Microsoft Fabric in a Day Lab Manual – **Lab 2**

Author: Will Crayger
Email: wcrayger@lucidbi.co
LinkedIn: LinkedIn.com/in/willcrayger/
Course Material: GitHub.com/Lucid-Will/FabCon-EU-Zero-To-Hero-with-Fabric

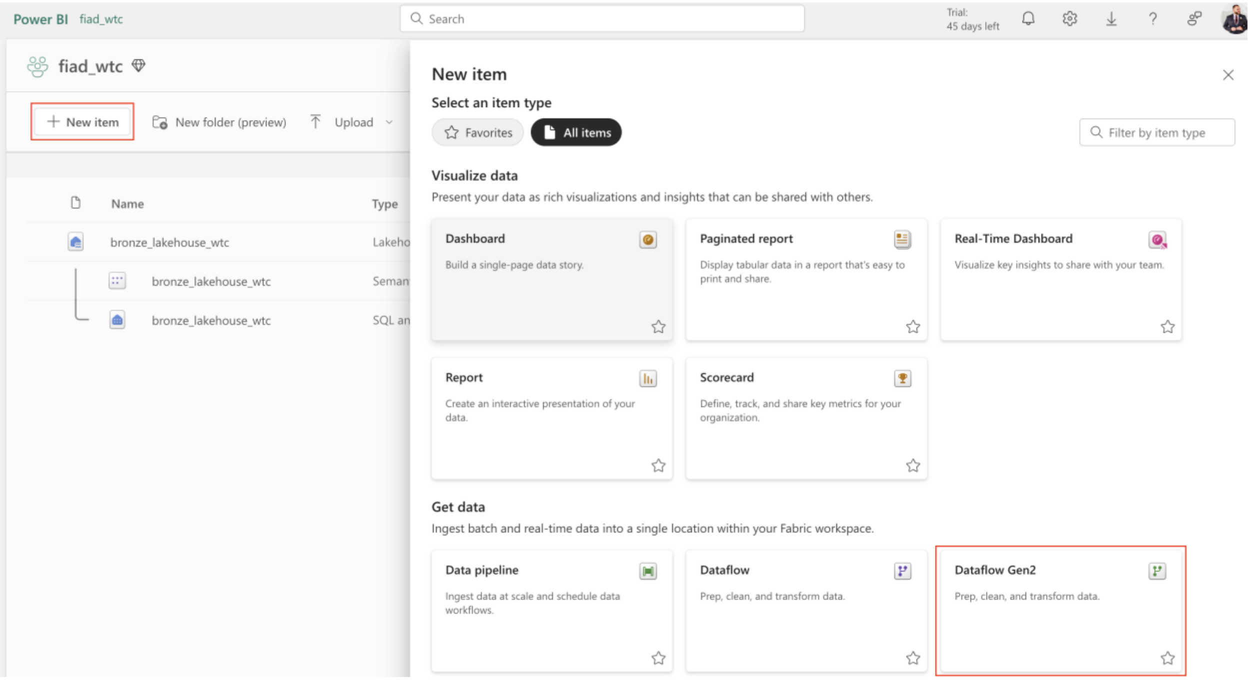
Lab 2: Dataflows Generation 2 (Gen2) – Creating a Location Table

Introduction

This lab will explore Fabric's low/no-code data integration workload, Dataflows gen2. The steps below will walk through creating a Dataflow, connecting it to a sample database, filtering by the necessary schema, and integrating base tables that will be used in the following labs.

Creating the Dataflow Gen2

Creating the Dataflow Gen2: Select **Workspaces** from the left-side navigation blade and click your workspace name **fiad_{your_initials}**, e.g. fiad_wtc. This is your central hub for creating and managing all Fabric workloads. Select the **New item** dropdown and select the **Dataflow Gen2** workload from the menu that appears. Rename your dataflow by selecting Dataflow 1 in the top-left corner, type **locations**, and pressing Enter.



Present your data as rich visualizations and insights that can be shared with others.

Dashboard: Build a single-page data story.

Paginated report: Display tabular data in a report that's easy to print and share.

Real-Time Dashboard: Visualize key insights to share with your team.

Report: Create an interactive presentation of your data.

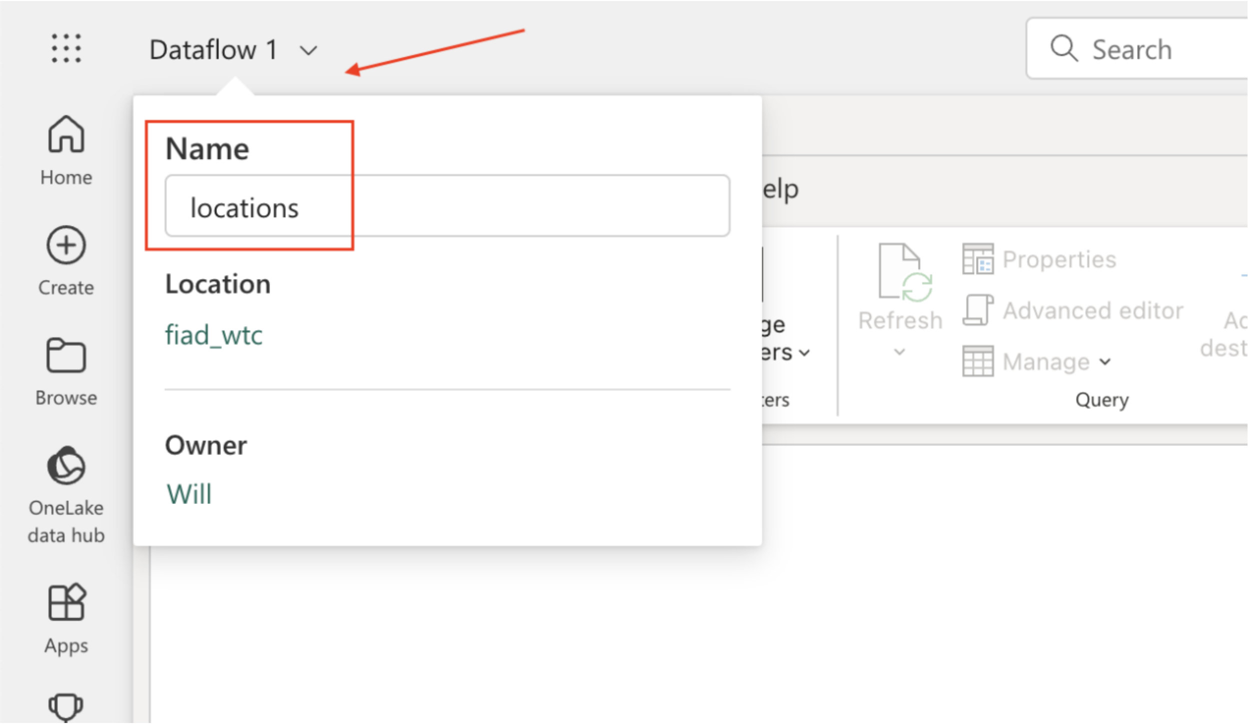
Scorecard: Define, track, and share key metrics for your organization.

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

Data pipeline: Ingest data at scale and schedule data workflows.

Dataflow: Prep, clean, and transform data.

Dataflow Gen2: Prep, clean, and transform data.



Name

locations

Location

fiad_wtc

Owner

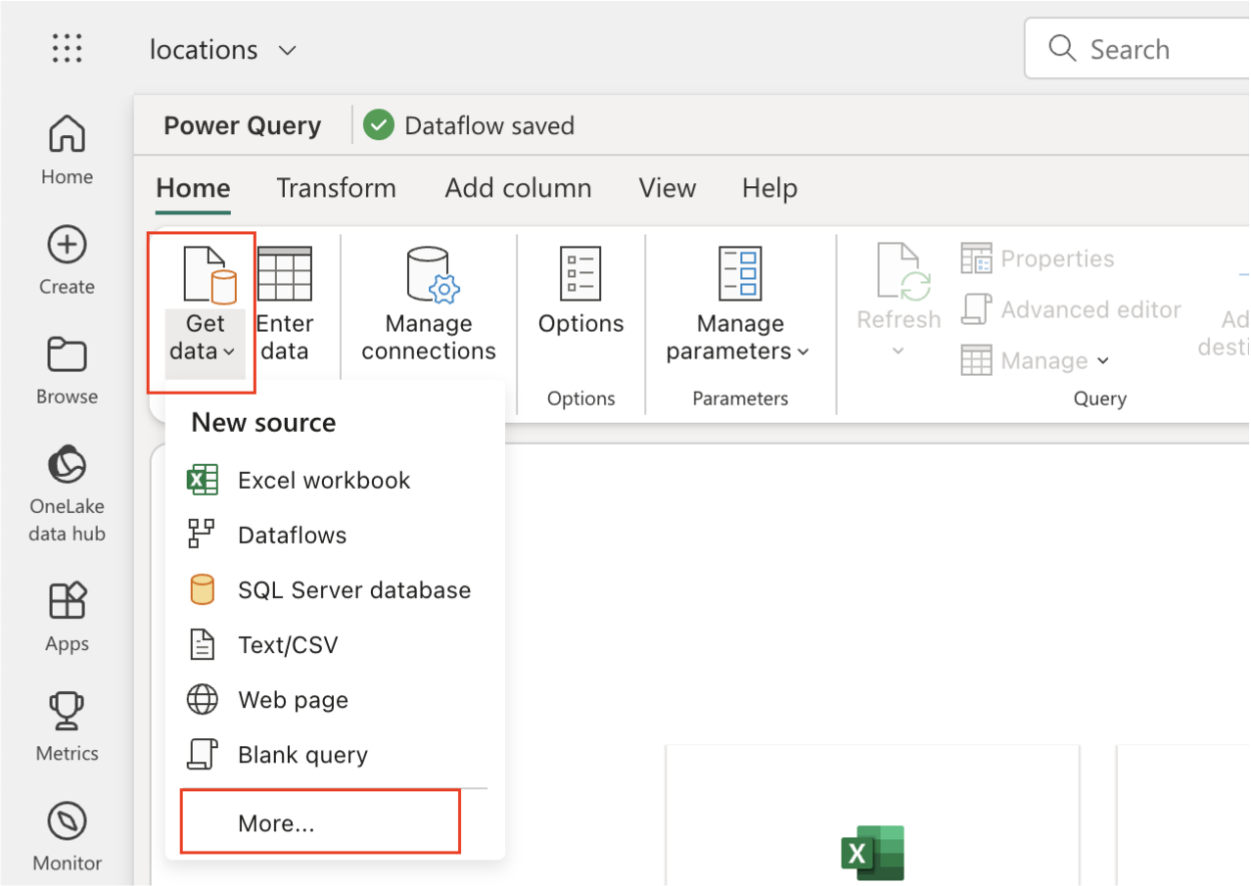
Will

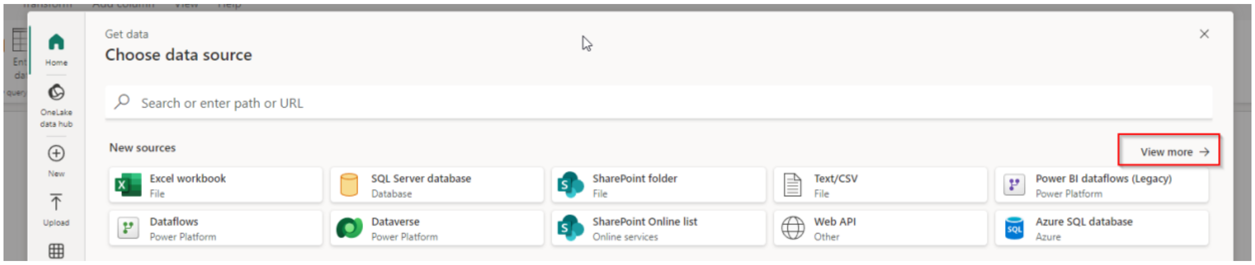
Connecting to Source Tables

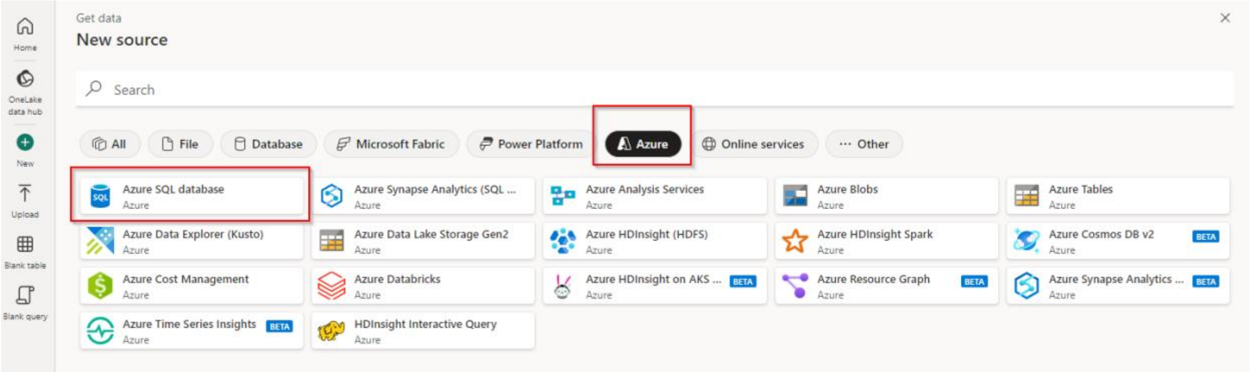
Connect to a Source: begin by selecting **Get Data** from the top left of your screen. This will open a window allowing you to explore available data sources. From there, choose **View More** under the **New Sources** section to expand the available options.

