Microsoft

Microsoft Technical Trainer

Enterprise Skills Initiative

DP-605 LAB 01

Get data in Power BI Desktop

This document is provided by the Microsoft Technical Trainer (MTT) team for attendees of the ESI training.

Summary

The information in this document is based on content reviewed by Microsoft as of the date shown and may change after that date to reflect evolving market requirements. This document does not guarantee that there will be no changes affecting customers after the stated date.

This document is provided for informational purposes only and carries no warranties of any kind.

You are responsible for complying with applicable copyright laws. Without Microsoft's prior consent, reproduction, storage, or redistribution of this document—whether in electronic or physical form, and for any purpose—is not permitted.

Microsoft owns the patents, trademarks, copyrights, and other intellectual property contained in this document. Except where expressly permitted in the document, no patent, trademark, copyright, or other intellectual property rights are granted to any other party.

Microsoft® is a trademark registered in the United States and/or other countries. The actual company and product names mentioned herein may be the trademarks of their respective owners.

Revision history

Date	Version	Author	Change description
2024.05.18	1.0.0	Jin Hwan Woo	Get data in Power BI Desktop
2025.01.03	1.1.0	Jin Hwan Woo	Applied lab update
2025.08.01	1.2.0	Jin Hwan Woo	Updated lab files; applied Power BI updates
2025.10.25	1.3.0	Jin Hwan Woo	Updated the lab to the English version

Contents

Overview

In this lab, you'll get started with Power BI Desktop, connect to data, and use data preview techniques to understand the characteristics and quality of your source data.

You will:

- Open Power BI Desktop.
- Connect to multiple data sources.
- Preview source data with Power Query.
- Use Power Query's data profiling techniques.

Duration: This lab takes about `30 minutes`.

Exercise 01: Get data in Power BI Desktop

Task 01: Start Power BI Desktop

In this task, you'll start by opening a Power BI starter file (.pbix). The starter file doesn't include any data, but it has been specifically configured to help you complete the lab.

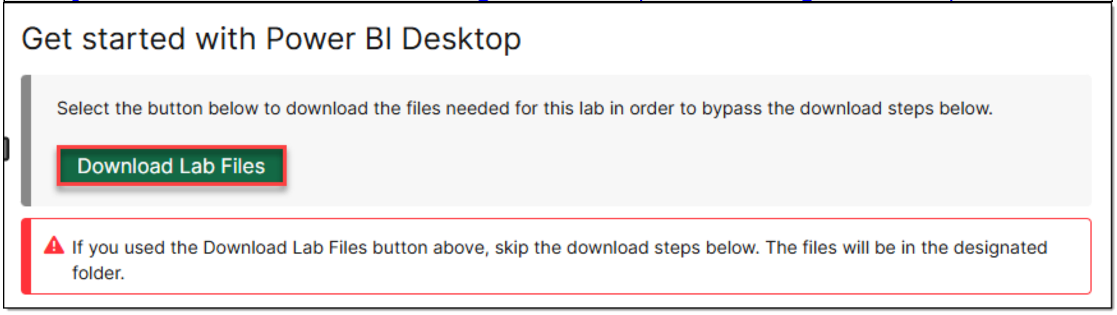
The starter file has the following report-level settings disabled:

- Data Load – Import relationships from data sources on first load
- Data Load – Detect new relationships after data is loaded

While enabling these two options can be helpful when you're developing a data model, they've been disabled to support the lab environment. In the "`Clean, transform, and load data in Power BI`" lab, you'll create relationships and learn why each relationship is added.

1. On the lab machine, open a browser and navigate to the following URL to download the ZIP file. Alternatively, in the lab environment, select the [**Instructions**] tab and then select [**Download Lab Files**] to download the files. In that case, the files will be saved to the "`Downloads`" folder.

```
https://github.com/MicrosoftLearning/PL-300-Microsoft-Power-BI-Data-Analyst/raw/Main/Allfiles/Labs/01-get-data-in-power-bi/01-get-data.zip
```

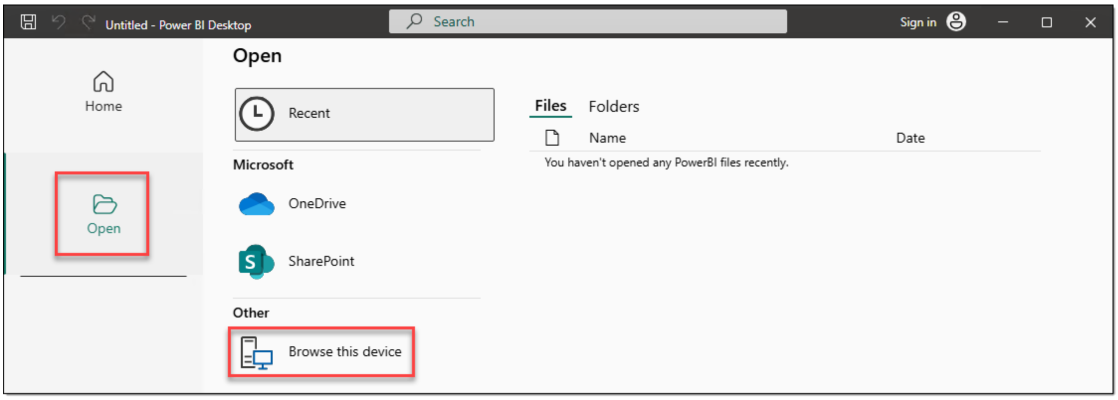


Get started with Power BI Desktop

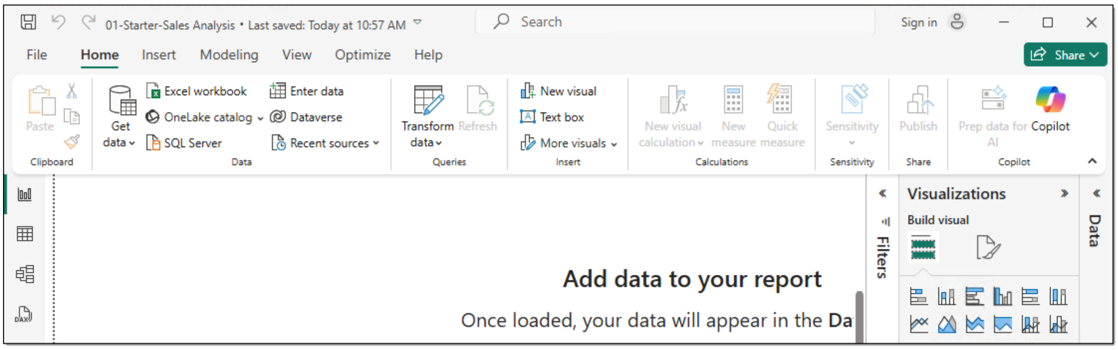
Select the button below to download the files needed for this lab in order to bypass the download steps below.

If you used the Download Lab Files button above, skip the download steps below. The files will be in the designated folder.

2. Start [**Power BI Desktop**]. Select [**Open – Browse this device**].



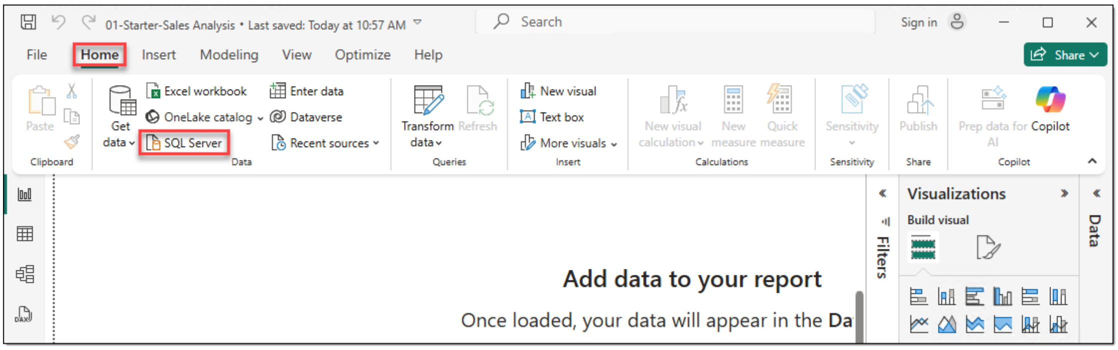
3. In the [**Open**] dialog, select the previously downloaded and extracted `01-Starter-Sales Analysis.pbix` file. If Power BI prompts you to sign in or ask for an email address, ignore the prompt (select Skip if shown).



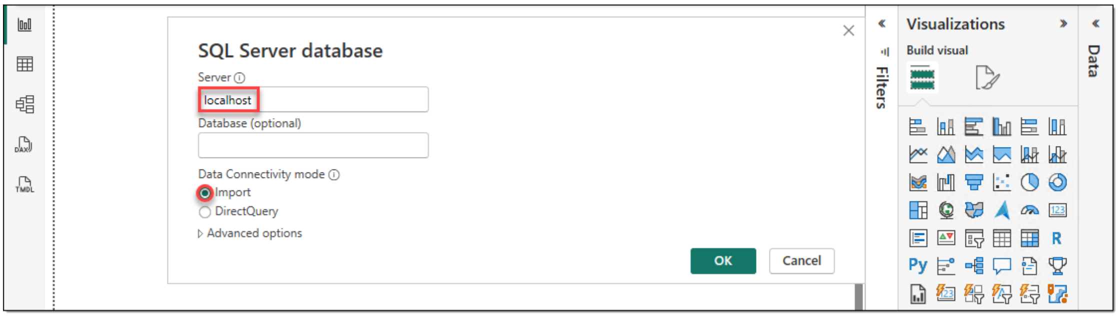
Task 02: Get data from SQL Server

In this task, you'll connect to a SQL Server database and use Power Query to create queries that import tables.

1. On the [**Home**] ribbon, select [**Data – SQL Server**].



2. In the [**SQL Server database**] dialog, configure the settings as follows, and then select [**OK**]. For this lab, you'll connect to SQL Server using localhost so the gateway data source won't be discovered. This approach isn't recommended for production solutions.
   - Server: `localhost`
   - Data Connectivity mode: `Import`



SQL Server database

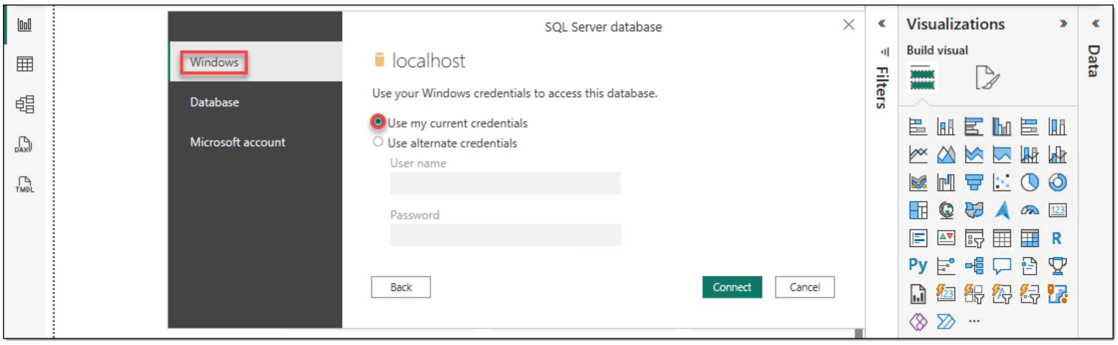
Server ⓘ
localhost

Database (optional)

Data Connectivity mode ⓘ
● Import
○ DirectQuery

▷ Advanced options

3. In the [**SQL Server database**] dialog, select "`Use my current credentials`", and then select [**Connect**]. If the [**Encryption Support**] dialog appears, select [**OK**].



SQL Server database

localhost

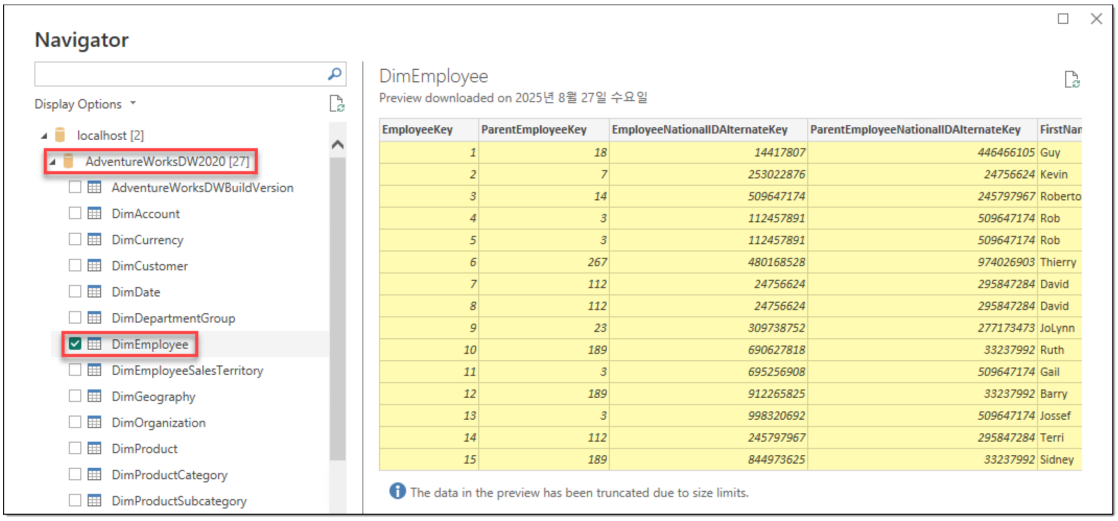
Use your Windows credentials to access this database.

