

Microsoft

Microsoft Technical Trainer

Enterprise Skills Initiative

DP-605 LAB 02

Clean, transform, and load data in Power BI

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.

© 2025 Microsoft Corporation. All rights reserved.

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
2024.05.19	0.8.0	Jin Hwan Woo	Load transformed data in Power BI Desktop
2024.05.21	1.0.0	Jin Hwan Woo	Document formatting
2025.01.03	1.1.0	Jin Hwan Woo	Update lab content
2025.08.01	1.2.0	Jin Hwan Woo	Apply Power BI updates
2025.10.26	1.3.0	Jin Hwan Woo	Updated the lab to the English version

Contents

OVERVIEW	5
EXERCISE 01: GET STARTED	5
EXERCISE 02: CONFIGURE THE SALESPERSON QUERY	5
EXERCISE 03: CONFIGURE THE SALESPERSONREGION QUERY	10
EXERCISE 04: CONFIGURE THE PRODUCT QUERY	11
EXERCISE 05: CONFIGURE THE RESELLER QUERY	14
EXERCISE 06: CONFIGURE THE REGION QUERY	17
EXERCISE 07: CONFIGURE THE SALES QUERY	18
EXERCISE 08: CONFIGURE THE TARGETS QUERY	22
EXERCISE 09: CONFIGURE THE COLORFORMATS QUERY	27
EXERCISE 10: UPDATE THE PRODUCT QUERY	27
EXERCISE 11. UPDATE THE COLORFORMATS QUERY	29
EXERCISE 12: CLEAN UP	30

Overview

In this lab, you'll begin shaping the data model by cleaning and transforming the data. You'll then apply the queries and load each one as a table into the model.

You will:

- Apply a variety of transformations.
- Load queries into the data model.

Duration: About 45 minutes.

EXERCISE 01: Get started

In this task, you'll set up the environment for the lab. If you completed the previous lab on the **same** virtual machine, you can skip this task.

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/02-transform-data-power-bi/02-transform-data.zip>

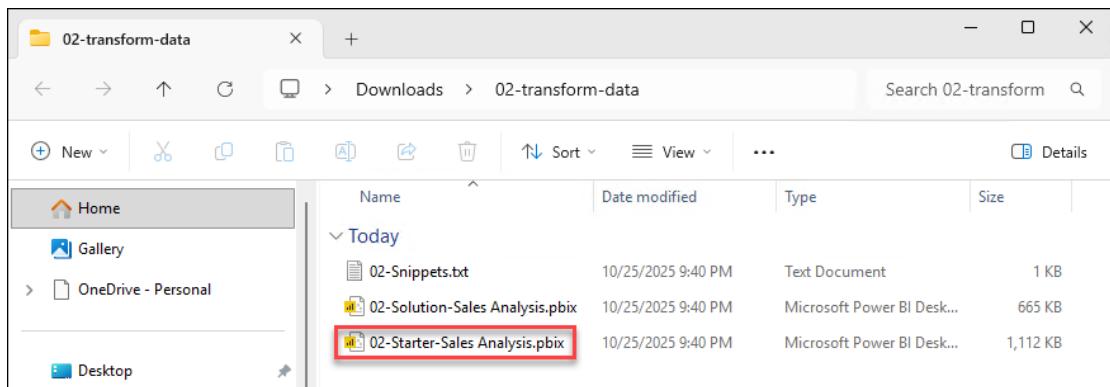
Get started

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

Download Lab Files

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

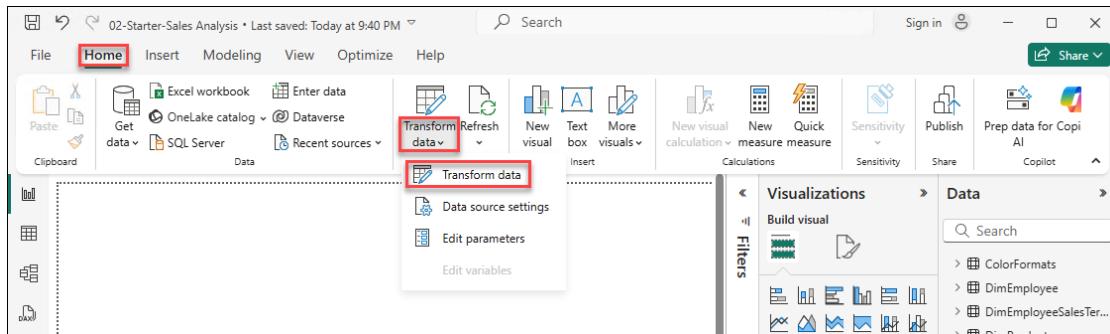
2. Extract the downloaded ZIP file, then open **02-Starter-Sales Analysis.pbix**. If prompted to enter an email address, select [**Cancel**].



EXERCISE 02: Configure the Salesperson query

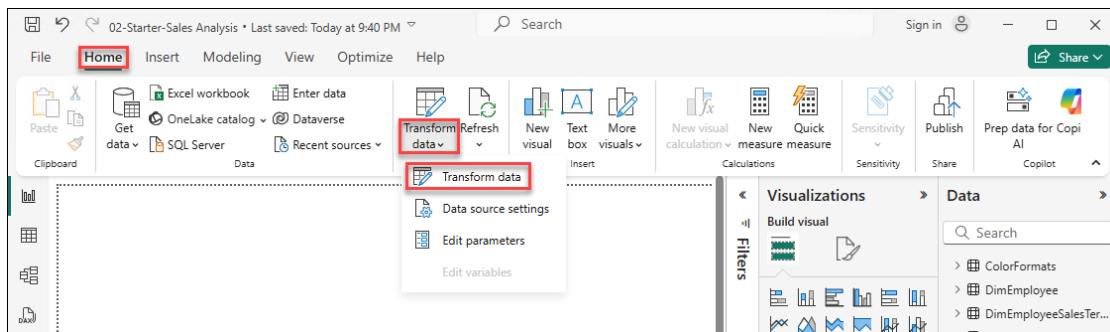
In this task, you'll use **Power Query Editor** to configure the **Salesperson** query. When renaming columns, it's important to follow the guide exactly and use the precise names provided.