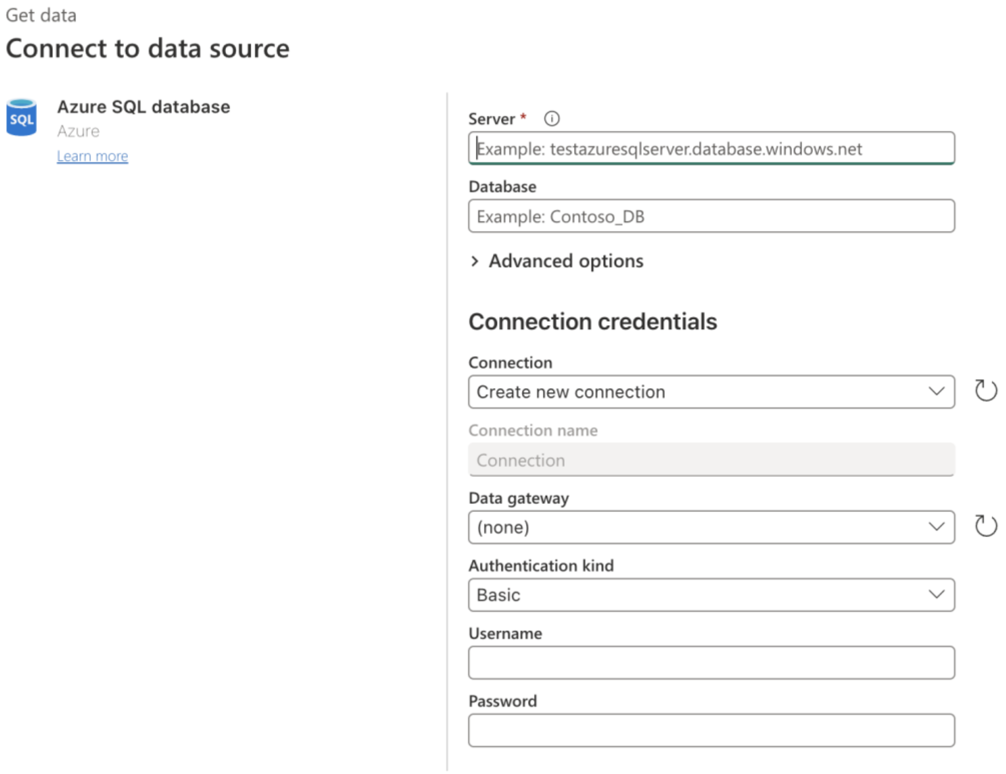
Next, select **Azure** from the slicer. Once filtered, choose **Azure SQL database** from the list of options presented. You will then be prompted to input connection settings. Using to the “Source Connection Details” file, complete the **Server** and **Database** fields, and then set the **Authentication Kind** to **Basic** before completing the **Username** and **Password** fields. Click **Next**.

Add Source Tables: Tick checkboxes for the **Application.Cities** and **Application.StateProvinces** tables. Click **Create**.


locations
Search
Home
Create
Browse
OneLake data hub
Apps
Metrics
Monitor
Power Query
Dataflow saved
Home
Transform
Add column
View
Help
Get data
Enter data
Manage connections
Options
Manage parameters
Refresh
Properties
Advanced editor
Manage
Options
Parameters
Query
New source
Excel workbook
Dataflows
SQL Server database
Text/CSV
Web page
Blank query
More...


Home
OneLake data hub
New
Upload
Get data
Choose data source
Search or enter path or URL
New sources
View more →
Excel workbook
File
SQL Server database
Database
SharePoint folder
File
Text/CSV
File
Power BI dataflows (Legacy)
Power Platform
Dataflows
Power Platform
Dataverse
Power Platform
SharePoint Online list
Online services
Web API
Other
Azure SQL database
Azure


Home
OneLake data hub
New
Upload
Blank table
Blank query
Get data
New source
Search
All
File
Database
Microsoft Fabric
Power Platform
Azure
Online services
··· Other
Azure SQL database
Azure
Azure Synapse Analytics (SQL ...
Azure
Azure Analysis Services
Azure
Azure Blobs
Azure
Azure Tables
Azure
Azure Data Explorer (Kusto)
Azure
Azure Data Lake Storage Gen2
Azure
Azure HDInsight (HDFS)
Azure
Azure HDInsight Spark
Azure
Azure Cosmos DB v2
BETA
Azure
Azure Cost Management
Azure
Azure Databricks
Azure
Azure HDInsight on AKS ...
BETA
Azure
Azure Resource Graph
BETA
Azure
Azure Synapse Analytics ...
BETA
Azure
Azure Time Series Insights
BETA
Azure
HDInsight Interactive Query
Azure



Get data

Connect to data source

Azure SQL database
Azure
Learn more

Server * ⓘ
Example: testazuresqlserver.database.windows.net

Database
Example: Contoso_DB

› Advanced options

Connection credentials

Connection
Create new connection

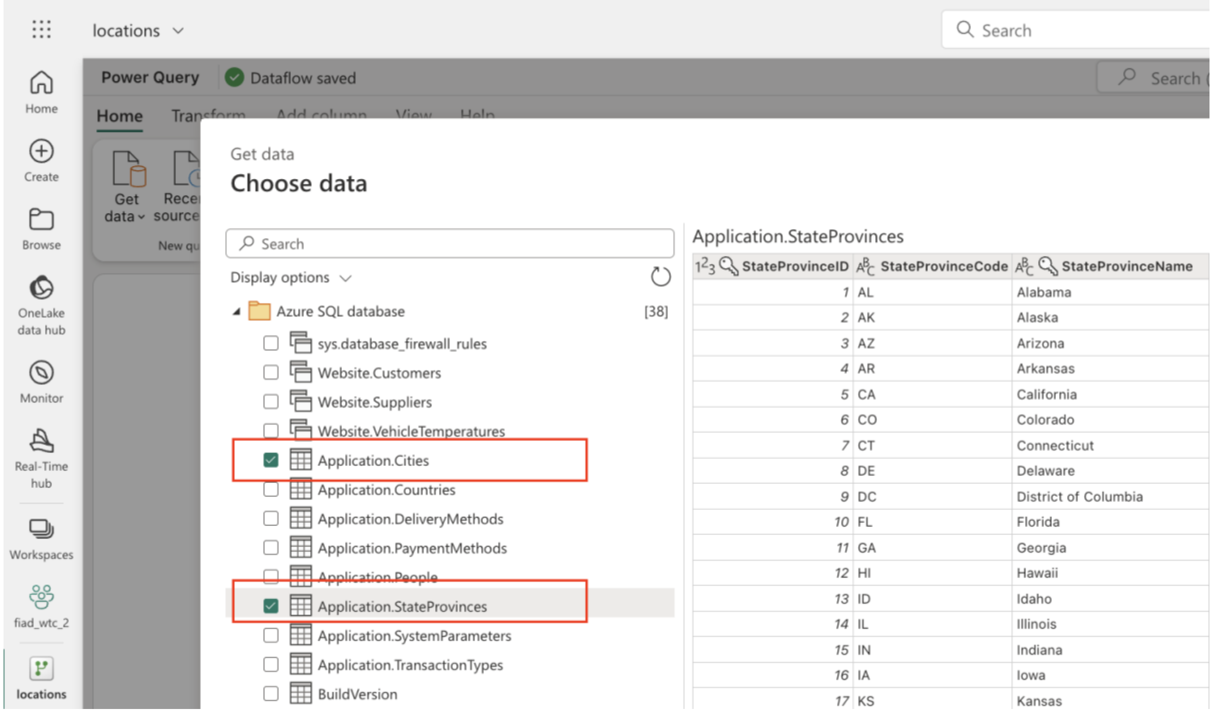
Connection name
Connection

Data gateway
(none)

Authentication kind
Basic

Username

Password



Get data

Choose data

Search

Display options

◢ Azure SQL database [38]

- ☐ sys.database_firewall_rules
- ☐ Website.Customers
- ☐ Website.Suppliers
- ☐ Website.VehicleTemperatures
- ☑ Application.Cities
- ☐ Application.Countries
- ☐ Application.DeliveryMethods
- ☐ Application.PaymentMethods
- ☐ Application.People
- ☑ Application.StateProvinces
- ☐ Application.SystemParameters
- ☐ Application.TransactionTypes
- ☐ BuildVersion

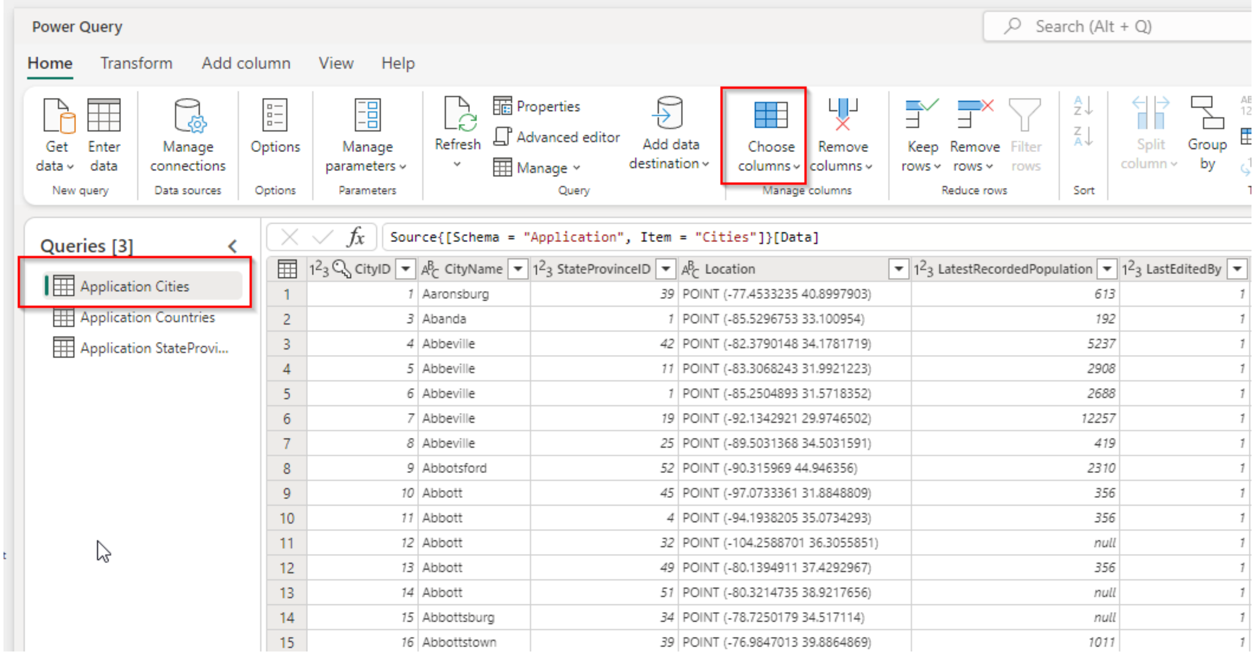
Application.StateProvinces

StateProvinceID	StateProvinceCode	StateProvinceName
1	AL	Alabama
2	AK	Alaska
3	AZ	Arizona
4	AR	Arkansas
5	CA	California
6	CO	Colorado
7	CT	Connecticut
8	DE	Delaware
9	DC	District of Columbia
10	FL	Florida
11	GA	Georgia
12	HI	Hawaii
13	ID	Idaho
14	IL	Illinois
15	IN	Indiana
16	IA	Iowa
17	KS	Kansas

