

Code Issues 6 Pull requests Actions Projects Security Insights

Main ▾

...

PL-300-Microsoft-Power-BI-Data-Analyst / Instructions / 01-prepare-data-with-power-query-in-power-bi-desktop.md



bpmoring Minor update to instructions ✓



1 contributor

349 lines (176 sloc) | 15.7 KB

...

lab	
title	module
Prepare Data in Power BI Desktop	Module 2 - Get Data in Power BI

## Prepare Data in Power BI Desktop

The estimated time to complete the lab is 45 minutes

In this lab you commence the development of a Power BI Desktop solution for the Adventure Works company. It involves connecting to source data, previewing the data, and using data preview techniques to understand the characteristics and quality of the source data.

In this lab you learn how to:

- Open Power BI Desktop
- Set Power BI Desktop options
- Connect to source data
- Preview source data

- Use data preview techniques to better understand the data

## Lab story

This lab is one of many in a series of labs that was designed as a complete story from data preparation to publication as reports and dashboards. You can complete the labs in any order. However, if you intend to work through multiple labs, we suggest you do them in the following order:

1. Prepare Data in Power BI Desktop
2. Load Data in Power BI Desktop
3. Model Data in Power BI Desktop
4. Create DAX Calculations in Power BI Desktop, Part 1
5. Create DAX Calculations in Power BI Desktop, Part 2
6. Design a Report in Power BI Desktop, Part 1
7. Design a Report in Power BI Desktop, Part 2
8. Create a Power BI Dashboard
9. Perform Data Analysis in Power BI Desktop
10. Enforce Row-Level Security

## Exercise 1: Prepare Data

---

In this exercise you will create eight Power BI Desktop queries. Six queries will source data from SQL Server, and two from CSV files.

### Task 1: Save the Power BI Desktop file

In this task you will first save the Power BI Desktop file.

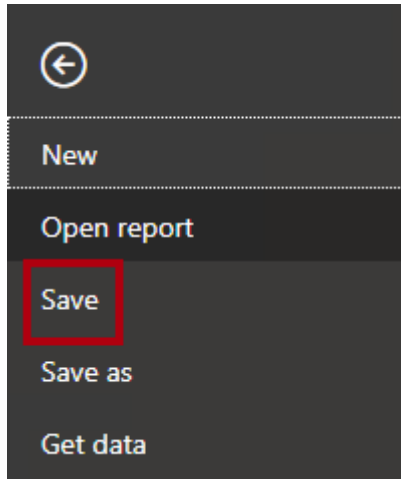
1. To open the Power BI Desktop, on the taskbar, click the Microsoft Power BI Desktop shortcut.



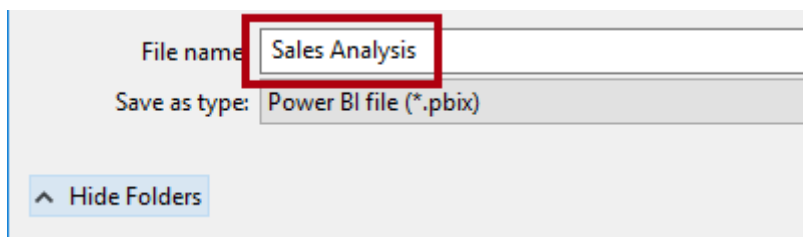
2. To close the getting started window, at the top-right of the window, click X.



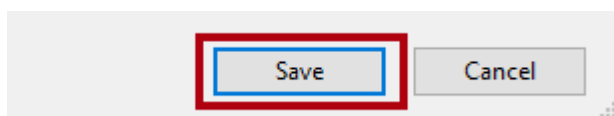
3. To save the file, click the **File** ribbon tab to open the backstage view.
4. Select **Save**.



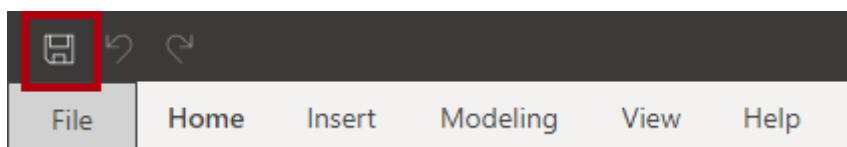
5. In the **Save As** window, navigate to the **D:\PL300\MySolution** folder.
6. In the **File Name** box, enter **Sales Analysis**.



