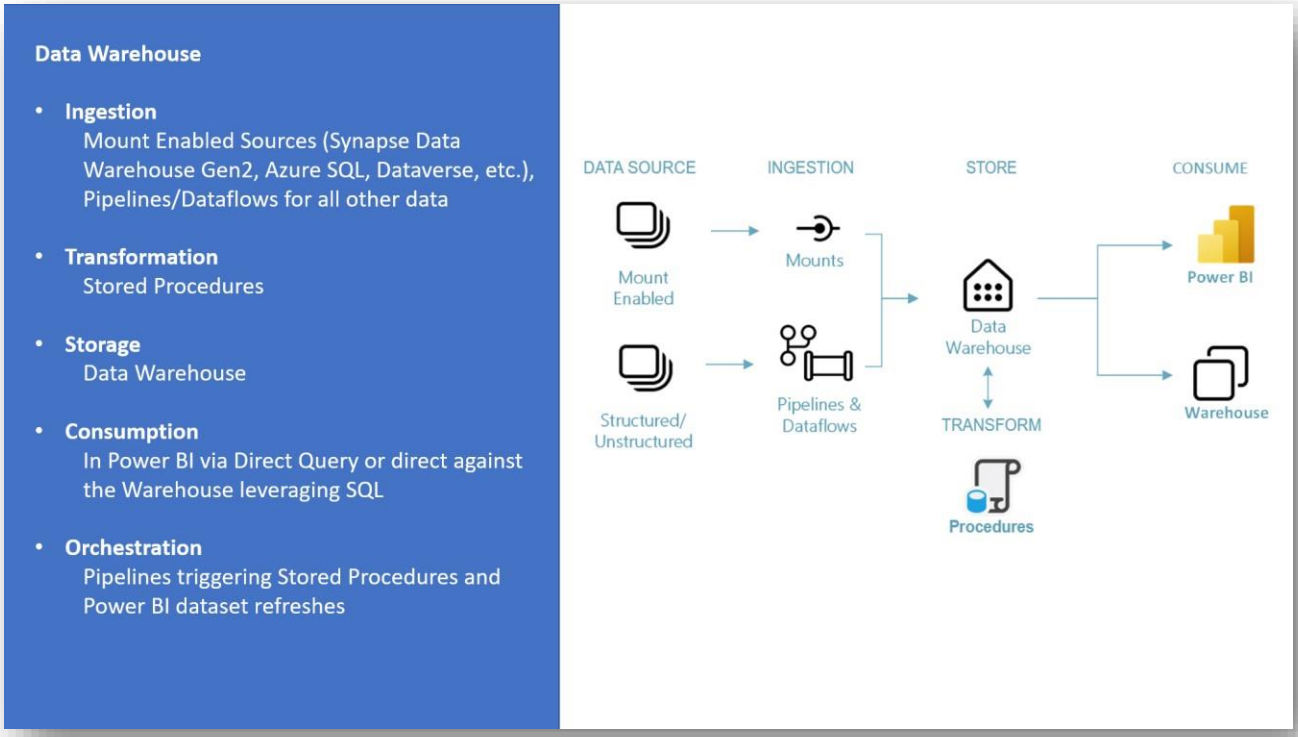
Tutorial

Data Warehouse

Published: July 2024



Contents

Introduction

What is Fabric?

Fabric provides a one-stop shop for all the analytical needs for every enterprise. It covers the complete spectrum of services including data movement, data lake, data engineering, data integration and data science, real time analytics, and business intelligence. With Fabric, there is no need to stitch together different services from multiple vendors. Instead, the customer enjoys an end-to-end, highly integrated, single comprehensive product that is easy to understand, onboard, create and operate. There is no other product on the market that offers the breadth, depth, and level of integration that Fabric offers. Additionally, Microsoft Purview is included by default in every tenant to meet compliance and governance needs.

To get an overview over the components and concepts of Fabric read Fabric - Overview and Concepts.

Purpose of this tutorial

While many concepts in Fabric may be familiar to data and analytics professionals it can be challenging to apply those concepts in a new environment. This tutorial has been designed to walk step-by-step through an end-to-end scenario from data acquisition to data consumption to build a basic understanding of the Fabric UX, the various workloads and their integration points, and the Fabric professional and citizen developer experiences.

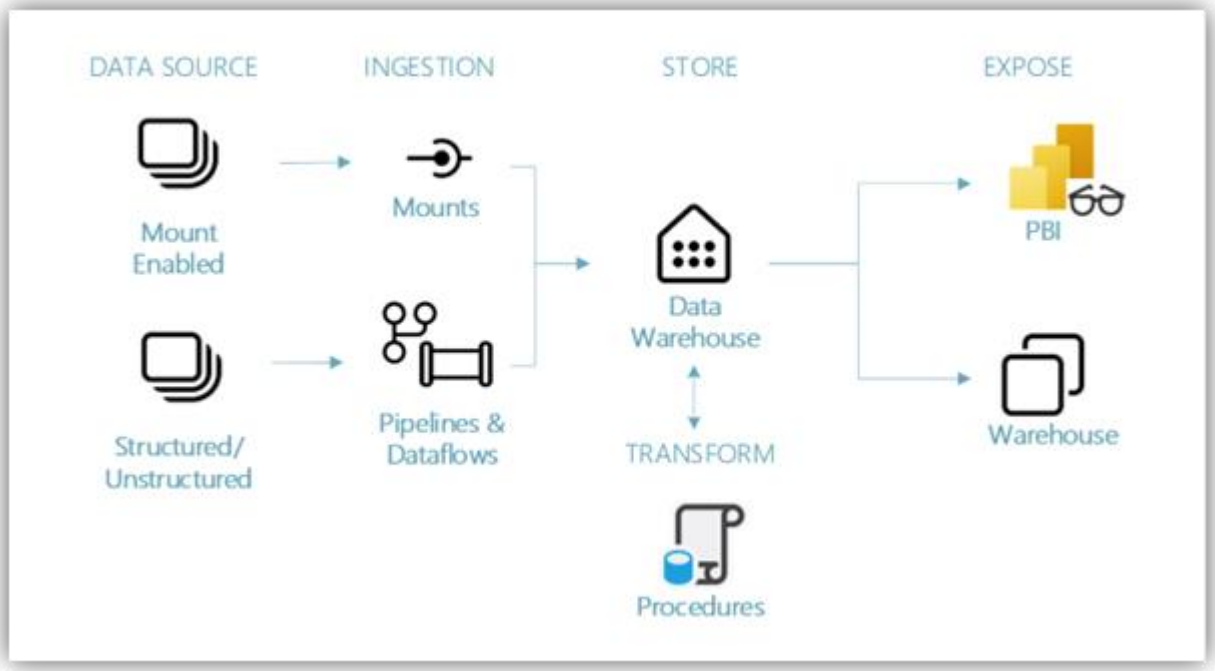
The tutorials are not intended to be a reference architecture, an exhaustive list of features and functionality, or a recommendation of specific best practices.

The data warehouse tutorial

In this tutorial, you will take on the role of a data warehouse developer at the fictional Wide World Importers company and complete the following steps:

- Sign into your Power BI online account , or if you don't have an account yet, sign up for a free trial.
- Build and implement an end-to-end data warehouse for your organization: ○ Enable Fabric in your tenant ○ Create a Fabric workspace ○ Quickly create a data warehouse
    - Ingest data from source to the data warehouse dimensional model ○ Transform the data to create aggregated datasets using T-SQL
    - Perform orchestration, data ingestion, and data transformation with pipelines ○ Query the data warehouse using T-SQL and a visual query editor ○ Create Power BI report using DirectLake mode to analyze the data in place
- Cleanup resources by deleting the workspace and other items

The data warehouse end-to-end architecture



Data Sources – Fabric makes it easy and quick to connect to Azure Data Services, other cloud platforms, and on-premises data sources to ingest data from.

Ingestion – With 200+ native connectors as part of the Fabric pipeline and with drag and drop data transformation with dataflow, you can quickly build insights for your organization. Shortcut is a new feature in Fabric that provides a way to connect to existing data without having to copy or move it – more details about Shortcut later in this tutorial.

Transform and Store – Fabric standardizes on Delta Lake format, that means all the engines of Fabric can read and work on the same dataset stored in OneLake – no need for data duplicity. This storage allows you to build a data warehouse or data mesh based on your organizational need. For transformation, you can choose either low-code or nocode experience with pipelines/dataflows or use T-SQL for a code first experience.

Consume – Data from the data warehouse can be consumed by Power BI, industry leading business intelligence tool, for reporting and visualization. Each data warehouse comes with a built-in TDS/SQL endpoint for easily connecting to and querying data from other reporting tools, when needed. When a data warehouse is created a secondary item, called a default dataset, will be automatically generated at the same time with the same name of the data warehouse to start visualizing data with just a couple of mouse clicks.

The sample data

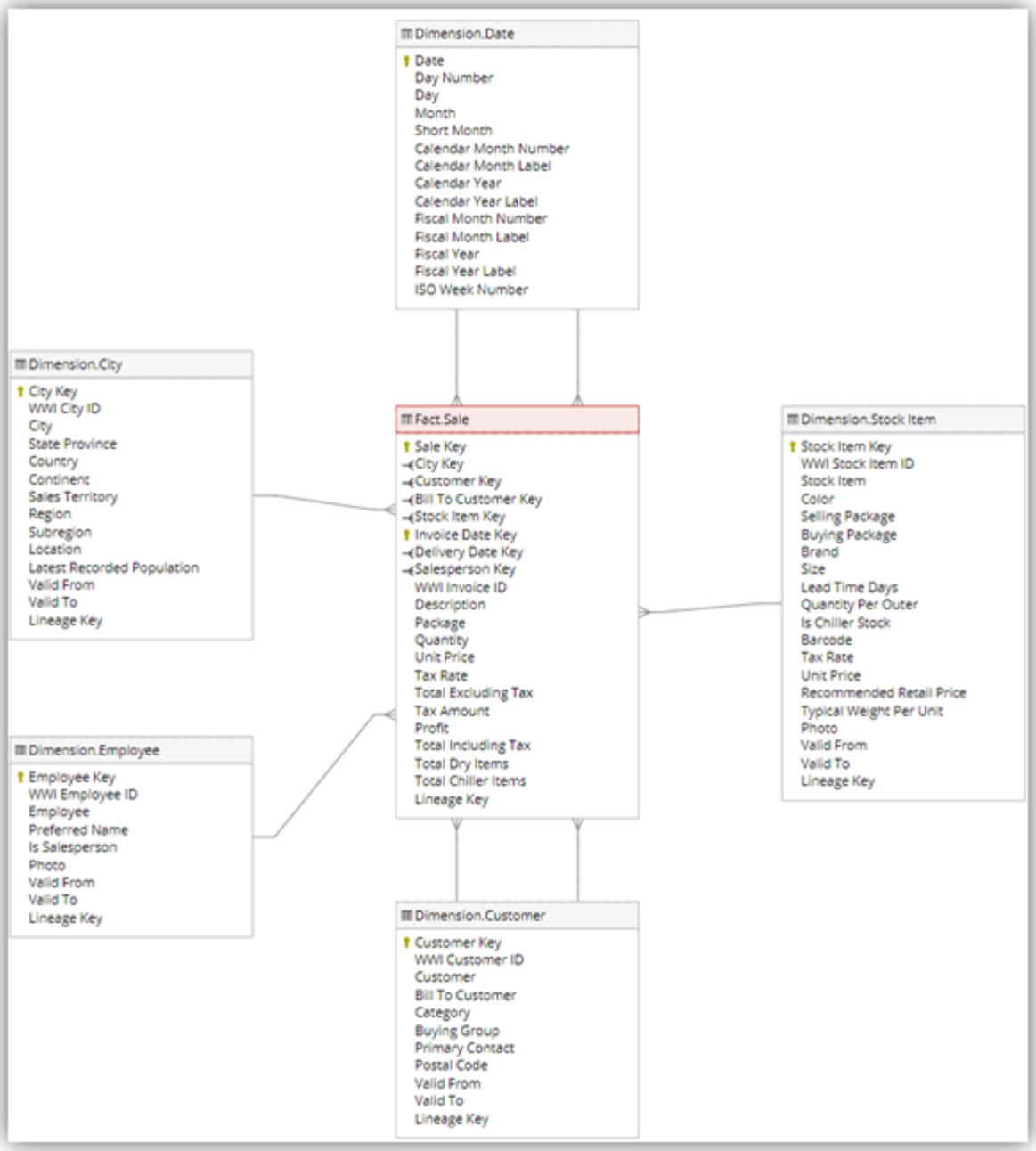
For sample data, we are going to use Wide World Importers (WWI) sample database. For our data warehouse end-to-end scenario, we have generated sufficient data for a sneak peek into the scale and performance capabilities of the Fabric platform.

Wide World Importers (WWI) is a wholesale novelty goods importer and distributor operating from the San Francisco Bay area. As a wholesaler, WWI's customers are mostly companies who resell to individuals. WWI sells to retail customers across the United States including specialty stores, supermarkets, computing stores, tourist attraction shops, and some individuals. WWI also sells to other wholesalers via a network of agents who promote the products on WWI's behalf. You can learn more about their company profile and operation here.

Typically, you would bring data from transactional systems (or line of business applications) into a data lake or data warehouse staging area, however for simplicity of this tutorial, we are going to use the dimensional model provided by WWI as our initial data source. We are going to use it as the source to ingest the data into a data warehouse and transform it through T-SQL.

The data model

While the WWI dimensional model contains multiple fact tables, for simplicity in explanation we will focus on the Sale Fact table and its related dimensions only, as below, to demonstrate this end-to-end data warehouse scenario:



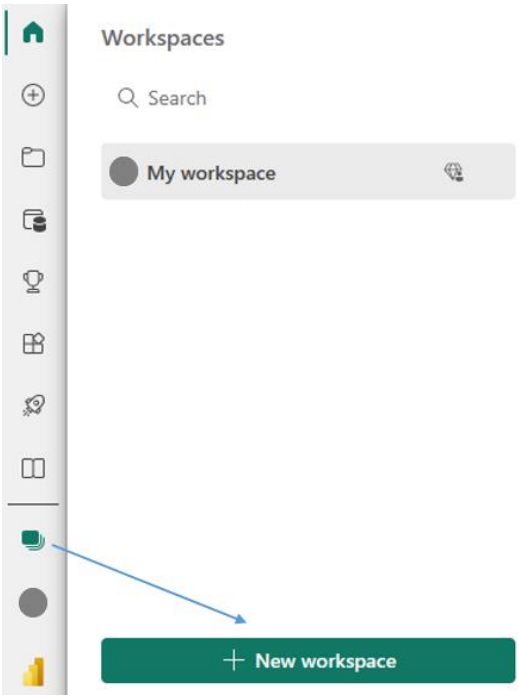
Module 1: Create a workspace

Before you can begin building the warehouse, you will need to create a workspace where you will build out the remainder of the tutorial. In this module, you will learn to:

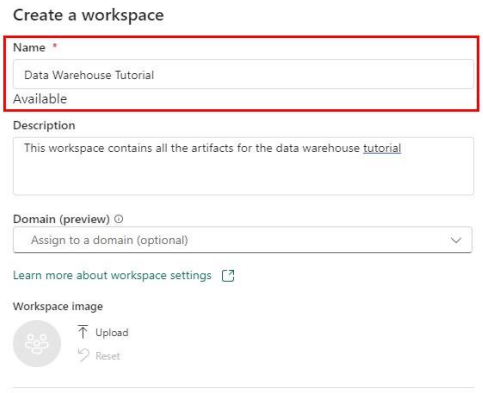
- Create a workspace

The workspace will contain all the artifacts needed for data warehousing including Data Factory pipelines, the data warehouse, Power BI datasets, and reports.

1. Sign in to Power BI.
2. Select **Workspaces > New Workspace**.



3. Fill out the **Create a workspace** form as follows:
    a. **Name:** Enter *Data Warehouse Tutorial*, and some characters for uniqueness.



    b. **Description**: Optionally, enter a description for the workspace.
    c. Expand the **Advanced** section.

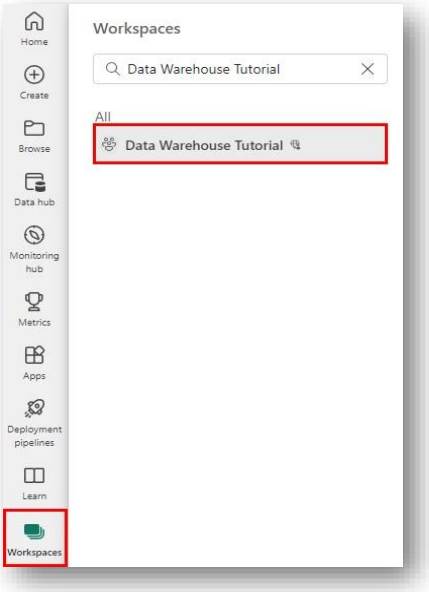
d. Choose **Premium Capacity** in the **License Mode** section.
e. Choose a premium capacity you have access to.
f. Select **Apply**. The workspace will be created and opened.

