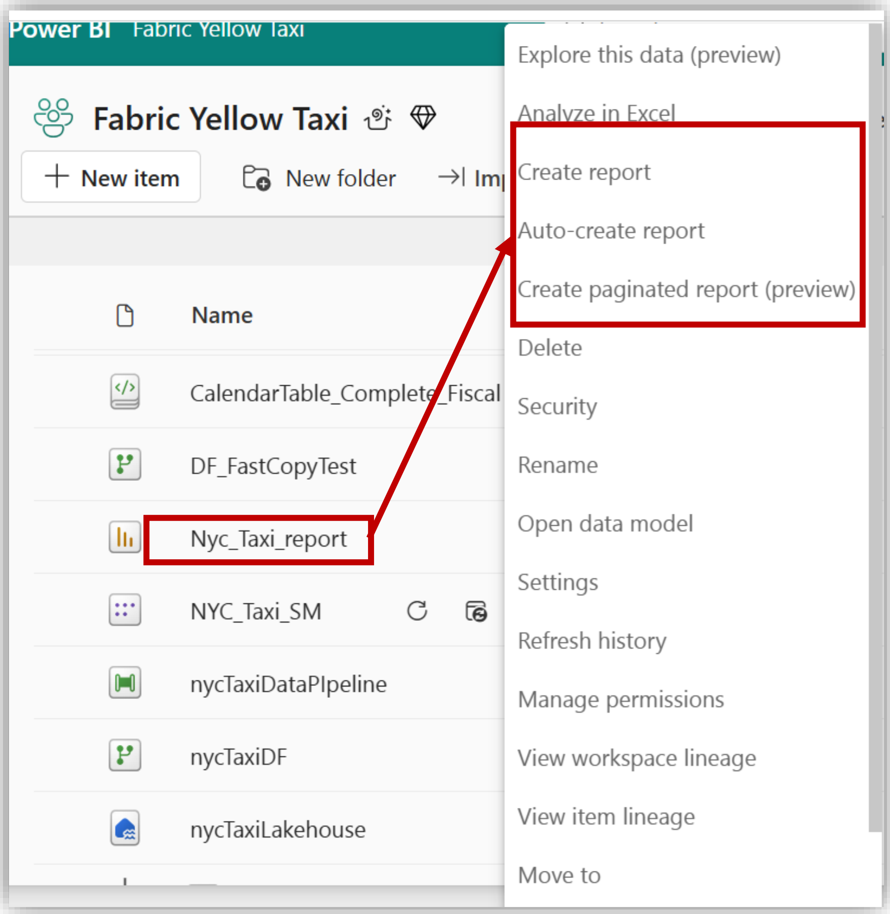
Format
Currency

5. Create the following measures
    a. Tip Amount = SUM(nyctaxidatayellow[tipAmount])
       Format: Currency
    b. Total Passengers = SUM(nyctaxidatayellow[passengerCount])
       Format: Whole Number
    c. Total Trip Distance = SUM(nyctaxidatayellow[tripDistance])
       Format: Decimal Number
    d. Total Trips = COUNTROWS(nyctaxidatayellow)
       Format: Whole Number
    e. Trip Amount = SUM(nyctaxidatayellow[totalAmount])
       Format: Currency

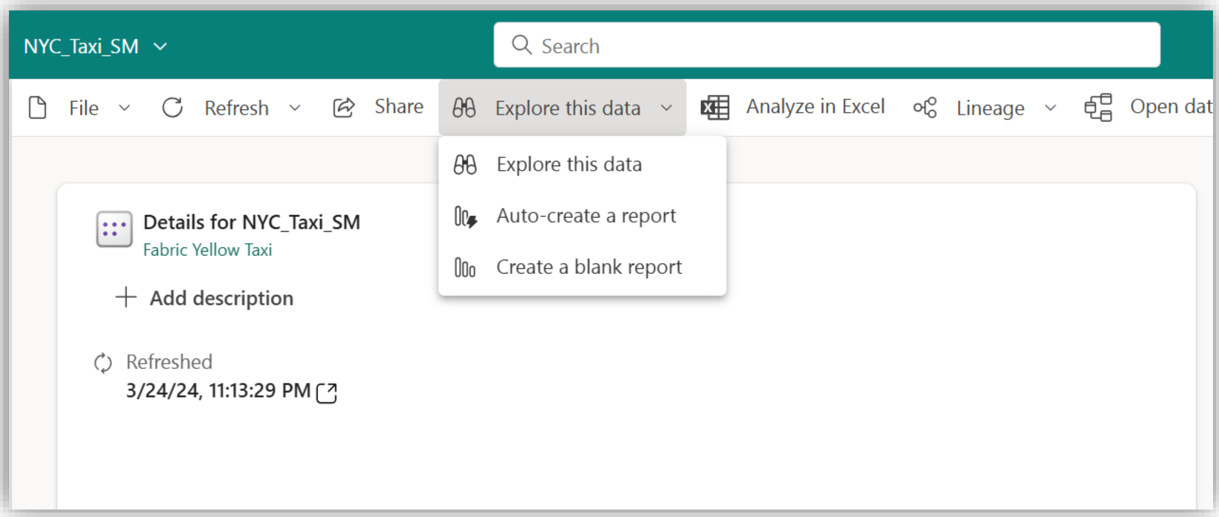
Section 5: Report Creation

For report creation, we are using the custom semantic model created in the last step.

Select the three dots on the semantic model:



You can also directly open the semantic model and create reports from there.



NYC_Taxi_SM
Search
File
Refresh
Share
Explore this data
Analyze in Excel
Lineage
Open dat
Explore this data
Auto-create a report
Create a blank report
Details for NYC_Taxi_SM
Fabric Yellow Taxi
Add description
Refreshed
3/24/24, 11:13:29 PM

Quizz

Data landing

What are the 3 artifacts that are created with a new LakeHouse:

- Lakehouse; SQL Endpoint; Default Semantic Model (Correct answer)
- Lakehouse; Notebook; Semantic Model
- SQL Endpoint; Semantic Model, Report

Identify what type of data can be stored both in a lakehouse and a warehouse

- Structure data (Correct Answer)
- Semi-structured Data
- Unstructured Data

What's the primary developer persona for a Warehouse

- SQL Engineer (Correct Answer)
- Data Scientist
- Power BI Developer

Semantic model

What is the storage option to build a semantic model on top of Onelake:

- Direct Query
- Import
- Composite models
- Direct Lake (Correct Answer)

Identify the limitation using Semantic Models on top of Onelake storage:

- DAX measures are not supported
- Calculated columns and tables are not supported (Correct answer)
- Relationships between tables are not supported

What is the option needs to be enable to edit Direct Lake semantic models with external tools

- Row Level Security
- XMLA read-write option for the capacity (Correct Answer)
- Is a default option with Direct Lake