7. Click **Save**.



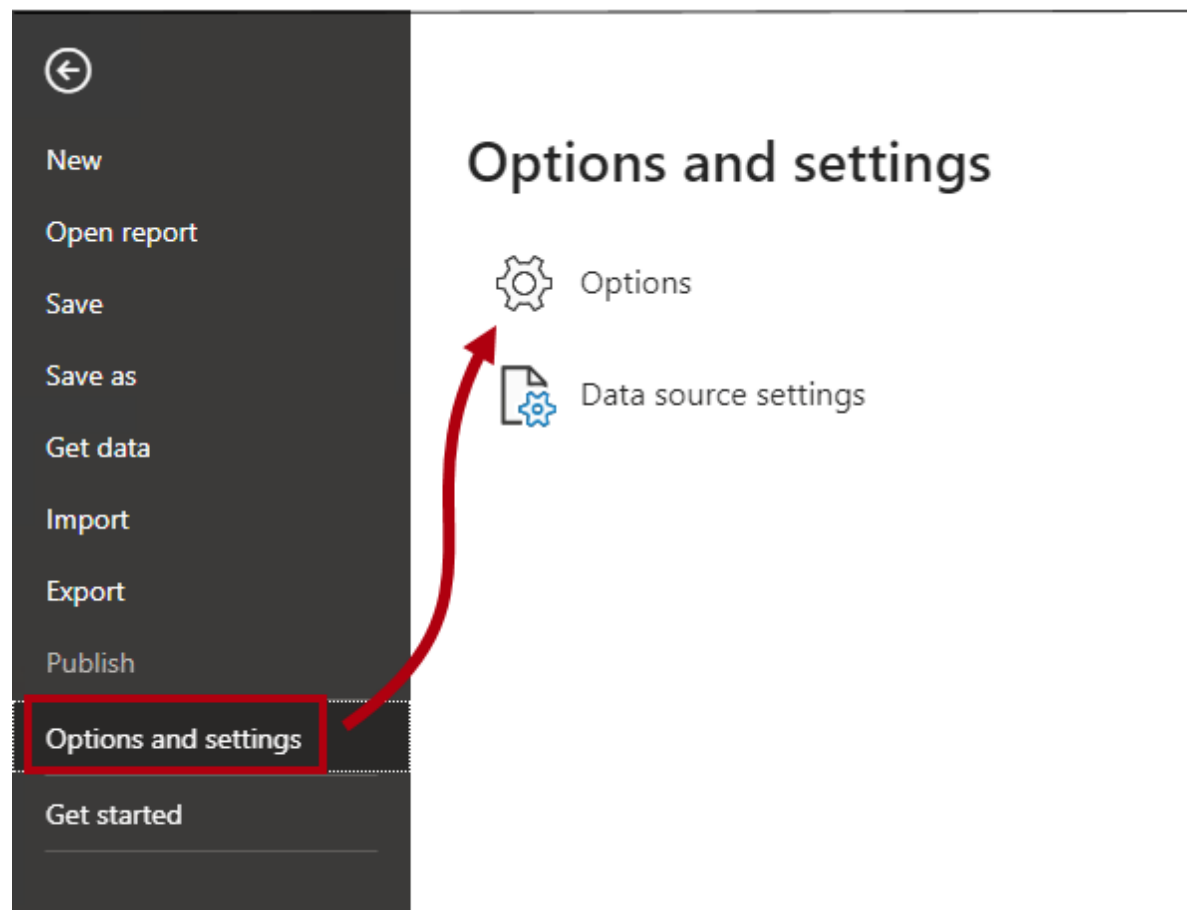
Tip: You can also save the file by click the **Save** icon located at the top-left.