Module 2: Build your first data warehouse

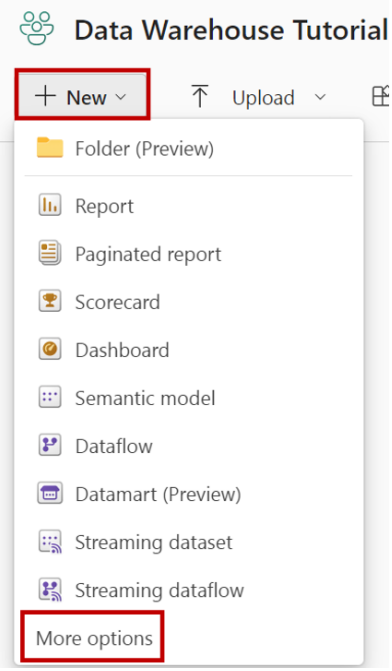
The intent of this module is to quickly build end to end journey of building a data warehouse, ingesting data for a table and then using the data warehouse for creating a report.

Create a data warehouse

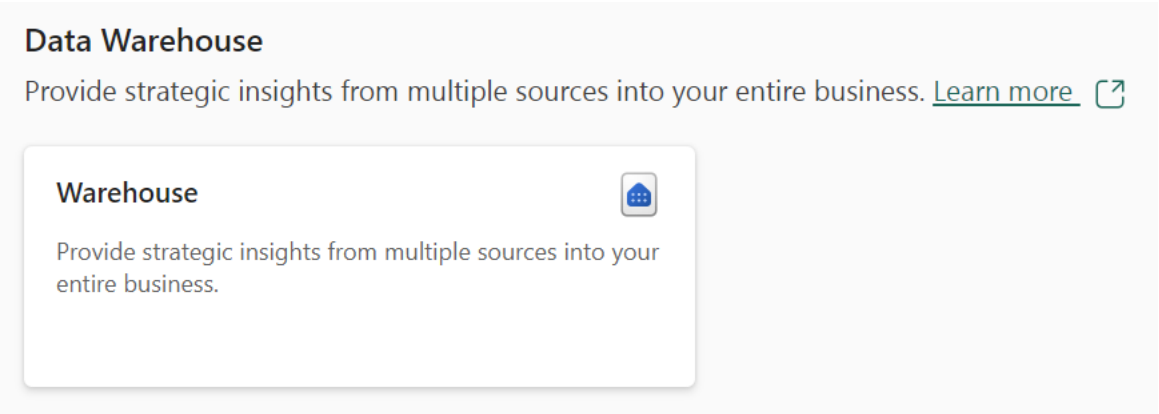
1. In the Power BI service select **Workspaces** in the left-hand menu.
2. Search for the workspace you create in Module 1 by typing in the search textbox at the top and click on your workspace to open it.



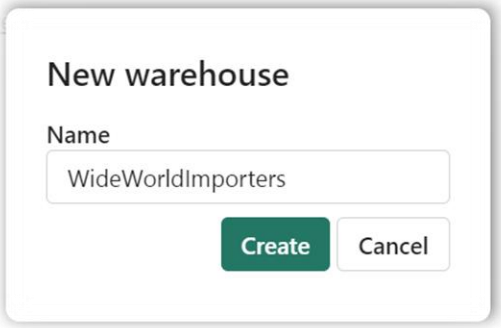
3. In the upper left corner, select **New > More Options** to display a full list of available items.



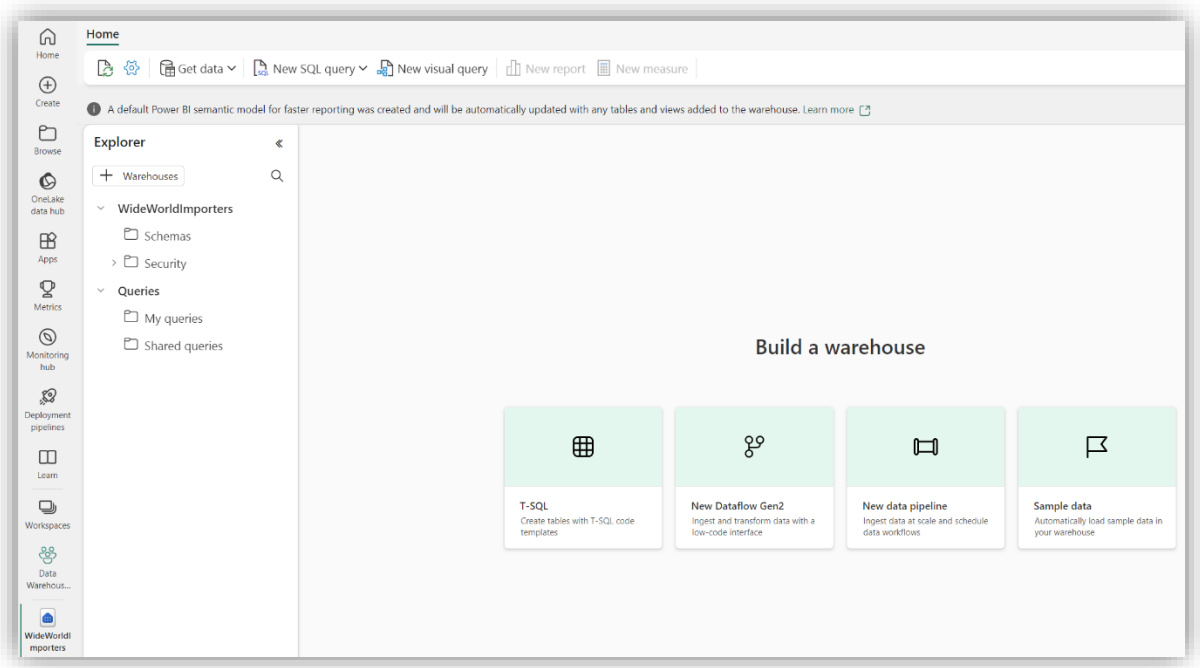
5. In the **Data warehouse** section, select **Warehouse**.



6. On the **New warehouse** dialog, enter **WideWorldImporters** as the name.
7. Select **Create**.

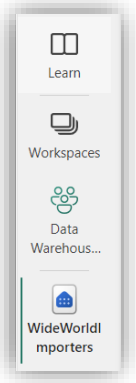


When provisioning is complete the **Build a warehouse** landing page will be shown.

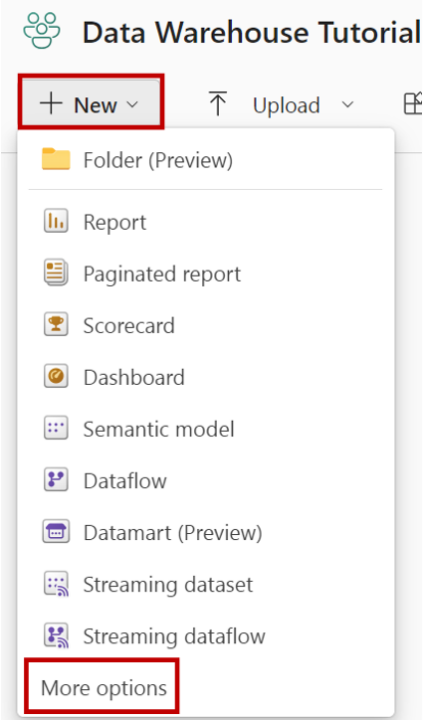


Data ingestion

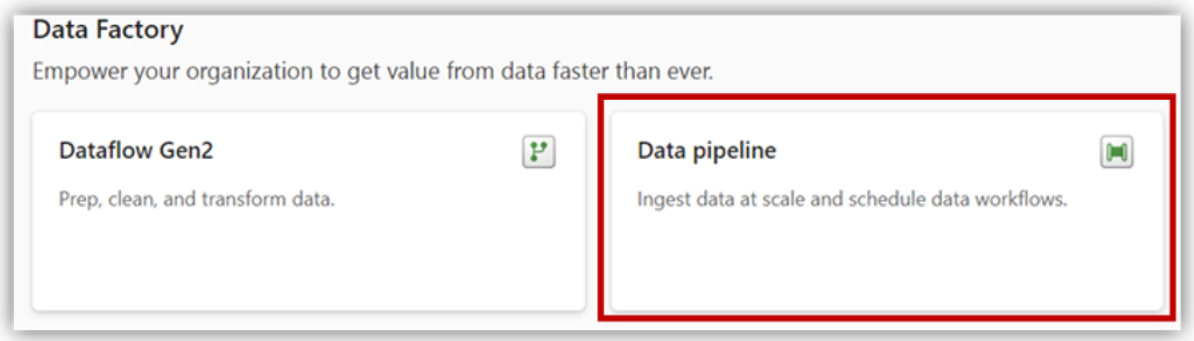
1. Select **Data Warehouse Tutorial** in the left-hand navigation menu to return to the workspace artifact view.



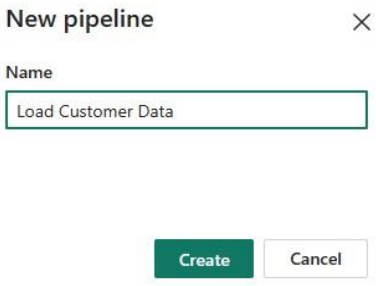
2. In the upper left corner, select **New > More Options** to display a full list of available items.



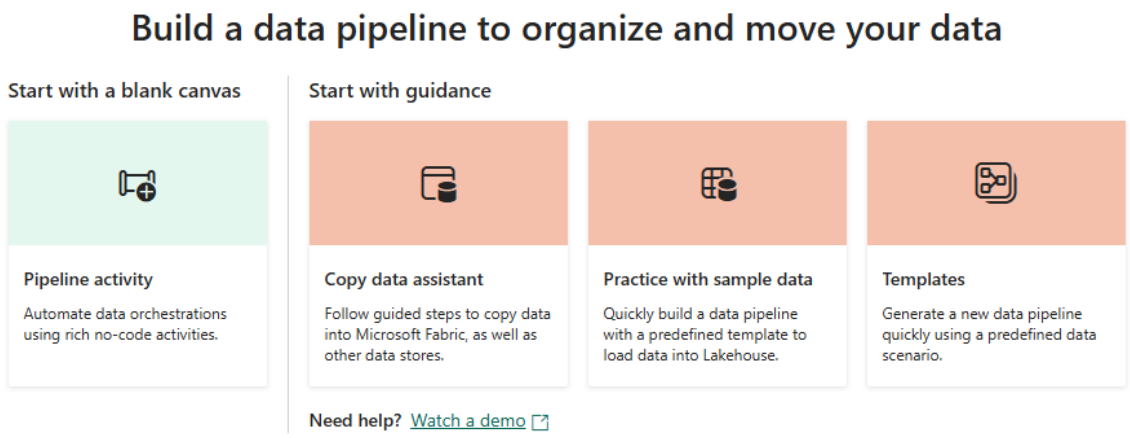
9. In the **Data Factory** section, select **Data pipeline**.



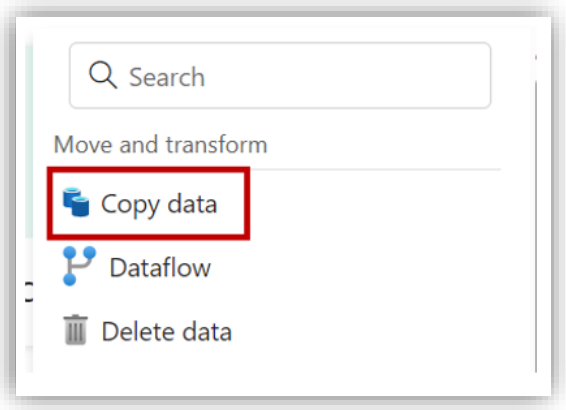
10. On the **New pipeline** dialog, enter **Load Customer Data** as the name.



11. Select **Create**.
12. Select **Add pipeline activity** from the **Start building your data pipeline** landing page.

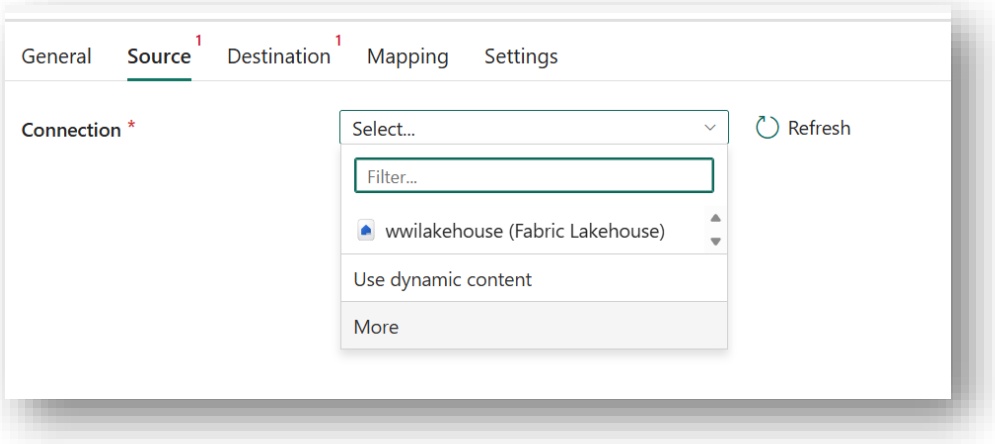


13. Select **Copy data** from the **Move & transform** section.

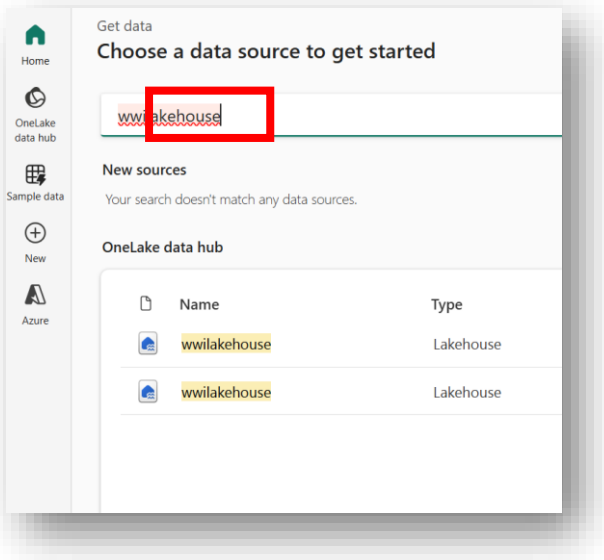


14. If necessary, select the newly created Copy data activity from the design canvas and follow the steps below to configure it.
15. On the **General** page, enter **CD Load dimension_customer** as the **Name**.

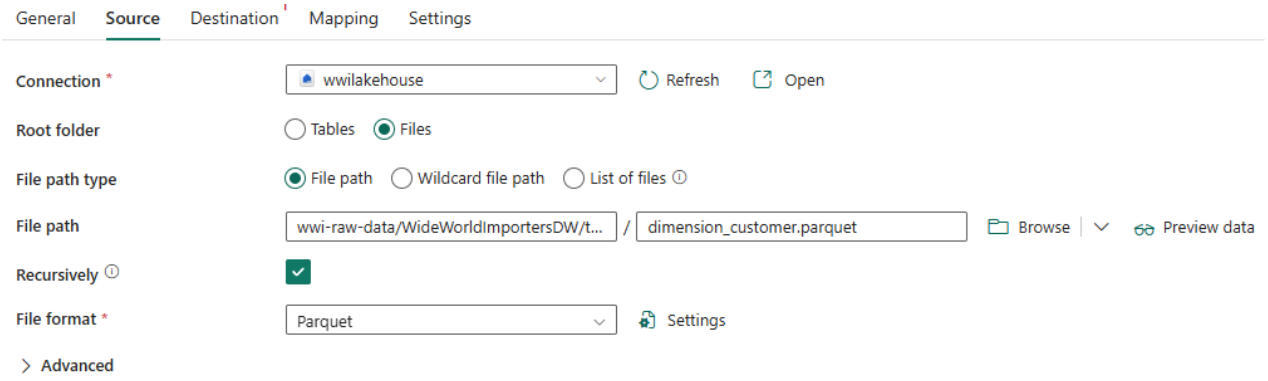
16. On the **Source** page Under **Connection** , select **More**



17. Type **wwilakehouse** in the search box to select the lakehouse created in the lakehouse tutorial. Make sure you are the owner of the lakehouse and is the one you created in the prior tutorial as multiple names will show up.