Transforming Application Cities

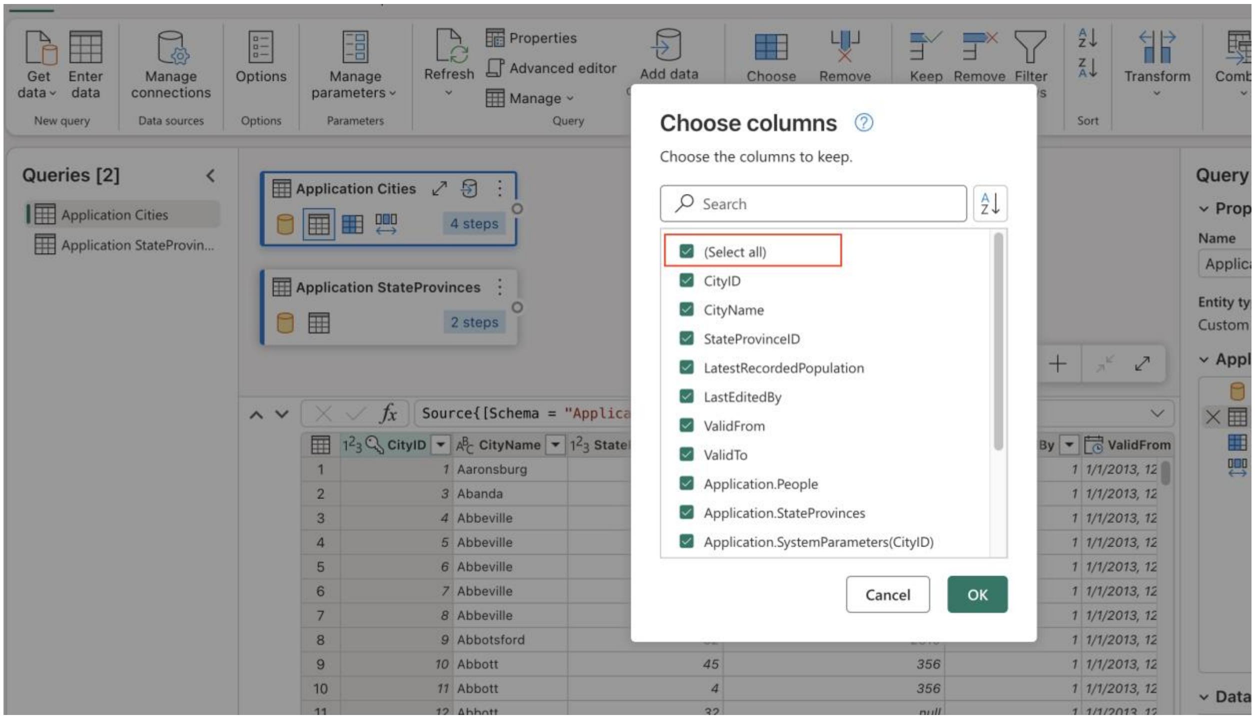
Now that we've connected to the sample database and created queries for the tables we'll be working with, we can start the data transformation process. We'll begin by working with the **Application Cities** table before moving on to joining the **Application StateProvinces** table.

Apply transformations to Cities: Start by selecting **Application Cities** from the queries pane on the left. Navigate to the **Home** tab in the top ribbon and select **Choose Columns**. In the dialog box that appears, uncheck the **Select All** box. Then, check the boxes for the following fields: **CityID**, **CityName**, and **StateProvinceID**. Click **OK**. Reorganize the columns of the table to shuffle the Id fields to the beginning by clicking and dragging **StateProvinceId** to the second position.



```
Source{[Schema = "Application", Item = "Cities"]}[Data]
```

	CityID	CityName	StateProvinceID	Location	LatestRecordedPopulation	LastEditedBy
1	1	Aaronsburg	39	POINT (-77.4533235 40.8997903)	613	1
2	3	Abanda	1	POINT (-85.5296753 33.100954)	192	1
3	4	Abbeville	42	POINT (-82.3790148 34.1781719)	5237	1
4	5	Abbeville	11	POINT (-83.3068243 31.9921223)	2908	1
5	6	Abbeville	1	POINT (-85.2504893 31.5718352)	2688	1
6	7	Abbeville	19	POINT (-92.1342921 29.9746502)	12257	1
7	8	Abbeville	25	POINT (-89.5031368 34.5031591)	419	1
8	9	Abbotsford	52	POINT (-90.315969 44.946356)	2310	1
9	10	Abbott	45	POINT (-97.0733361 31.8848809)	356	1
10	11	Abbott	4	POINT (-94.1938205 35.0734293)	356	1
11	12	Abbott	32	POINT (-104.2588701 36.3055851)	null	1
12	13	Abbott	49	POINT (-80.1394911 37.4292967)	356	1
13	14	Abbott	51	POINT (-80.3214735 38.9217656)	null	1
14	15	Abbottsburg	34	POINT (-78.7250179 34.517114)	null	1
15	16	Abbottstown	39	POINT (-76.9847013 39.8864869)	1011	1



Source{[Schema = "Applica

Choose columns

Choose the columns to keep.

Search

- ☑ (Select all)
- ☑ CityID
- ☑ CityName
- ☑ StateProvinceID
- ☑ LatestRecordedPopulation
- ☑ LastEditedBy
- ☑ ValidFrom
- ☑ ValidTo
- ☑ Application.People
- ☑ Application.StateProvinces
- ☑ Application.SystemParameters(CityID)

Cancel OK

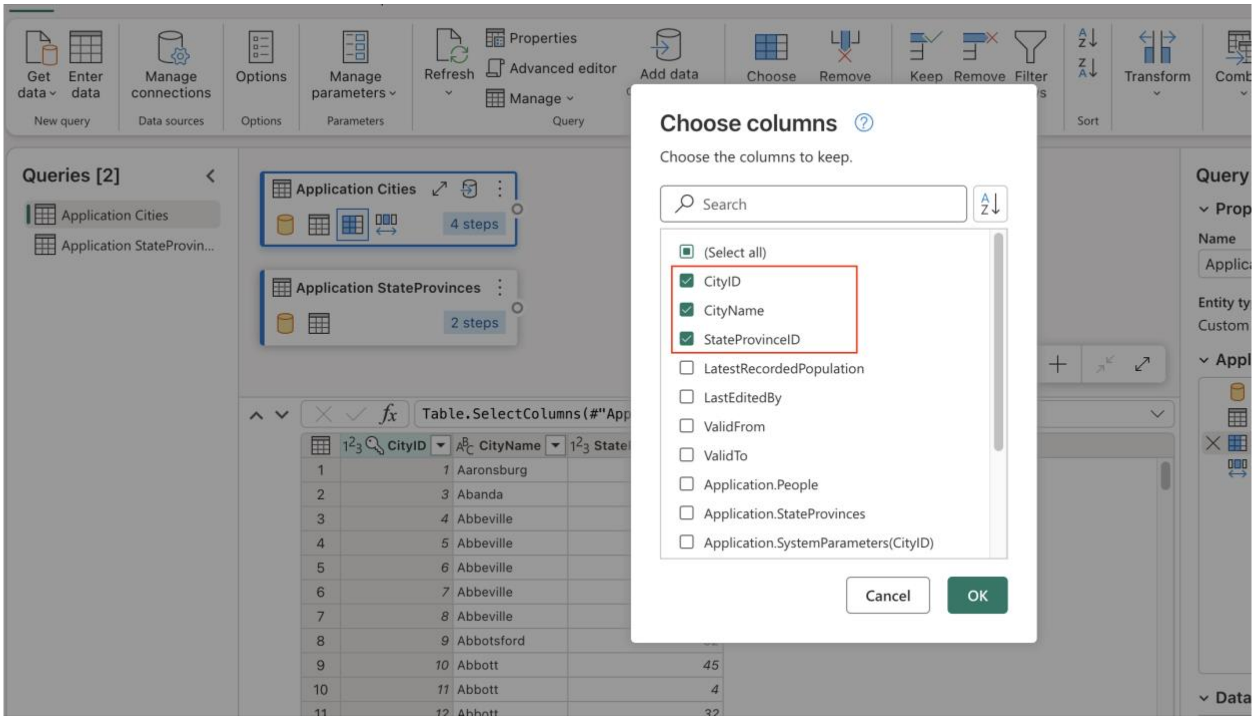


Table.SelectColumns(#"App

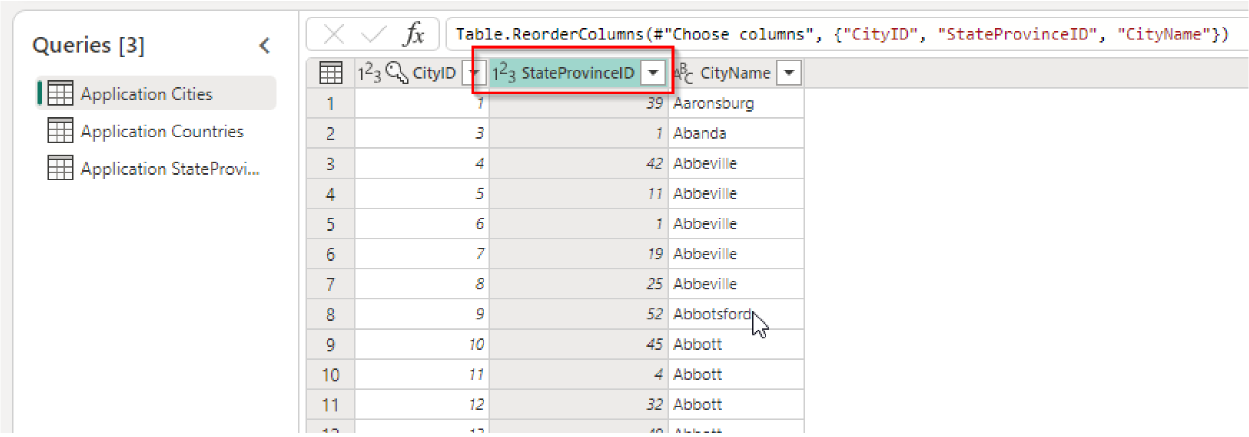
Choose columns

Choose the columns to keep.

Search

- ▣ (Select all)
- ☑ CityID
- ☑ CityName
- ☑ StateProvinceID
- □ LatestRecordedPopulation
- □ LastEditedBy
- □ ValidFrom
- □ ValidTo
- □ Application.People
- □ Application.StateProvinces
- □ Application.SystemParameters(CityID)

Cancel OK

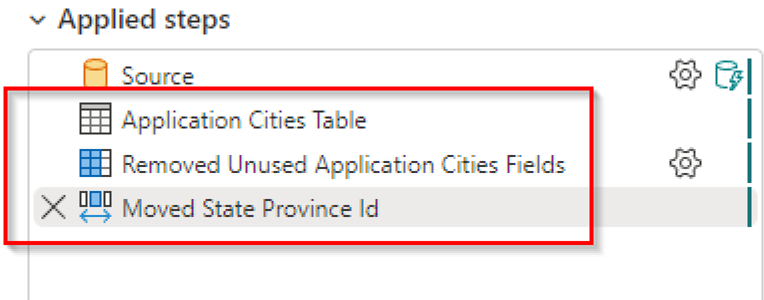


```
Table.ReorderColumns(#"Choose columns", {"CityID", "StateProvinceID", "CityName"})
```

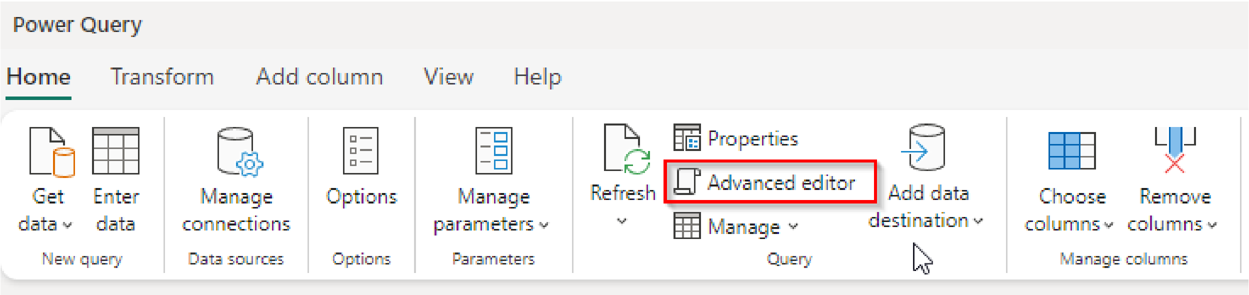
	CityID	StateProvinceID	CityName
1	1	39	Aaronsburg
2	3	1	Abanda
3	4	42	Abbeville
4	5	11	Abbeville
5	6	1	Abbeville
6	7	19	Abbeville
7	8	25	Abbeville
8	9	52	Abbotsford
9	10	45	Abbott
10	11	4	Abbott
11	12	32	Abbott