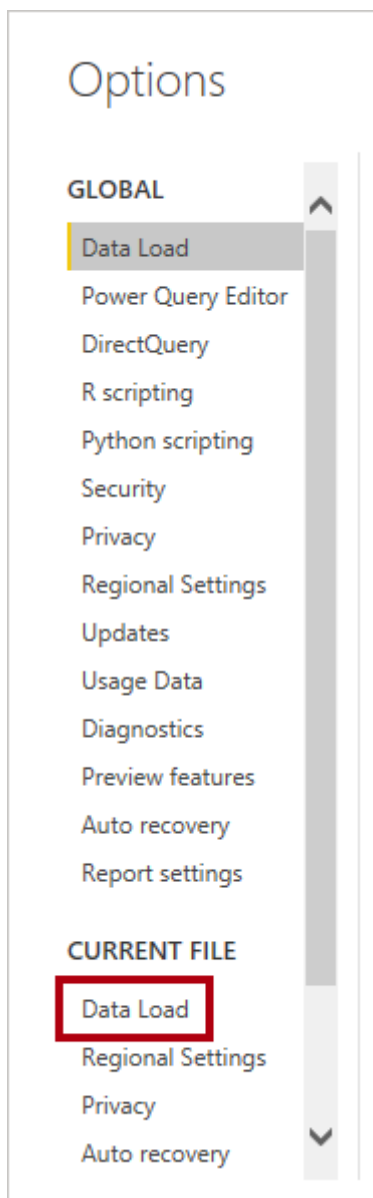
## Task 2: Set Power BI Desktop options

In this task you will set Power BI Desktop options.

1. In Power BI Desktop, click the **File** ribbon tab to open the backstage view.
2. At the left, select **Options and Settings**, and then select **Options**.

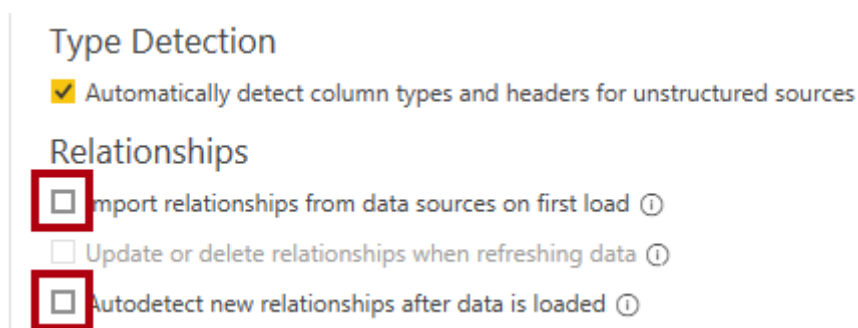


3. In the **Options** window, at the left, in the **Current File** group, select **Data Load**.



The **Data Load** settings for the current file allow setting options that determine default behaviors when modeling.

4. In the **Relationships** group, uncheck the two options that are already checked.



While having these two options enabled can be helpful when developing a data model, you disabled them earlier to support the lab experience. When you create relationships in the **Load Data in Power BI Desktop** lab, you'll learn why you are adding each one.