● Use my current credentials
○ Use alternate credentials

User name

Password

4. In the [**Navigator**] pane, expand the `AdventureWorksDW2020` database, and then select the `DimEmployee` table.
   - The `AdventureWorksDW2020` database is based on the `AdventureWorksDW2017` sample database and has been modified to support the learning objectives of this course lab.
   - When you select `DimEmployee`, a preview shows sample columns and rows from the table.



Navigator

Display Options ▾

- localhost [2]
  - AdventureWorksDW2020 [27]
    - □ AdventureWorksDWBuildVersion
    - □ DimAccount
    - □ DimCurrency
    - □ DimCustomer
    - □ DimDate
    - □ DimDepartmentGroup
    - ☑ DimEmployee
    - □ DimEmployeeSalesTerritory
    - □ DimGeography
    - □ DimOrganization
    - □ DimProduct
    - □ DimProductCategory
    - □ DimProductSubcategory

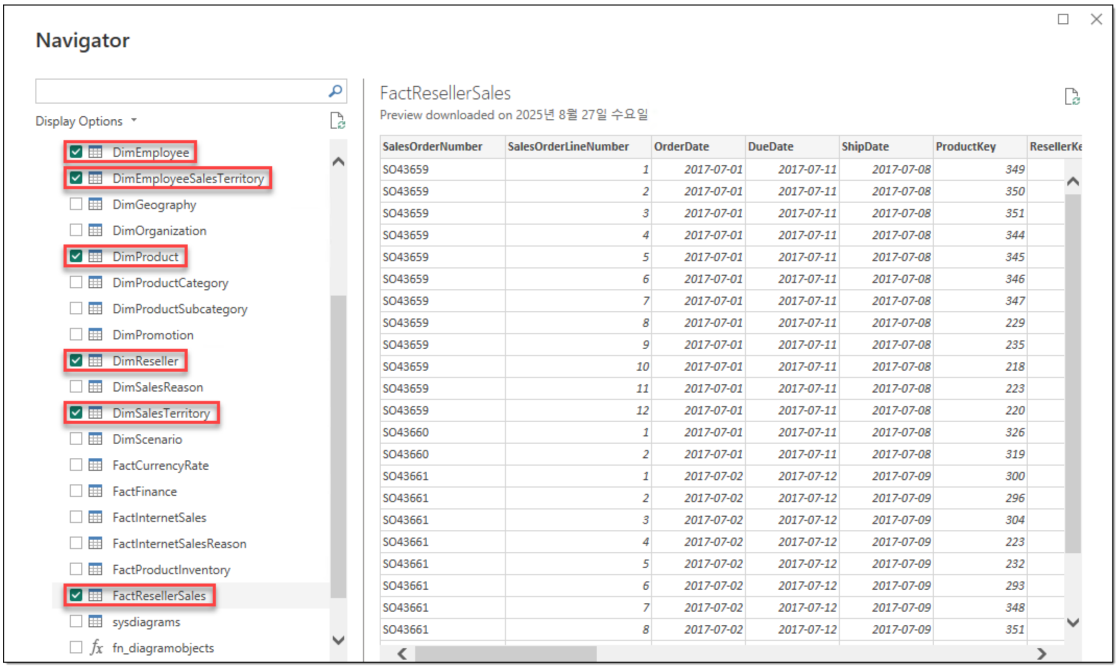
DimEmployee

Preview downloaded on 2025년 8월 27일 수요일

EmployeeKey	ParentEmployeeKey	EmployeeNationalIDAlternateKey	ParentEmployeeNationalIDAlternateKey	FirstNar
1	18	14417807	446466105	Guy
2	7	253022876	24756624	Kevin
3	14	509647174	245797967	Roberto
4	3	112457891	509647174	Rob
5	3	112457891	509647174	Rob
6	267	480168528	974026903	Thierry
7	112	24756624	295847284	David
8	112	24756624	295847284	David
9	23	309738752	277173473	JoLynn
10	189	690627818	33237992	Ruth
11	3	695256908	509647174	Gail
12	189	912265825	33237992	Barry
13	3	998320692	509647174	Jossef
14	112	245797967	295847284	Terri
15	189	844973625	33237992	Sidney

ⓘ The data in the preview has been truncated due to size limits.

5. In the [**Navigator**] pane, select the following six tables, and then select [**Transform Data**]:
   - `DimEmployee`
   - `DimEmployeeSalesTerritory`
   - `DimProduct`
   - `DimReseller`
   - `DimSalesTerritory`
   - `FactResellerSales`



- ☑ DimEmployee
- ☑ DimEmployeeSalesTerritory
- □ DimGeography
- □ DimOrganization
- ☑ DimProduct
- □ DimProductCategory
- □ DimProductSubcategory
- □ DimPromotion
- ☑ DimReseller
- □ DimSalesReason
- ☑ DimSalesTerritory
- □ DimScenario
- □ FactCurrencyRate
- □ FactFinance
- □ FactInternetSales
- □ FactInternetSalesReason
- □ FactProductInventory
- ☑ FactResellerSales
- □ sysdiagrams
- □ fn_diagramobjects

FactResellerSales

Preview downloaded on 2025년 8월 27일 수요일

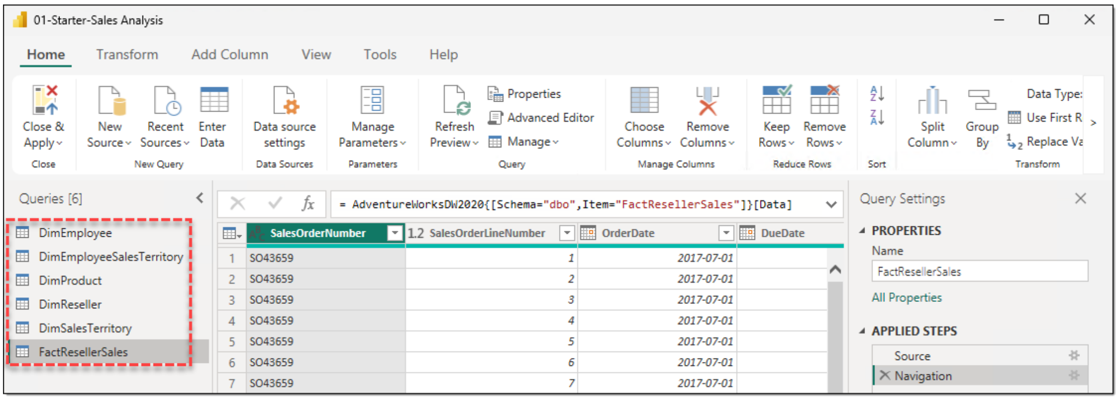
SalesOrderNumber	SalesOrderLineNumber	OrderDate	DueDate	ShipDate	ProductKey	ResellerKe
SO43659	1	2017-07-01	2017-07-11	2017-07-08	349	
SO43659	2	2017-07-01	2017-07-11	2017-07-08	350	
SO43659	3	2017-07-01	2017-07-11	2017-07-08	351	
SO43659	4	2017-07-01	2017-07-11	2017-07-08	344	
SO43659	5	2017-07-01	2017-07-11	2017-07-08	345	
SO43659	6	2017-07-01	2017-07-11	2017-07-08	346	
SO43659	7	2017-07-01	2017-07-11	2017-07-08	347	
SO43659	8	2017-07-01	2017-07-11	2017-07-08	229	
SO43659	9	2017-07-01	2017-07-11	2017-07-08	235	
SO43659	10	2017-07-01	2017-07-11	2017-07-08	218	
SO43659	11	2017-07-01	2017-07-11	2017-07-08	223	
SO43659	12	2017-07-01	2017-07-11	2017-07-08	220	
SO43660	1	2017-07-01	2017-07-11	2017-07-08	326	
SO43660	2	2017-07-01	2017-07-11	2017-07-08	319	
SO43661	1	2017-07-02	2017-07-12	2017-07-09	300	
SO43661	2	2017-07-02	2017-07-12	2017-07-09	296	
SO43661	3	2017-07-02	2017-07-12	2017-07-09	304	
SO43661	4	2017-07-02	2017-07-12	2017-07-09	223	
SO43661	5	2017-07-02	2017-07-12	2017-07-09	232	
SO43661	6	2017-07-02	2017-07-12	2017-07-09	293	
SO43661	7	2017-07-02	2017-07-12	2017-07-09	348	
SO43661	8	2017-07-02	2017-07-12	2017-07-09	351	

You're now connected to the data, and [**Power Query Editor**] opens for the next task.

Task 03: Preview data in Power Query Editor

In this task, you'll get oriented with [**Power Query Editor**] and profile the data. This review will help you decide how to clean and transform the data later. You'll examine both the dimension tables (prefixed with `Dim`) and the fact table (prefixed with `Fact`).

1. In the [**Power Query Editor**] window, look at the [**Queries**] pane on the left. The [**Queries**] pane contains one query for each table you selected.



Queries [6]

- DimEmployee
- DimEmployeeSalesTerritory
- DimProduct
- DimReseller
- DimSalesTerritory
- FactResellerSales

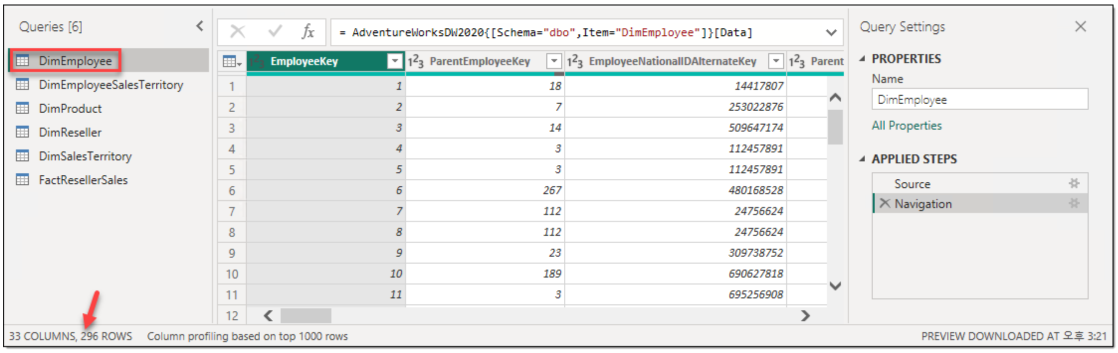
```
= AdventureWorksDW2020{[Schema="dbo",Item="FactResellerSales"]}[Data]
```

	SalesOrderNumber	1.2 SalesOrderLineNumber	OrderDate	DueDate
1	SO43659	1	2017-07-01	
2	SO43659	2	2017-07-01	
3	SO43659	3	2017-07-01	
4	SO43659	4	2017-07-01	
5	SO43659	5	2017-07-01	
6	SO43659	6	2017-07-01	
7	SO43659	7	2017-07-01	

Name: FactResellerSales

APPLIED STEPS:
- Source
- Navigation

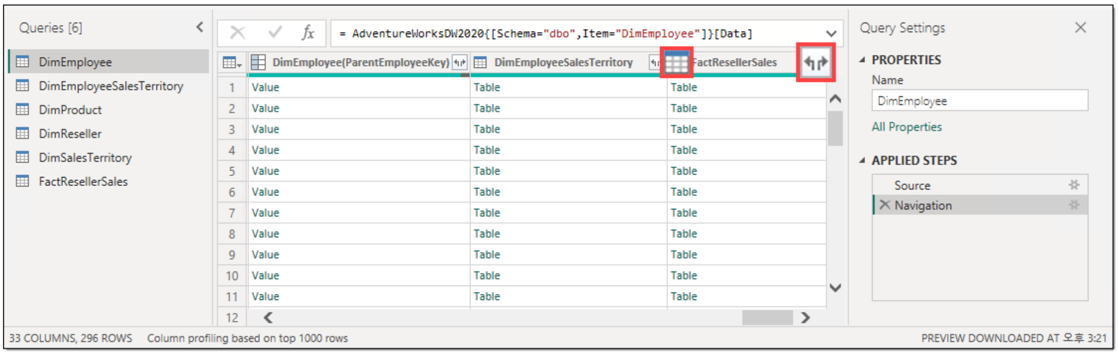
2. In the [**Power Query Editor**], select the `DimEmployee` query in the [**Queries**] pane.
   - The `DimEmployee` table in SQL Server stores one row per employee. A subset of these rows represents salespeople relevant to the model you'll build.
   - In the lower-left of the status bar, you can see basic table statistics. For this table, there are `33 columns` and `296 rows`.



```
= AdventureWorksDW2020{[Schema="dbo",Item="DimEmployee"]}[Data]
```