1. In [**Power BI Desktop**], on the [**Home**] ribbon, select [**Queries – Transform data – Transform data**].



2. In [Power Query Editor], on the [Home] ribbon, select [Close – Close & Apply – Close & Apply] to apply changes and close the editor. This action initializes the queries and commits connection information and metadata.
- **Important:** In this lab environment, skipping this step can cause errors when you apply changes at the final step.

3. In [Power BI Desktop], on the [Home] ribbon, select [Queries – Transform data – Transform data] again.



4. In [Power Query Editor], select the DimEmployee query in the [Queries] pane. In [Query Settings], rename it to Salesperson and press Enter. Then confirm in the [Queries] pane that the query name has been updated.
- **Note:** The query name determines the model table name. Use concise, user-friendly names.

5. On the [Home] ribbon, select [Manage Columns – Choose Columns – Go to Column]. The [Go to Column] menu is helpful when a table has many columns. Alternatively, you can scroll to the right in the table to find the column.

6. In the [Go to Column] dialog, select the [AZ] sort button, then choose [Name]. Next, select the SalesPersonFlag column and select [OK].

7. In the SalesPersonFlag column, select the [filter] icon and keep only "TRUE" to include salespeople only.

8. In [Query Settings], confirm that a "Filtered Rows" step was added under "APPLIED STEPS".

- Each transformation you create adds a new step in the query logic. You can edit or delete steps as needed.
- You can also select any step to preview the query results at that point in the transformation.

9. To remove unnecessary columns, on the [Home] ribbon select [Manage Columns – Choose Columns – Choose Columns].

The screenshot shows the Power BI Desktop interface. The ribbon at the top has 'Home' selected. In the center, there's a query editor window displaying a table with columns: LeaveHours, CurrentFlag, SalesPersonFlag, and DepartmentName. The 'Choose Columns' button in the ribbon is highlighted with a red box. On the right side, there's a 'Query Settings' pane and a 'PROPERTIES' pane where 'Name' is set to 'Salesperson'.

10. In the [Choose Columns] dialog, clear "(Select All Columns)", then select only the following six columns, and select [OK].

- EmployeeKey
- EmployeeNationalIDAlternateKey
- FirstName
- LastName
- Title
- EmailAddress

This screenshot shows the 'Choose Columns' dialog box. It lists various columns from a table, with checkboxes next to them. The columns 'EmployeeKey', 'EmployeeNationalIDAlternateKey', 'FirstName', 'LastName', 'Title', and 'EmailAddress' have their checkboxes checked, while others like 'MiddleName' and 'Phone' do not. The 'Query Settings' pane on the right shows the step 'Filtered Rows' under 'APPLIED STEPS'.

11. In [Query Settings], confirm that a step named "Removed Other Columns" has been added under "APPLIED STEPS".

This screenshot shows the Power BI query editor. The 'Applied Steps' pane on the right lists steps: 'Source', 'Navigation', 'Filtered Rows', and 'Removed Other Columns'. A red arrow points to the 'Removed Other Columns' step. The main area shows a table with columns: LastName, Title, and EmailAddress, containing data for various employees.

12. To create a single column that combines two columns, select FirstName and LastName (hold <Ctrl> while selecting), then right-click and choose [Merge Columns].
- You can apply many common transformations by right-clicking a column header and choosing an action from the context menu.
 - Even more transformations are available on the ribbon.

The screenshot shows the Power BI Data Editor interface. A context menu is open over the FirstName and LastName columns. The 'Merge Columns' option is highlighted with a red box. Other options visible in the menu include Copy, Remove Columns, Remove Other Columns, Add Column From Examples..., Remove Duplicates, Remove Errors, Replace Values..., Fill, Change Type, Transform, Group By..., Unpivot Columns, Unpivot Other Columns, Unpivot Only Selected Columns, Move, and Merge Columns.

13. In the [Merge Columns] dialog, configure the settings as follows, then select [OK].

- Separator: Space
- New column name (optional): Salesperson

The screenshot shows the 'Merge Columns' dialog box. It has fields for 'Separator' (set to 'Space') and 'New column name (optional)' (set to 'Salesperson'). There are 'OK' and 'Cancel' buttons at the bottom. The background shows the Power BI Data Editor interface with the merged columns displayed.

14. Confirm that the FirstName and LastName columns have been merged into the new Salesperson column.

The screenshot shows the Power BI Data Editor interface with the merged 'Salesperson' column. The 'EmployeeNationalIDAlternateKey' column is also present. The 'Salesperson' column is highlighted with a red box. The 'Merged Columns' step is listed in the Applied Steps section of the ribbon.

15. To rename a column, double-click EmployeeNationalIDAlternateKey, change the name to "EmployeeID", and press Enter.

The screenshot shows the Power BI Data Editor interface with the renamed 'EmployeeID' column. The 'EmployeeNationalIDAlternateKey' column is also present. The 'EmployeeID' column is highlighted with a red box. The 'Merged Columns' step is listed in the Applied Steps section of the ribbon.

16. Using the same method, rename EmailAddress to "UPN". UPN stands for User Principal Name.

The screenshot shows the Power Query Editor interface. In the left pane, there's a list of queries. The main area displays a table with three columns: Salesperson, Title, and UPN. The table has 18 rows of data. The 'UPN' column is currently selected, indicated by a red box around its header.

- In the lower-left status bar of [Power Query Editor], confirm that it shows 5 columns and 18 rows.

This screenshot is similar to the previous one, but a red arrow points to the status bar at the bottom left, which displays the text "5 COLUMNS, 18 ROWS".