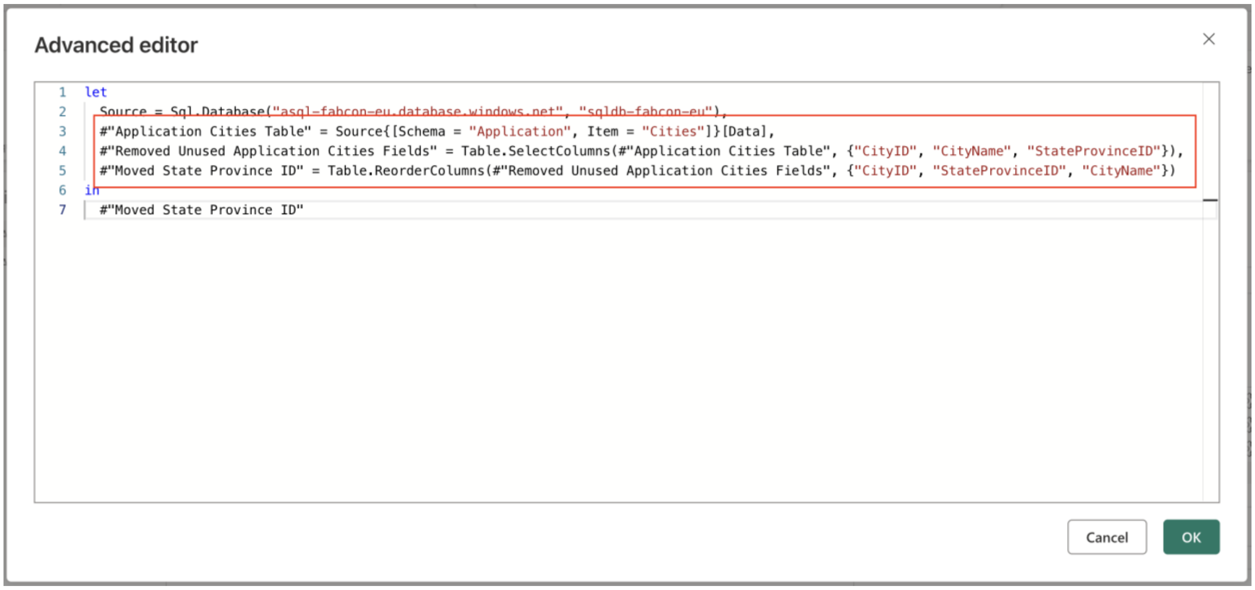
To enhance the readability of the transformation steps, edit the names of the **Applied steps**. Right-click on each step and select **Rename** to give meaningful descriptions:

- Rename **Navigation 1** to **Application Cities Table**
- Rename **Choose Columns** to **Removed Unused Application Cities Fields**
- Rename **Reordered Columns** to **Moved State Province ID**



After completing these steps, open the **Advanced Editor** to review the **Mashup (M code)** that has been written as the steps were applied. Pay close attention to the naming conventions we've applied, noting how they improve code navigation and readability.

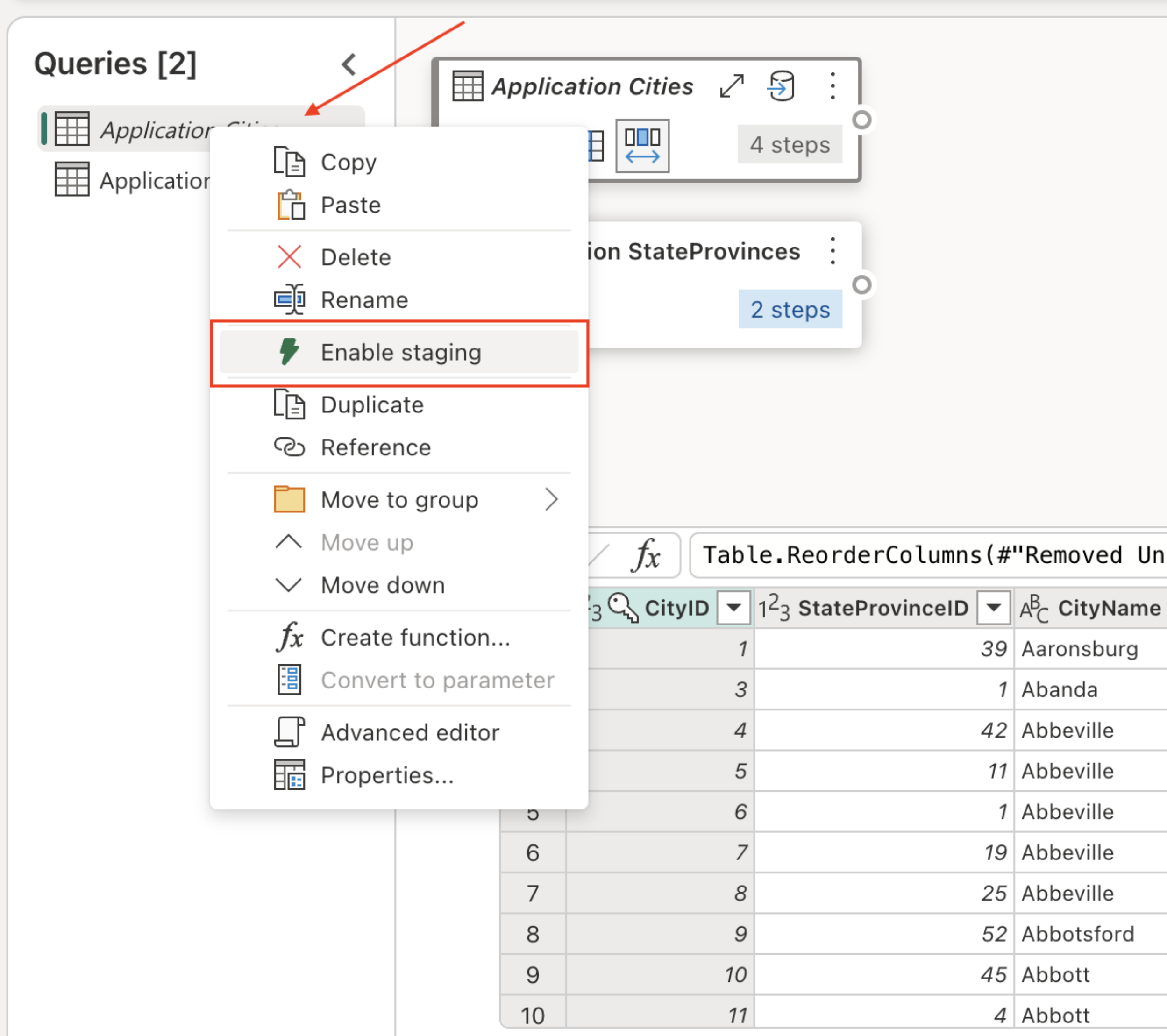




Advanced editor

```
let
  Source = Sql.Database("asql-fabcon-eu.database.windows.net", "sqldb-fabcon-eu"),
  #"Application Cities Table" = Source{[Schema = "Application", Item = "Cities"]}[Data],
  #"Removed Unused Application Cities Fields" = Table.SelectColumns(#"Application Cities Table", {"CityID", "CityName", "StateProvinceID"}),
  #"Moved State Province ID" = Table.ReorderColumns(#"Removed Unused Application Cities Fields", {"CityID", "StateProvinceID", "CityName"})
in
  #"Moved State Province ID"
```

Disable dataflow staging: Right click Application Cities from the Queries blade and disable **Enable Staging**.



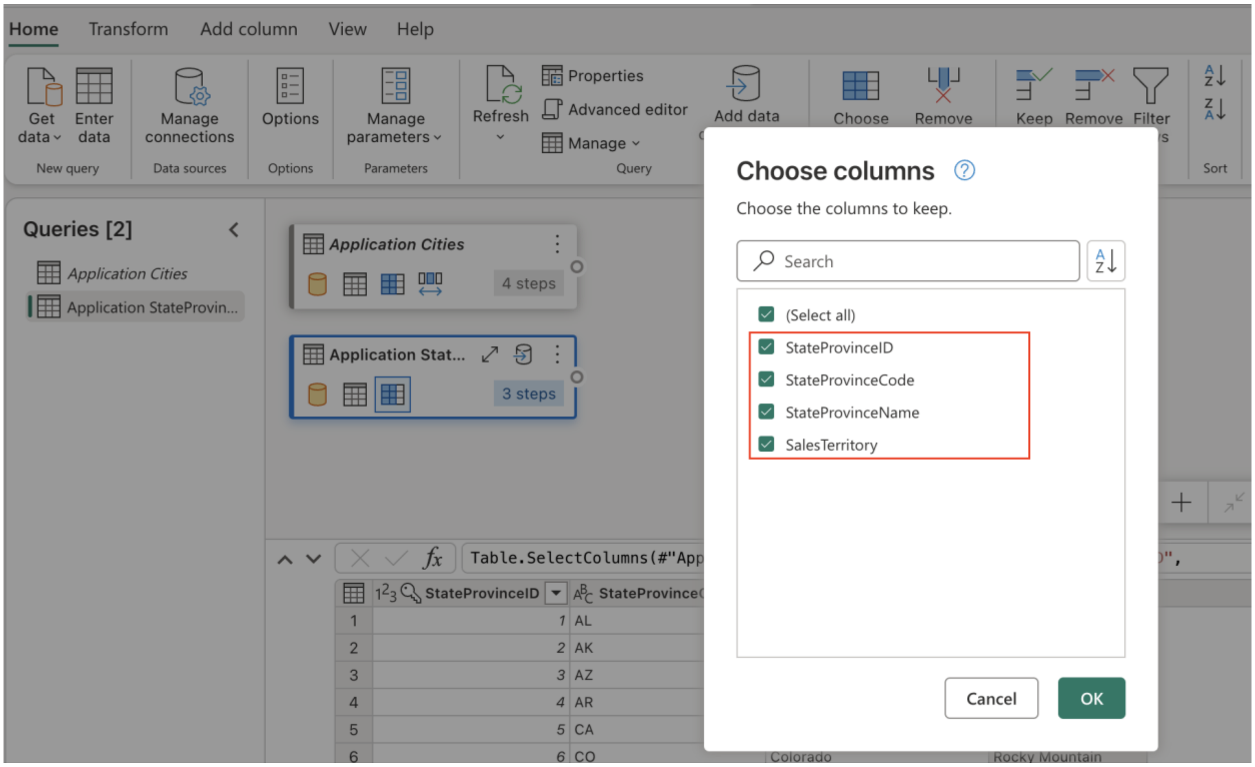
```
Table.ReorderColumns(#"Removed Un
```

	CityID	StateProvinceID	CityName
	1	39	Aaronsburg
	3	1	Abanda
	4	42	Abbeville
	5	11	Abbeville
5	6	1	Abbeville
6	7	19	Abbeville
7	8	25	Abbeville
8	9	52	Abbotsford
9	10	45	Abbott
10	11	4	Abbott

Transforming Application StateProvinces

Now that we've had a detailed walkthrough of how to transform the Application Cities table, we're going to repeat the process for Application StateProvinces with abbreviated instructions.