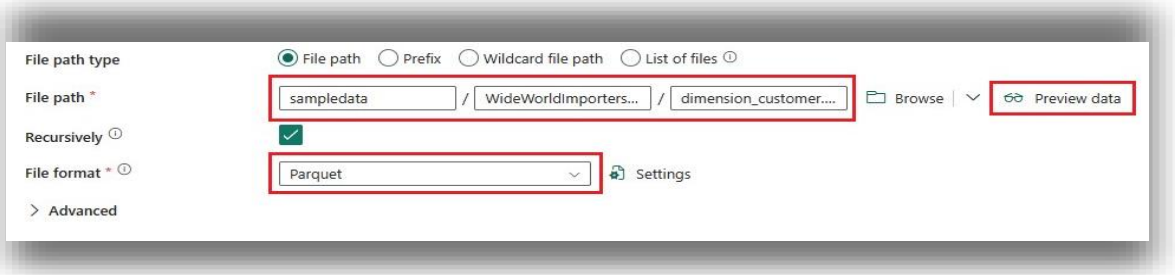


18. On the **File Path**, configure the settings as follows:

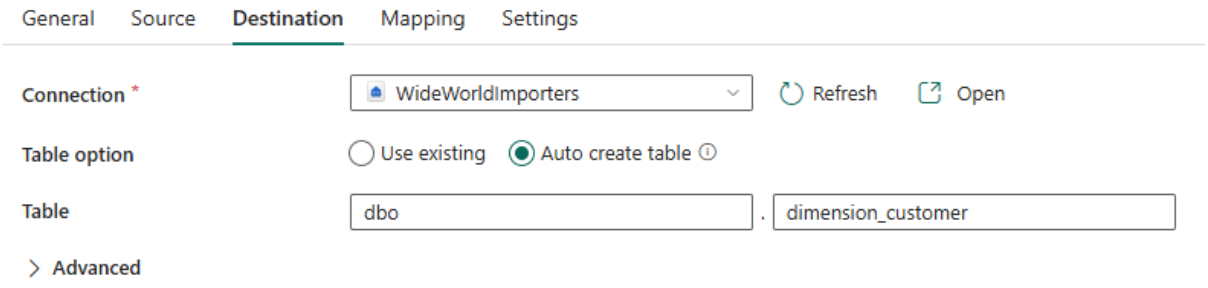


File path – Directory: wwi-raw-data/WideWorldImportersDW/tables
File path – File name: dimension_customer.parquet
File format: Parquet

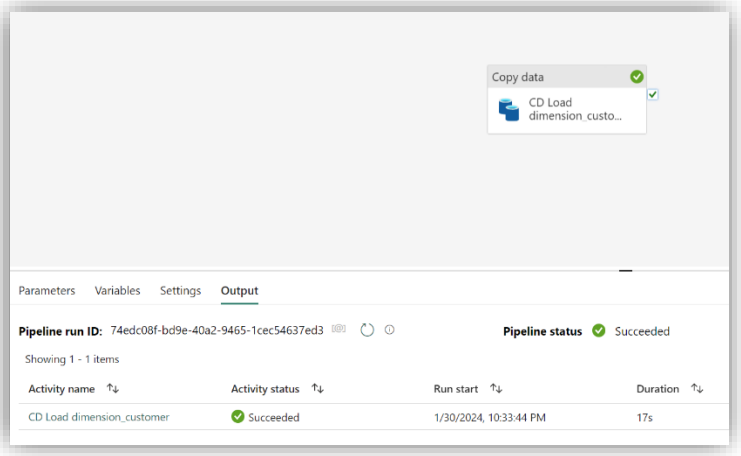
19. Select **Preview data** next to the **File path** setting to ensure there are no errors.



20. On the **Destination** page, select **Workspace** for the **Data store type**.
21. Select **Warehouse** for the **Workspace data store type**.
22. In the **Warehouse** drop down, select **WideWorldImporters** from the list.
23. Next to the **Table option** setting, select **Auto create table**
24. In the first box (schema name) next to the **Table** setting, enter **dbo**.
25. In the second box(table name) next to the **Table** setting, enter **dimension_customer**.



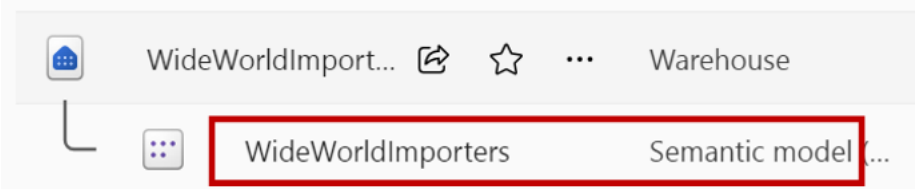
26. From the ribbon, select **Run**.
27. Select **Save and run** from the dialog box. The pipeline to load the dimension_customer table with start.
28. Monitor the copy activity's progress on the **Output** page and wait for it to complete.



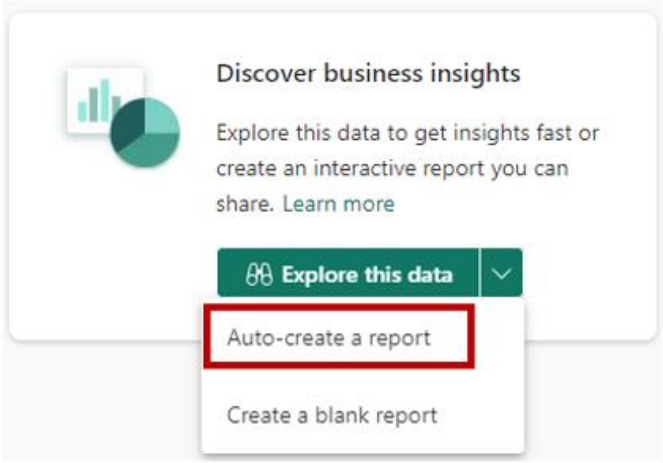
Building a report

1. Select **Data Warehouse Tutorial** in the left-hand navigation menu to return to the workspace artifact view.

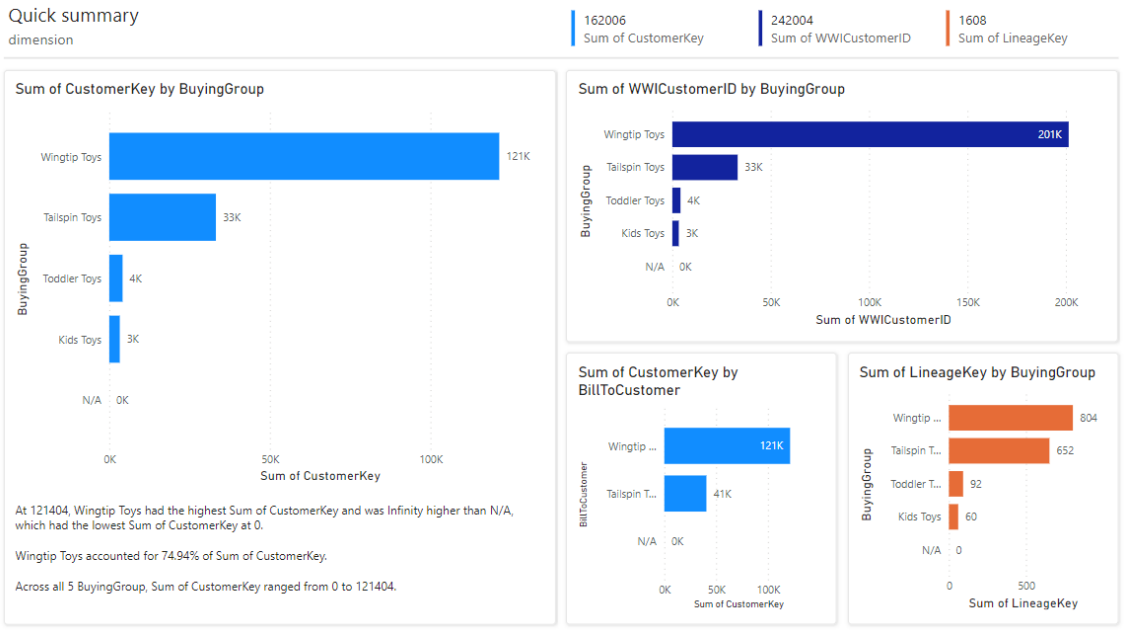
2. From the artifact list, select **WideWorldImporters** with the type of **Semantic Model (default)**.



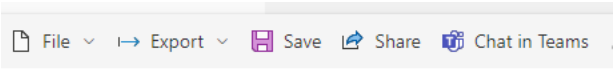
3. In the **Discover business insights** section, select **Explore this data > Auto-create a report**. A report will be generated from the dimension_customer table that was loaded in the previous section.



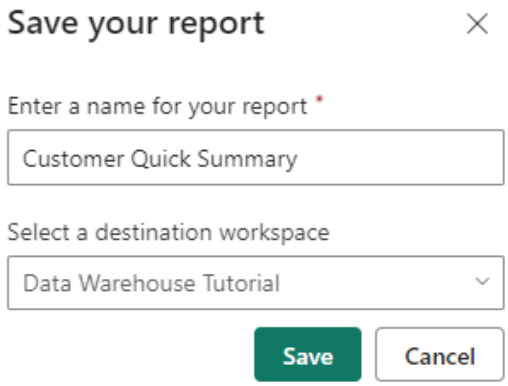
4. A report like the one shown below will be generated.



5. From the ribbon, select **Save.**



6.
7. Enter **Customer Quick Summary** in the name box.
8. Select **Save**.

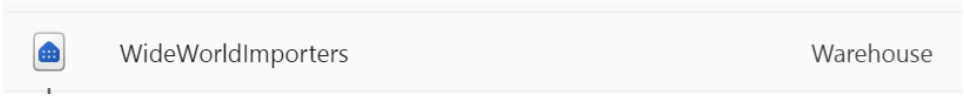


Module 3: Extending the solution

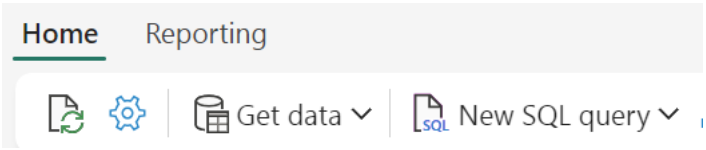
Now that you have see how to build a data warehouse, load a table, and generate a report it is time to extend the solution by exploring additional methods for loading data, querying data, and building reports.

Creating tables in the data warehouse

1. Select **Workspaces** in the left-hand menu of the Power BI service.
2. Select the workspace created in **Module 1: Getting started**, such as **Data Warehouse Tutorial**.
3. From the artifact list, select **WideWorldImporters** with the type of **Warehouse**.



4. From the ribbon, select **New SQL query**.



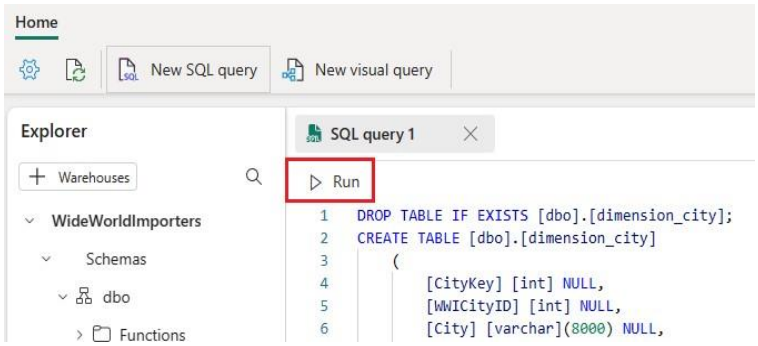
5. In the query editor, paste the code below.

   Note: In case of issues with copy/paste formatting, a text file containing the script called **Create Tables.txt** can be accessed from the Scripts folder.

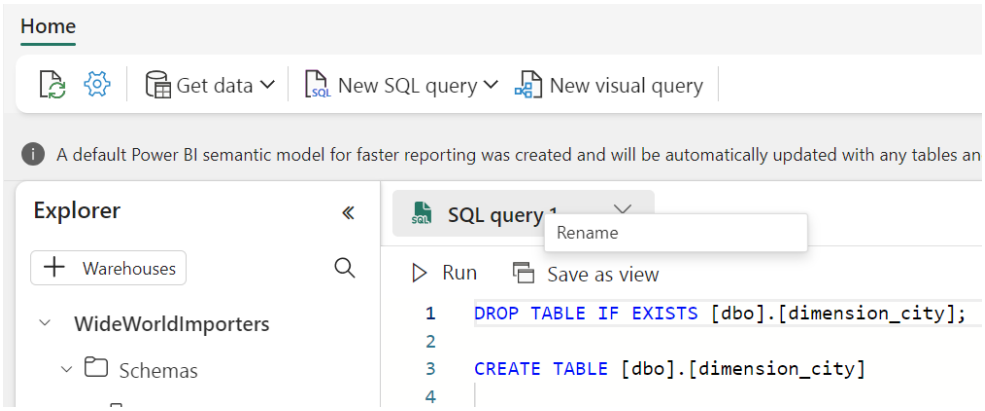
```
/*
1. Drop the dimension_city table if it already exists.
2. Create the dimension_city table.
3. Drop the fact_sale table if it already exists.
4. Create the fact_sale table.
*/
--dimension_city
DROP TABLE IF EXISTS [dbo].[dimension_city];
CREATE TABLE [dbo].[dimension_city]
    (
        [CityKey] [int] NULL,
        [WWICityID] [int] NULL,
        [City] [varchar](8000) NULL,
        [StateProvince] [varchar](8000) NULL,
        [Country] [varchar](8000) NULL,
        [Continent] [varchar](8000) NULL,
        [SalesTerritory] [varchar](8000) NULL,
        [Region] [varchar](8000) NULL,
        [Subregion] [varchar](8000) NULL,
        [Location] [varchar](8000) NULL,
        [LatestRecordedPopulation] [bigint] NULL,
        [ValidFrom] [datetime2](6) NULL,
        [ValidTo] [datetime2](6) NULL,
        [LineageKey] [int] NULL
    );


--fact_sale
DROP TABLE IF EXISTS [dbo].[fact_sale];
CREATE TABLE [dbo].[fact_sale]
    (
        [SaleKey] [bigint] NULL,
        [CityKey] [int] NULL,
        [CustomerKey] [int] NULL,
        [BillToCustomerKey] [int] NULL,
        [StockItemKey] [int] NULL,
```

```
    [InvoiceDateKey] [datetime2](6) NULL,
    [DeliveryDateKey] [datetime2](6) NULL,
    [SalespersonKey] [int] NULL,
    [WWIInvoiceID] [int] NULL,
    [Description] [varchar](8000) NULL,
    [Package] [varchar](8000) NULL,
    [Quantity] [int] NULL,
    [UnitPrice] [decimal](18, 2) NULL,
    [TaxRate] [decimal](18, 3) NULL,
    [TotalExcludingTax] [decimal](29, 2) NULL,
    [TaxAmount] [decimal](38, 6) NULL,
    [Profit] [decimal](18, 2) NULL,
    [TotalIncludingTax] [decimal](38, 6) NULL,
    [TotalDryItems] [int] NULL,
    [TotalChillerItems] [int] NULL,
    [LineageKey] [int] NULL,
    [Month] [int] NULL,
    [Year] [int] NULL,
    [Quarter] [int] NULL
);
```

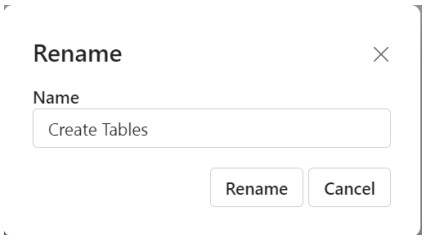
6. Select **Run** to execute the query.



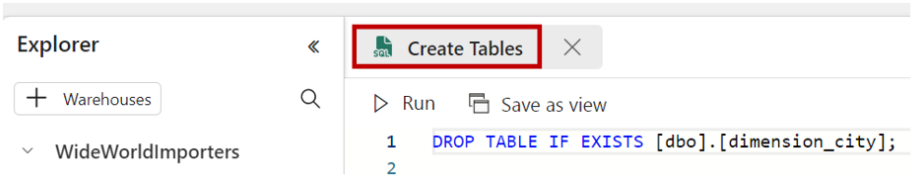
7. To save this query for reference later, right-click on the query tab just above the editor and select **Rename**.