EXERCISE 03: Configure the SalespersonRegion query

In this task, you'll configure the SalespersonRegion query.

- In the [Queries] pane, select **DimEmployeeSalesTerritory**. In [Query Settings], rename it to **"SalespersonRegion"**.

The screenshot shows the Power Query Editor with the 'DimEmployeeSalesTerritory' query selected in the Queries pane. In the Query Settings pane, the 'Name' field is highlighted with a red box and contains the value "SalespersonRegion".

- To remove the last two columns, select **DimEmployee** and **DimSalesTerritory** (hold <Ctrl> to multi-select), right-click, and choose [Remove Columns].

The screenshot shows the Power Query Editor with the 'DimEmployeeSalesTerritory' query selected. The 'DimEmployee' and 'DimSalesTerritory' columns are selected and highlighted with red boxes. A context menu is open over these columns, with the 'Remove Columns' option highlighted with a red box.

- In the lower-left status bar of [Power Query Editor], confirm that it shows 2 columns and 39 rows.

Queries [10]

- Salesperson
- SalespersonRegion**
- DimProduct
- DimReseller
- DimSalesTerritory
- FactResellerSales
- ResellerSalesTargets
- ColorFormats
- SQLInstance (localhost)
- Database (AdventureWorksDW2020)

	EmployeeKey	SalesTerritoryKey
1	272	1
2	272	2
3	272	3
4	272	4
5	272	5
6	272	6
7	277	1
8	277	2
9	277	3
10	277	4
11	277	5

2 COLUMNS, 39 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 9:45 PM

EXERCISE 04: Configure the Product query

In this task, you'll configure the Product query.

1. In the [Queries] pane, select DimProduct. In [Query Settings], rename it to "Product".

Queries [10]

- Salesperson
- SalespersonRegion
- DimProduct**
- DimReseller
- DimSalesTerritory
- FactResellerSales
- ResellerSalesTargets
- ColorFormats
- SQLInstance (localhost)
- Database (AdventureWorksDW2020)

	ProductKey	ProductAlternateKey	ProductSubcategoryKey	WeightUnit
1	AR-5381			null
2	BA-8327			null
3	BE-2349			null
4	BE-2908			null
5	BL-2036			null
6	CA-5965			null
7	CA-6738			null
8	CA-7457			null
9	CB-2903			null
10	CN-6137			null
11	CR-7833			null

2. Go to the FinishedGoodsFlag column, filter to keep only "TRUE" (finished goods), and select [OK].

Queries [10]

- Salesperson
- SalespersonRegion
- Product**
- DimReseller
- DimSalesTerritory
- FactResellerSales
- ResellerSalesTargets
- ColorFormats
- SQLInstance (localhost)
- Database (AdventureWorksDW2020)

	ProductName	StandardCost	FinishedGoodsFlag	Color
1	ABACUS	868.63	TRUE	Black
2	APACHE	868.63	TRUE	Red
3	SPORT 100, rouge	12.03	TRUE	Red

3. On the [Home] ribbon, select [Manage Columns – Choose Columns – Choose Columns].

02-Starter-Sales Analysis

Home Transform Add Column View Tools Help

Choose Columns

Queries [10]

- Salesperson
- SalespersonRegion
- Product**
- DimReseller

	ProductName	StandardCost	FinishedGoodsFlag	Color
1	ABACUS	868.63	TRUE	Black
2	APACHE	868.63	TRUE	Red
3	SPORT 100, rouge	12.03	TRUE	Red

4. In the [Choose Columns] dialog, select only the columns listed below, and then select [OK].

- **ProductKey**
- **EnglishProductName**

- StandardCost
- Color
- DimProductSubcategory

The screenshot shows the Power Query Editor interface. On the left, the 'Queries [10]' pane lists various tables like Salesperson, SalespersonRegion, Product, etc. The 'Product' query is currently selected. In the center, a 'Choose Columns' dialog is open over the query grid. The grid contains columns: 'Name', 'StandardCost', 'Color', and 'DimProductSubcategory'. A red dashed box highlights the 'DimProductSubcategory' column. On the right, the 'Query Settings' pane shows the 'APPLIED STEPS' section with 'Filtered Rows' selected.

5. The **DimProductSubcategory** column contains a **Value** link, which indicates a related table.

This screenshot shows the 'DimProductSubcategory' column expanded. The expanded view shows a table with columns: 'Name', 'StandardCost', 'Color', and 'DimProductSubcategory'. The 'DimProductSubcategory' column now contains a table of values. A red dashed box highlights this expanded view. The 'Query Settings' pane on the right shows 'Removed Other Columns' as the applied step.

6. Select the [**Expand**] button (two arrows) on the **DimProductSubcategory** column header. Configure as follows, then select [**OK**]:

- Columns to select: EnglishProductSubcategoryName, DimProductCategory
- Use original column name as prefix: Clear this option.

This screenshot shows the 'DimProductSubcategory' column expanded with the 'Expand' dialog open. The 'Search Columns to Expand' field is empty. The 'Columns to select' dropdown is set to 'EnglishProductSubcategoryName, DimProductCategory'. The 'Use original column name as prefix' checkbox is unchecked. The 'OK' button is highlighted. The 'Query Settings' pane on the right shows 'Removed Other Columns' as the applied step.

7. Confirm that two columns were added and the **DimProductSubcategory** column was removed as a result of the transformation.

- By selecting the two columns above, Power Query applied a join to the **DimProductSubcategory** table and included the selected fields. **DimProductSubcategory** is another related table in the data source.
- Column names in a query must always be unique. If you select "Use original column name as

`prefix`", the expanded column names are prefixed with the source table name (in this case, `DimProductSubcategory`). In this lab, that option was left **cleared** because the selected column names do not conflict with other columns in the `Product` query.