Apply transformations to StateProvinces: Select **Application StateProvinces** from the queries pane on the left then use **Choose columns** to keep only **StateProvinceID, StateProvinceCode, StateProvinceName** and **SalesTerritory**. Click **OK**.



```
Table.SelectColumns(#"App
```

	StateProvinceID	StateProvince
1	1	AL
2	2	AK
3	3	AZ
4	4	AR
5	5	CA
6	6	CO

Choose columns

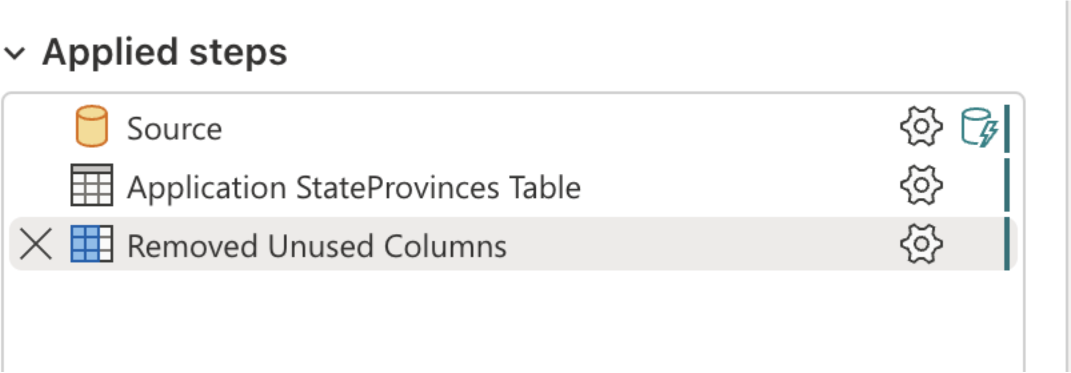
Choose the columns to keep.

Search

☑ (Select all)
☑ StateProvinceID
☑ StateProvinceCode
☑ StateProvinceName
☑ SalesTerritory

Cancel OK

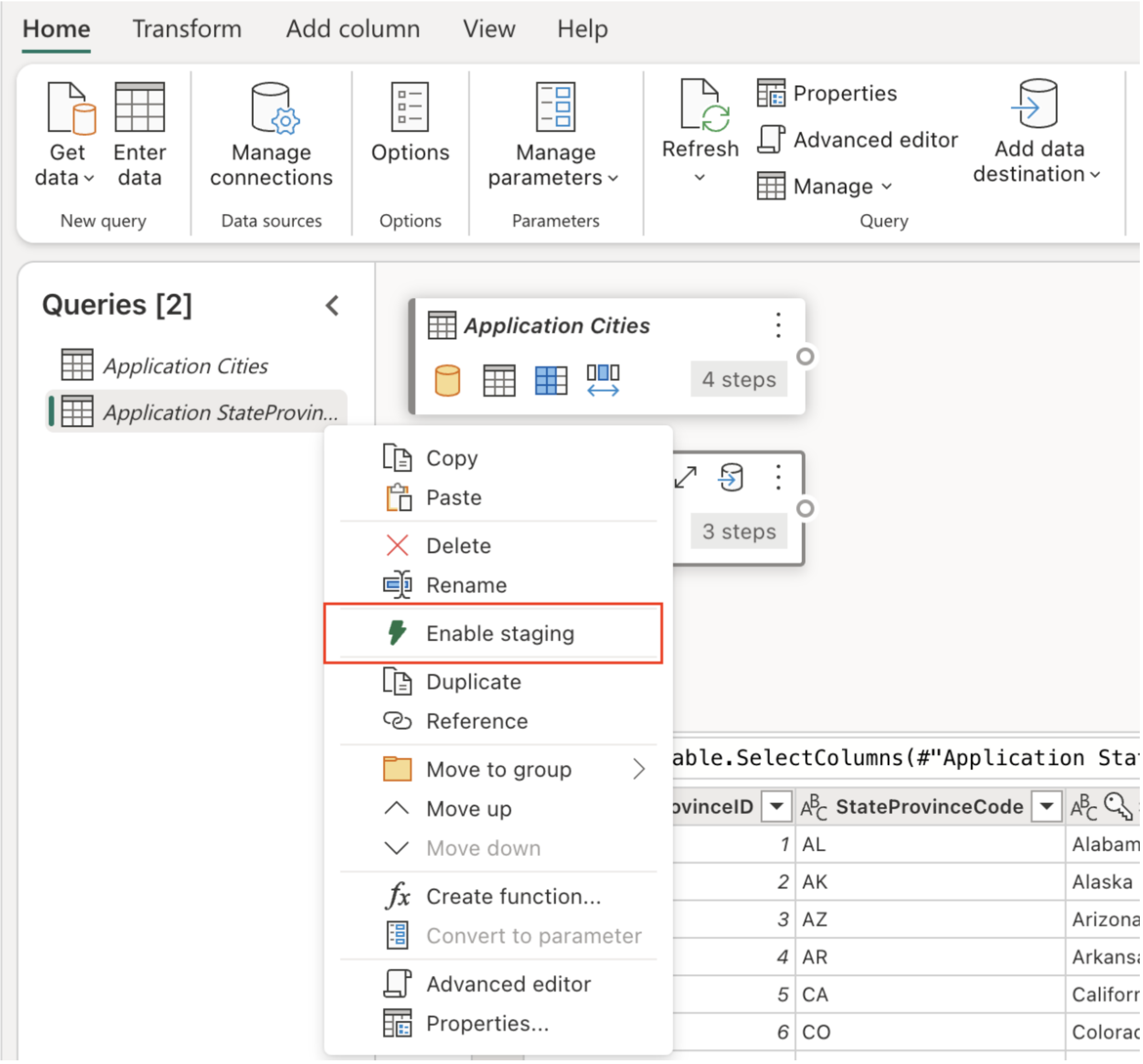
Apply meaningful names to applied steps:



Applied steps

- Source
- Application StateProvinces Table
- Removed Unused Columns

Disable Dataflow staging:



```
able.SelectColumns(#"Application Sta
```

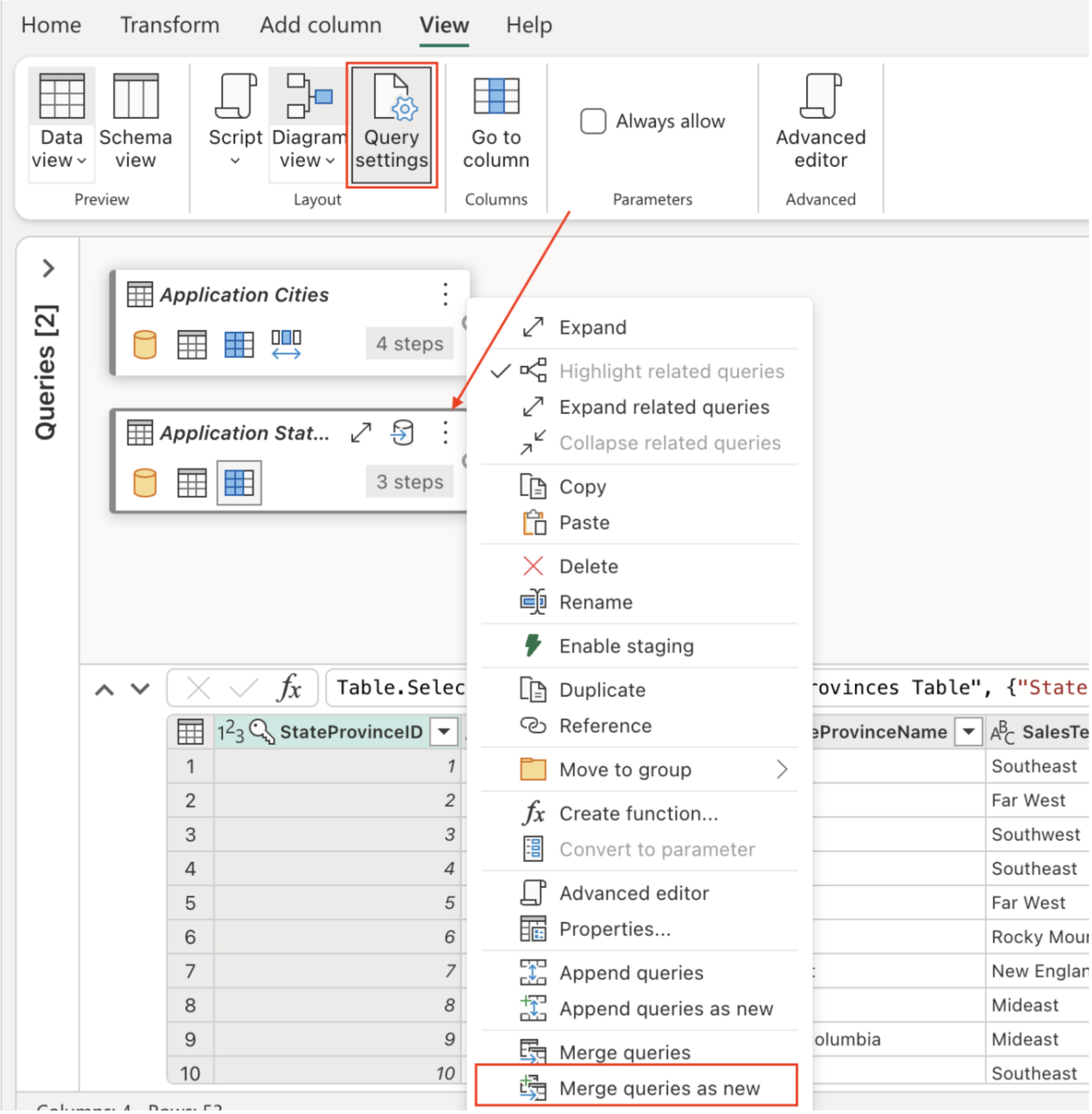
ovinceID	StateProvinceCode	
1	AL	Alabam
2	AK	Alaska
3	AZ	Arizona
4	AR	Arkansa
5	CA	Califorr
6	CO	Colorac

Merge Tables

Now that our tables are prepared, we're ready to join them together to create a single **locations** table. There are several ways to accomplish this. We'll use the diagram view to merge Application Cities and Application StateProvinces tables.

Merge tables using Diagram View: To merge the tables, start by navigating to the **View** tab in the top ribbon. From there, select the **Diagram View** toggle. In this view, you'll see the two tables represented in a visual interface.

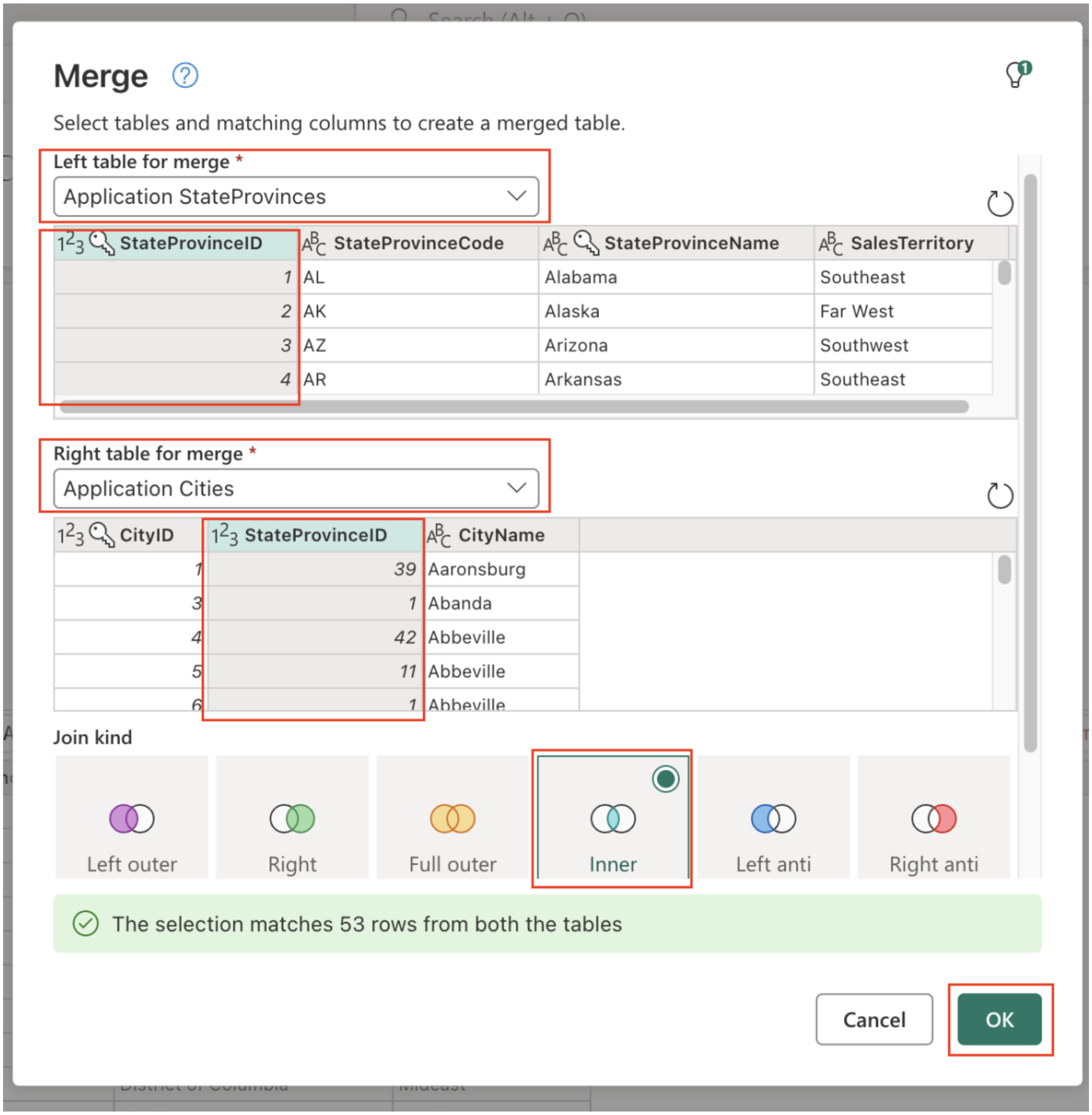
Next, click on the ellipsis (three dots) in the top right corner of the **Application StateProvinces** table. From the options, select **Merge queries as new**. Select **StateProvinceID** from the **Application StateProvinces** table. In the second dropdown menu, select **Application Cities** and choose **StateProvinceID** as the join field. For the join type, select **Inner** and then click **OK**.



```
Table.Selec ... rovinces Table", {"State
```

	StateProvinceID	eProvinceName	SalesTe
1	1		Southeast
2	2		Far West
3	3		Southwest
4	4		Southeast
5	5		Far West
6	6		Rocky Mour
7	7	:	New Englar
8	8		Mideast
9	9	olumbia	Mideast
10	10		Southeast

Columns: 4 Rows: 53



Merge

Select tables and matching columns to create a merged table.