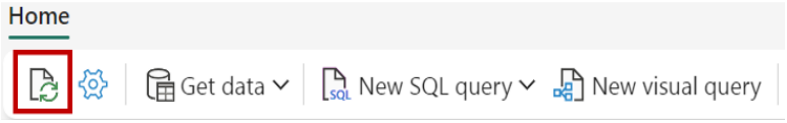
8. Type **Create Tables** to change the name of the query.



9. Click **Rename** to save the query with a given name



10. Validate the table was created successfully by clicking the **refresh** button on the ribbon.

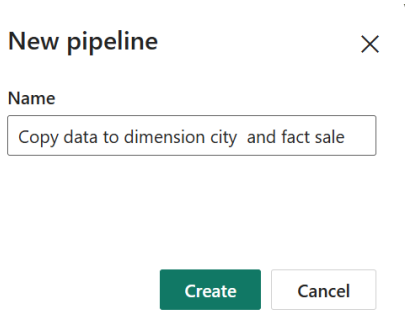


11. In the **Object explorer** verify that you can see the newly created **Create Tables** query, **fact_sale** table, and **dimension_city** table.

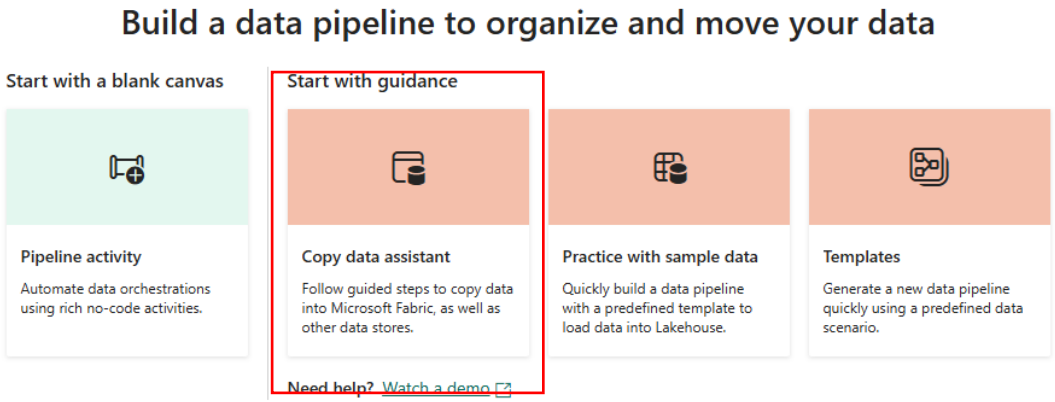
Loading data using Pipeline

1. From the Data factory experience, select **New Data Pipeline**

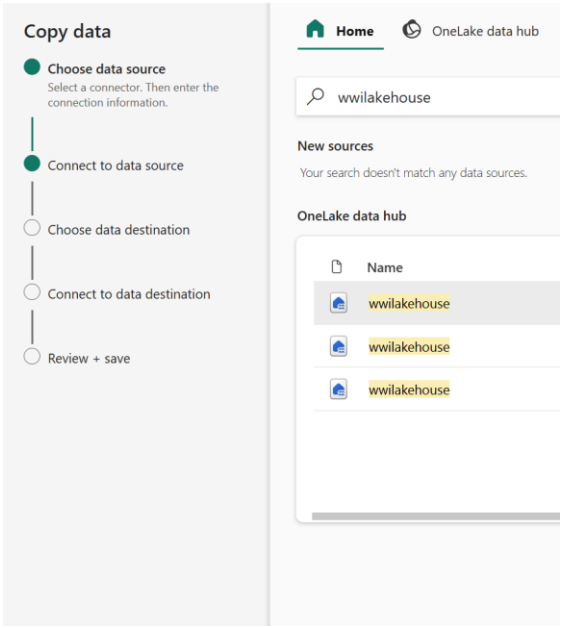
2. Name the Pipeline **Copy data to dimension city and fact sale**



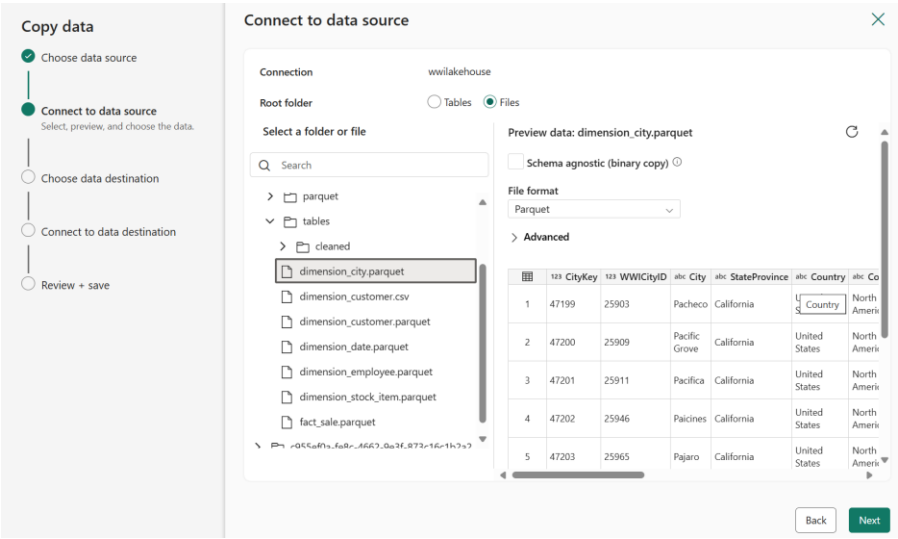
3. Click on 'Copy data assistant' in the middle of the canvas screen.



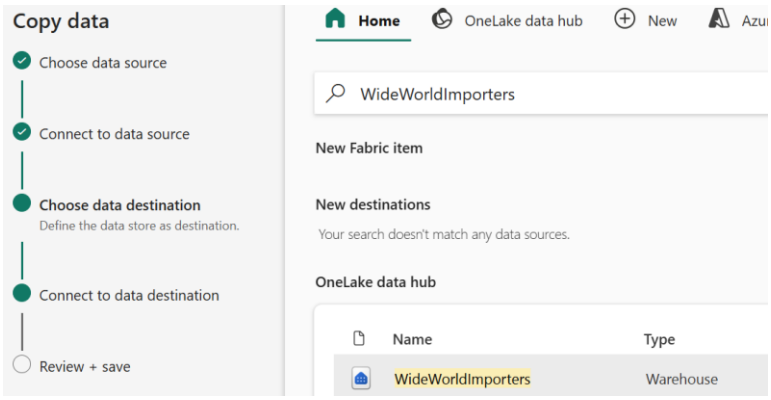
4. In the copy data assistant window, search for **wwilakehouse** and select the lakehouse you own as it will show multiple lakehouse



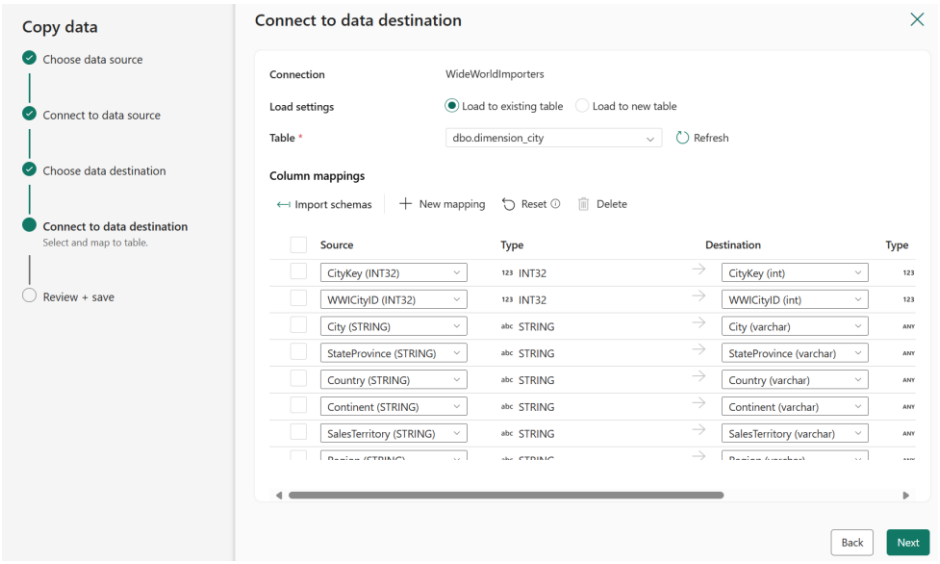
5. Select 'Files' for the Root Folder option and select the file **dimension_city.parquet** as the source file located at 'wwi-raw-data/WidWorldImportersDW/tables. Click next.



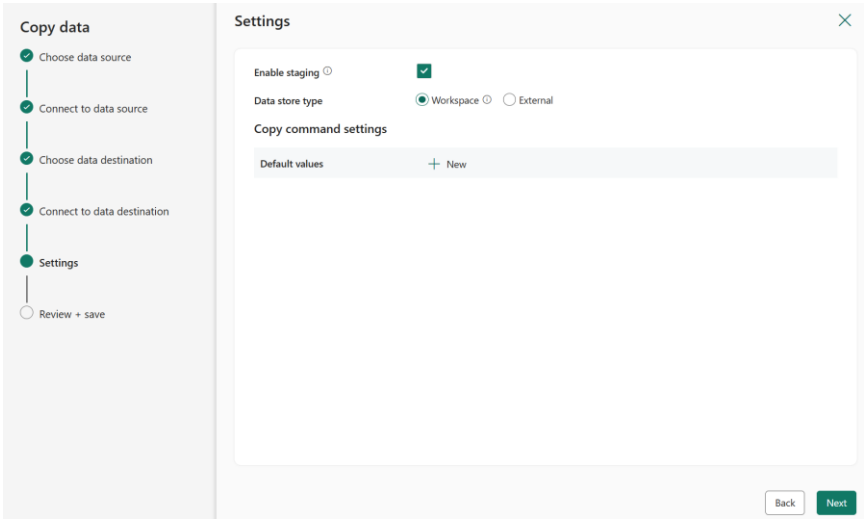
6. In destination choose warehouse **WideWorldImporters**. Select the one you own as multiple warehouses with the same name will show up



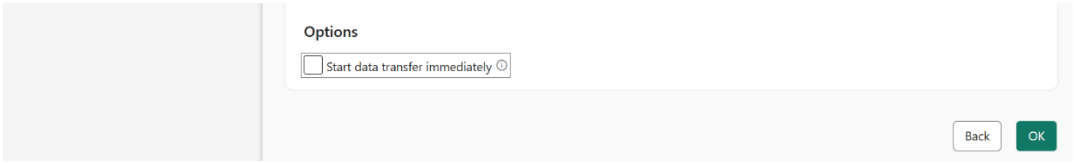
7. Leave selected 'Load to existing table' for Load Settings and ensure '**dbo.dimension_city'** is the table selected in the box next to 'Table'. Click next.



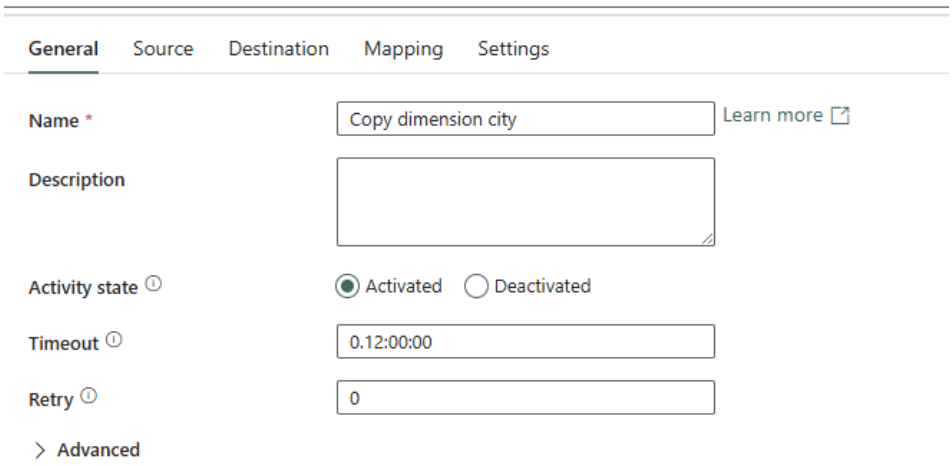
8. Make sure **enable staging** is enabled. Click next.



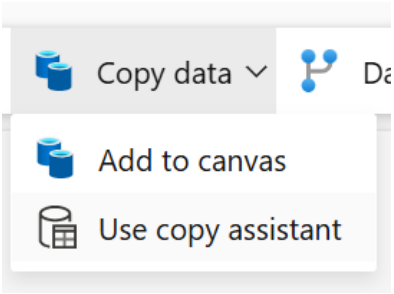
9. Make sure option **Start data transfer immediately** is unchecked as we will run it later. Click ok



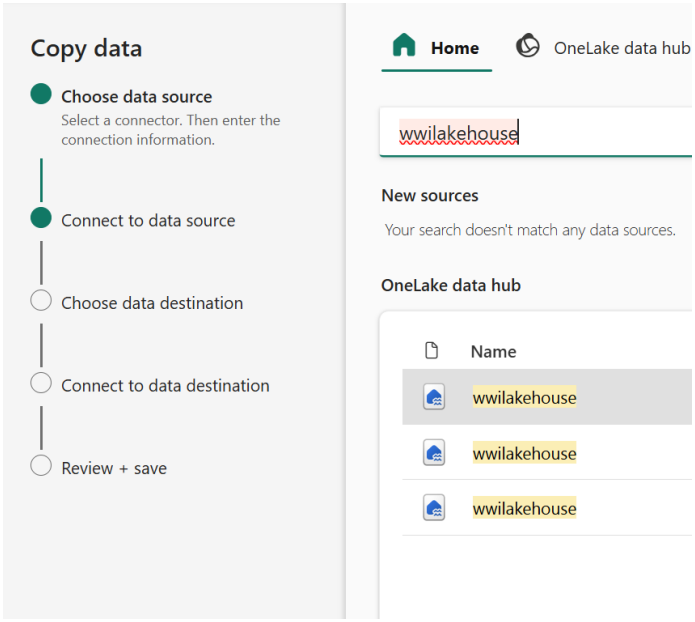
10. Rename the copy data activity as **Copy dimension city** in the general tab of the copy activity.