The screenshot shows the Power BI Data Editor interface. On the left, the 'Queries [10]' pane lists various tables like Salesperson, SalespersonRegion, Product, etc. The main area displays a table with columns: Color, EnglishProductName, EnglishProductSubcategoryName, and DimProductCategory. The DimProductCategory column contains values like 'Road Frames', 'Helmets', 'Socks', and 'Helmets'. A red dashed box highlights the DimProductCategory column. On the right, the 'Query Settings' pane shows the 'APPLIED STEPS' section with 'Expanded DimProductSub...' listed.

- Select the [Expand] icon on the `DimProductCategory` column, select `EnglishProductCategoryName` only, and then select [OK].

This screenshot shows the 'Search Columns to Expand' dialog box. It lists several options under 'Select All Columns': ProductCategoryKey, EnglishProductCategoryName (which is checked), ProductCategoryAlternateKey, SpanishProductCategoryName, FrenchProductCategoryName, and DimProductSubcategory. There is also a checkbox for 'Use original column name as prefix'. The 'OK' button is highlighted with a red box.

- Rename the following four columns as shown

Current column name	New column name
EnglishProductName	Product
StandardCost	Standard Cost
EnglishProductSubcategoryName	Subcategory
EnglishProductCategoryName	Category

This screenshot shows the Power BI Data Editor after renaming. The table now has four new columns: Product, Standard Cost, Color, and Subcategory. The original columns have been removed. A red box highlights the 'Renamed Columns' step in the 'APPLIED STEPS' section of the 'Query Settings' pane.

- In the lower-left status bar, confirm that it shows 6 columns and 397 rows.

This screenshot shows the Power BI Data Editor with the status bar at the bottom left indicating '6 COLUMNS, 397 ROWS'. A red arrow points to this status bar. The 'Query Settings' pane on the right shows the 'Renamed Columns' step in the 'APPLIED STEPS' section.

- On the [Home] ribbon, select [Close & Apply – Apply].

- Important:** If you want to apply all query transformations at the very end of the lab, some changes may not apply correctly. It's best to apply after completing transformations in each query.

EXERCISE 05: Configure the Reseller query

In this task, you'll configure the Reseller query.

1. In the [Queries] pane, select DimReseller. In [Query Settings], rename it to "Reseller".

2. On the [Home] ribbon, select [Manage Columns – Choose Columns – Choose Columns].

3. In the [Choose Columns] dialog, select the following columns, and then select [OK].

- ResellerKey
- BusinessType
- ResellerName
- DimGeography

The screenshot shows the 'Choose Columns' dialog for the 'Reseller' table. The 'DimReseller' step is selected in the navigation pane. The 'OK' button is highlighted.

4. Select the [Expand] icon to the right of the **DimGeography** column, select the following columns, and then select [OK].

- **City**
- **StateProvinceName**
- **EnglishCountryRegionName**
- Do not select the "Use original column name as prefix".

The screenshot shows the 'Table.SelectColumns' dialog for the 'Reseller' table. The 'DimGeography' column is expanded. The 'OK' button is highlighted.

5. Select the dropdown for the **BusinessType** column and review the distinct values. This column contains both "Ware House" and "Warehouse". Select [Cancel].

The screenshot shows the 'Table.ExpandRecordColumn' dialog for the 'Reseller' table. The 'BusinessType' column dropdown is open, showing 'Ware House' and 'Warehouse'. The 'OK' button is highlighted.

6. Right-click the **BusinessType** column, and then select [Replace Values...].

The screenshot shows the Power BI Data Editor interface. On the left, the 'Queries [10]' pane lists various tables like Salesperson, SalespersonRegion, Product, and Reseller. The 'Reseller' table is selected. In the main area, the 'BusinessType' column is highlighted with a red box. A context menu is open over this column, with the 'Replace Values...' option highlighted by another red box. The 'APPLIED STEPS' pane on the right shows a step named 'Expanded DimGeography'.

7. In the [Replace Values] dialog, configure as follows, and then select [OK].

- Value To Find: Ware House
- Replace With: Warehouse

The screenshot shows the 'Replace Values' dialog box. It has two input fields: 'Value To Find' containing 'Ware House' and 'Replace With' containing 'Warehouse', both of which are highlighted with red boxes. There is also an 'Advanced options' button. The background shows the Power BI Data Editor with the Reseller table and its columns.

8. Open the dropdown for the BusinessType column again and confirm that "Ware House" no longer appears. Select [Cancel].

The screenshot shows the Power BI Data Editor with the Reseller table. The 'BusinessType' column dropdown menu is open, displaying a list of values. An arrow points to the 'Warehouse' entry, which is highlighted with a red box. The 'APPLIED STEPS' pane on the right shows a step named 'Replaced Value'.

9. Rename the columns as follows. After renaming, verify in the lower-left status bar that there are **6 columns** and **701 rows**.

Current column name	New column name
BusinessType	Business Type
ResellerName	Reseller
StateProvinceName	State-Province
EnglishCountryRegionName	Country-Region

6 COLUMNS, 701 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 8:40 AM

EXERCISE 06: Configure the Region query

In this task, you'll configure the Region query.

1. In the [Queries] pane, select DimSalesTerritory. In [Query Settings], rename it to "Region".

2. Open the dropdown for SalesTerritoryAlternateKey, clear the value 0, and select [OK].

- Note: The value 0 represents headquarters and shouldn't be included as a region.

3. On the [Home] ribbon, select [Manage Columns – Choose Columns – Choose Columns].

4. In the [Choose Columns] dialog, select **only** the following columns, and then select [OK].

- **SalesTerritoryKey**
- **SalesTerritoryRegion**
- **SalesTerritoryCountry**
- **SalesTerritoryGroup**

5. Rename the columns as follows. Then verify in the lower-left status bar that there are **4 columns** and **10 rows**.

Current column name	New column name
SalesTerritoryRegion	Region
SalesTerritoryCountry	Country
SalesTerritoryGroup	Group

EXERCISE 07: Configure the Sales query

In this task, you'll configure the Sales query.