Left table for merge *

Application StateProvinces

StateProvinceID	StateProvinceCode	StateProvinceName	SalesTerritory
1	AL	Alabama	Southeast
2	AK	Alaska	Far West
3	AZ	Arizona	Southwest
4	AR	Arkansas	Southeast

Right table for merge *

Application Cities

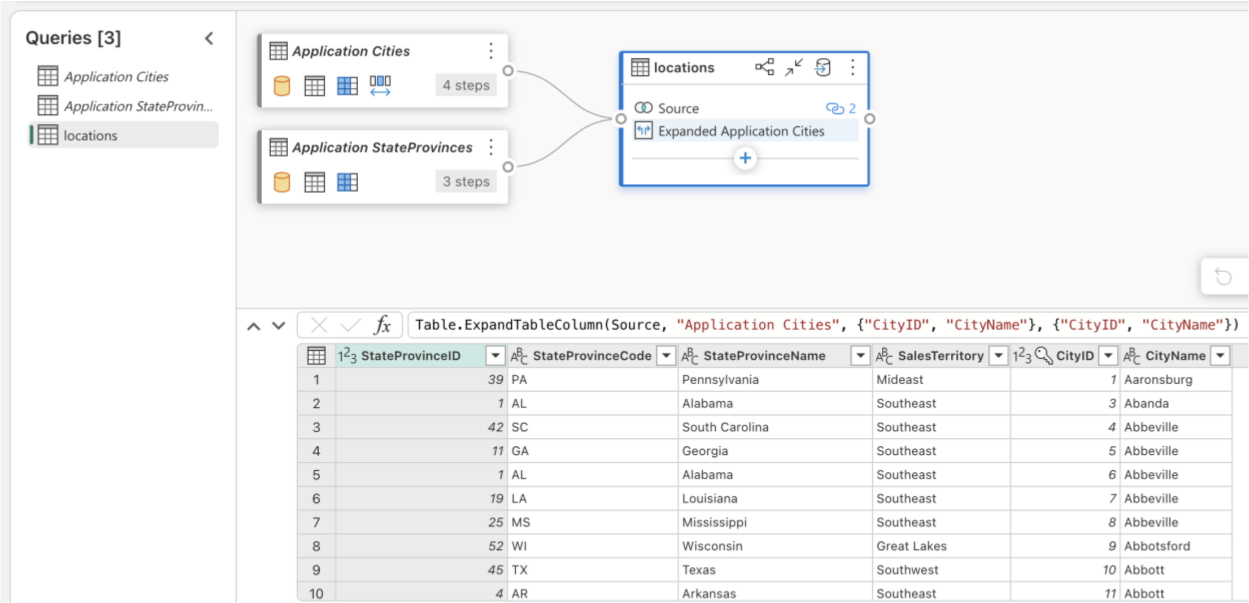
CityID	StateProvinceID	CityName
1	39	Aaronsburg
3	1	Abanda
4	42	Abbeville
5	11	Abbeville
6	1	Abbeville

Join kind

○ Left outer ○ Right ○ Full outer ● Inner ○ Left anti ○ Right anti

The selection matches 53 rows from both the tables

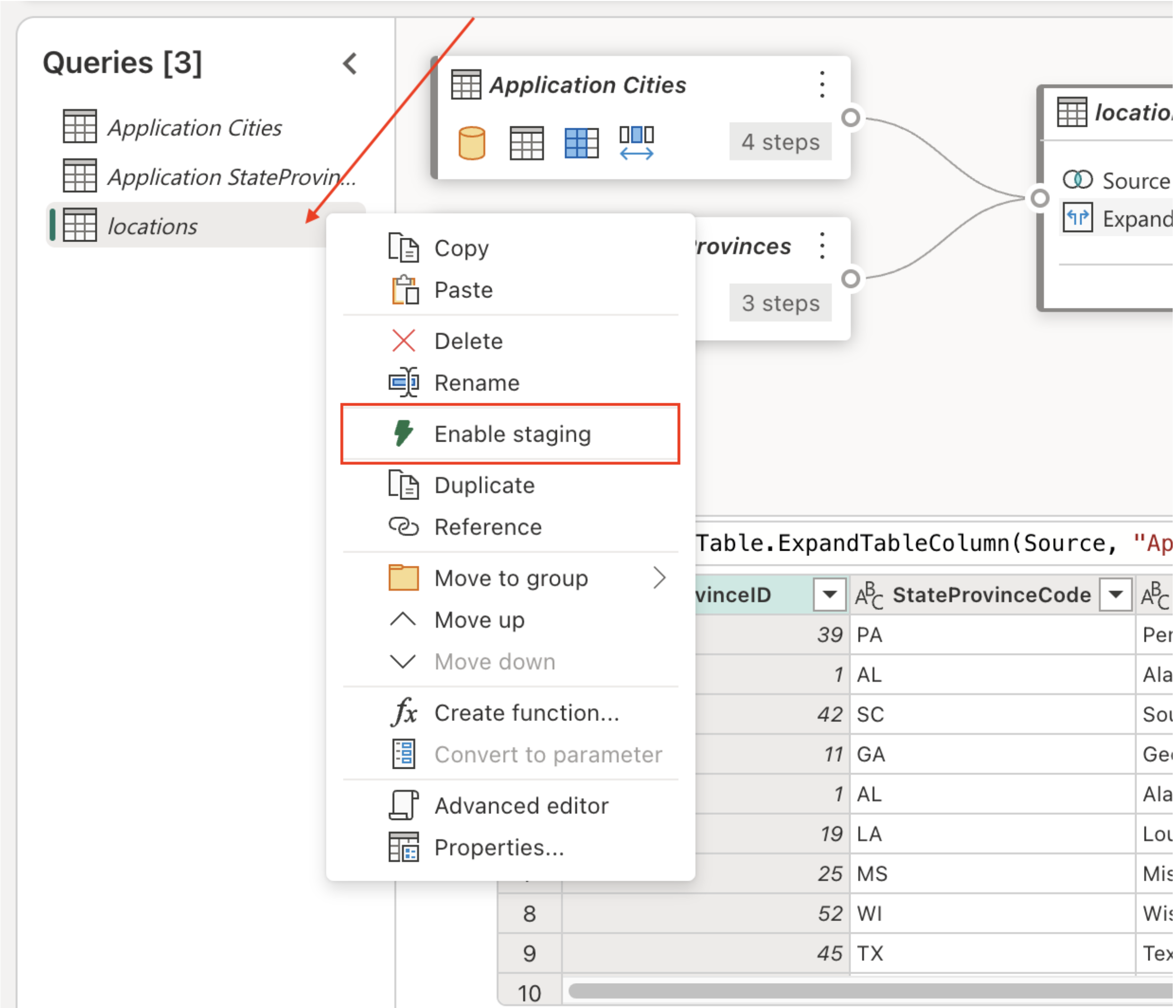
Review the newly created table: At this point, your diagram view should show the two **Application** tables merging into the **Merge** table. Right-click this new table, select Rename, type **locations** and press enter. In the table preview at the bottom, expand the contents of the **Application Cities** column by clicking ⇆, unchecking the **StateProvinceID** box (as this would result in a duplicate column), and pressing OK. Reorganize the columns by moving the **CityID** field to the beginning.



```
Table.ExpandTableColumn(Source, "Application Cities", {"CityID", "CityName"}, {"CityID", "CityName"})
```

	StateProvinceID	StateProvinceCode	StateProvinceName	SalesTerritory	CityID	CityName
1	39	PA	Pennsylvania	Mideast	1	Aaronsburg
2	1	AL	Alabama	Southeast	3	Abanda
3	42	SC	South Carolina	Southeast	4	Abbeville
4	11	GA	Georgia	Southeast	5	Abbeville
5	1	AL	Alabama	Southeast	6	Abbeville
6	19	LA	Louisiana	Southeast	7	Abbeville
7	25	MS	Mississippi	Southeast	8	Abbeville
8	52	WI	Wisconsin	Great Lakes	9	Abbotsford
9	45	TX	Texas	Southwest	10	Abbott
10	4	AR	Arkansas	Southeast	11	Abbott

Disable dataflow staging: To finalize, expand the **Queries** pane on the left and disable staging for the **locations** table.



```
Table.ExpandTableColumn(Source, "Ap
```

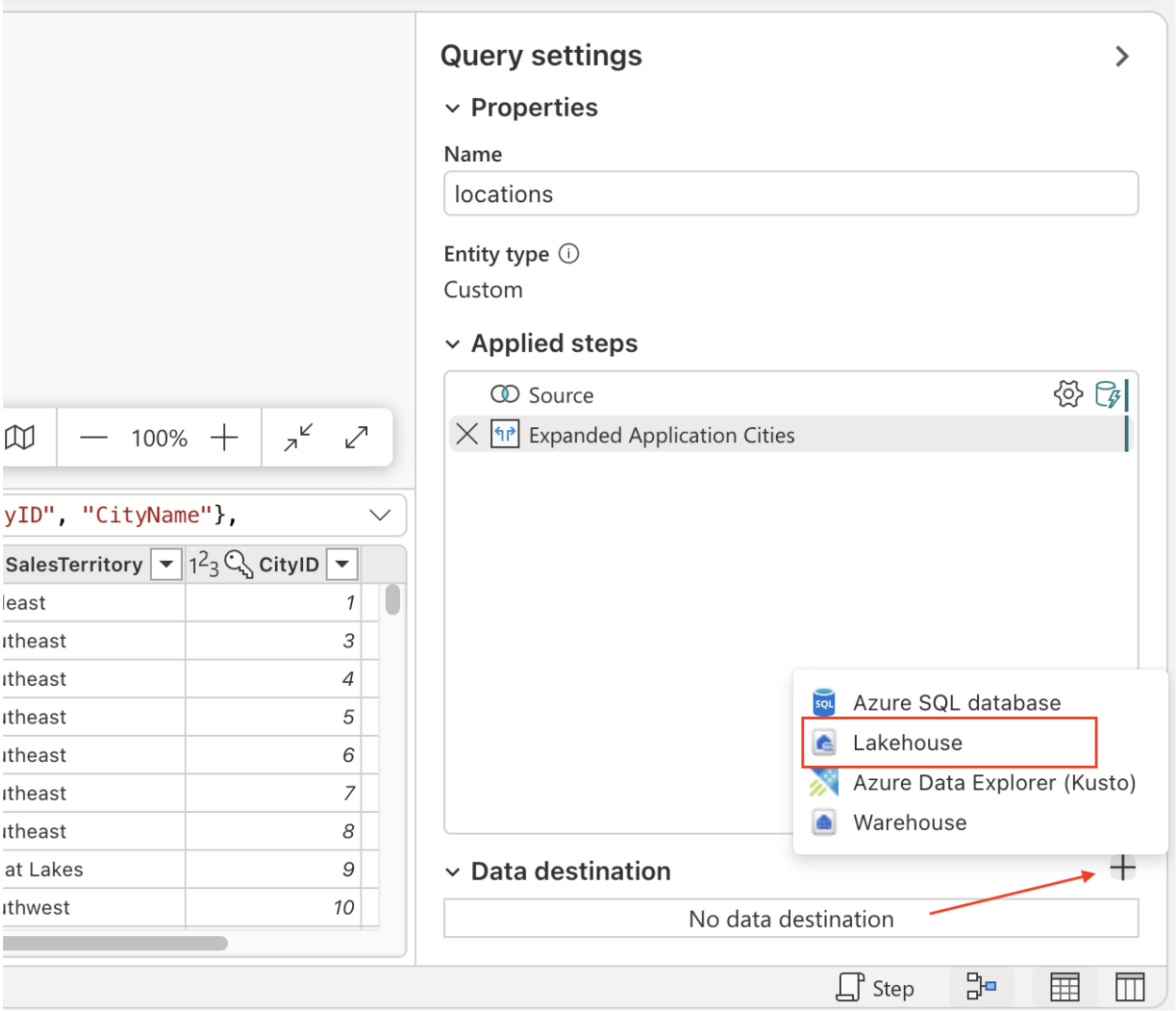
	vinceID	StateProvinceCode	
	39	PA	Per
	1	AL	Ala
	42	SC	Sou
	11	GA	Ge
	1	AL	Ala
	19	LA	Lou
	25	MS	Mis
8	52	WI	Wis
9	45	TX	Tex
10			

Writing to a Destination

With the transformation process complete, we now have the final version of the **locations** table. The next step is to set our Lakehouse as the destination to the dataflow and load the **locations** table.