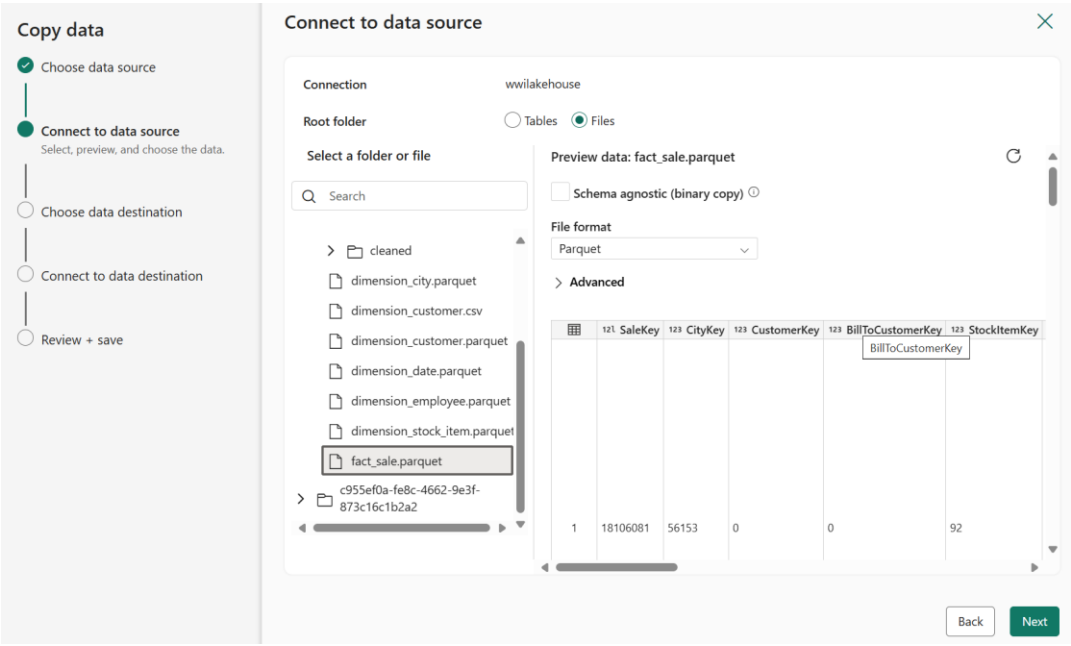
11. Add one more copy data activity to the canvas. Choose **Use copy assistant** option



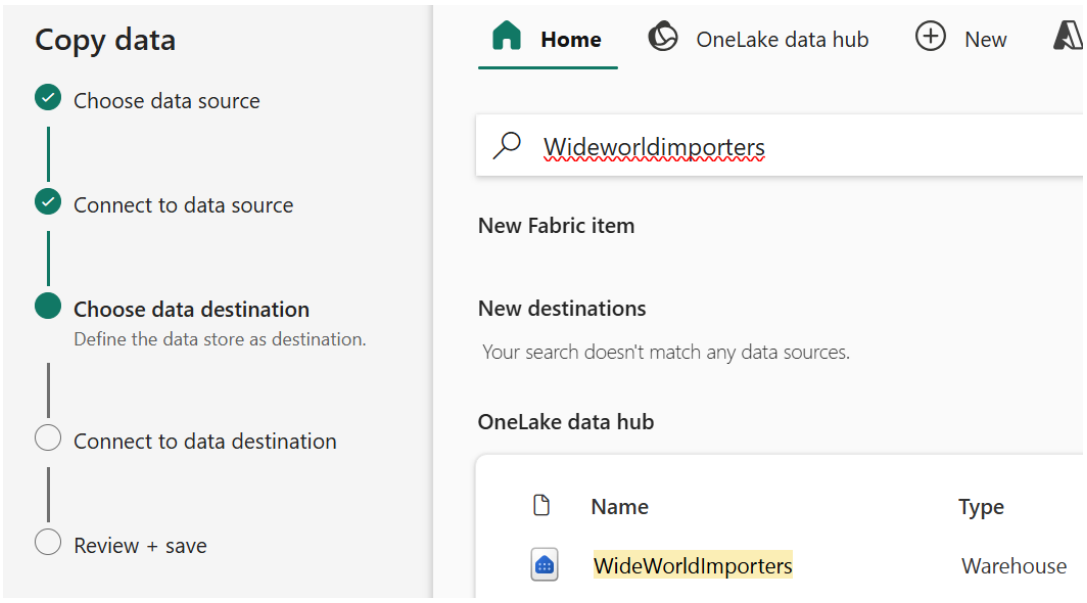
12. Select **wwilakehouse** as the lakehouse source. Make sure to select the one you own ad multiple lakehouse will show up.



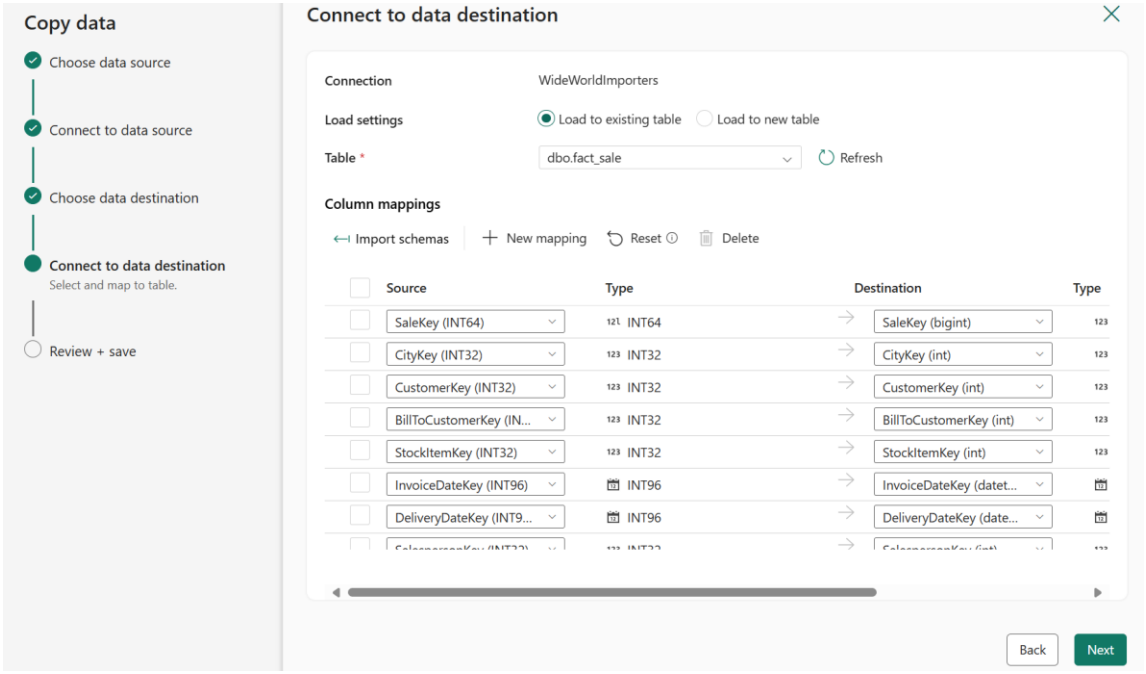
13. Select **fact_sale.parquet** as the source file located at 'wwi-raw-data/WidWorldImportersDW/tables.. Click next.

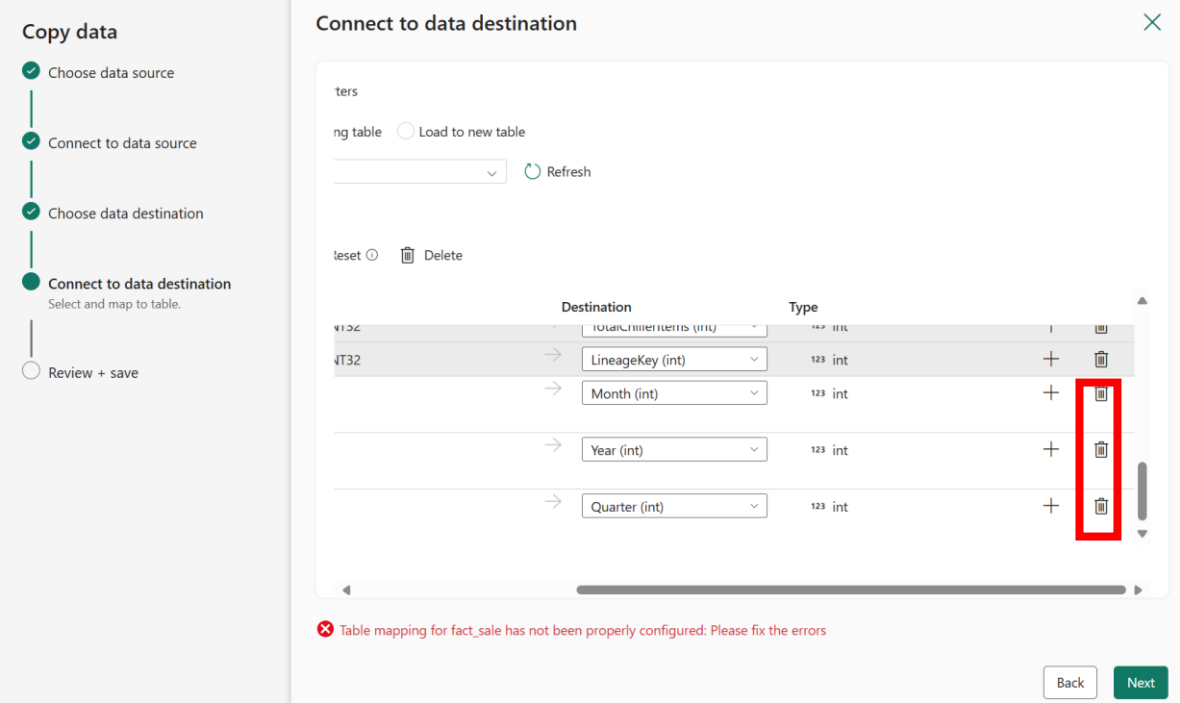


14. Choose **Wideworksimporters** warehouse as the target destination . Make sure to select the one you own as multiples will show up.

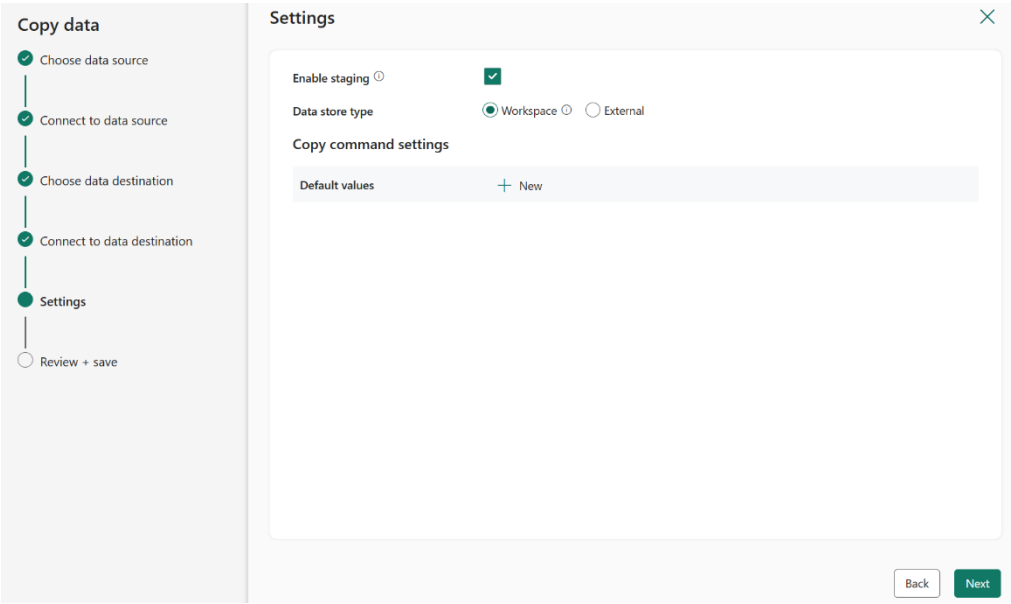


15. In the mappings tab , Make sure to delete the mapping for **Month , Year and Quarter** as the source file doesn't contain these columns. Click Next

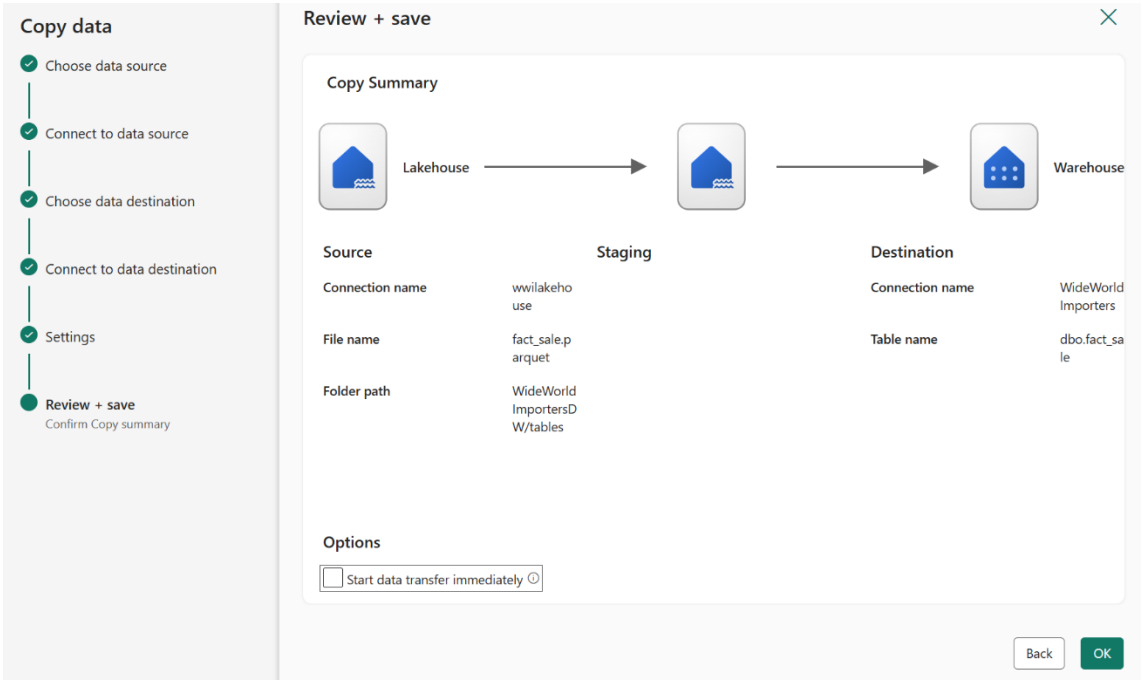




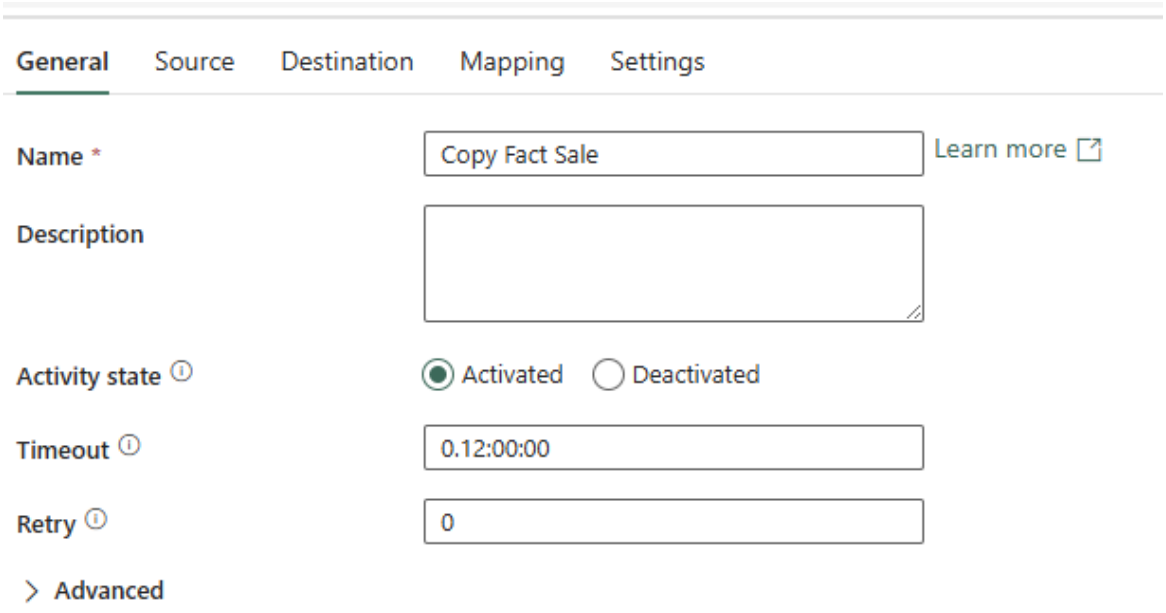
16. Make sure the **enable staging option** is checked, Click Next



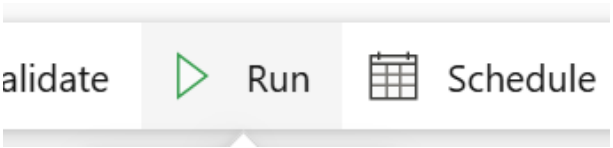
17. Make sure **Start data transfer immediately** checkbook is unchecked, Click ok



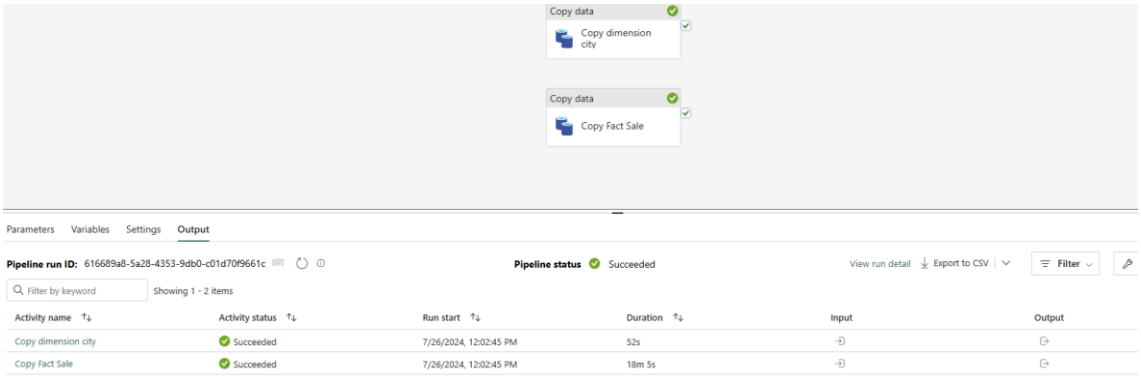
18. Rename the copy activity as **Copy Fact Sale**



19. Now that we have both the copy activity defined, Run the entire pipeline by clicking the Run option from the ribbon



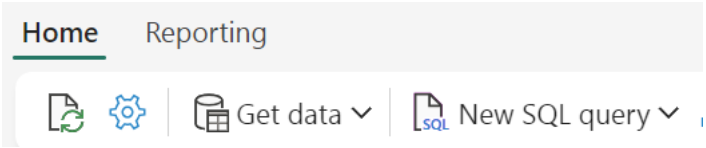
20. Monitor the pipeline for successful completion.