	EmployeeKey	ParentEmployeeKey	EmployeeNationalIDAlternateKey	Parent
1	1	18	14417807	
2	2	7	253022876	
3	3	14	509647174	
4	4	3	112457891	
5	5	3	112457891	
6	6	267	480168528	
7	7	112	24756624	
8	8	112	24756624	
9	9	23	309738752	
10	10	189	690627818	
11	11	3	695256908	
12				

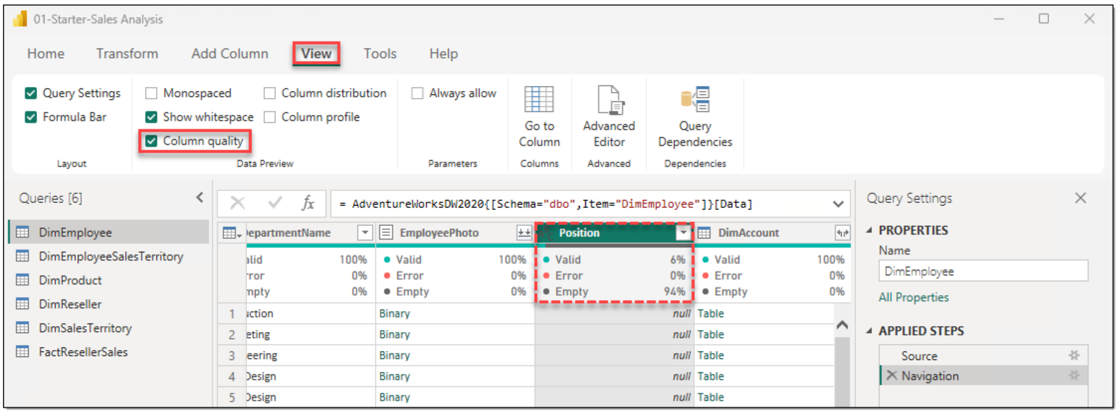
3. In the data preview, scroll to the right to review all columns. The last five columns contain **Table** or **Value** links.
   - These five columns indicate relationships to other tables in the database.
   - You can use these columns to join tables; you'll perform those joins in the "`Clean, transform, and load data in Power BI`" lab.



```
= AdventureWorksDW2020{[Schema="dbo",Item="DimEmployee"]}[Data]
```

	DimEmployee(ParentEmployeeKey)	DimEmployeeSalesTerritory	FactResellerSales
1	Value	Table	Table
2	Value	Table	Table
3	Value	Table	Table
4	Value	Table	Table
5	Value	Table	Table
6	Value	Table	Table
7	Value	Table	Table
8	Value	Table	Table
9	Value	Table	Table
10	Value	Table	Table
11	Value	Table	Table
12			

4. To access column quality, on the [**View**] ribbon, select [**Data Preview** – **Column quality**]. The column quality feature lets you quickly see the percentage of **Valid**, **Error**, or **Empty** values found in each column.
   - Confirm that the **Position** column has **94%** empty rows.

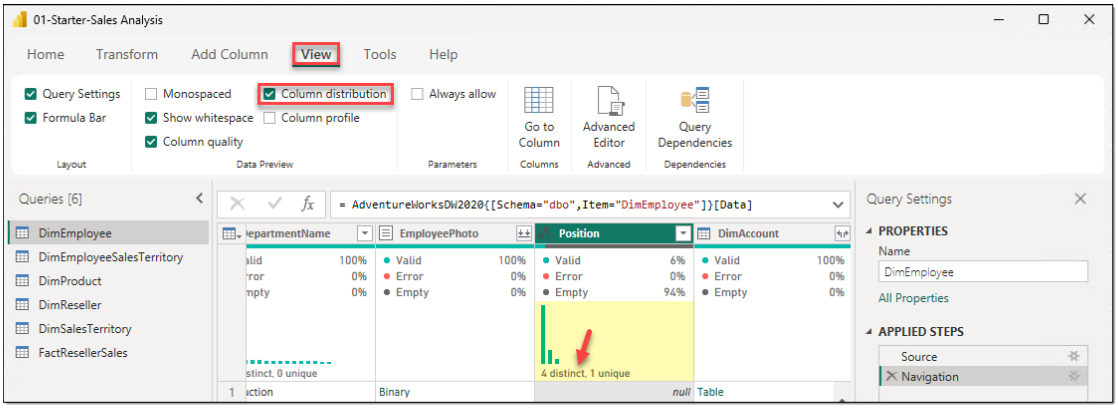


☑ Query Settings
☑ Formula Bar
☐ Monospaced
☑ Show whitespace
☑ Column quality
☐ Column distribution
☐ Column profile
☐ Always allow

```
= AdventureWorksDW2020{[Schema="dbo",Item="DimEmployee"]}[Data]
```

	ɪepartmentName	EmployeePhoto	Position	DimAccount
	ılid 100% ror 0% npty 0%	● Valid 100% ● Error 0% ● Empty 0%	● Valid 6% ● Error 0% ● Empty 94%	● Valid 100% ● Error 0% ● Empty 0%
1	ıction	Binary	null	Table
2	eting	Binary	null	Table
3	eering	Binary	null	Table
4	Design	Binary	null	Table
5	Design	Binary	null	Table

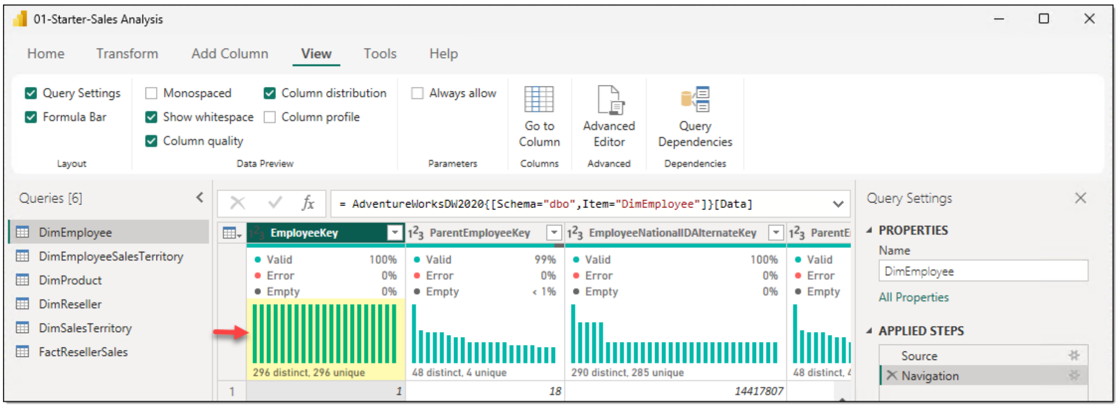
5. To review column distribution, on the [**View**] ribbon select [**Data Preview** – **Column distribution**].
   - Confirm that the Position column shows **4 distinct** values and **1 unique** value.



☑ Query Settings ☑ Formula Bar □ Monospaced ☑ Show whitespace ☑ Column quality ☑ Column distribution □ Column profile □ Always allow

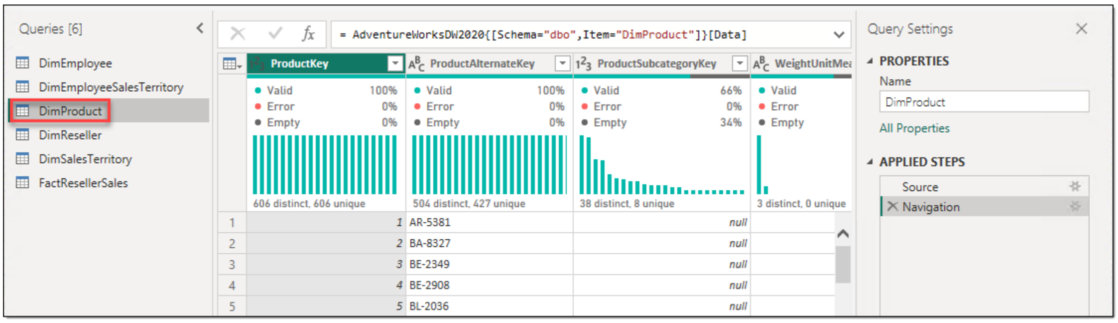
```
= AdventureWorksDW2020{[Schema="dbo",Item="DimEmployee"]}[Data]
```

6. Review the distribution for the `EmployeeKey` column. Confirm that it shows **296 distinct values** and **296 unique values**.
   - When the **distinct count** and **unique count** are the same, the column contains only unique values. *Distinct* means the number of different values present; *unique* means values that appear exactly once in the column. For example, if a column has five **A** values, three **B** values, and one **C** value, the **distinct count** is **3**, while the **unique count** is **1**.
   - Having unique columns in some model tables is important for modeling. You'll use such unique columns to create one-to-many relationships in the "`Configure a semantic model in Power BI Desktop`" lab.



```
= AdventureWorksDW2020{[Schema="dbo",Item="DimEmployee"]}[Data]
```

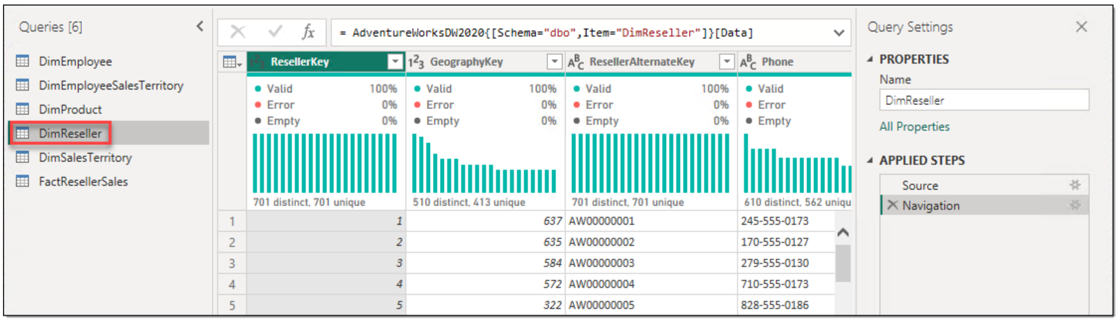
7. In the [**Queries**] pane, select the `DimProduct` query. The `DimProduct` table contains one row for each product that the company sells.



```
= AdventureWorksDW2020{[Schema="dbo",Item="DimProduct"]}[Data]
```

	ProductKey	ProductAlternateKey	ProductSubcategoryKey	WeightUnitMea
1	1	AR-5381	null	
2	2	BA-8327	null	
3	3	BE-2349	null	
4	4	BE-2908	null	
5	5	BL-2036	null	

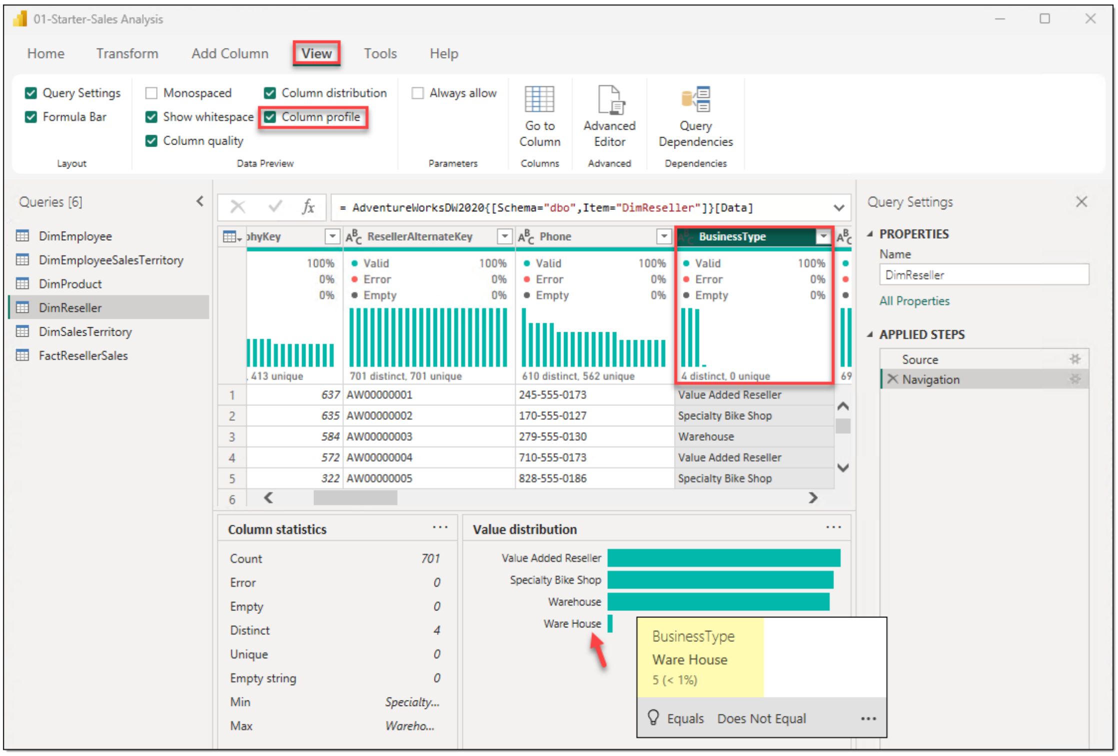
8. In the [**Queries**] pane, select the `DimReseller` query. The `DimReseller` table contains reseller information, with one row per reseller. Resellers sell, distribute, or add value to Adventure Works products.