5. Click **OK**.

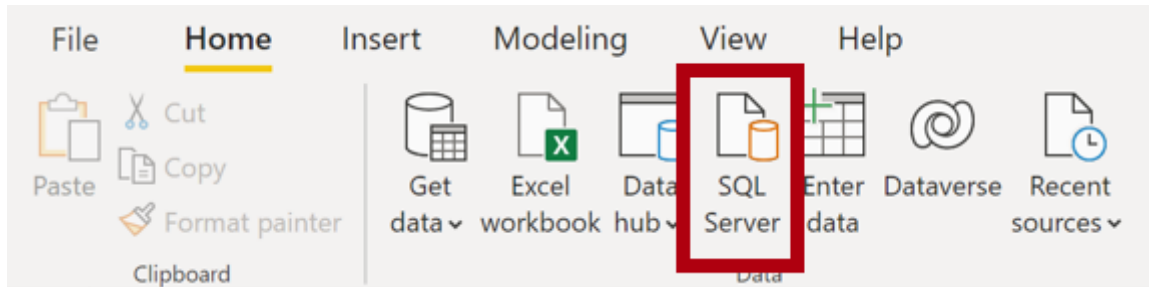


6. Save the Power BI Desktop file.

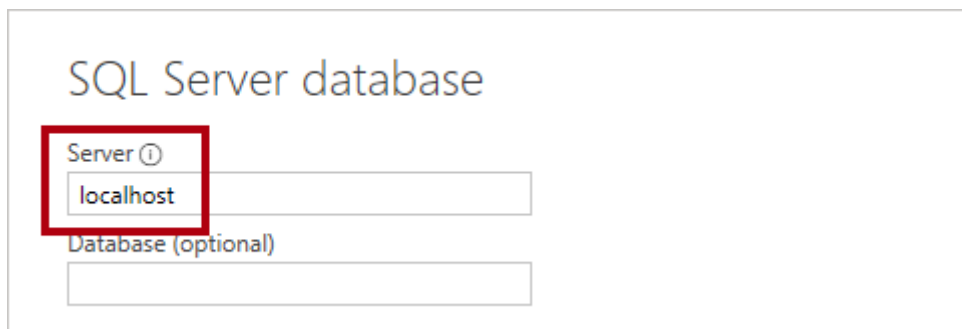
### Task 3: Get data from SQL Server

In this task you will create queries based on SQL Server tables.

1. On the **Home** ribbon tab, from inside the **Data** group, click **SQL Server**.



2. In the **SQL Server Database** window, in the **Server** box, enter **localhost**.



In this lab you'll connect to the SQL Server database by using **localhost**. This isn't a recommended practice when creating your own solutions. It's because gateway data sources cannot resolve **localhost**.

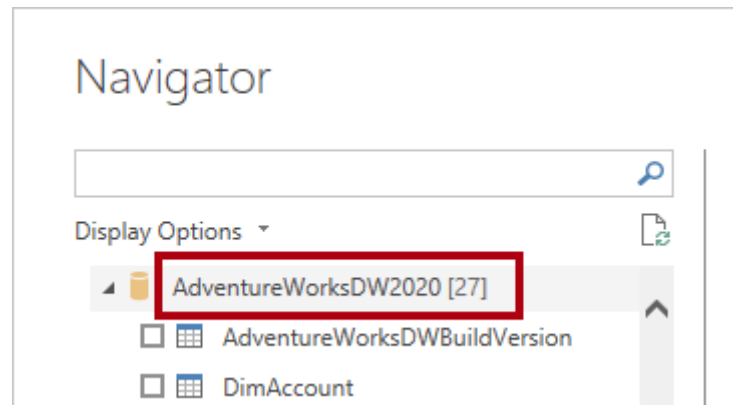
3. Click **OK**.



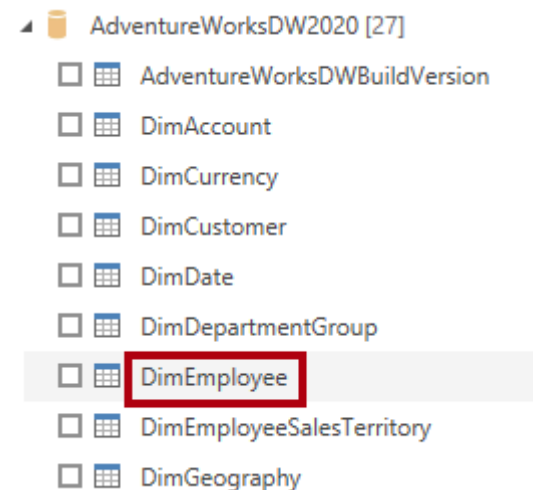
4. If prompted for credentials, in the **SQL Server Database** window, select **Use my current credentials**. Then **Connect**.

5. In the **Navigator** window, at the left, expand the **AdventureWorksDW2020** database.

The **AdventureWorksDW2020** database is based on the **AdventureWorksDW2017** sample database. It has been modified to support the learning objectives of the course labs.



6. Select—but don't check—the **DimEmployee** table.



7. In the right pane, notice a preview of the table data.

The preview data allows you to determine the columns and a sample of rows.

8. To create queries, select the checkbox next to the following six tables:

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

9. To apply transformations to the data of the selected tables, click **Transform Data**.

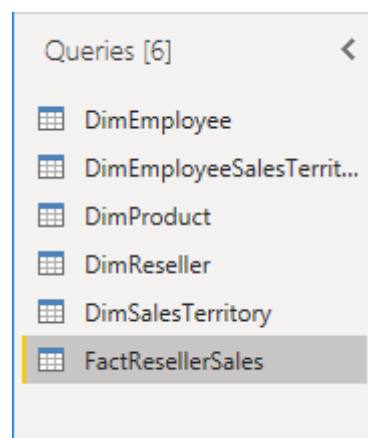
You won't be transforming the data in this lab. The objectives of this lab focus on exploring and profiling the data in the **Power Query Editor** window.