1. In the [Queries] pane, select FactResellerSales. In [Query Settings], rename it to "Sales".

SalesOrderNumber	SalesOrderLineNumber	OrderDate	DueDate
SO43659	1	7/1/2017	7/1/2017
SO43659	2	7/1/2017	7/1/2017
SO43659	3	7/1/2017	7/1/2017
SO43659	4	7/1/2017	7/1/2017
SO43659	5	7/1/2017	7/1/2017
SO43659	6	7/1/2017	7/1/2017
SO43659	7	7/1/2017	7/1/2017
SO43659	8	7/1/2017	7/1/2017
SO43659	9	7/1/2017	7/1/2017
SO43659	10	7/1/2017	7/1/2017
SO43659	11	7/1/2017	7/1/2017

2. On the [Home] ribbon, select [Manage Columns – Choose Columns – Choose Columns].

SalesOrderNumber	SalesOrderLineNumber	OrderDate	DueDate
SO43659	1	7/1/2017	7/1/2017
SO43659	2	7/1/2017	7/1/2017
SO43659	3	7/1/2017	7/1/2017

3. In the [Choose Columns] dialog, select only the following columns, and then select [OK].

- SalesOrderNumber
- OrderDate
- ProductKey
- ResellerKey
- EmployeeKey
- SalesTerritoryKey
- OrderQuantity
- UnitPrice
- TotalProductCost
- SalesAmount
- DimProduct

Search Columns
<input type="checkbox"/> (Select All Columns)
<input checked="" type="checkbox"/> SalesOrderNumber
<input type="checkbox"/> SalesOrderLineNumber
<input checked="" type="checkbox"/> OrderDate
<input type="checkbox"/> DueDate
<input type="checkbox"/> ShipDate
<input checked="" type="checkbox"/> ProductKey
<input checked="" type="checkbox"/> ResellerKey
<input type="checkbox"/> PromotionKey
<input checked="" type="checkbox"/> EmployeeKey
<input checked="" type="checkbox"/> SalesTerritoryKey
<input checked="" type="checkbox"/> OrderQuantity
<input checked="" type="checkbox"/> UnitPrice
<input checked="" type="checkbox"/> TotalProductCost
<input checked="" type="checkbox"/> SalesAmount
<input type="checkbox"/> DimDate(OrderDate)
<input type="checkbox"/> DimDate(ShipDate)
<input type="checkbox"/> DimEmployee
<input checked="" type="checkbox"/> DimProduct
<input type="checkbox"/> DimPromotion

4. In a previous data preparation lab, you observed that some FactResellerSales rows have missing TotalProductCost values. To help address this, include the DimProduct column so you can reference the product's StandardCost. Select the [Expand] icon on DimProduct, select StandardCost only, and then select [OK].

The screenshot shows the Power BI Query Editor interface. On the left, there's a list of queries and tables. In the center, a table view shows rows of data. On the right, the 'Query Settings' pane is open, showing properties like 'Name' (set to 'Sales') and applied steps ('Removed Other Columns'). A 'Custom Column' dialog is overlaid, with a red box highlighting the 'Custom Column' button on the ribbon. Inside the dialog, a list of columns from the 'DimProduct' table is shown, with the 'StandardCost' checkbox checked. At the bottom right of the dialog is an 'OK' button.

5. To add a new column, on the [Add Column] ribbon select [General – Custom Column].

This screenshot shows the Power BI Query Editor with the 'Add Column' dialog open. A red box highlights the 'Custom Column' button on the ribbon. The 'Custom Column' dialog is displayed, listing columns from the 'DimProduct' table. The 'StandardCost' checkbox is checked. The 'OK' button is located at the bottom right of the dialog.

6. In the [Custom Column] dialog, configure as follows, then select [OK].

- New column name: **Cost**
- Custom column formula: This expression checks whether **TotalProductCost** is missing. If the value is **null**, it returns **OrderQuantity × StandardCost**; if it isn't **null**, it returns **TotalProductCost**.
- You can run the following prompt in Copilot Chat to verify these M query expressions: "Write an M query for use in Power BI's Custom Column that meets the following conditions. – If the TotalProductCost is null, it returns OrderQuantity × StandardCost; if it isn't null, it returns TotalProductCost."

```
if [TotalProductCost] = null then [OrderQuantity] * [StandardCost] else [TotalProductCost]
```

This screenshot shows the 'Custom Column' dialog in the Power BI Query Editor. A red box highlights the 'Cost' input field. The 'Custom column formula' field contains the M code: 'if [TotalProductCost] = null then [OrderQuantity] * [StandardCost] else [TotalProductCost]'. The 'Available columns' list on the right includes 'SalesOrderNumber', 'OrderDate', 'ProductKey', 'ResellerKey', 'EmployeeKey', 'SalesTerritoryKey', 'OrderQuantity', and 'Product'. At the bottom right of the dialog is an 'OK' button.

7. With **<Ctrl>** pressed, select the **TotalProductCost** and **StandardCost** columns, right-click, and choose **[Remove Columns]**.

The screenshot shows the Power BI Data Editor interface with the 'Sales' table selected. A context menu is open over the 'StandardCost' column, with the 'Remove Columns' option highlighted.

8. Rename the columns as follows.

Current column name	New column name
OrderQuantity	Quantity
UnitPrice	Unit Price
SalesAmount	Sales

The screenshot shows the Power BI Data Editor interface with the 'Sales' table selected. The 'Quantity' column is highlighted. The 'Applied Steps' pane shows the 'Renamed Columns' step.

9. To change the data type, select the [1.2] icon to the left of the **Quantity** column and choose **[Whole Number]**.

- Configuring the correct data type is important.
- If a column contains numeric values, selecting the proper type is also essential for performing mathematical calculations.

The screenshot shows the Power BI Data Editor interface with the 'Quantity' column selected. The data type dropdown shows 'Whole Number' selected, highlighted with a red box.