21. Validate in the warehouse WideWorldImporters that the tables have been loaded successfully by navigating back to your data warehouse.

Data transformation using a stored procedure

1. From the **Home** tab of the ribbon, select **New SQL query**.

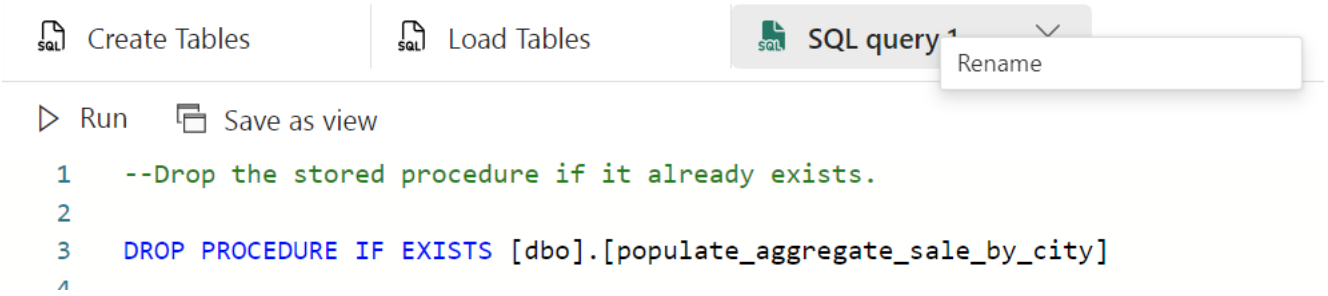


2. In the query editor, paste the code below.
   Note: In case of issues with copy/paste formatting, a text file containing the script called **Create Aggregate Procedure.txt** from the Scripts folder .

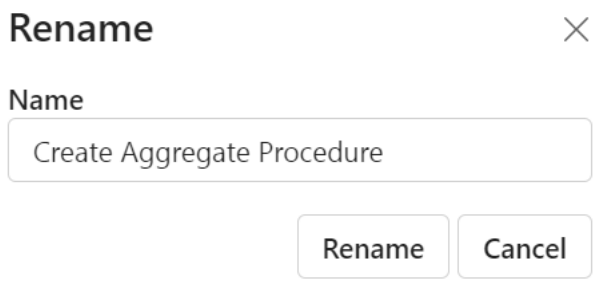
```
--Drop the stored procedure if it already exists.
DROP PROCEDURE IF EXISTS [dbo].[populate_aggregate_sale_by_city]
GO


--Create the populate_aggregate_sale_by_city stored procedure.
CREATE PROCEDURE [dbo].[populate_aggregate_sale_by_city]
AS
BEGIN
    --If the aggregate table already exists, drop it. Then create the table.
    DROP TABLE IF EXISTS [dbo].[aggregate_sale_by_date_city];
    CREATE TABLE [dbo].[aggregate_sale_by_date_city]
        (
            [Date] [DATETIME2](6),
            [City] [VARCHAR](8000),
            [StateProvince] [VARCHAR](8000),
            [SalesTerritory] [VARCHAR](8000),
            [SumOfTotalExcludingTax] [DECIMAL](38,2),
            [SumOfTaxAmount] [DECIMAL](38,6),
            [SumOfTotalIncludingTax] [DECIMAL](38,6),
            [SumOfProfit] [DECIMAL](38,2)
```

```
        );

    --Reload the aggregated dataset to the table.
    INSERT INTO [dbo].[aggregate_sale_by_date_city]
    SELECT
        FS.[InvoiceDateKey] AS [Date],
        DC.[City],
        DC.[StateProvince],
        DC.[SalesTerritory],
        SUM(FS.[TotalExcludingTax]) AS [SumOfTotalExcludingTax],
        SUM(FS.[TaxAmount]) AS [SumOfTaxAmount],
        SUM(FS.[TotalIncludingTax]) AS [SumOfTotalIncludingTax],
        SUM(FS.[Profit]) AS [SumOfProfit]
    FROM [dbo].[fact_sale] AS FS
    INNER JOIN [dbo].[dimension_city] AS DC
        ON FS.[CityKey] = DC.[CityKey]
    GROUP BY
        FS.[InvoiceDateKey],
        DC.[City],
        DC.[StateProvince],
        DC.[SalesTerritory]
    ORDER BY
        FS.[InvoiceDateKey],
        DC.[StateProvince],
        DC.[City];
END
```

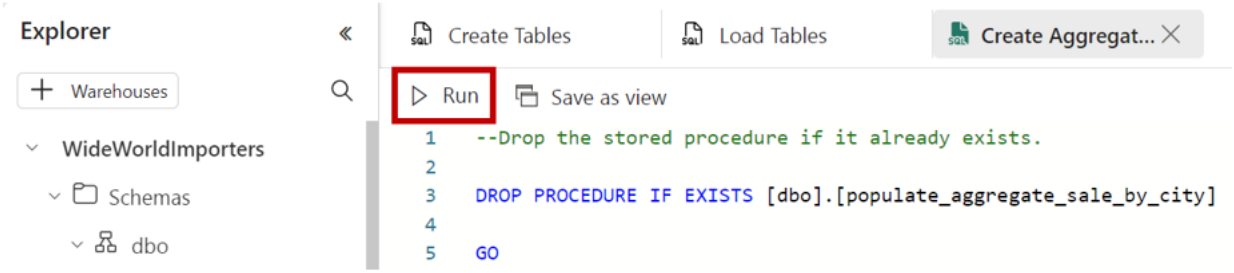
1. To save this query for reference later, right-click on the query tab just above the editor and select **Rename**.



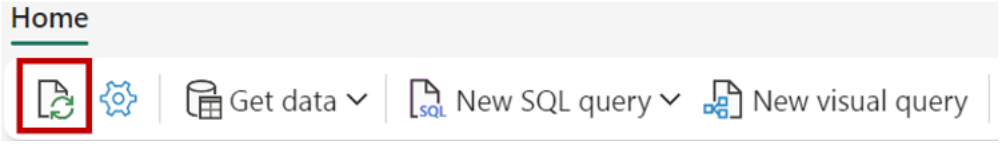
2. Type **Create Aggregate Procedure** to change the name of the query.



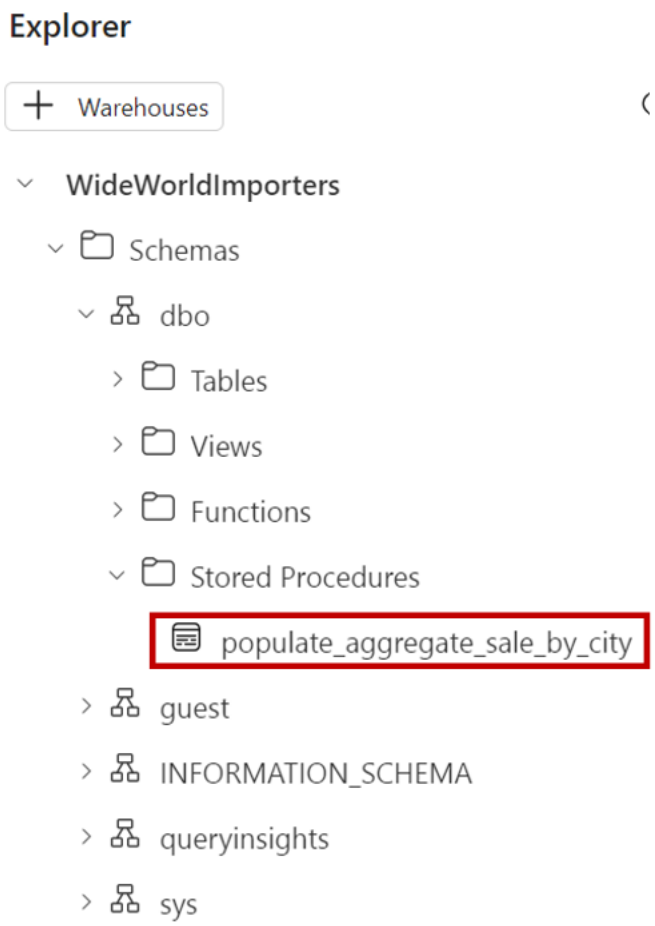
3. Click on **Rename** to save the query with given name
4. Select **Run** to execute the query.



5. Click the **refresh** button on the ribbon.



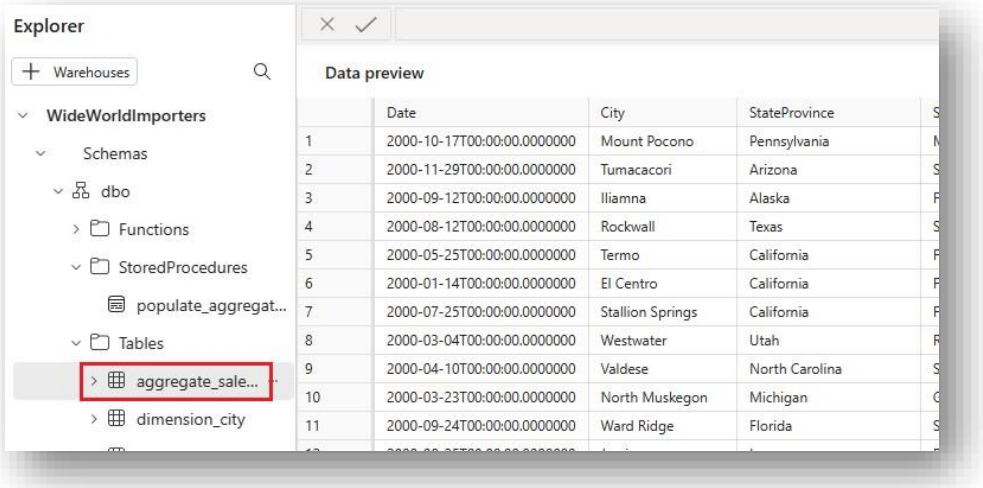
6. In the **Object explorer** verify that you can see the newly created stored procedure by expanding the **StoredProcedures** node under the **dbo** schema.



7. From the **Home** tab of the ribbon, select **New SQL query**.
8. In the query editor, paste the code below.
   Note: In case of issues with copy/paste formatting, a text file containing the script called **Run Aggregate Procedure.txt** can be accessed from the Scripts folder .

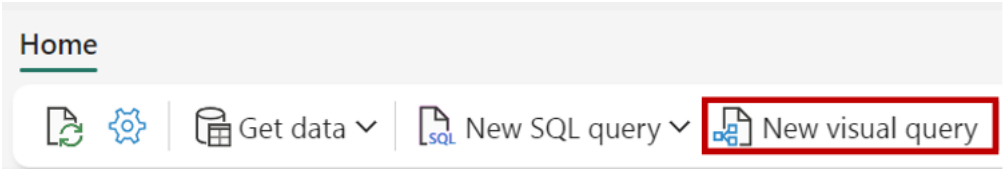
```
--Execute the stored procedure to create the aggregate table.
EXEC [dbo].[populate_aggregate_sale_by_city];
```

9. To save this query for reference later, right-click on the query tab just above the editor and select **Rename**.
10. Type **Run Create Aggregate Procedure** to change the name of the query.
11. Select **Run** to execute the query.
12. Click the **refresh** button on the ribbon. The query will take between 2 and 3 minutes to execute.
13. In the **Object explorer**, load the data preview to validate the data loaded successfully by clicking on the **aggregate_sale_by_city** table in the **Explorer**.

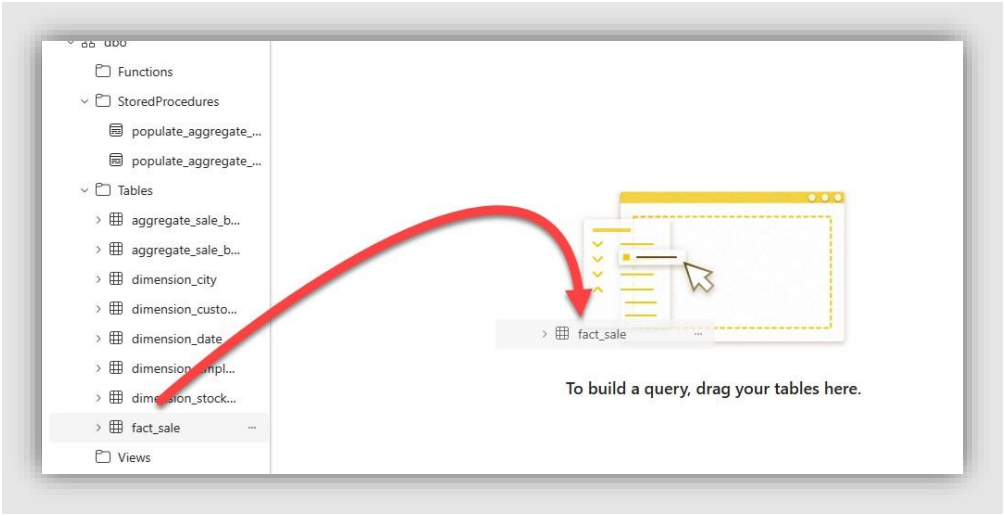


Using the visual query builder

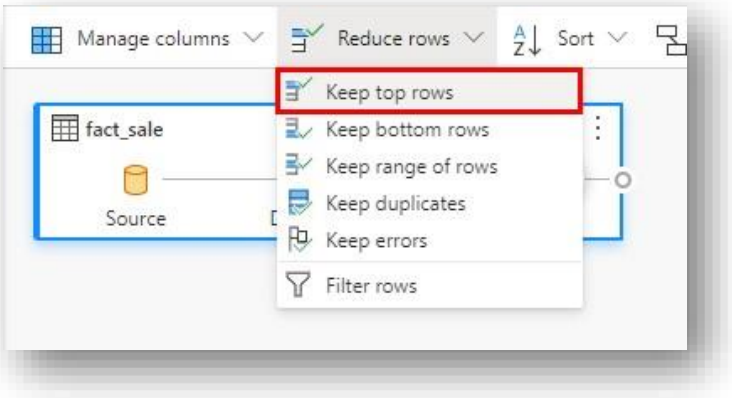
1. From the **Home** tab of the ribbon, select **New visual query**.



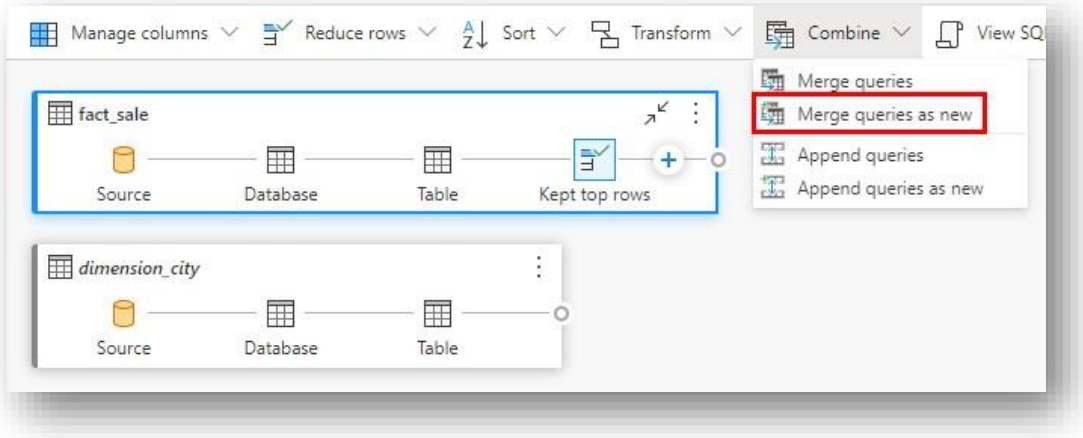
2. Drag the **fact_sale** table from the explorer to the query design pane.