Add dataflow destination: To begin adding a destination to the dataflow, expand the **Query settings** pane to the right, and click the “+” button in the bottom right corner. From the available options, select **Lakehouse**. Leave the default settings as they are and click **Next**.

Note: You may need to re-authenticate as the cached credentials may have expired. Use the organizational account credentials assigned to you to re-authenticate.



```
yID", "CityName"},
```

SalesTerritory	CityID
ıeast	1
ıtheast	3
ıtheast	4
ıtheast	5
ıtheast	6
ıtheast	7
ıtheast	8
at Lakes	9
ıthwest	10

Query settings

Properties

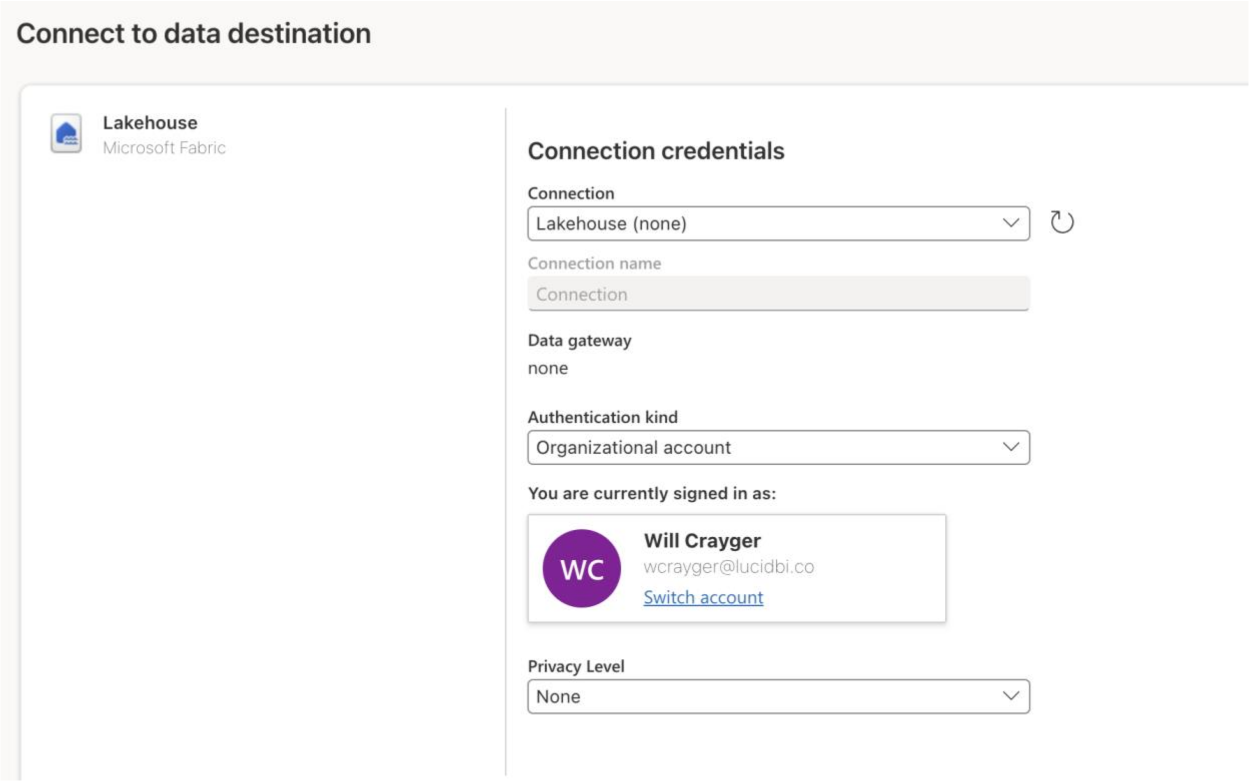
Name: locations

Entity type: Custom

Applied steps
- Source
- Expanded Application Cities

Data destination

No data destination



Connect to data destination

Lakehouse
Microsoft Fabric

Connection credentials

Connection: Lakehouse (none)

Connection name: Connection

Data gateway: none

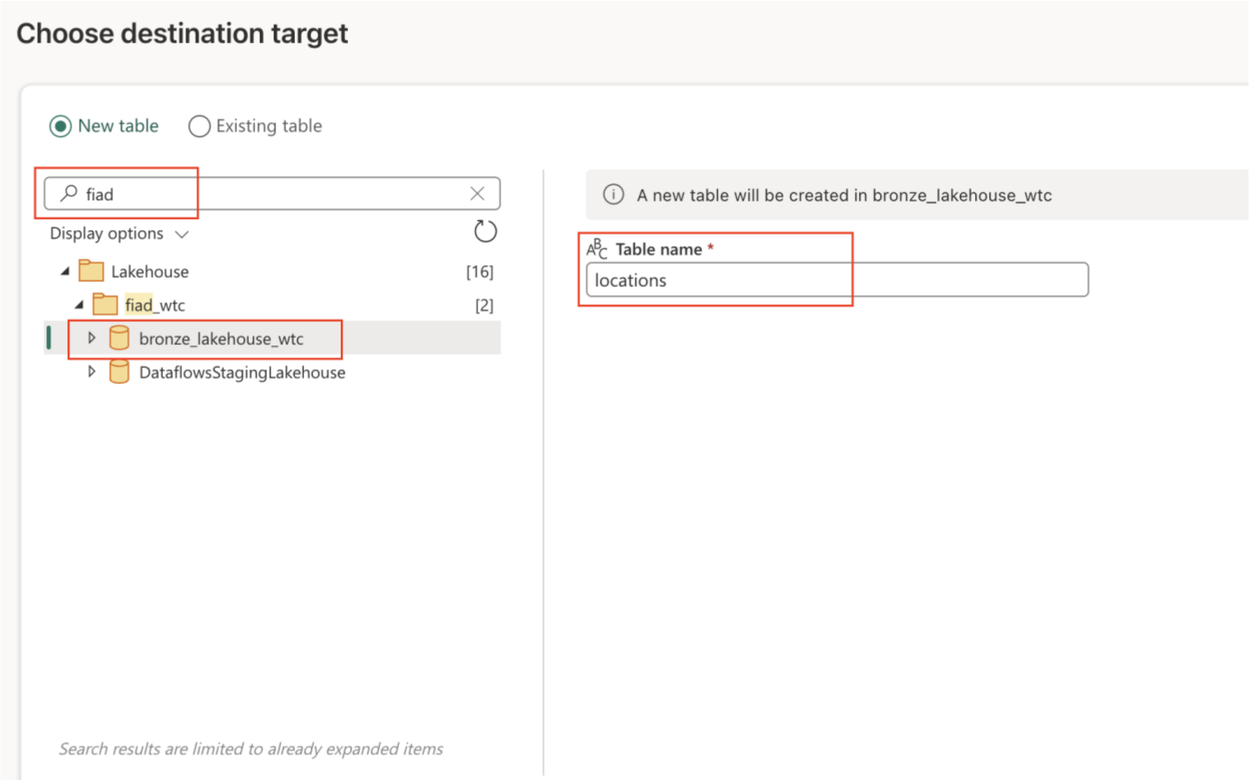
Authentication kind: Organizational account

You are currently signed in as:

Will Crayger
wcrayger@lucidbi.co
Switch account

Privacy Level: None

Navigate to your **Fabric in a Day Workspace** and expand the folder. Select the Lakehouse that you created earlier, labeled as **bronze_lakehouse_{your initials}**, and click **Next**.



Choose destination target

● New table ○ Existing table

fiad

Display options

- Lakehouse [16]
  - fiad_wtc [2]
    - bronze_lakehouse_wtc
    - DataflowsStagingLakehouse

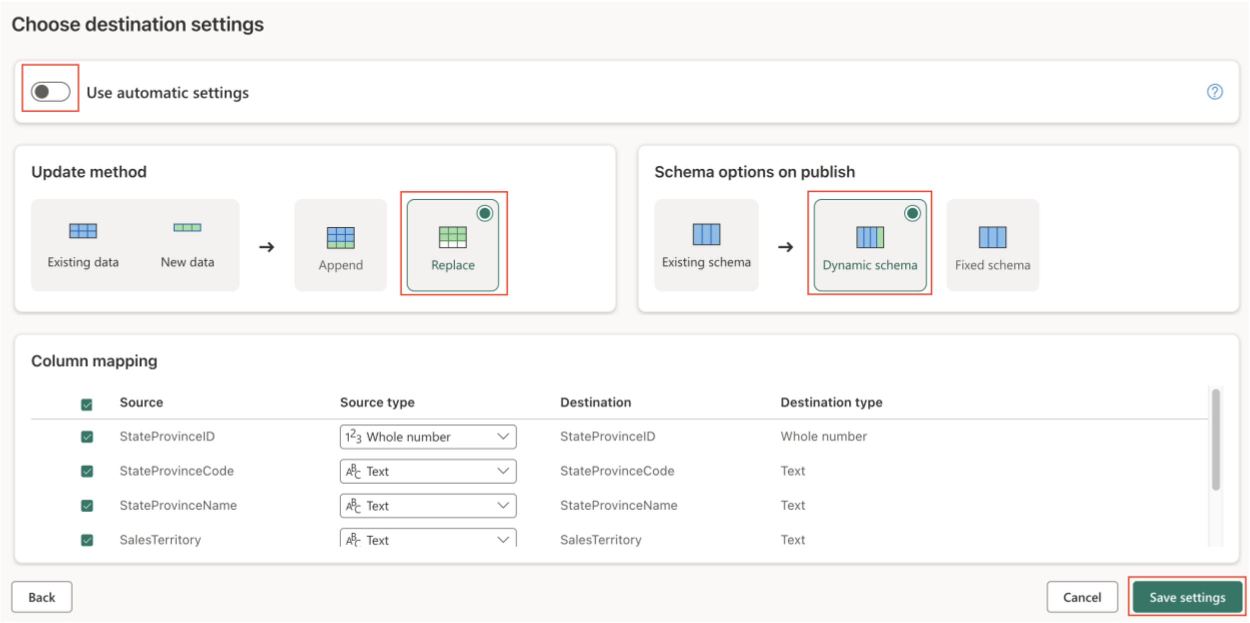
Search results are limited to already expanded items

A new table will be created in bronze_lakehouse_wtc

Table name *

locations

Untoggle **Use automatic settings** at the top, ensure **Replace** is selected, and click **Save Settings**. To complete the development of the dataflow, click the **Publish** button down the bottom-right, which will initiate loading the **locations** table into your Lakehouse.