## Task 4: Preview SQL Server queries

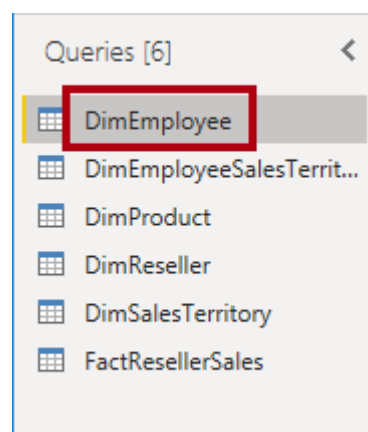
In this task you will preview the data of the SQL Server queries. First, you will learn relevant information about the data. You will also use column quality, column distribution, and column profile tools to understand the data and to assess data quality.

1. In the **Power Query Editor** window, at the left, notice the **Queries** pane.



The **Queries** pane contains one query for each table you checked.

2. Select the first query—**DimEmployee**.



The **DimEmployee** table in the SQL Server database stores one row for each employee. A subset of the rows from this table represents the salespeople, which will be relevant to the model you'll develop.

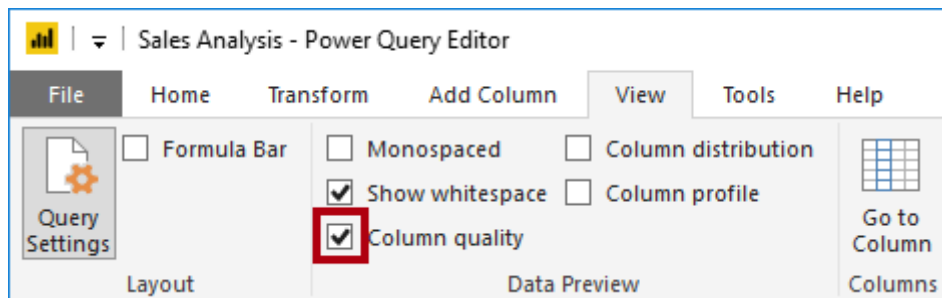
3. At the bottom left, in the status bar, notice the table statistics—the table has 33 columns, and 296 rows.



4. In the data preview pane, scroll horizontally to review all columns.
5. Notice that the last five columns contain **Table** or **Value** links.

These five columns represent relationships to other tables in the database. They can be used to join tables together. You'll join tables in the **Load Data in Power BI Desktop** lab.

6. To assess column quality, on the **View** ribbon tab, from inside the **Data Preview** group, check **Column Quality**.

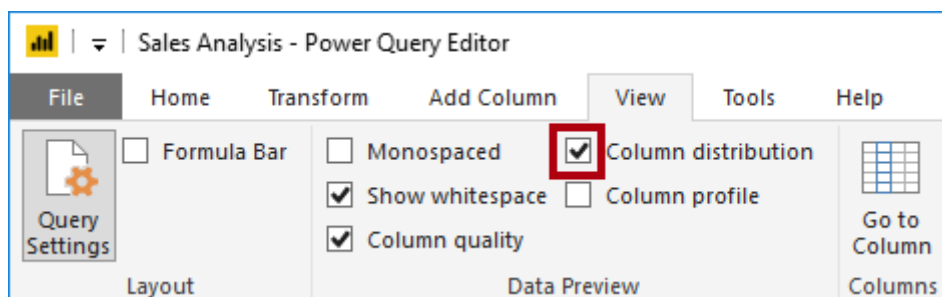


The column quality feature allows you to easily determine the percentage of valid, error, or empty values found in columns.

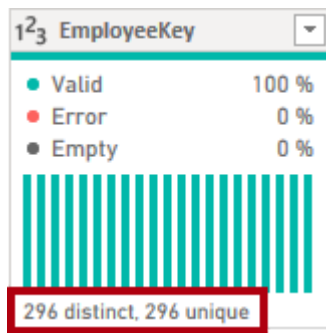
7. For the **Position** column (sixth last column), notice that 94% of rows are empty (null).



8. To assess column distribution, on the **View** ribbon tab, from inside the **Data Preview** group, check **Column Distribution**.