10. Using the same method, change the **datatype** of the following **three** columns to "**Fixed decimal number**".

The "Fixed decimal number" type supports 19 digits and allows higher precision to avoid rounding errors.

It's important to use this type for monetary values or exchange rates.

- Unit Price**
- Sales**
- Cost**

Queries [10]

Salesperson
SalespersonRegion
Product
Reseller
Region
Sales
ResellerSalesTargets
ColorFormats
SQLInstance (localhost)
Database (AdventureWorksDW2020)

Unit Price | Sales | Cost

	Unit Price	Sales	Cost
1	2,024.99	2,024.99	1,898.09
2	2,024.99	6,074.97	5,694.27
3	2,024.99	2,024.99	1,898.09
4	2,039.99	2,039.99	1,912.15
5	2,039.99	2,039.99	1,912.15
6	2,039.99	4,079.98	3,824.30
7	2,039.99	2,039.99	1,912.15
8	28.84	86.52	95.16
9	28.84	28.84	31.72
10	5.70	34.20	20.40
11	5.19	10.38	11.42
12	20.19	80.76	48.12
13	419.46	419.46	413.15

Query Settings

Properties

Name: Sales

Applied Steps

- Source
- Navigation
- Removed Other Columns
- Expanded DimProduct
- Added Custom
- Removed Columns
- Renamed Columns
- Changed Type

11. In the lower-left status bar, confirm that it shows **10 columns** and **999+ rows**. Each query loads up to **1,000** rows as preview data. Then, on the [Home] ribbon, select [Close & Apply – Apply] to apply your changes.

02-Starter-Sales Analysis

Home Transform Add Column View Tools Help

New Source Recent Sources Enter Data Data source settings Manage Parameters Refresh Advanced Editor Choose Columns Remove Rows Keep Rows Remove Rows Reduce Rows Sort Split Column Group By Use First Row Replace Value Transform

New Query Data Sources Parameters Query Manage Columns Manage Rows

SalespersonRegion Product Reseller Region Sales ResellerSalesTargets ColorFormats SQLInstance (localhost) Database (AdventureWorksDW2020)

A₁ SalesOrderNumber B₁ OrderDate C₁ ProductKey D₁ ResellerKey

SalesOrderNumber	OrderDate	ProductKey	ResellerKey
SO43659	7/1/2017	349	
SO43659	7/1/2017	350	
SO43659	7/1/2017	351	
SO43659	7/1/2017	344	
SO43659	7/1/2017	345	
SO43659	7/1/2017	346	
SO43659	7/1/2017	347	
SO43659	7/1/2017	229	
SO43659	7/1/2017	235	
SO43659	7/1/2017	218	

Query Settings

Properties

Name: Sales

Applied Steps

- Source
- Navigation
- Removed Other Columns
- Expanded DimProduct
- Added Custom

10 COLUMNS, 999+ ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED AT 9:10 AM

EXERCISE 08: Configure the Targets query

In this task, you'll configure the Targets query.

1. In the [Queries] pane, select **ResellerSalesTargets**. Select [Edit Credentials]. In the [Access Web content] dialog, choose [Anonymous], and then select [Connect].

Queries [10]

Salesperson
SalespersonRegion
Product
Reseller
Region
Sales
ResellerSalesTargets
ColorFormats
SQLInstance (localhost)
Database (AdventureWorksDW2020)

Please specify how to connect. **Edit Credentials**

Access Web content

Anonymous

Windows

Basic

Web API

Organizational account

https://raw.githubusercontent.com/MicrosoftLearn...

Select which level to apply these settings to
https://raw.githubusercontent.com/

Connect Cancel

2. In [Query Settings], rename it to "Targets".

Queries [10]

- Salesperson
- SalespersonRegion
- Product
- Reseller
- Region
- Sales
- Targets**
- ColorFormats
- SQLInstance (localhost)
- Database (AdventureWorksDW2020)

Query Settings

- Properties: Name (Targets)
- Applied Steps: Promoted Headers, Changed Type

3. To unpivot the 12 monthly columns (labeled M01, M02, ...), select the Year and EmployeeID columns, right-click, and choose [Unpivot Other Columns].

Queries [10]

- Salesperson
- SalespersonRegion
- Product
- Reseller
- Region
- Sales
- Targets**
- ColorFormats
- SQLInstance (localhost)
- Database (AdventureWorksDW2020)

Query Settings

- Properties: Name (Targets)
- Applied Steps: Promoted Headers, Changed Type

4. After unpivoting, confirm that the original column names (M01, M02, ...) appear in the "Attribute" column and the values appear in the "Value" column.

Queries [10]

- Salesperson
- SalespersonRegion
- Product
- Reseller
- Region
- Sales
- Targets**
- ColorFormats
- SQLInstance (localhost)
- Database (AdventureWorksDW2020)

Query Settings

- Properties: Name (Targets)
- Applied Steps: Promoted Headers, Changed Type, Unpivoted Other Columns

5. In the "Value" column, open the dropdown, filter out the "-" values, and select [OK]. In the previous lab, the source CSV used a hyphen (-) to represent a value of 0.

Queries [10]

- Salesperson
- SalespersonRegion
- Product
- Reseller
- Region
- Sales
- Targets**
- ColorFormats
- SQLInstance (localhost)
- Database (AdventureWorksDW2020)

Query Settings

- Properties: Name (Targets)
- Applied Steps: Promoted Headers, Changed Type, Unpivoted Other Columns

6. Rename the columns as follows:

Current column name	New column name
Attribute	MonthNumber
Value	Target

The screenshot shows the Power BI desktop interface with the 'Targets' query selected. In the main editor area, the 'MonthNumber' column is highlighted with a red box. The 'APPLIED STEPS' pane on the right shows the 'Renamed Columns' step has been applied.