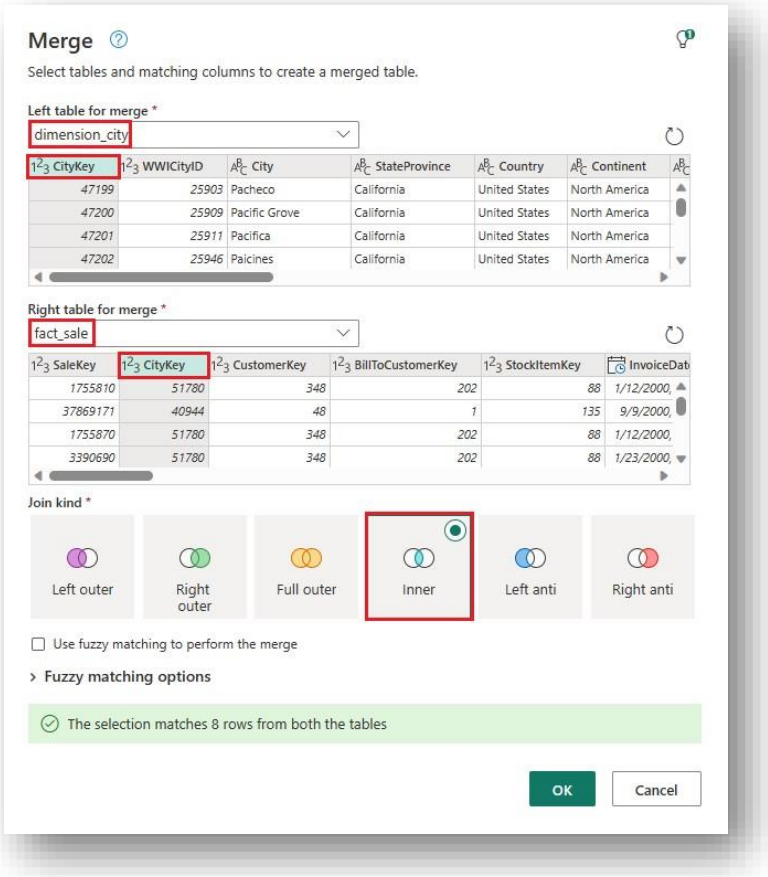
3. Limit the dataset size by selecting **Reduce rows > Keep top rows** from the transformations ribbon.



4. In the **Keep top rows** dialog enter **10,000**.
5. Select **OK**.
6. Drag the **dimension_city** table from the explorer to the query design pane.
7. From the transformations ribbon, select the dropdown next to **Combine** and select **Merge queries as new**.

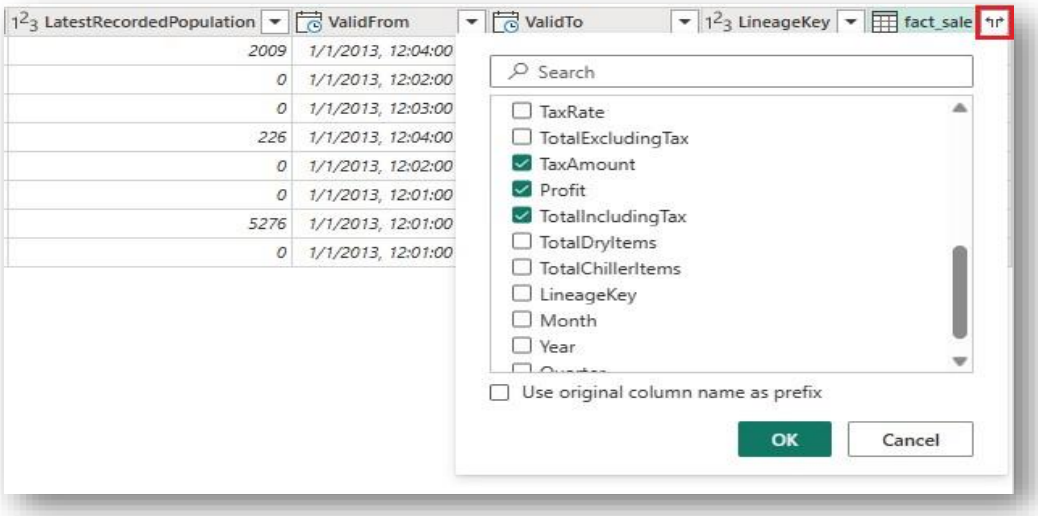


8. On the **Merge** settings page:
    a. **Left table for merge:** dimension_city
    b. **Right table for merge:** fact_sale
    c. Select the **CityKey** field in the **dimension_city** table by clicking on the column name in the header row to indicate the join column.
    d. Select the **CityKey** field in the **fact_sale** table by clicking on the column name in the header row to indicate the join column.
    e. **Join kind:** Inner



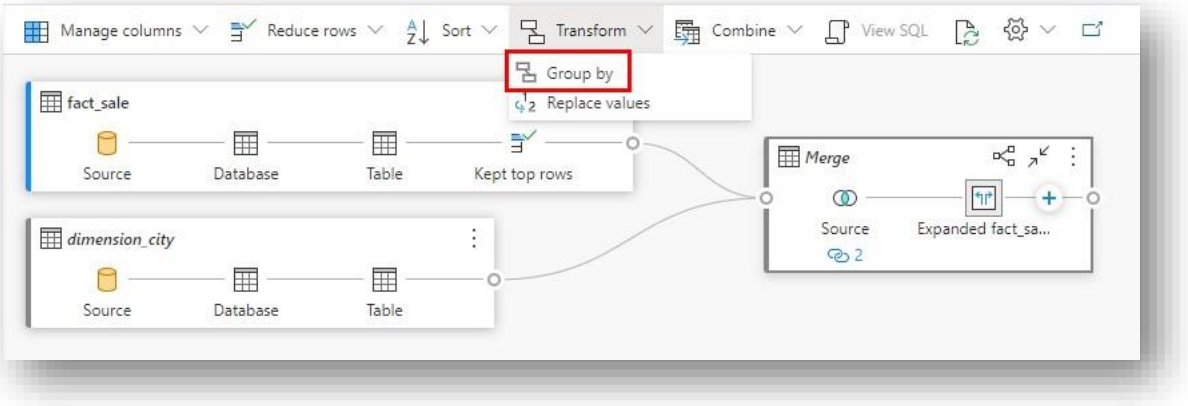
Select **OK**.

2. With the **Merge** step selected, select the **Expand** button next to **fact_sale** on the header of the data grid then select only **TaxAmount, Profit,** and **TotalIncludingTax.**

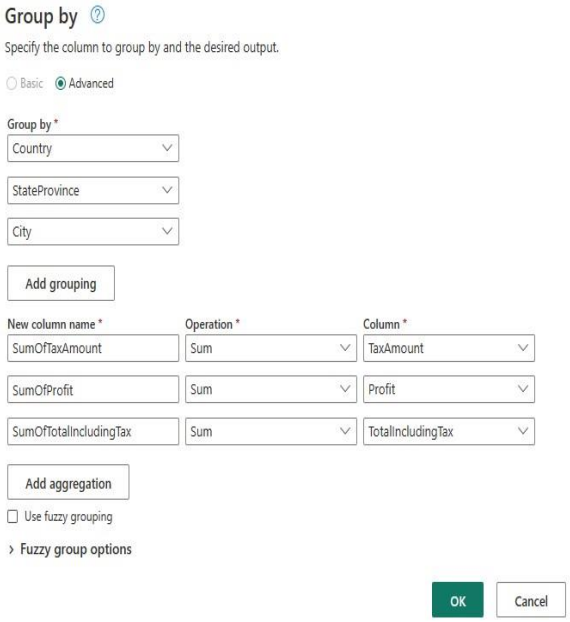


9. Select **OK**.

10. Select **Transform > Group by** from the transformations ribbon.

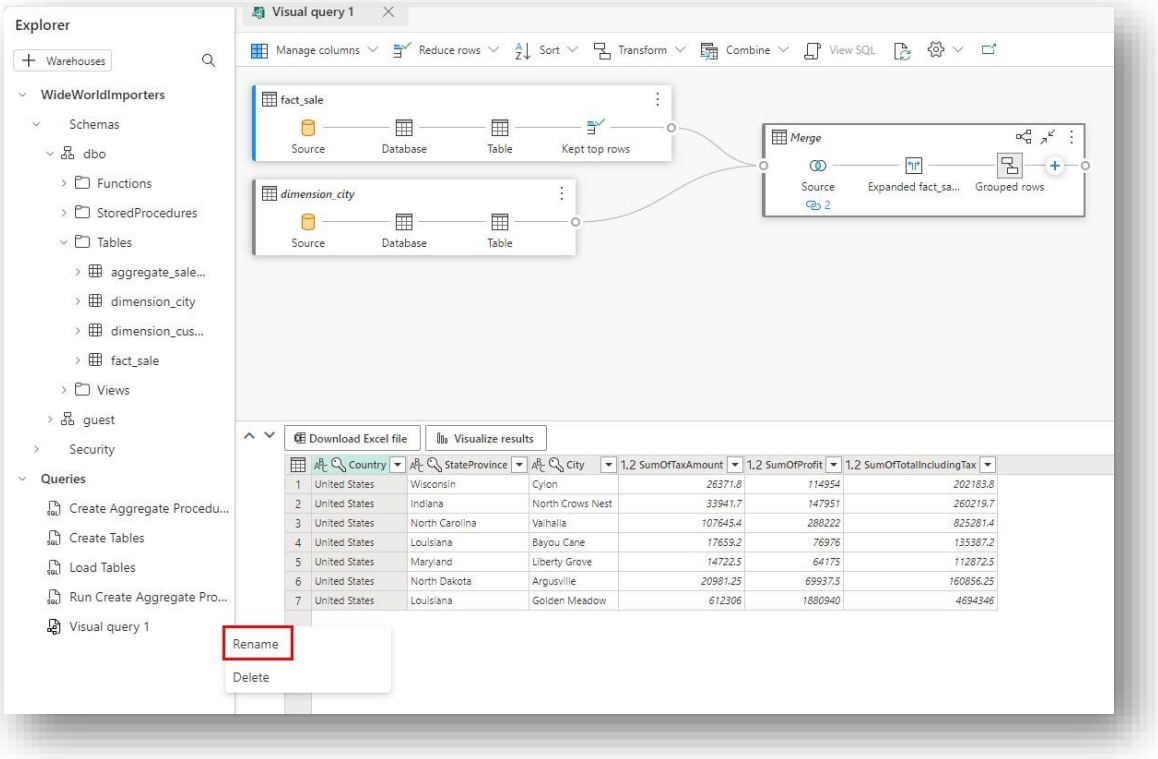


11. On the **Group by** settings page:
    a. Change to **Advanced**.
    b. **Group by** (if necessary, select **Add grouping** to add additional group by columns)**:** i. Country
        ii. StateProvince
        iii. City
    c. **New column name** (if necessary, select **Add aggregation** to add additional aggregate columns and operations)**:**
        i. **SumOfTaxAmount** with **Operation** of **Sum** and **Column** of **TaxAmount**
        ii. **SumOfProfit** with **Operation** of **Sum** and **Column** of **Profit** iii. **SumOfTotalIncludingTax** with **Operation** of **Sum** and **Column** of **TotalIncludingTax**



Select **OK**.

12. Right-click on **Visual query 1** in the explorer and select **Rename**.

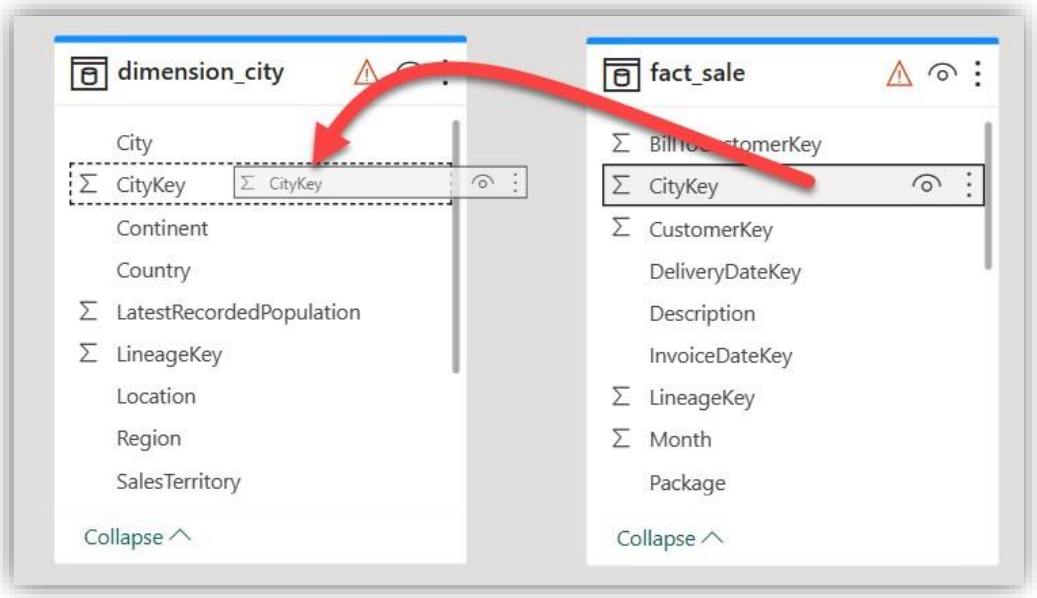


13. Type **Sales Summary** to change the name of the query.
14. Press **Enter** on the keyboard or click off anywhere outside the tab to save the change.

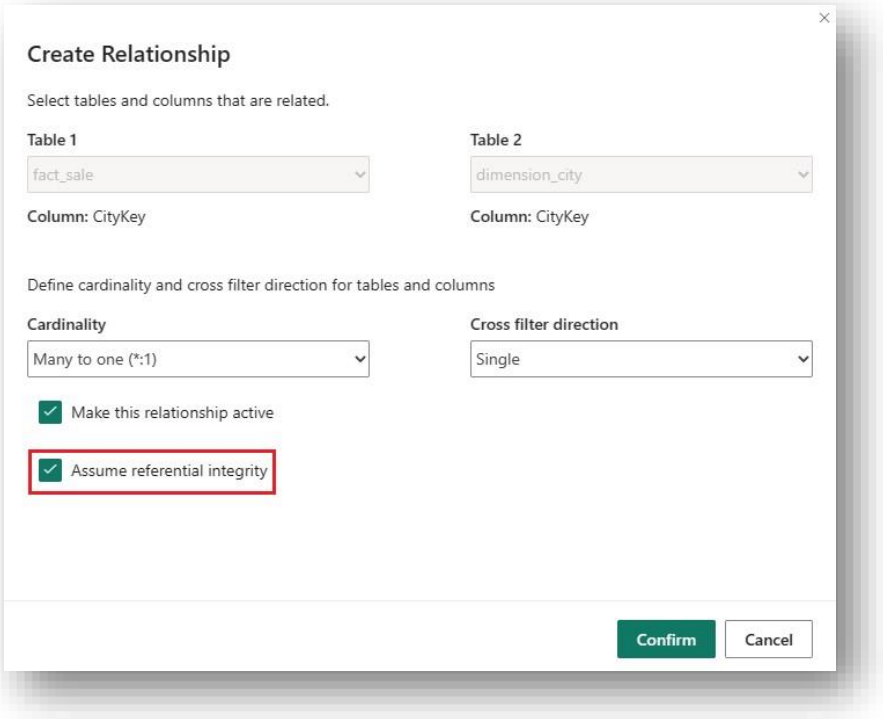
Create a Power BI report

1. Select the **Model** view from the options in the bottom left corner, just outside the canvas.

2. From the **fact_sale** table, drag the **CityKey** field and drop it on the **CityKey** field in the **dimension_city** table to create a relationship.