= AdventureWorksDW2020{[Schema="dbo",Item="DimReseller"]}[Data]

	ResellerKey	GeographyKey	ResellerAlternateKey	Phone
	● Valid 100% ● Error 0% ● Empty 0%	● Valid 100% ● Error 0% ● Empty 0%	● Valid 100% ● Error 0% ● Empty 0%	● Valid 100% ● Error 0% ● Empty 0%
	701 distinct, 701 unique	510 distinct, 413 unique	701 distinct, 701 unique	610 distinct, 562 uniqu
1	1	637	AW00000001	245-555-0173
2	2	635	AW00000002	170-555-0127
3	3	584	AW00000003	279-555-0130
4	4	572	AW00000004	710-555-0173
5	5	322	AW00000005	828-555-0186

9. On the [**View**] ribbon, select [**Data Preview – Column profile**]. Then select the `BusinessType` column header and confirm that a new pane appears below the data preview.
   - In the data preview, review the **Column statistics** and **Value distribution**.
   - In the "`Value distribution`" area, confirm there are two labels: "`Warehouse`" and "`Ware House`". "`Ware House`" is a misspelling of "`Warehouse`".
   - Hover over the "`Ware House`" bar to confirm that **5 rows** contain this value. You'll fix the labels for these five rows by applying a transformation in the "`Clean, transform, and load data in Power BI Desktop`" lab.



= AdventureWorksDW2020{[Schema="dbo",Item="DimReseller"]}[Data]

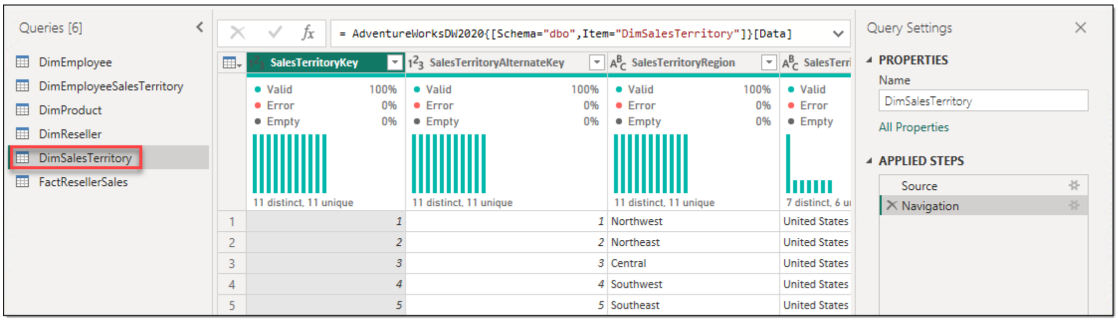
	phyKey	ResellerAlternateKey	Phone	BusinessType
	100% 0% 0%	● Valid 100% ● Error 0% ● Empty 0%	● Valid 100% ● Error 0% ● Empty 0%	● Valid 100% ● Error 0% ● Empty 0%
	413 unique	701 distinct, 701 unique	610 distinct, 562 unique	4 distinct, 0 unique
1	637	AW00000001	245-555-0173	Value Added Reseller
2	635	AW00000002	170-555-0127	Specialty Bike Shop
3	584	AW00000003	279-555-0130	Warehouse
4	572	AW00000004	710-555-0173	Value Added Reseller
5	322	AW00000005	828-555-0186	Specialty Bike Shop
6				

Column statistics

Count	701
Error	0
Empty	0
Distinct	4
Unique	0
Empty string	0
Min	Specialty...
Max	Wareho...

Value distribution

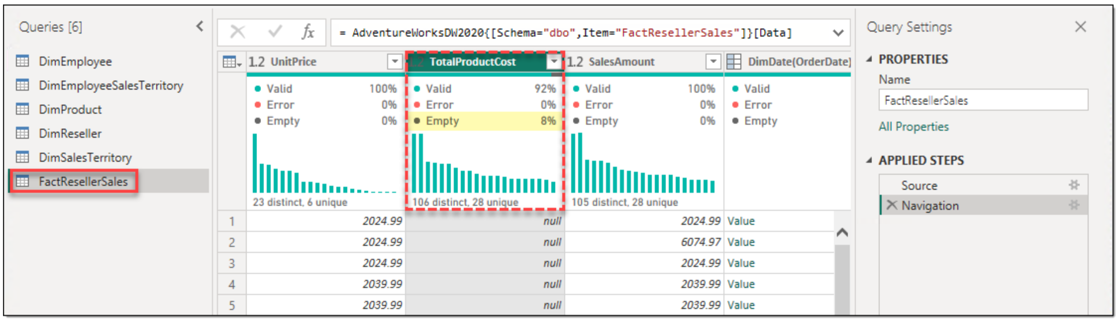
10. In the [**Queries**] pane, select the `DimSalesTerritory` query. The `DimSalesTerritory` table contains one row per sales territory, including the company headquarters.
    - **Regions** are assigned to a **Country**, and **Countries** are assigned to a **Group**.
    - In the "`Configure a semantic model in Power BI Desktop`" lab, you'll create a hierarchy that supports analysis at the **region**, **country**, and **group** levels.



```
= AdventureWorksDW2020{[Schema="dbo",Item="DimSalesTerritory"]}[Data]
```

	SalesTerritoryKey	SalesTerritoryAlternateKey	SalesTerritoryRegion	SalesTerri
1	1	1	Northwest	United States
2	2	2	Northeast	United States
3	3	3	Central	United States
4	4	4	Southwest	United States
5	5	5	Southeast	United States

11. In the [**Queries**] pane, select the `FactResellerSales` query. The `FactResellerSales` table contains one row per sales order, and each sales order includes one or more **line items**.
    - Review the **column quality** for the `TotalProductCost` column and confirm that **8%** of rows are empty.
    - Missing values in `TotalProductCost` represent a data quality issue. To address this, in the "`Clean, transform, and load data in Power BI Desktop`" lab you'll apply a transformation that fills the missing values by using the `standard cost` stored in the related `DimProduct` table.



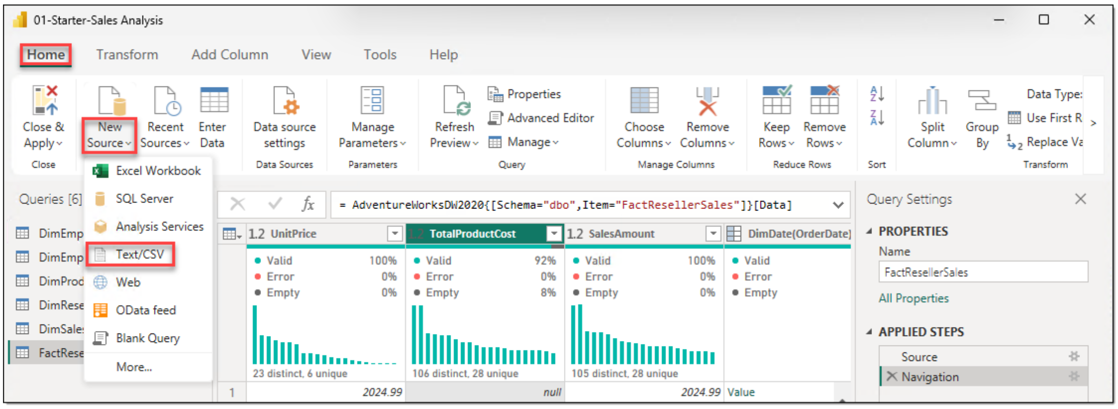
```
= AdventureWorksDW2020{[Schema="dbo",Item="FactResellerSales"]}[Data]
```

	UnitPrice	TotalProductCost	SalesAmount	DimDate(OrderDate)
1	2024.99	null	2024.99	Value
2	2024.99	null	6074.97	Value
3	2024.99	null	2024.99	Value
4	2039.99	null	2039.99	Value
5	2039.99	null	2039.99	Value

Task 04: Get data from a CSV file

In this task, you'll create a new query from a CSV file.

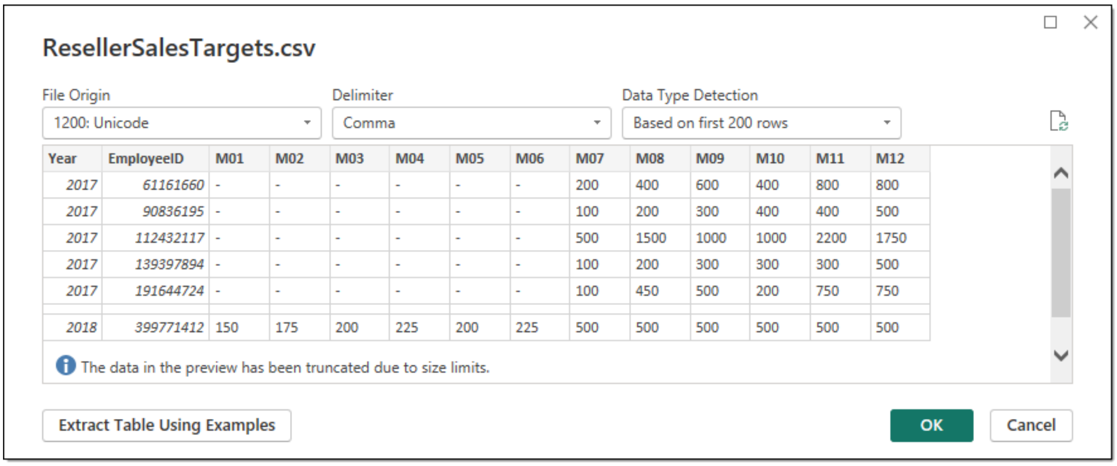
1. On the [**Home**] ribbon, select [**New Query** – **New Source** – **Text/CSV**]. Select the previously downloaded `ResellerSalesTargets.csv` file, and then select [**Open**].



```
= AdventureWorksDW2020{[Schema="dbo",Item="FactResellerSales"]}[Data]
```

	UnitPrice	TotalProductCost	SalesAmount	DimDate(OrderDate)
1	2024.99	null	2024.99	Value

2. In the `ResellerSalesTargets.csv` dialog, review the preview data, and then select [**OK**].



ResellerSalesTargets.csv

File Origin: 1200: Unicode

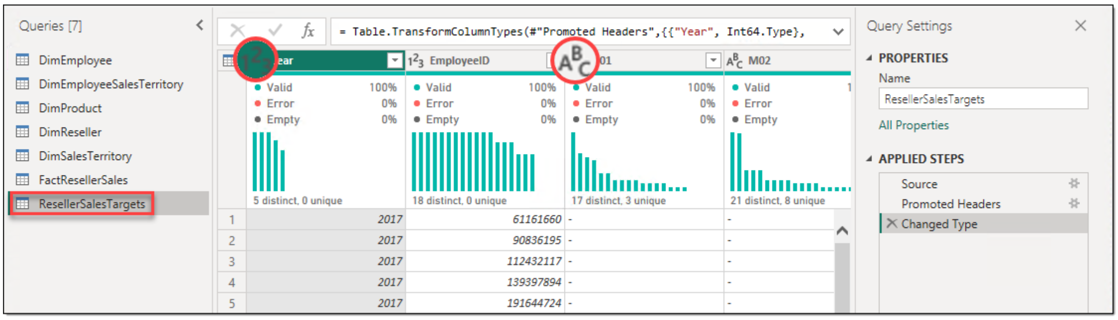
Delimiter: Comma

Data Type Detection: Based on first 200 rows

Year	EmployeeID	M01	M02	M03	M04	M05	M06	M07	M08	M09	M10	M11	M12
2017	61161660	-	-	-	-	-	-	200	400	600	400	800	800
2017	90836195	-	-	-	-	-	-	100	200	300	400	400	500
2017	112432117	-	-	-	-	-	-	500	1500	1000	1000	2200	1750
2017	139397894	-	-	-	-	-	-	100	200	300	300	300	500
2017	191644724	-	-	-	-	-	-	100	450	500	200	750	750
2018	399771412	150	175	200	225	200	225	500	500	500	500	500	500

The data in the preview has been truncated due to size limits.

3. In the [**Queries**] pane, confirm that the `ResellerSalesTargets` query has been added. The `ResellerSalesTargets` CSV file contains one row per salesperson per year.
   - Each row records **12 monthly sales targets** (in thousands).
   - **Adventure Works** uses a fiscal year that starts on **July 1**.
   - Verify that none of the columns contain empty values. Where a monthly target is missing, a **hyphen (–)** is stored instead.
   - Review the icon to the left of each column header. The icon indicates the column's data type: **123** means **Whole Number** (integer), and **ABC** means **Text**.