7. Right-click the MonthNumber column, and then select [Replace Values...] to prepare the values.

The screenshot shows the 'Targets' query in the Power BI editor. A context menu is open over the 'MonthNumber' column, with the 'Replace Values...' option highlighted with a red box. The 'APPLIED STEPS' pane shows the 'Renamed Columns' step.

8. In the [Replace Values] dialog, enter "M" in "Value To Find", leave "Replace With" blank, and select [OK]. This removes the character "M" from the values.

The screenshot shows the 'Replace Values' dialog box. The 'Value To Find' field contains 'M' with a red box around it. The 'Replace With' field is empty with a red box around it. The 'OK' button is visible at the bottom right.

9. Select the icon to the left of the MonthNumber column and change the data type to "Whole Number".

The screenshot shows the 'Targets' query in the Power BI editor. The 'MonthNumber' column's data type dropdown is open, showing various options like Decimal Number, Fixed decimal number, Whole Number, Percentage, Date, Time, Duration, Text, True/False, Binary, and Using Locale... The 'Whole Number' option is highlighted with a red box.

10. Now create a Date column by applying a transformation derived from the Year and MonthNumber columns.

Go to the [Add Column] ribbon, then select [General – Column From Examples – From All Columns].

11. In the first grid cell of "Column1", enter "7/1/2017" and press Enter. For this first row, Year = 2017 and MonthNumber = 7.

12. Confirm that the grid cells update with the expected values. This feature combines the values from the Year and MonthNumber columns. Also review the formula shown in the query grid. Select [OK].

13. Rename the "Merged" column to "TargetMonth".

14. Select the Year and MonthNumber columns, right-click, and choose [Remove Columns].

Queries [10]

	Year	EmployeeID	MonthNumber
1	2017	61161660	
2	2017	61161660	
3	2017	61161660	
4	2017	61161660	
5	2017	61161660	
6	2017	61161660	
7	2017	90836195	
8	2017	90836195	
9	2017	90836195	
10	2017	90836195	
11	2017	90836195	

Query Settings

PROPERTIES

- Remove Columns
- Remove Other Columns
- Add Column From Examples...
- Remove Duplicates
- Remove Errors
- Replace Values...
- Fill
- Change Type
- Transform

15. Change the **Target** column's data type to "Fixed decimal number", and change the **TargetMonth** column's data type to "Date".

Queries [10]

	EmployeeID	Target	TargetMonth
1	61161660	200.00	
2	61161660	400.00	
3	61161660	600.00	
4	61161660	400.00	
5	61161660	800.00	
6	61161660	800.00	
7	90836195	100.00	
8	90836195	200.00	
9	90836195	300.00	
10	90836195	400.00	
11	90836195	400.00	
12	90836195	500.00	
13	112432117	500.00	
14	112432117	1,500.00	
15	112432117	1,000.00	
16	112432117	1,000.00	10/1/2017
17	112432117	2,200.00	11/1/2017

Query Settings

PROPERTIES

- Name: Targets
- All Properties

APPLIED STEPS

- Source
- Promoted Headers
- Changed Type
- Unpivoted Other Columns
- Filtered Rows
- Renamed Columns
- Replaced Value
- Changed Type1
- Inserted Merged Column
- Renamed Columns1
- Removed Columns
- Changed Type2

16. Select the **Target** column. Then, on the **[Transform]** ribbon, select **[Number Column – Standard – Multiply]**.
- Note:** In the previous lab, **Target** values were stored in thousands, so you'll multiply by 1,000 here.

02-Starter-Sales Analysis

Home **Transform** Add Column View Tools Help

Queries [10]

	EmployeeID	Target	TargetMonth
1	61161660	200.00	
2	61161660	400.00	
3	61161660	600.00	
4	61161660	400.00	
5	61161660	800.00	
6	61161660	800.00	
7	90836195	100.00	
8	90836195	200.00	

Query Settings

PROPERTIES

- Name: Targets
- All Properties

APPLIED STEPS

- Source
- Promoted Headers
- Changed Type

17. In the **[Multiply]** dialog, enter **1000**, and then select **[OK]**.

Queries [10]

	EmployeeID	Target	TargetMonth
1	61161660	200.00	7/1/2017
2	61161660	400.00	8/1/2017

Multiply

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

Value: **1000**

OK Cancel

Query Settings

PROPERTIES

- Name: Targets

18. Confirm in the lower-left status bar that there are **3 columns** and **809 rows**. Then, on the **[Home]** ribbon, select **[Close & Apply – Apply]**.

EXERCISE 09: Configure the ColorFormats query

In this task, you'll configure the **ColorFormats** query.

1. In the [Queries] pane, select **ColorFormats**. Because the first row contains the column names, on the [Home] ribbon select [Transform – Use First Row as Headers – Use First Row as Headers].

2. Confirm in the lower-left status bar that there are **3 columns** and **10 rows**.

EXERCISE 10: Update the Product query

In this task, you'll merge the **Product** and **ColorFormats** queries.

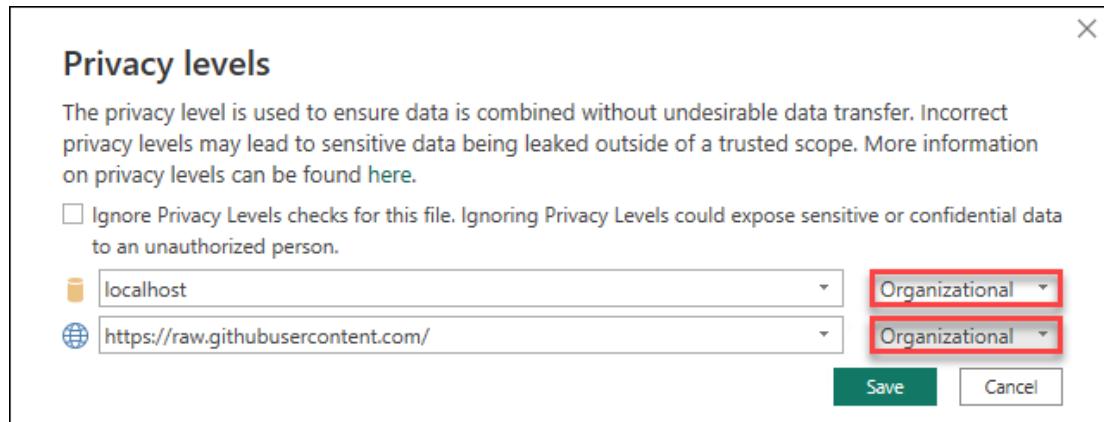
- In the [Queries] pane, select Product. On the [Home] ribbon, select [Combine – Merge Queries – Merge Queries].
 - Note:** Merging queries lets you bring together data from different sources (such as SQL Server and CSV files).