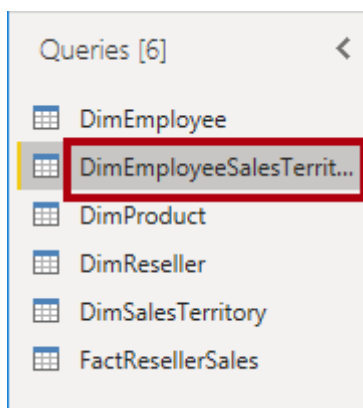


9. Review the **Position** column again, and notice that there are four distinct values, and one unique value.
10. Review the column distribution for the **EmployeeKey** (first) column—there are 296 distinct values, and 296 unique values.



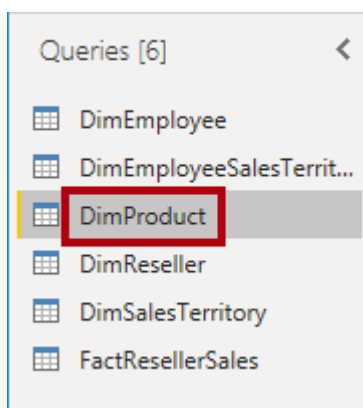
When the distinct and unique counts are the same, it means the column contains unique values. When modeling, it's important that some model tables have unique columns. These unique columns can be used to create one-to-many relationships, which you will do in the **Model Data in Power BI Desktop** lab.

11. In the **Queries** pane, select the **DimEmployeeSalesTerritory** query.



The **DimEmployeeSalesTerritory** table stores one row for each employee and the sales territory regions they manage. The table supports relating many regions to a single employee. Some employees manage one, two, or possibly more regions. When you model this data, you'll need to define a many-to-many relationship.

12. In the **Queries** pane, select the **DimProduct** query.



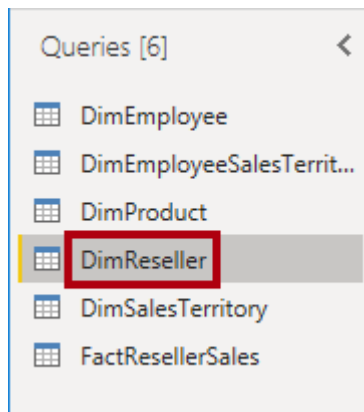
The **DimProduct** table contains one row per product sold by the company.

13. Horizontally scroll to reveal the last columns.

14. Notice the **DimProductSubcategory** column.

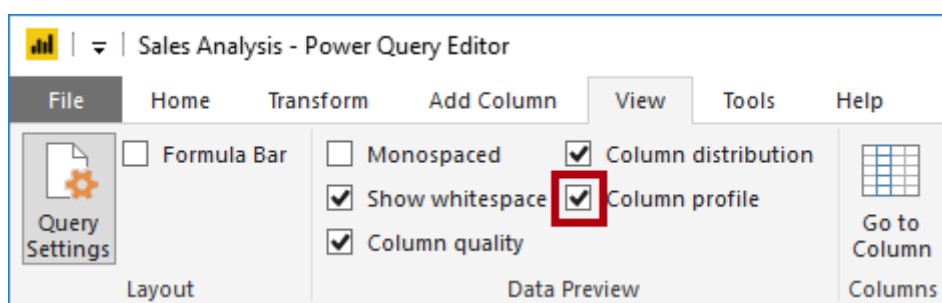
When you add transformations to this query in the **Load Data in Power BI Desktop** lab, you'll use the **DimProductSubcategory** column to join tables.

15. In the **Queries** pane, select the **DimReseller** query.



The **DimReseller** table contains one row per reseller. Resellers sell, distribute, or value add to the Adventure Works products.