Choose destination settings

Use automatic settings

Update method

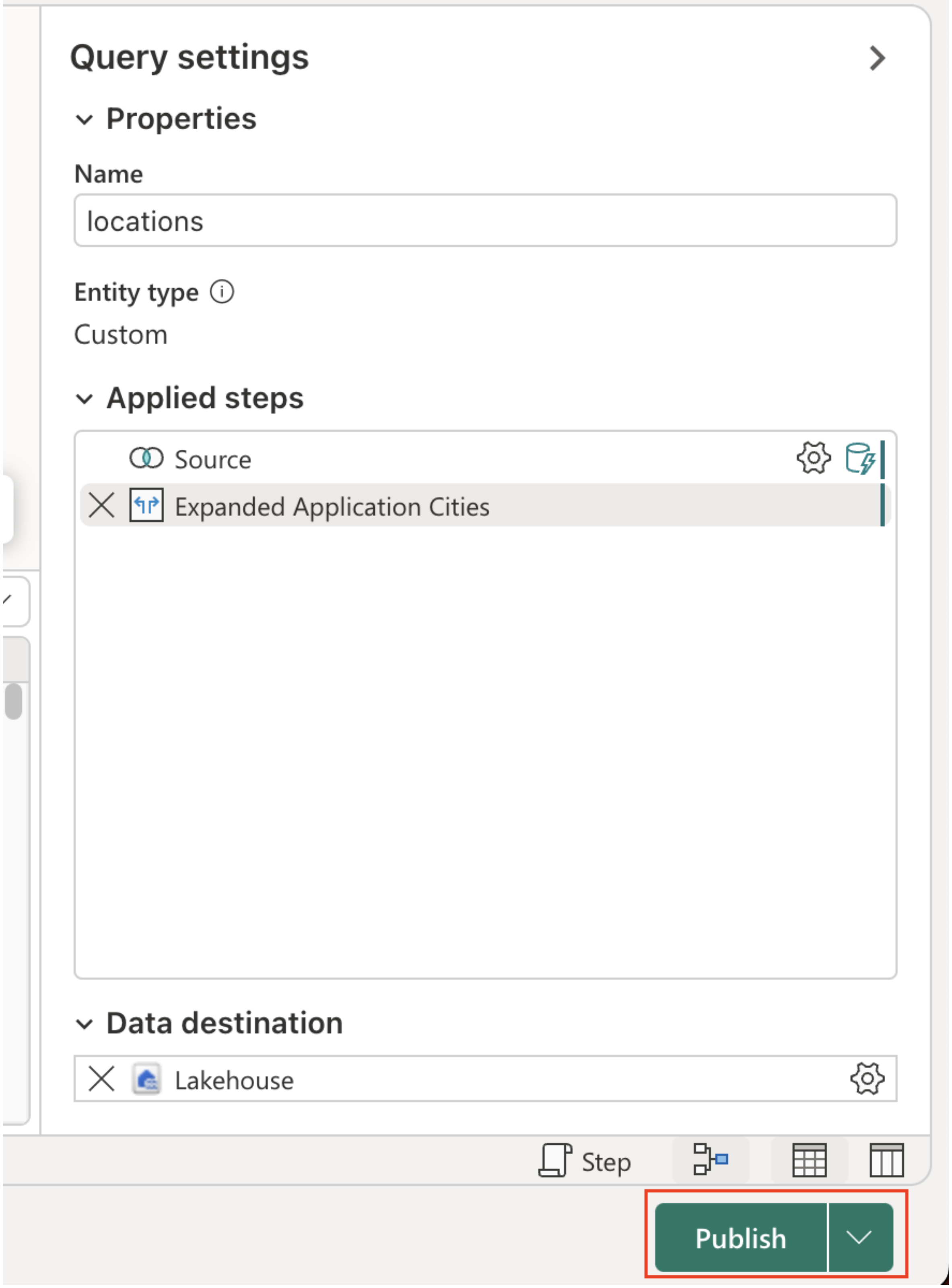
Existing data, New data → Append, ● Replace

Schema options on publish

Existing schema → ● Dynamic schema, Fixed schema

Column mapping

☑	Source	Source type	Destination	Destination type
☑	StateProvinceID	Whole number	StateProvinceID	Whole number
☑	StateProvinceCode	Text	StateProvinceCode	Text
☑	StateProvinceName	Text	StateProvinceName	Text
☑	SalesTerritory	Text	SalesTerritory	Text



Query settings

Properties

Name

locations

Entity type

Custom

Applied steps

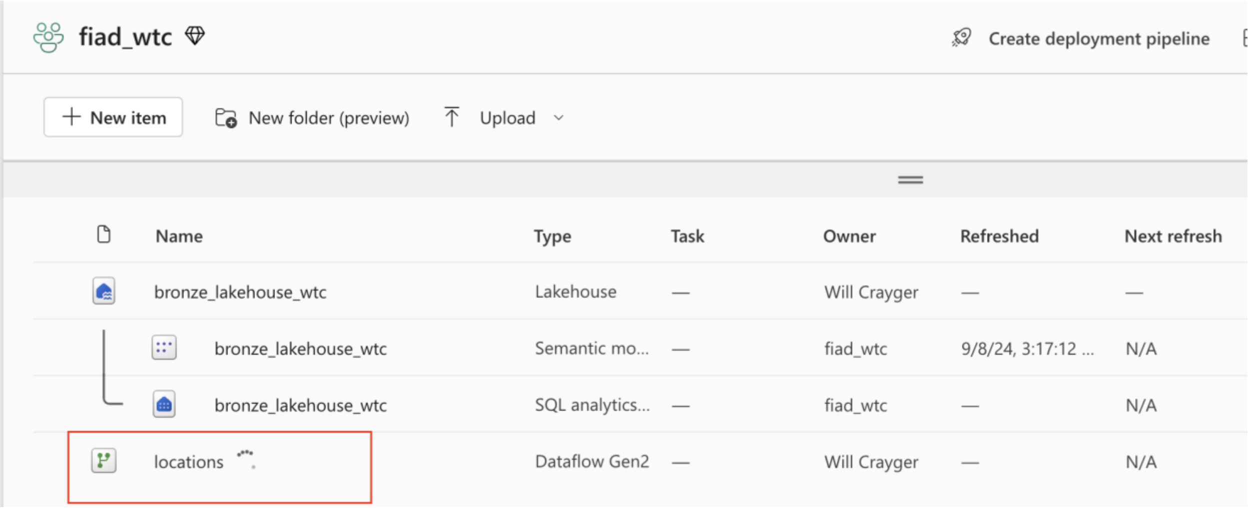
Source

Expanded Application Cities

Data destination

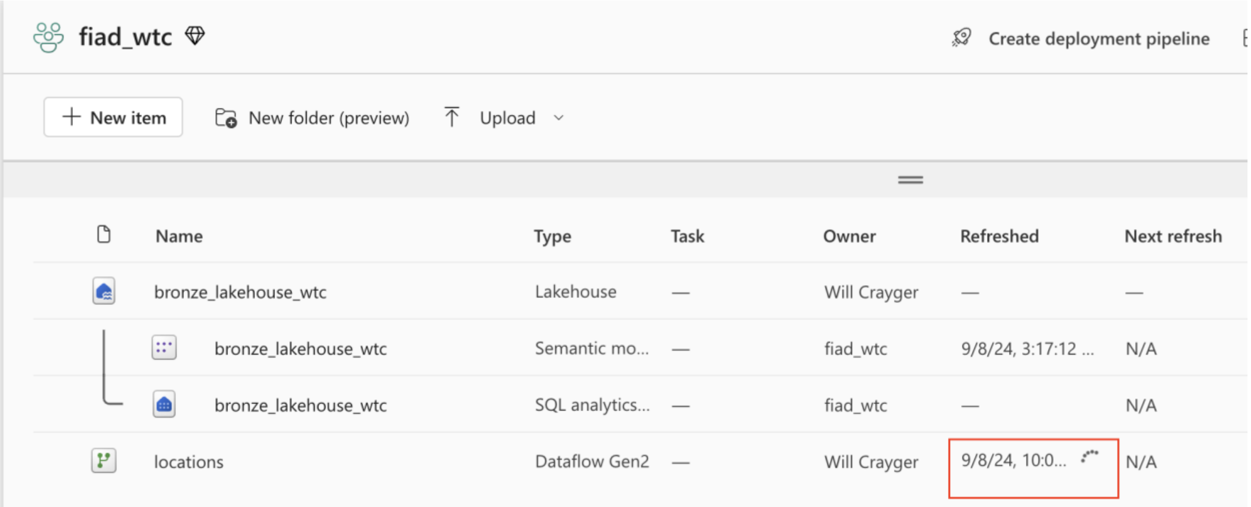
Lakehouse

Once the publishing process begins, you'll see two different loading symbols. The first "swirl" should be quick, and it indicates that the metadata of the dataflow is being saved. You may also notice other artifacts appear in your workspace during this process, which are temporary and will automatically disappear upon completion.

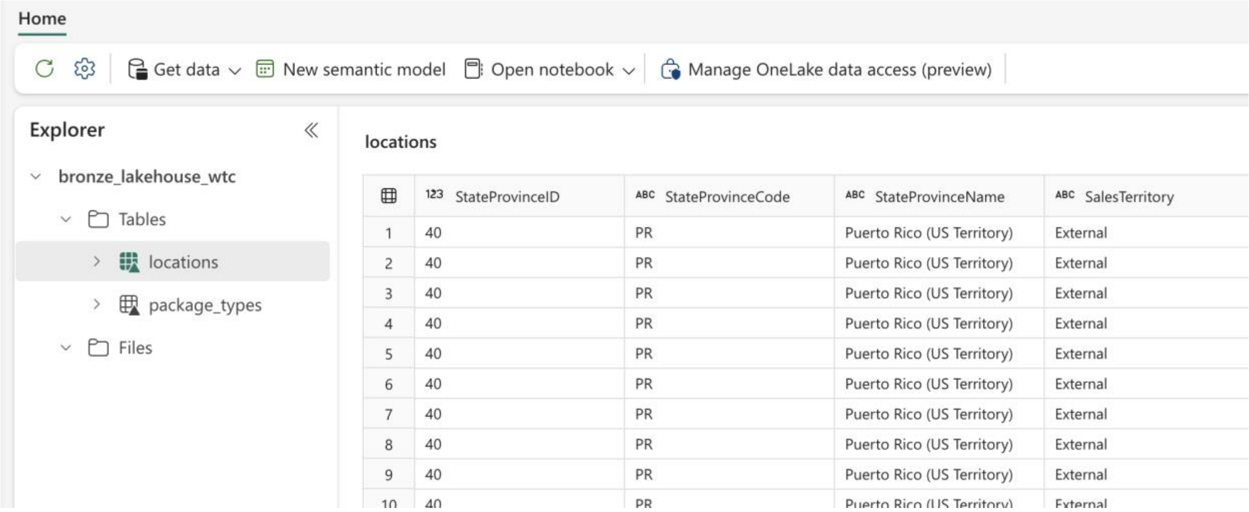


Name	Type	Task	Owner	Refreshed	Next refresh
bronze_lakehouse_wtc	Lakehouse	—	Will Crayger	—	—
bronze_lakehouse_wtc	Semantic mo...	—	fiad_wtc	9/8/24, 3:17:12 ...	N/A
bronze_lakehouse_wtc	SQL analytics...	—	fiad_wtc	—	N/A
locations	Dataflow Gen2	—	Will Crayger	—	N/A

The second "swirl" represents the execution of the dataflow. After it finishes, navigate to the **Lakehouse**. You should now see your **locations** table under the **Tables** section of the **Lakehouse Explorer**. Click on the table to preview the data.



Name	Type	Task	Owner	Refreshed	Next refresh
bronze_lakehouse_wtc	Lakehouse	—	Will Crayger	—	—
bronze_lakehouse_wtc	Semantic mo...	—	fiad_wtc	9/8/24, 3:17:12 ...	N/A
bronze_lakehouse_wtc	SQL analytics...	—	fiad_wtc	—	N/A
locations	Dataflow Gen2	—	Will Crayger	9/8/24, 10:0...	N/A



locations

	StateProvinceID	StateProvinceCode	StateProvinceName	SalesTerritory
1	40	PR	Puerto Rico (US Territory)	External
2	40	PR	Puerto Rico (US Territory)	External
3	40	PR	Puerto Rico (US Territory)	External
4	40	PR	Puerto Rico (US Territory)	External
5	40	PR	Puerto Rico (US Territory)	External
6	40	PR	Puerto Rico (US Territory)	External
7	40	PR	Puerto Rico (US Territory)	External
8	40	PR	Puerto Rico (US Territory)	External
9	40	PR	Puerto Rico (US Territory)	External
10	40	PR	Puerto Rico (US Territory)	External

Note: If the locations table is missing or you see "Undefined" in your Lakehouse, please refresh your browser.

You have now successfully completed this lab.