Queries [7]
- DimEmployee
- DimEmployeeSalesTerritory
- DimProduct
- DimReseller
- DimSalesTerritory
- FactResellerSales
- ResellerSalesTargets

```
= Table.TransformColumnTypes(#"Promoted Headers",{{"Year", Int64.Type},
```

	Year	EmployeeID	M01	M02
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid Error Empty
	5 distinct, 0 unique	18 distinct, 0 unique	17 distinct, 3 unique	21 distinct, 8 unique
1	2017	61161660	-	-
2	2017	90836195	-	-
3	2017	112432117	-	-
4	2017	139397894	-	-
5	2017	191644724	-	-

Query Settings

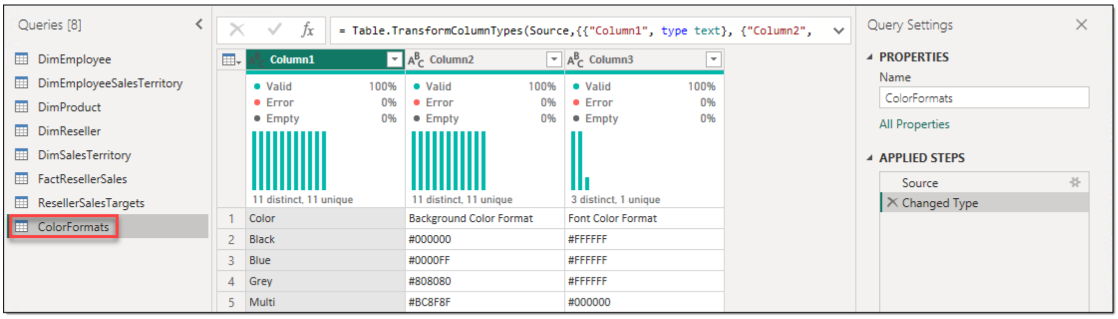
PROPERTIES

Name: ResellerSalesTargets

All Properties

APPLIED STEPS
- Source
- Promoted Headers
- Changed Type

4. Repeat the same steps to import the `ColorFormats.csv` file. The `ColorFormats` CSV file contains one row per product color. Each row specifies background and font color formatting by using **HEX** codes.



Queries [8]
- DimEmployee
- DimEmployeeSalesTerritory
- DimProduct
- DimReseller
- DimSalesTerritory
- FactResellerSales
- ResellerSalesTargets
- ColorFormats

```
= Table.TransformColumnTypes(Source,{{"Column1", type text}, {"Column2",
```

	Column1	Column2	Column3
	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%	Valid 100% Error 0% Empty 0%
	11 distinct, 11 unique	11 distinct, 11 unique	3 distinct, 1 unique
1	Color	Background Color Format	Font Color Format
2	Black	#000000	#FFFFFF
3	Blue	#0000FF	#FFFFFF
4	Grey	#808080	#FFFFFF
5	Multi	#BC8F8F	#000000

Query Settings

PROPERTIES

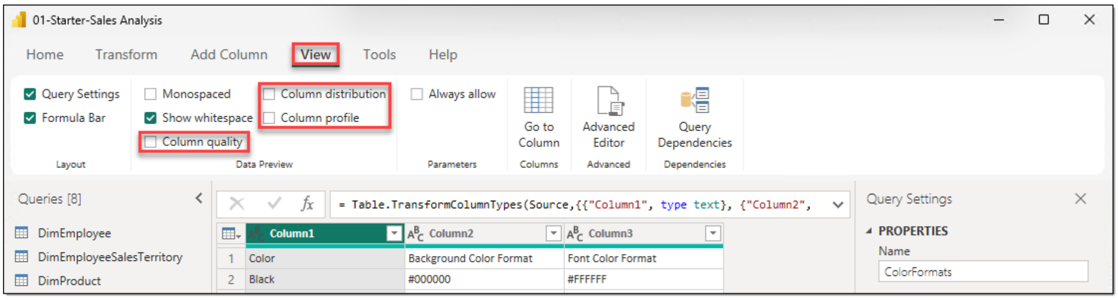
Name: ColorFormats

All Properties

APPLIED STEPS
- Source
- Changed Type

Task 05: Finish the lab

1. On the [**View**] ribbon, in [**Data Preview**], clear the check boxes for "`Column quality`", "`Column distribution`", and "`Column profile`".



```
= Table.TransformColumnTypes(Source,{{"Column1", type text}, {"Column2",
```

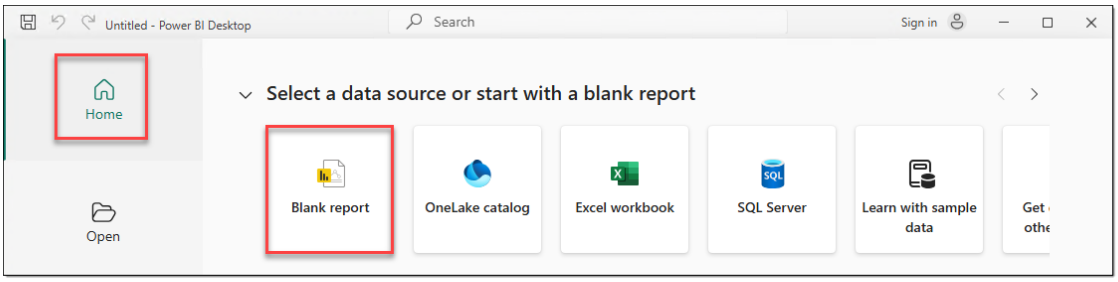
	Column1	Column2	Column3
1	Color	Background Color Format	Font Color Format
2	Black	#000000	#FFFFFF

Exercise 02: Use dynamic M query parameters in Power BI Desktop (Optional)

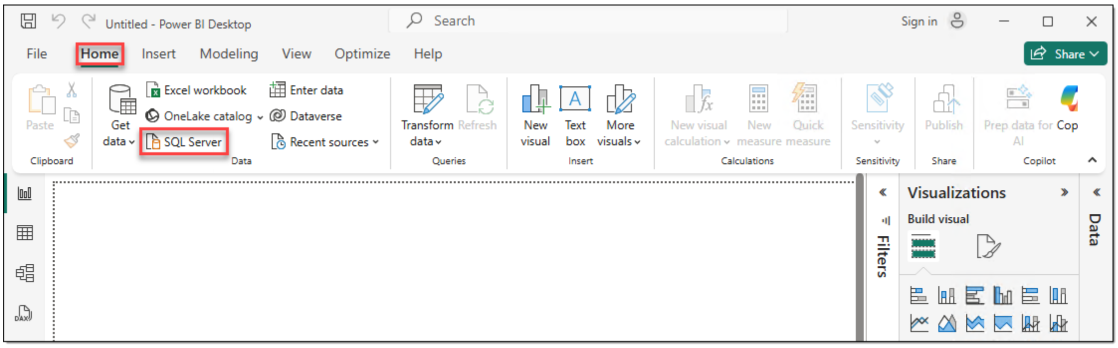
In this exercise, you'll create and use dynamic M query parameters in Power BI Desktop. This exercise is optional.

Task 01: Create a dynamic report for a single value

1. Open Power BI Desktop, then select [**Home – Blank report**].



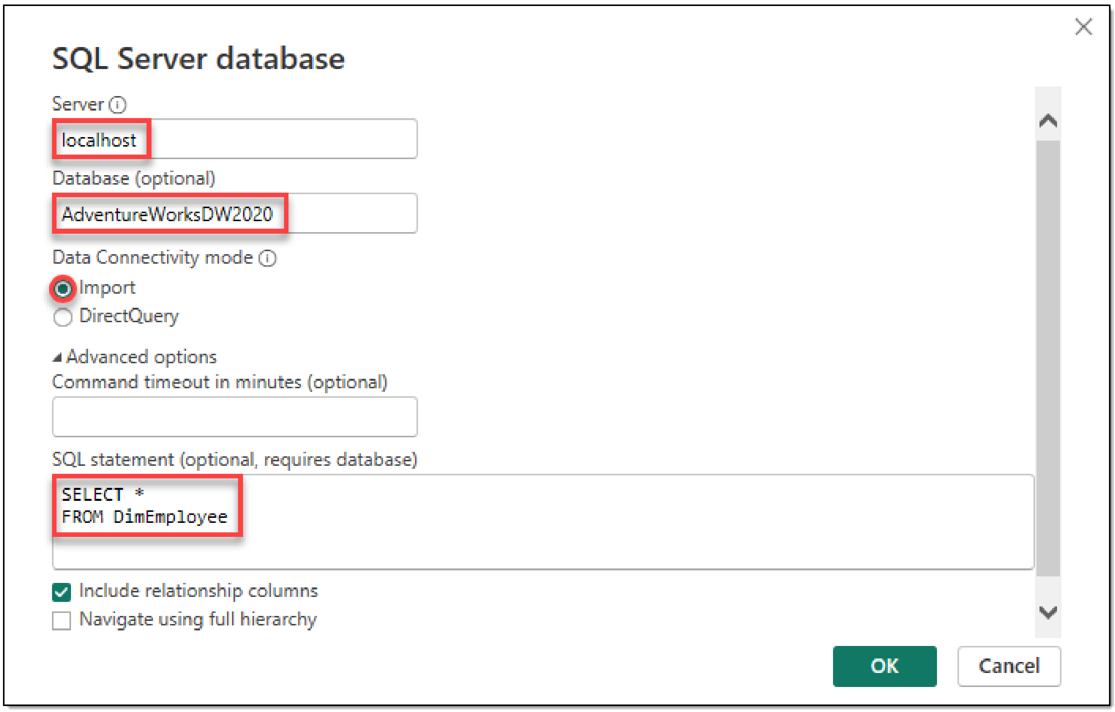
2. In Power BI Desktop, on the [**Home**] ribbon, select [**Data – SQL Server**].



3. In the [**SQL Server database**] dialog, configure the settings as follows, and then select [**OK**].

- Server: `localhost`
- Database: `AdventureWorksDW2020`
- Data Connectivity mode: `Import`
- Expand "`Advanced options`", and in **SQL statement**, enter the SQL query provided by the lab instructions.

```
SELECT *
FROM DimEmployee
```



SQL Server database

Server ⓘ

localhost

Database (optional)

AdventureWorksDW2020

Data Connectivity mode ⓘ

● Import

○ DirectQuery

◢ Advanced options

Command timeout in minutes (optional)

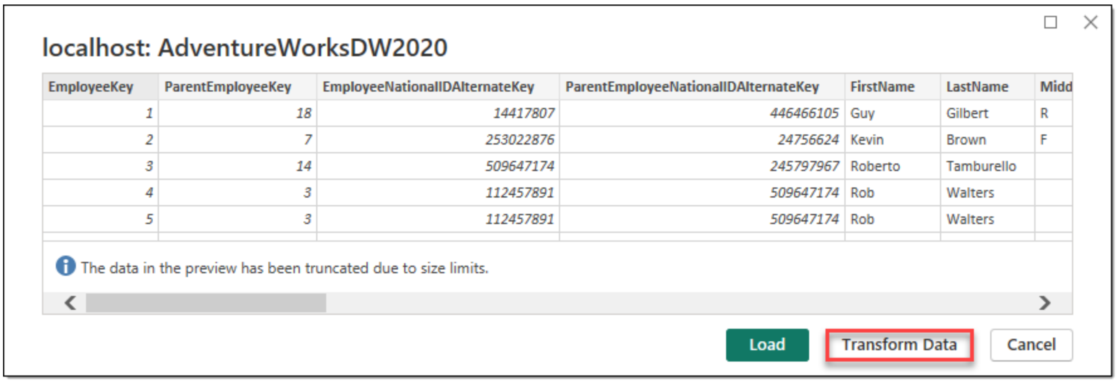
SQL statement (optional, requires database)

```
SELECT *
FROM DimEmployee
```

☑ Include relationship columns

☐ Navigate using full hierarchy

4. In the [localhost: AdventureWorksDW2020] dialog, select [Transform Data].

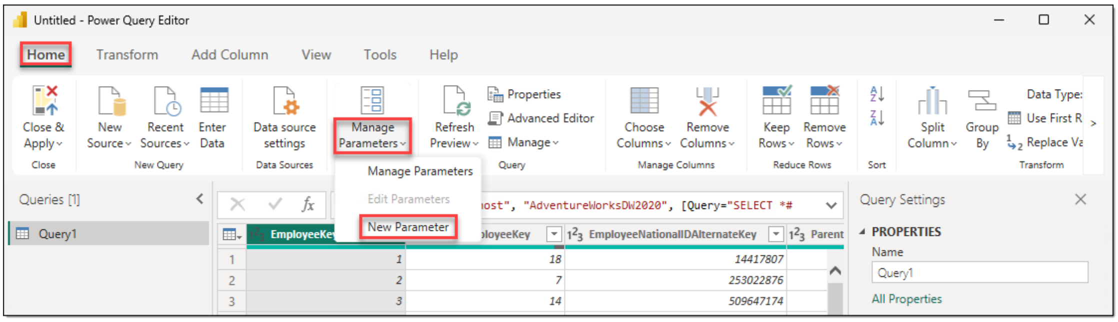


localhost: AdventureWorksDW2020

EmployeeKey	ParentEmployeeKey	EmployeeNationalIDAlternateKey	ParentEmployeeNationalIDAlternateKey	FirstName	LastName	Midd
1	18	14417807	446466105	Guy	Gilbert	R
2	7	253022876	24756624	Kevin	Brown	F
3	14	509647174	245797967	Roberto	Tamburello	
4	3	112457891	509647174	Rob	Walters	
5	3	112457891	509647174	Rob	Walters	

ⓘ The data in the preview has been truncated due to size limits.