16. To view column values, on the **View** ribbon tab, from inside the **Data Preview** group, check **Column Profile**.

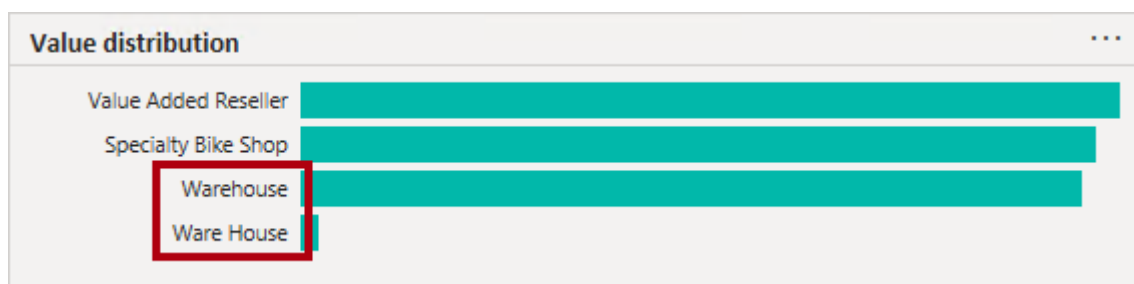


17. Select the **BusinessType** column header.

18. Notice the new pane beneath the data preview pane.

19. Review the column statistics and value distribution in the data preview pane.

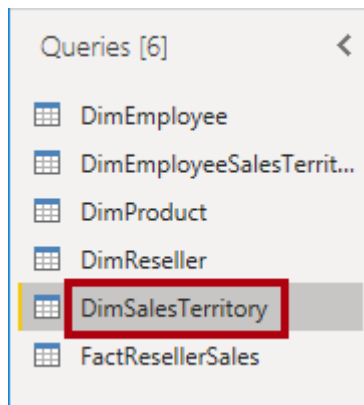
20. Notice the data quality issue: there are two labels for warehouse (**Warehouse**, and the misspelled **Ware House**).



21. Hover the cursor over the **Ware House** bar, and notice that there are five rows with this value.

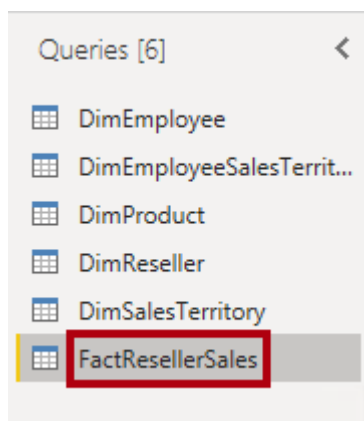
You'll apply a transformation to relabel these five rows in the **Load Data in Power BI Desktop** lab.

22. In the **Queries** pane, select the **DimSalesTerritory** query.



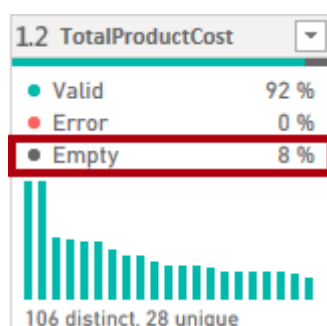
The **DimSalesTerritory** table contains one row per sales region, including **Corporate HQ** (headquarters). Regions are assigned to a country, and countries are assigned to groups. In the **Model Data in Power BI Desktop** lab, you'll create a hierarchy to support analysis at region, country, or group level.

23. In the **Queries** pane, select the **FactResellerSales** query.



The **FactResellerSales** table contains one row per sales order line—a sales order contains one or more line items.