The screenshot shows the Power Query Editor interface. The ribbon is at the top with 'Home' selected. In the center, there's a table grid with columns: ProductKey, Product, Standard Cost, and Color. The 'Color' column has some red cells. To the left, the 'Queries [10]' pane lists 'Salesperson', 'SalespersonRegion', 'Product' (which is selected), 'Reseller', 'Region', and 'Sales'. On the right, the 'Query Settings' pane shows 'Name: Product' under 'PROPERTIES'.

- In the [Merge] dialog, configure as follows, then select [OK]:
 - In the Product query grid, select the Color column header.
 - For the second table, choose the ColorFormats query.
 - In ColorFormats, select the Color column header.
 - Join Kind: Left Outer (all from Product, matching from ColorFormats).

The screenshot shows the 'Merge' dialog. It has two tables: 'Product' and 'ColorFormats'. The 'Color' column is selected in both. The 'Join Kind' dropdown is set to 'Left Outer (all from first, matching from second)'. At the bottom, there are 'OK' and 'Cancel' buttons.

- If the [Privacy levels] dialog appears, set all sources to "Organizational", and then select [Save].
 - Privacy levels control whether data can be shared between sources.
 - Setting each source to "Organizational" allows data to be shared when needed. "Private" sources can't share data with other sources.
 - This doesn't mean a "private" source can't be shared at all—it means the Power Query engine won't share data **between sources**.



4. Select the [Expand] icon on the newly added **ColorFormats** column, select "Background Color Format" and "Font Color Format", and then select [OK].

5. Confirm in the lower-left status bar that there are 8 columns and 397 rows. On the [Home] ribbon, select [Close & Apply – Apply].

EXERCISE 11. Update the ColorFormats query

In this task, you'll update the **ColorFormats** query so that it doesn't load.

1. In the [Queries] pane, select **ColorFormats**. In [Query Settings], select "All properties".

Queries [10]

Color	Background Color Format	Font Color Format
1 Black	#000000	#FFFFFF
2 Blue	#0000FF	#FFFFFF
3 Grey	#808080	#FFFFFF
4 Multi	#B6B6BF	#000000
5 NA	#DCDCDC	#000000
6 Red	#FF0000	#FFFFFF
7 Silver	#COCOCO	#000000
8 Silver/Black	#696969	#FFFFFF
9 White	#FFFFFF	#000000
10 Yellow	#FFFF00	#000000

Query Settings

- PROPERTIES
 - Name: ColorFormats
 - All Properties
- APPLIED STEPS
 - Source
 - Changed Type
 - Promoted Headers
 - Changed Type1

- In the [Query Properties] dialog, clear "Enable load to report", and then select [OK].
 - Disabling load prevents this query from being loaded to the data model as a table.
 - Because this query was merged into the **Product** query, which loads to the model, there's no need to load this query.

Queries [10]

Query Properties

Name: ColorFormats

Description:

Enable load to report

⚠️ Disabling load will remove the table from the report, and any visuals that use its columns will be broken.

Include in report refresh

OK Cancel

Query Settings

- PROPERTIES
 - Name: ColorFormats
 - All Properties
- APPLIED STEPS
 - Source
 - Changed Type
 - Promoted Headers
 - Changed Type1

EXERCISE 12: Clean up

In this task, you'll complete the lab.

- In the [Queries] pane, verify that the following **eight** queries are present and correctly named.

- Salesperson
- SalespersonRegion
- Product
- Reseller
- Region
- Sales
- Targets
- ColorFormats

Queries [10]

EmployeeKey	EmployeeID	Salesperson	Title
1	272	502097814	Stephen Jiang
2	277	112432117	Brian Welcker
3	281	841560125	Michael Blythe
4	282	191644724	Linda Mitchell
5	283	615389812	Jillian Carson
6	284	234474252	Garrett Vargas
7	285	716374314	Tsvi Reiter
8	286	61161660	Pamela Anzman-Wolfe
9	287	139397894	Shu Ito
10	288	399771412	José Saravia
11	289	987554265	David Campbell

Query Settings

- PROPERTIES
 - Name: Salesperson
 - All Properties
- APPLIED STEPS
 - Source
 - Navigation
 - Filtered Rows
 - Removed Other Columns
 - Merged Columns
 - Renamed Columns

- To load the data model, on the [Home] ribbon select [Close – Close & Apply – Close & Apply]. All queries with "Enable load" selected will now be loaded into the data model.

The screenshot shows the Power BI Desktop interface with the 'Data' pane open. The 'Salesperson' table is currently selected. The 'Home' ribbon tab is active. The 'Close & Apply' button, located in the top-left corner of the ribbon, is highlighted with a red box.

3. In the [Data] pane of [Power BI Desktop], confirm that **seven** tables are loaded into the data model as shown.
Save the Power BI Desktop file.

The screenshot shows the Power BI Desktop interface with the 'Data' ribbon tab active. The 'Home' ribbon tab is also highlighted. The 'Data' pane on the right side lists seven tables: Product, Region, Reseller, Sales, Salesperson, SalespersonRegion, and Targets. A red dashed box highlights the list of tables in the Data pane.