5. In [Power Query Editor], on the [Home] ribbon, select [Parameters – Manage Parameters – New Parameter].

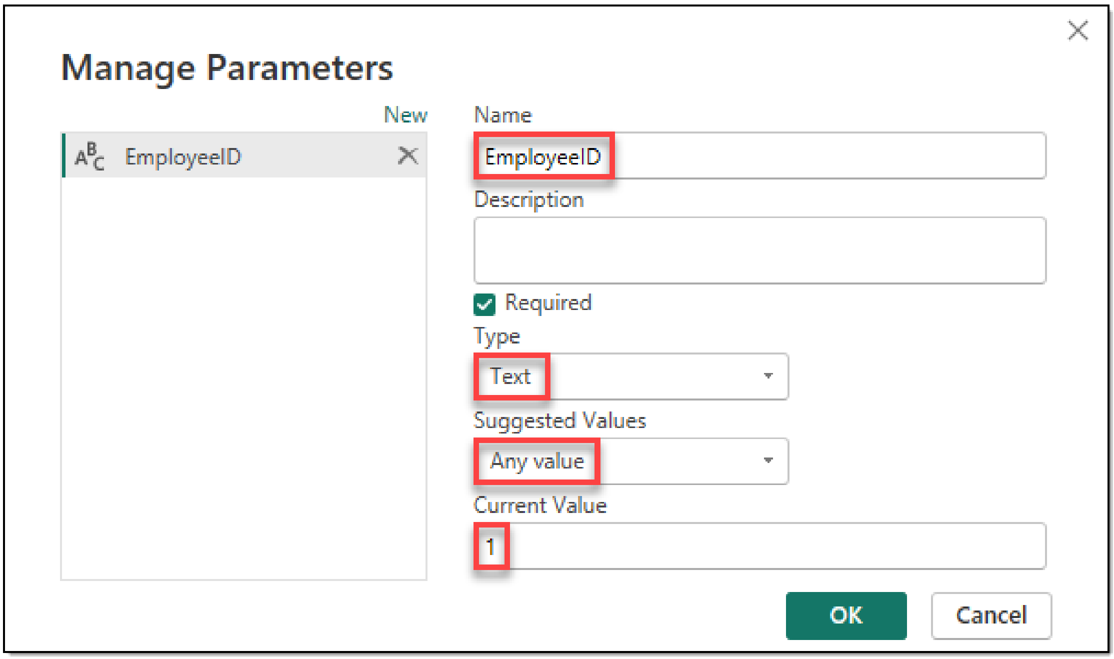


ost", "AdventureWorksDW2020", [Query="SELECT *#

EmployeeKe	loyeeKey	EmployeeNationalIDAlternateKey	Parent
1	18	14417807	
2	7	253022876	
3	14	509647174	

6. In the [Manage Parameters] dialog, configure the settings as follows, and then select [OK].

- Name: `EmployeeID`
- Type: `Text`
- Suggested Values: `Any value`
- Current Value: `1`



Manage Parameters

Name: EmployeeID

Description:

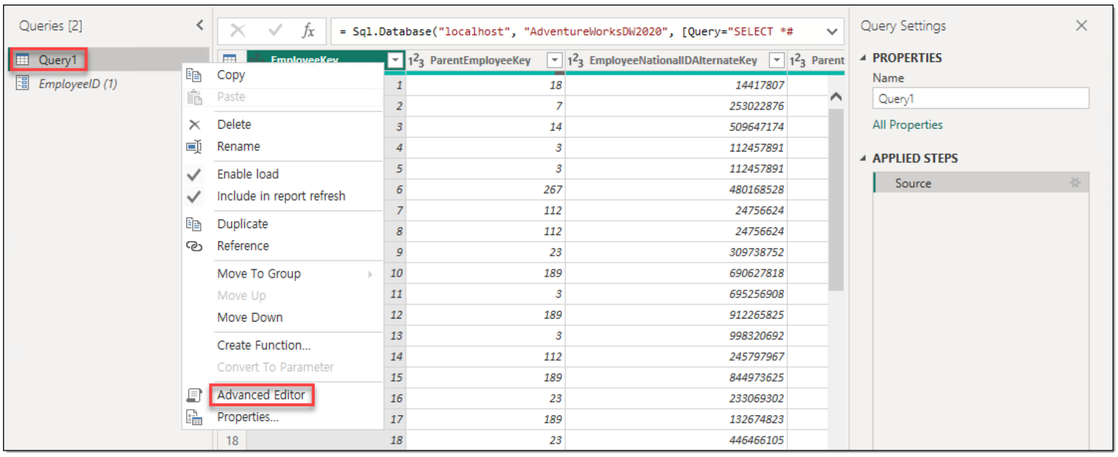
☑ Required

Type: Text

Suggested Values: Any value

Current Value: 1

7. In the [**Queries**] pane, right-click **Query1**, and then select [**Advanced Editor**].



```
= Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT *#
```

	ParentEmployeeKey	EmployeeNationalIDAlternateKey
1	18	14417807
2	7	253022876
3	14	509647174
4	3	112457891
5	3	112457891
6	267	480168528
7	112	24756624
8	112	24756624
9	23	309738752
10	189	690627818
11	3	695256908
12	189	912265825
13	3	998320692
14	112	245797967
15	189	844973625
16	23	233069302
17	189	132674823
18	23	446466105

Query Settings

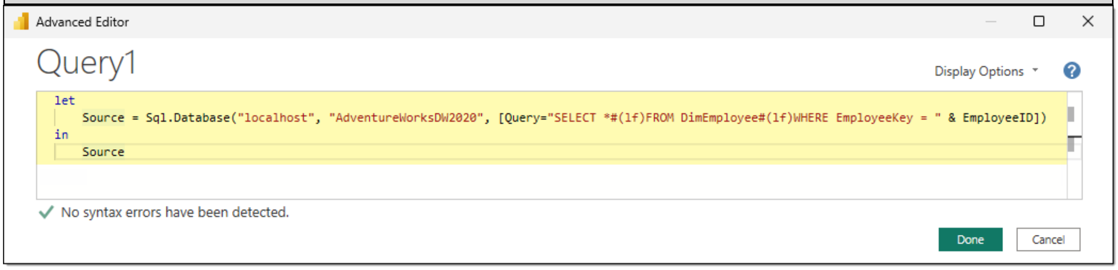
PROPERTIES

Name: Query1

APPLIED STEPS: Source

8. In the [**Advanced Editor**], replace the existing M code with the following, and then select [**Done**].

```
let
   Source = Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT
*#(lf)FROM DimEmployee#(lf)WHERE EmployeeKey = " & EmployeeID])
in
   Source
```



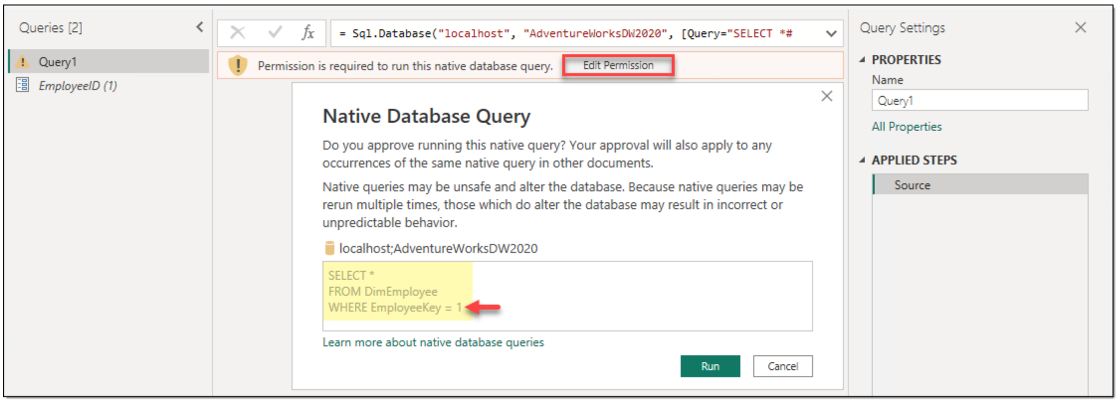
Advanced Editor

Query1

```
let
    Source = Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT *#(lf)FROM DimEmployee#(lf)WHERE EmployeeKey = " & EmployeeID])
in
    Source
```

✓ No syntax errors have been detected.

9. If you see the message "`Permission is required to run this native database query`", select [**Edit Permission**]. In the [**Native Database Query**] dialog, verify that the `WHERE` clause is shown correctly, and then select [**Run**].



= Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT *#

Permission is required to run this native database query. Edit Permission

Native Database Query

Do you approve running this native query? Your approval will also apply to any occurrences of the same native query in other documents.

Native queries may be unsafe and alter the database. Because native queries may be rerun multiple times, those which do alter the database may result in incorrect or unpredictable behavior.

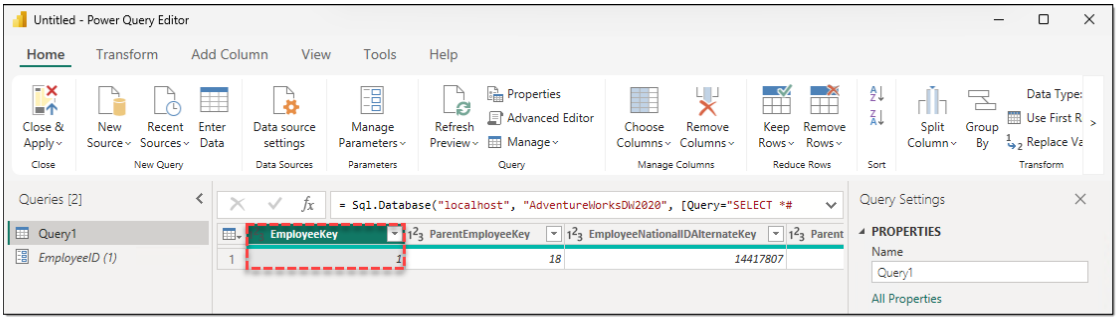
localhost;AdventureWorksDW2020

```
SELECT *
FROM DimEmployee
WHERE EmployeeKey = 1
```

Learn more about native database queries

Run Cancel

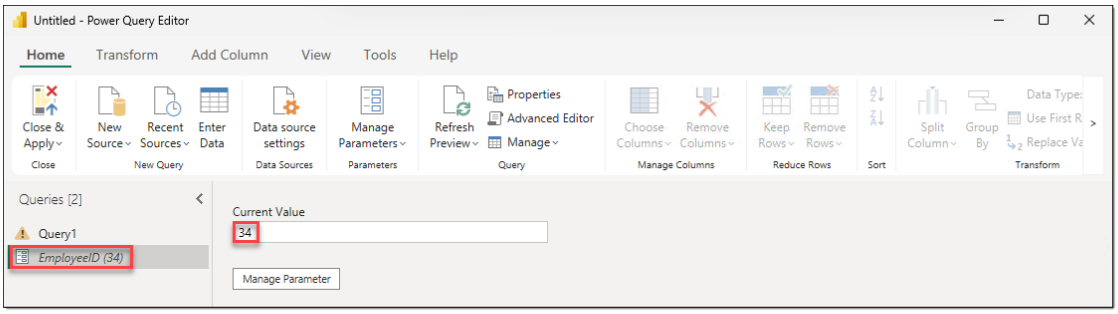
10. In the data preview, confirm that only the row(s) matching the `EmployeeKey` value you entered in the parameter are displayed.



= Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT *#

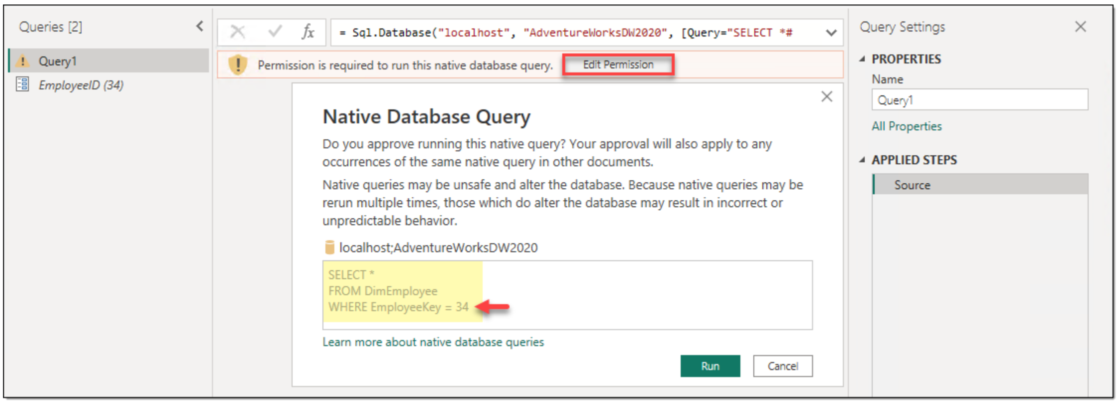
	EmployeeKey	ParentEmployeeKey	EmployeeNationalIDAlternateKey	Parent
1	1	18	14417807	

11. In the [**Queries**] pane, select the `EmployeeID` parameter, and change its value to a different number.



Current Value
34

12. In the [**Queries**] pane, select `Query1`, then select [**Edit Permission**]. In the [**Native Database Query**] dialog, verify that the `WHERE` clause reflects the updated value, and then select [**Run**].



= Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT *#

Permission is required to run this native database query. Edit Permission

Native Database Query

Do you approve running this native query? Your approval will also apply to any occurrences of the same native query in other documents.

Native queries may be unsafe and alter the database. Because native queries may be rerun multiple times, those which do alter the database may result in incorrect or unpredictable behavior.

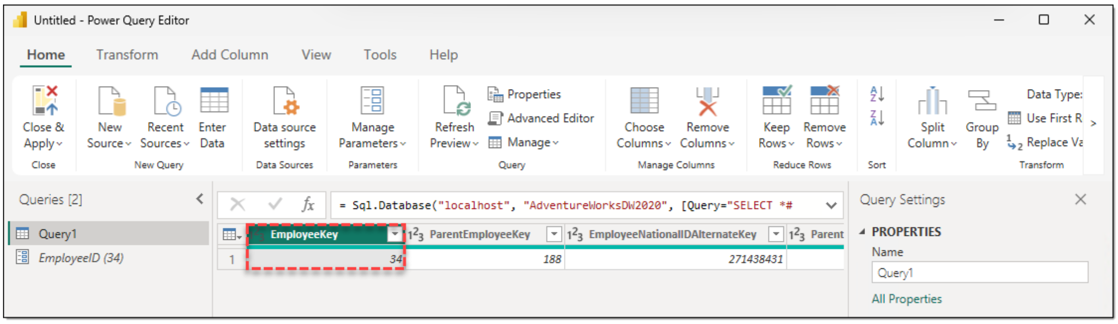
localhost;AdventureWorksDW2020

```
SELECT *
FROM DimEmployee
WHERE EmployeeKey = 34
```

Learn more about native database queries

Run Cancel

13. In the data preview, confirm that only the rows for the selected `EmployeeKey` value are displayed.



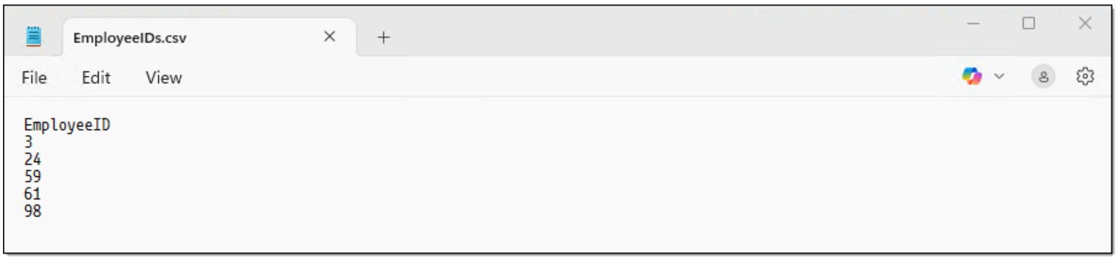
```
= Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT *#
```

	EmployeeKey	ParentEmployeeKey	EmployeeNationalIDAlternateKey	Parent
1	34	188	271438431	

Task 02: Create a dynamic report for multiple values

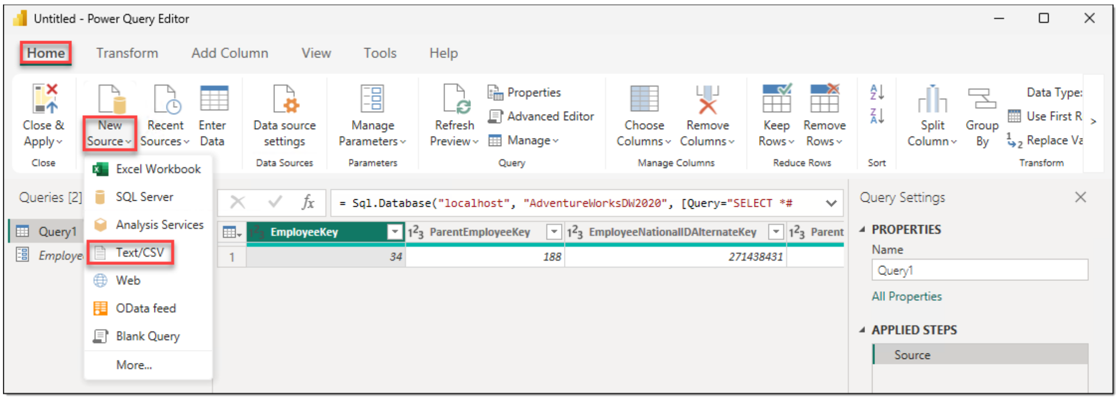
In the previous task, only the data that matched a single EmployeeKey was displayed. In this task, you'll proceed as follows to view data for **multiple** EmployeeKeys.

1. Open **Notepad** and enter the data shown below. Save the file as EmployeeIDs.csv. When saving, set "Save as type" to "All files (*.*)".



```
EmployeeID
3
24
59
61
98
```

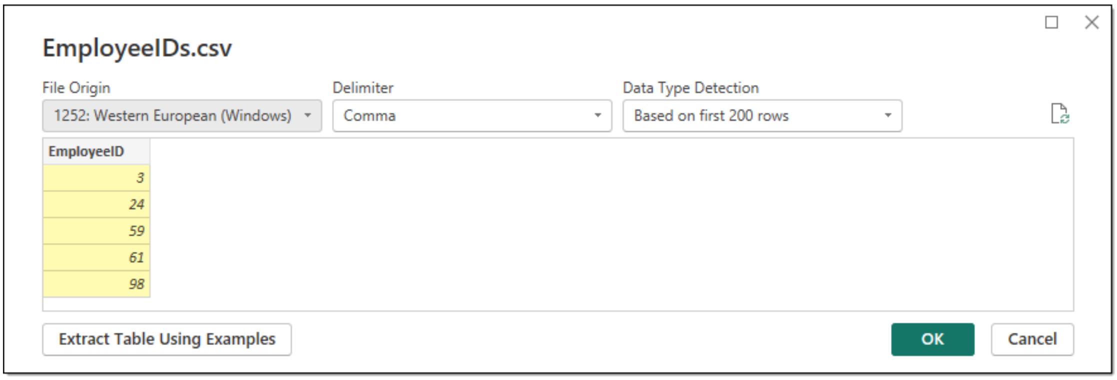
2. In [**Power Query Editor**], on the [**Home**] ribbon, select [**New Query** – **New Source** – **Text/CSV**]. Select the EmployeeIDs.csv file you created earlier, and then select [**Open**].



```
= Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT *#
```

	EmployeeKey	ParentEmployeeKey	EmployeeNationalIDAlternateKey	Parent
1	34	188	271438431	

3. In the [**EmployeeIDs.csv**] dialog, review that the entered data is displayed correctly, and then select [**OK**].



EmployeeIDs.csv

File Origin: 1252: Western European (Windows)

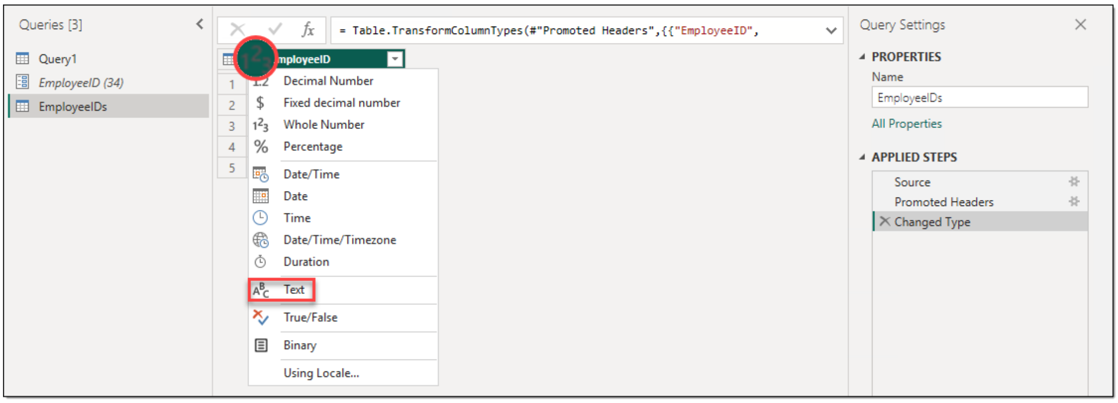
Delimiter: Comma

Data Type Detection: Based on first 200 rows

EmployeeID
3
24
59
61
98

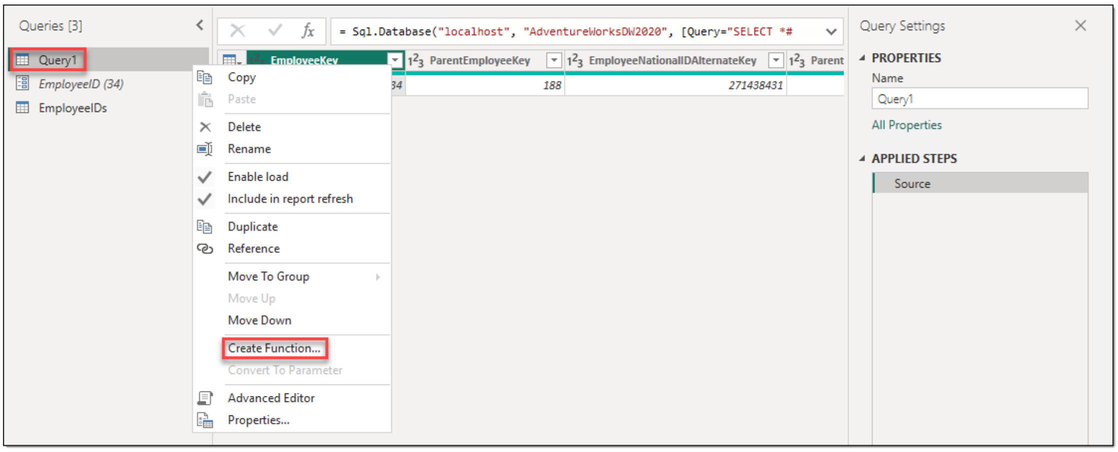
4. In the [**Queries**] pane, select the added EmployeeIDs query. Select 123 to the left of the EmployeeID column

name, and change the `Data type` to `Text`. When the [**Change Column Type**] dialog appears, select [**Replace current**].



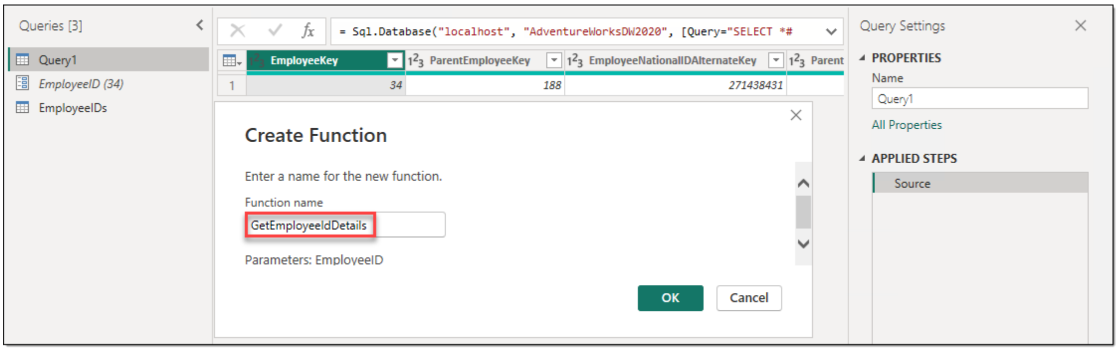
```
= Table.TransformColumnTypes(#"Promoted Headers",{{"EmployeeID",
```

5. Now apply this to the query you imported with the SQL statement. In the [**Queries**] pane, right-click "`Query1`", and then select [**Create Function…**].



```
= Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT *#
```

6. In the [**Create Function**] dialog, set the function name to "`GetEmployeeIdDetails`", and then select [**OK**].



```
= Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT *#
```

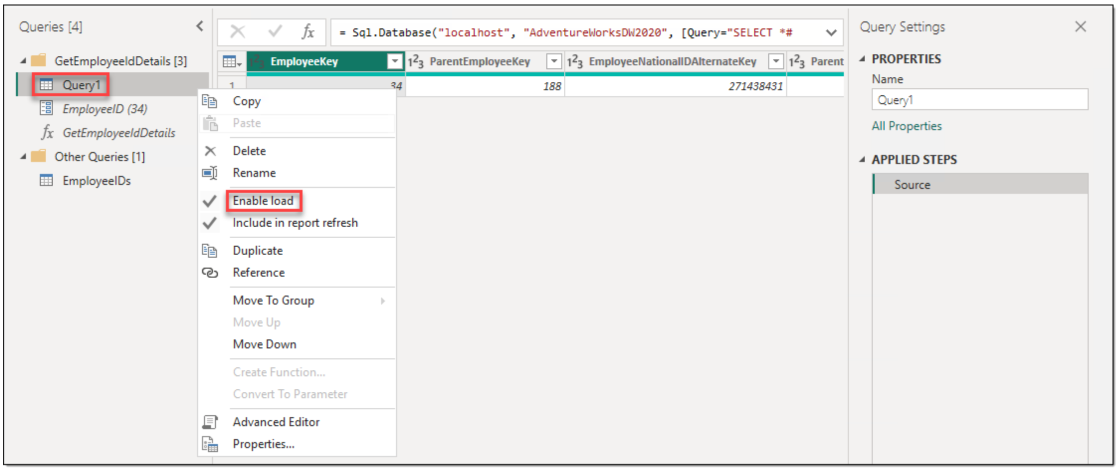
Create Function

Enter a name for the new function.