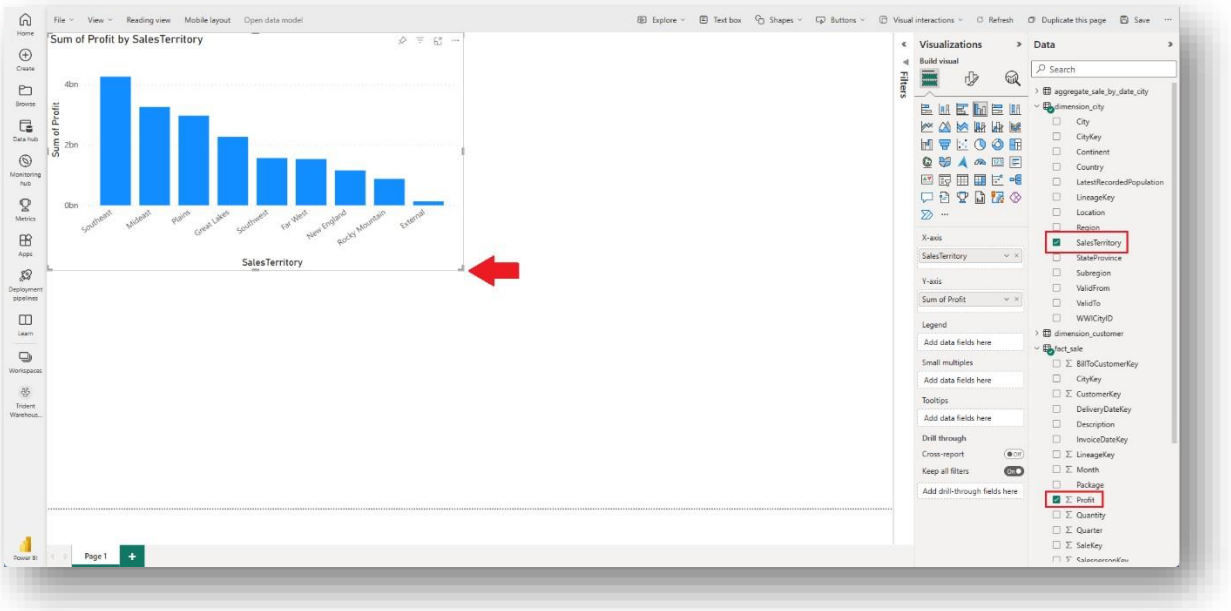


3. On the **Create Relationship** settings:
    a. Table 1 will be populated with fact_sale and the column of CityKey.
    b. Table 2 will be populated with dimension_city and the column of CityKey.
    c. Cardinality: **Many to one (*:1)**
    d. Cross filter direction: **Single**
    e. Leave the box next to **Make this relationship active** checked.
    f. Check the box next to **Assume referential integrity.**

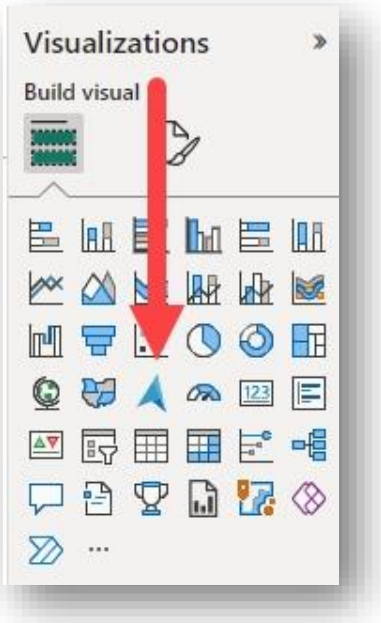


Select **Confirm**.

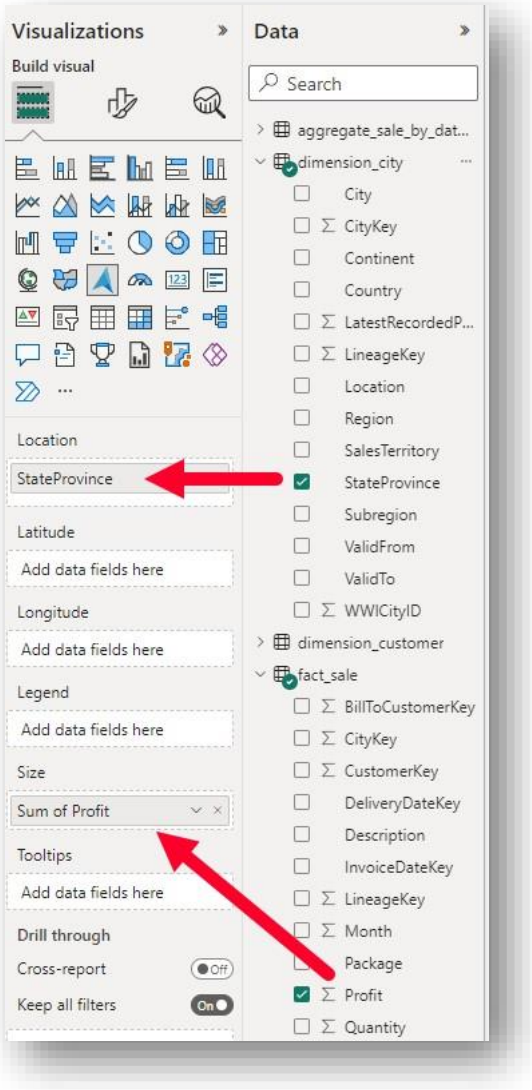
4. From the **Home** tab of the ribbon, select **New report**.
5. Build a column chart visual:
    a. On the **Data** pane, expand **fact_sales** and check the box next to **Profit**. This will create a column chart and add the field to the Y-axis.
    b. On the **Data** pane, expand **dimension_city** and check the box next to **SalesTerritory**. This will add the field to the X-axis.
    c. Reposition and resize the column chart to take up the top left quarter of the canvas by dragging the anchor points on the corners of the visual.



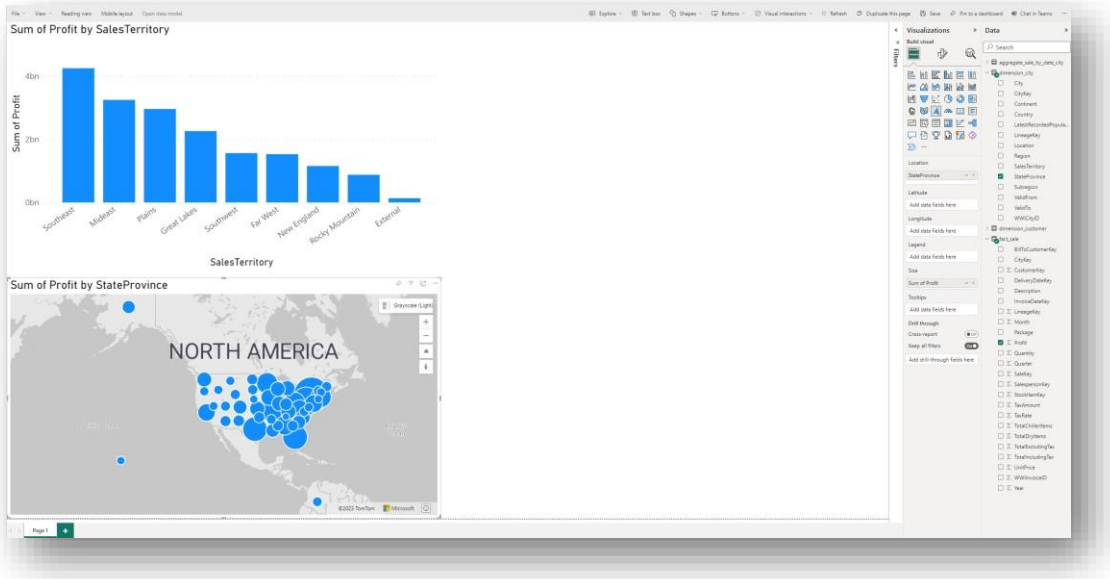
6. Click anywhere on the blank canvas (or press the Esc key) so the column chart visual is no longer selected.
7. Build a map visual:
    a. On the **Visualizations** pane, select the **Azure Map for Power BI** visual. **Azure Map** visual needs to be enabled by PowerBI admin



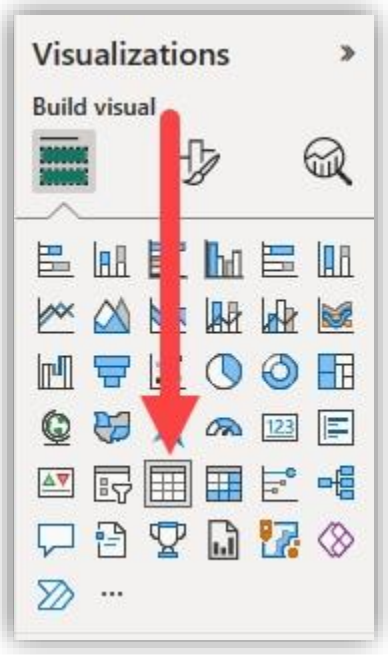
    b. From the **Data** pane, drag **StateProvince** from the **dimension_city** table to the **Location** bucket on the **Visualizations** pane.
    c. From the **Data** pane, drag **Profit** from the **fact_sale** table to the **Size** bucket on the **Visualizations** pane.



d. If necessary, reposition and resize the map to take up the bottom left quarter of the canvas by dragging the anchor points on the corners of the visual.



8. Click anywhere on the blank canvas (or press the Esc key) so the map visual is no longer selected.
9. Build a table visual:
    a. On the **Visualizations** pane, select the **Table** visual.

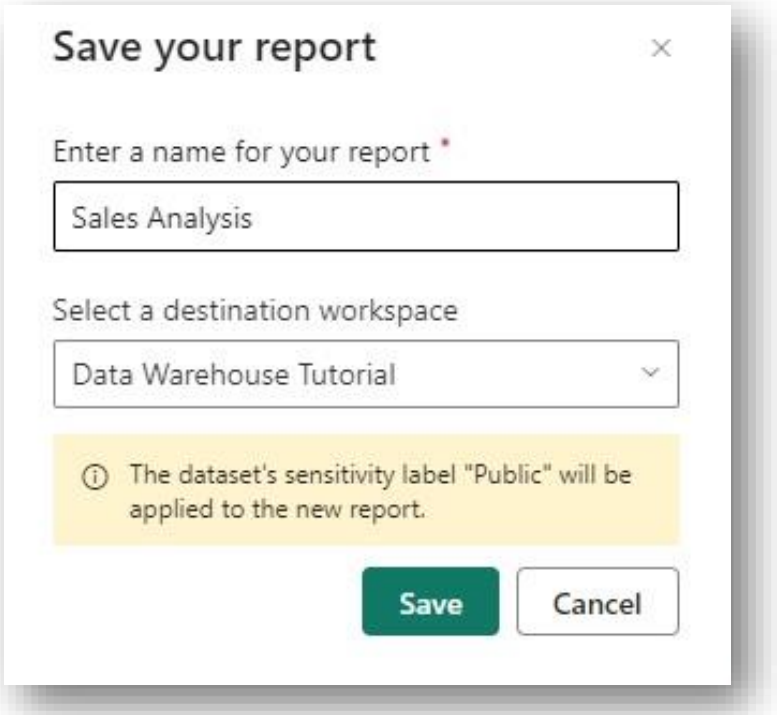


    b. From the **Data** pane, check the box next to **SalesTerritory** on the **dimension_city** table.
    c. From the **Data** pane, check the box next to **StateProvince** on the **dimension_city** table.
    d. From the **Data** pane, check the box next to **Profit** on the **fact_sale** table.
    e. From the **Data** pane, check the box next to **TotalExcludingTax** on the **fact_sale** table.

f. Reposition and resize the column chart to take up the right half of the canvas by dragging the anchor points on the corners of the visual.

SalesTerritory	StateProvince	Sum of Profit	Sum of TotalExcludingTax
External	Puerto Rico (US Territory)	133,582,372.50	258,769,045.00
Far West	Alaska	314,778,451.00	663,162,614.50
Far West	California	550,018,742.50	1,182,776,335.00
Far West	Hawaii	59,785,440.00	122,872,640.00
Far West	Nevada	87,382,492.00	176,047,816.00
Far West	Oregon	106,145,152.00	218,259,200.00
Far West	Washington	411,052,982.00	770,145,928.00
Great Lakes	Illinois	416,926,103.50	932,228,425.00
Great Lakes	Indiana	284,942,917.75	572,777,509.40
Great Lakes	Michigan	427,138,787.20	918,841,502.40
Great Lakes	Ohio	732,510,197.00	1,562,912,470.00
Great Lakes	Wisconsin	399,256,195.20	833,223,166.80
Mideast	Maryland	98,287,073.50	176,450,988.00
Mideast	New Jersey	555,537,109.50	1,070,499,457.00
Mideast	New York	1,482,546,963.90	2,681,614,839.40
Mideast	Pennsylvania	1,112,083,758.10	2,196,160,127.20
New England	Connecticut	47,040,918.40	88,664,246.40
New England	Maine	160,358,720.00	332,808,752.00
New England	Massachusetts[E]	709,774,846.30	1,533,238,994.20
New England	New Hampshire	85,134,049.00	255,283,250.00
New England	Vermont	151,622,175.50	271,023,963.00
Plains	Iowa	465,282,023.00	799,006,004.00
Plains	Kansas	424,714,480.50	842,428,141.80
Plains	Minnesota	742,880,436.00	1,560,372,792.00
Plains	Missouri	899,121,511.50	1,570,464,279.00
Plains	Nebraska	58,827,072.00	121,401,856.00
Plains	North Dakota	206,578,188.00	497,327,800.00
Plains	South Dakota	164,298,658.40	330,239,697.60
Rocky Mountain	Colorado	358,748,678.90	734,110,535.40
Rocky Mountain	Idaho	104,988,915.00	224,493,403.00
Rocky Mountain	Montana	104,259,472.00	218,817,288.00
Rocky Mountain	Utah	149,874,176.00	297,357,792.00
Rocky Mountain	Wyoming	158,427,288.00	344,167,674.00
Total		**17,980,267,019.75**	**36,186,702,997.75**

10. From the ribbon, select **File > Save**.
11. Enter the name of your report as **Sales Analysis**.
12. Select **Save**.

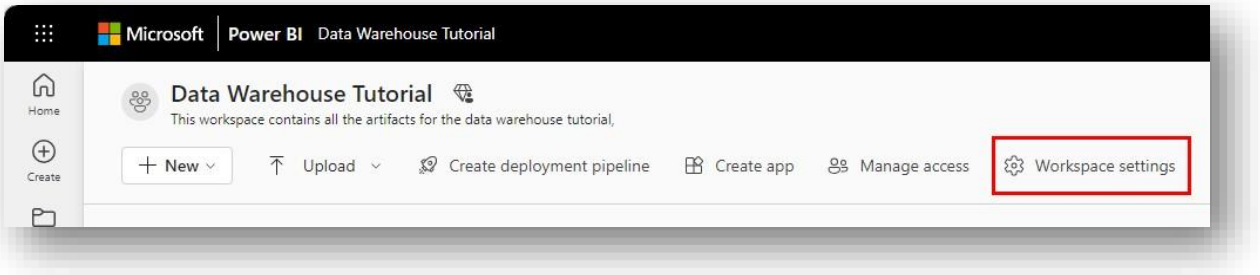


Module 4: Clean up resources

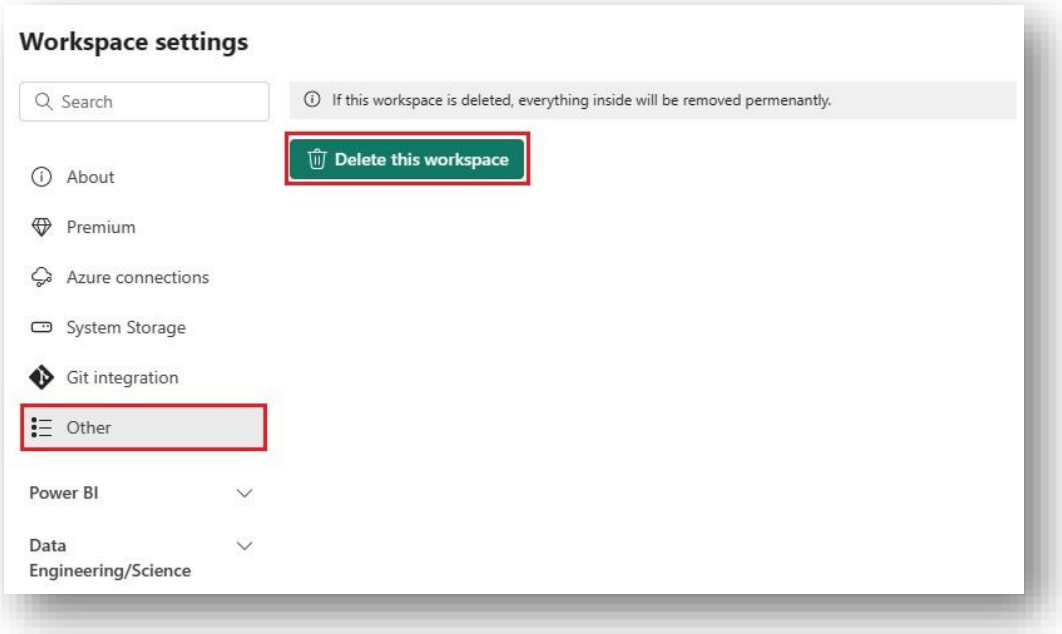
You can delete individual reports, pipelines, warehouses, and other items or remove the entire workspace.

1. Select **Data Warehouse Tutorial** in the left-hand navigation menu to return to the workspace artifact view.

2. Below the workspace name and description at the top of the workspace header, select **Workspace settings**.



3. Select **Other > Delete this workspace**.



4. Select **Delete** on the warning.