Function name: GetEmployeeIdDetails

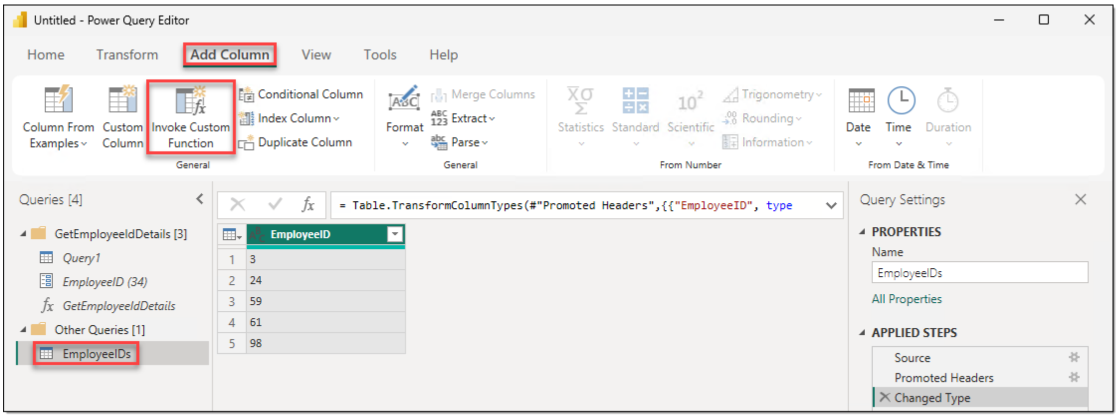
Parameters: EmployeeID

7. Since you no longer need to view the base query table, it's best to disable loading for this query. In the [**Queries**] pane, right-click "`Query1`", and clear [**Enable load**].



```
= Sql.Database("localhost", "AdventureWorksDW2020", [Query="SELECT *#
```

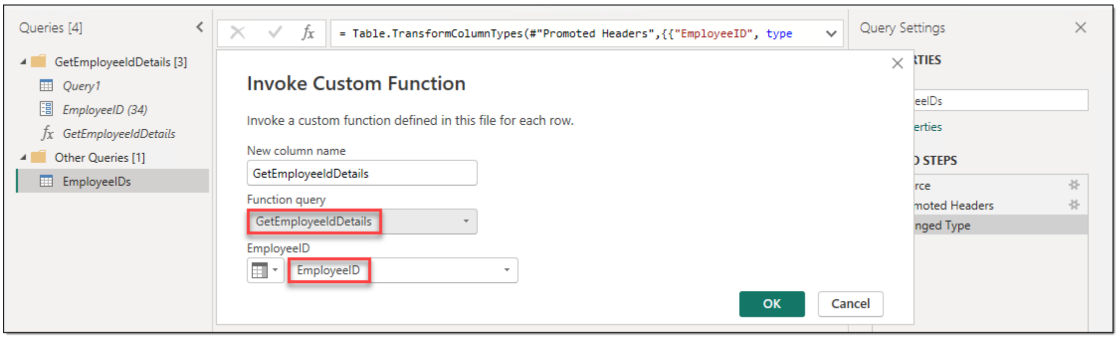
8. To return only the data for the `EmployeeIDs` listed in the CSV, call the function you created. In the [**Queries**] pane, select the "`EmployeeIDs`" query. On the [**Add Column**] ribbon, select [**General – Invoke Custom Function**].



```
= Table.TransformColumnTypes(#"Promoted Headers",{{"EmployeeID", type
```

9. In the [**Invoke Custom Function**] dialog, configure the settings as follows, and then select [**OK**].

- New column name: No need to specify; it will be set automatically when you choose the function query.
- Function query: `GetEmployeeIdDetails`
- EmployeeID: `EmployeeID`



Invoke Custom Function

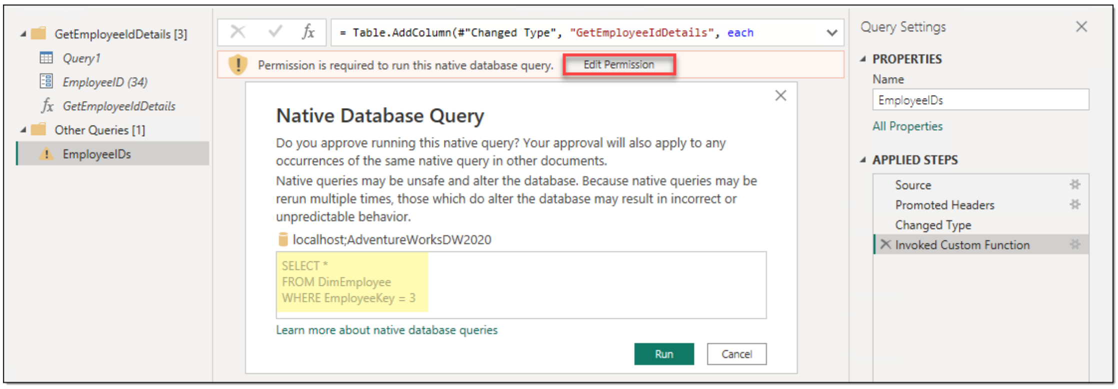
Invoke a custom function defined in this file for each row.

New column name: GetEmployeeIdDetails

Function query: GetEmployeeIdDetails

EmployeeID: EmployeeID

10. In the [**Queries**] pane, select the `EmployeeIDs` query, then select [**Edit Permissions**]. In the [**Native Database Query**] dialog, select [**Run**]. If you see "`Information is required about data privacy.`", select [**Continue**]. In the [**Privacy levels**] dialog, select "`Ignore privacy levels checks for this file`", and then select [**Save**].



```
= Table.AddColumn(#"Changed Type", "GetEmployeeIdDetails", each
```

Permission is required to run this native database query.

Native Database Query

Do you approve running this native query? Your approval will also apply to any occurrences of the same native query in other documents.

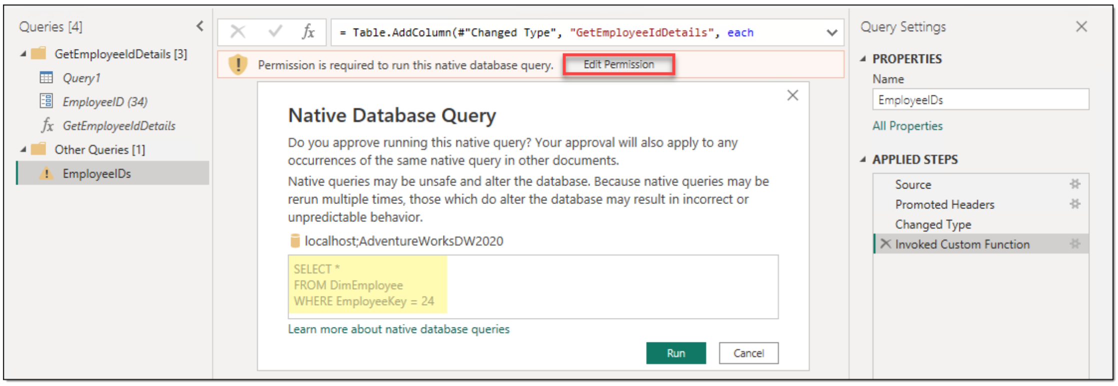
Native queries may be unsafe and alter the database. Because native queries may be rerun multiple times, those which do alter the database may result in incorrect or unpredictable behavior.

localhost;AdventureWorksDW2020

```
SELECT *
FROM DimEmployee
WHERE EmployeeKey = 3
```

Learn more about native database queries

11. The permission prompt appears again. Because you entered **five** EmployeeIDs in the **CSV** file, you'll see this permission message **five times**. Each time, select [**Edit Permission**], and then select [**Run**].



```
= Table.AddColumn(#"Changed Type", "GetEmployeeIdDetails", each
```

Permission is required to run this native database query.

Native Database Query

Do you approve running this native query? Your approval will also apply to any occurrences of the same native query in other documents.

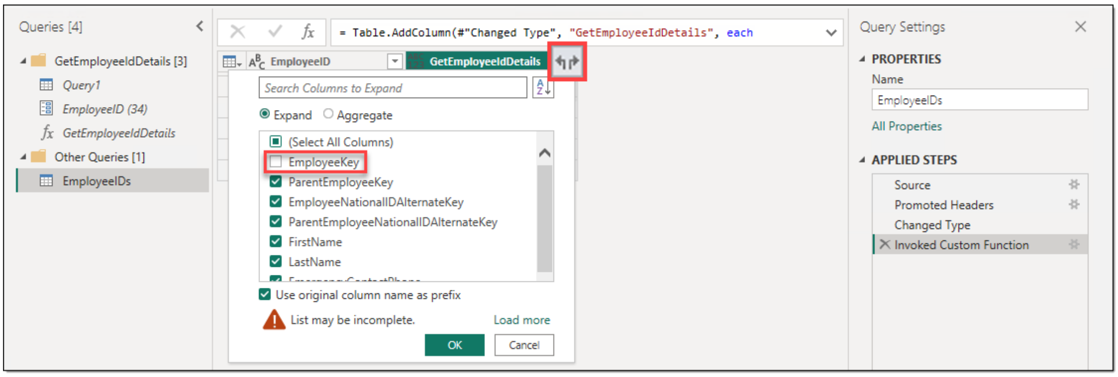
Native queries may be unsafe and alter the database. Because native queries may be rerun multiple times, those which do alter the database may result in incorrect or unpredictable behavior.

localhost;AdventureWorksDW2020

```
SELECT *
FROM DimEmployee
WHERE EmployeeKey = 24
```

Learn more about native database queries

12. In the data preview, you should now see results similar to the screenshot. Select the [**expand**] icon (two arrows) on the `GetEmployeeIdDetails` column. In the expand dialog, **clear** the `EmployeeKey` column (you already have an `EmployeeID` column), and then select [**OK**].



```
= Table.AddColumn(#"Changed Type", "GetEmployeeIdDetails", each
```

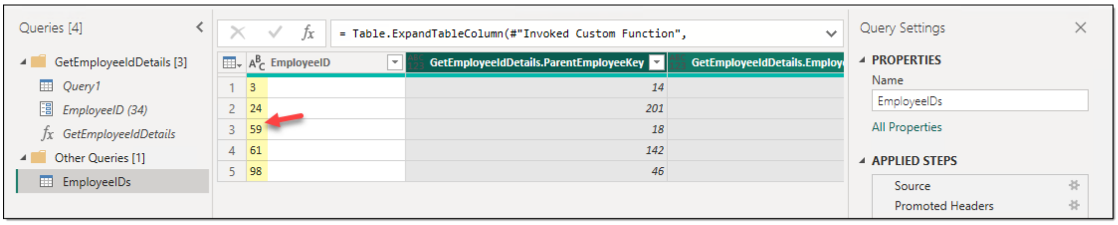
● Expand ○ Aggregate

- ■ (Select All Columns)
- □ EmployeeKey
- ☑ ParentEmployeeKey
- ☑ EmployeeNationalIDAlternateKey
- ☑ ParentEmployeeNationalIDAlternateKey
- ☑ FirstName
- ☑ LastName

☑ Use original column name as prefix

List may be incomplete.

13. Confirm that all rows for the `EmployeeIDs` you entered in the CSV file are now displayed.



```
= Table.ExpandTableColumn(#"Invoked Custom Function",
```

	EmployeeID	GetEmployeeIdDetails.ParentEmployeeKey	GetEmployeeIdDetails.Employ
1	3	14	
2	24	201	
3	59	18	
4	61	142	
5	98	46	