24. Review the column quality for the **TotalProductCost** column, and notice that 8% of the rows are empty.

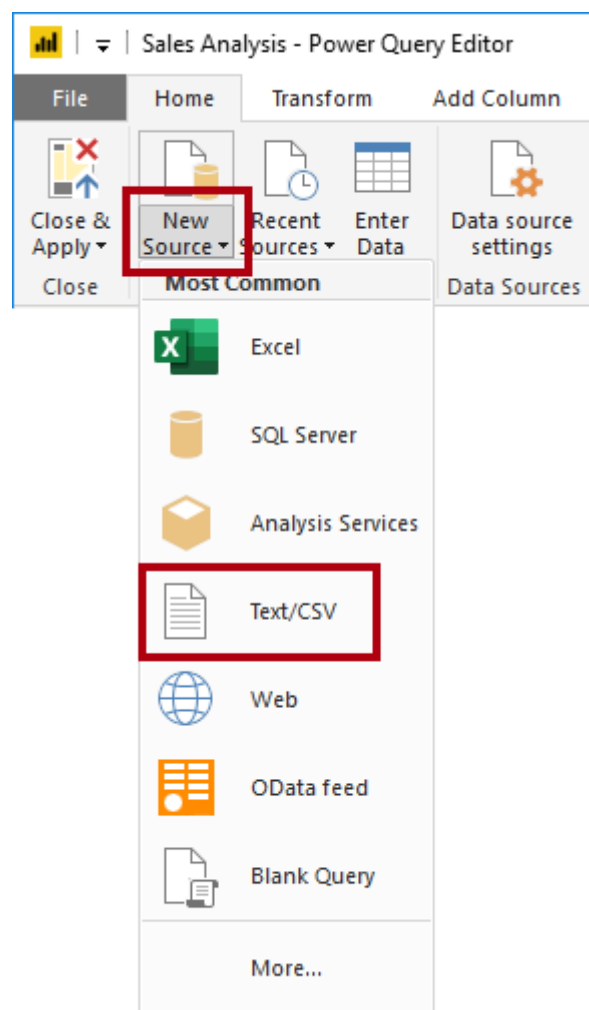


Missing **TotalProductCost** column values is a data quality issue. To address the issue, in the **Load Data in Power BI Desktop** lab, you'll apply transformations to fill in missing values by using the product standard cost, which is stored in the related **DimProduct** table.

## Task 5: Get data from a CSV file

In this task you will create a query based on a CSV file.

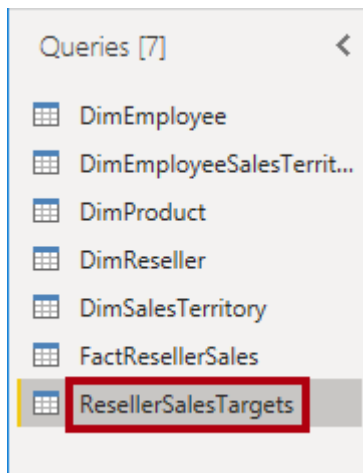
1. To add a new query, in the **Power Query Editor** window, on the **Home** ribbon tab, from inside the **New Query** group, click the **New Source** down-arrow, and then select **Text/CSV**.



2. In the **Open** window, navigate to the **D:\PL300\Resources** folder, and select the **ResellerSalesTargets.csv** file.
3. Click **Open**.
4. In the **ResellerSalesTargets.csv** window, review the preview data.
5. Click **OK**.



6. In the **Queries** pane, notice the addition of the **ResellerSalesTargets** query.

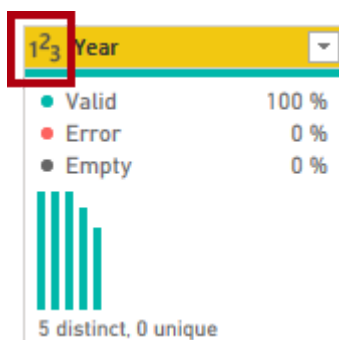


The **ResellerSalesTargets** CSV file contains one row per salesperson, per year. Each row records 12 monthly sales targets (expressed in thousands). Note that the business year for the Adventure Works company commences on July 1.

7. Notice that no column contains empty values.

When there isn't a monthly sales target, a hyphen character is stored instead.

8. Review the icons in each column header, to the left of the column name.



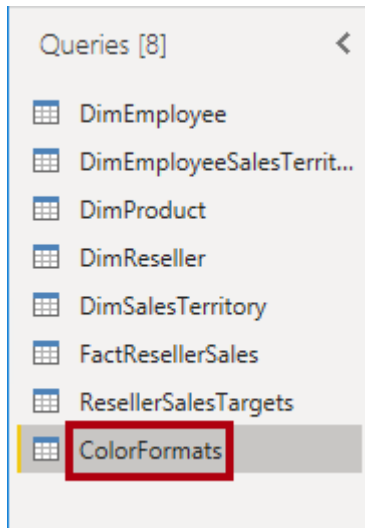
The icons represent the column data type. **123** is whole number, and **ABC** is text.

You'll apply many transformations to achieve a different shaped result consisting of only three columns: **Date**, **EmployeeKey**, and **TargetAmount** in the **Load Data in Power BI Desktop** lab.

## Task 6: Get additional data from a CSV file

In this task you will create an additional query based on a different CSV file.

1. Use the steps in the previous task to create a query based on the **D:\PL300\Resources\ColorFormats.csv** file.

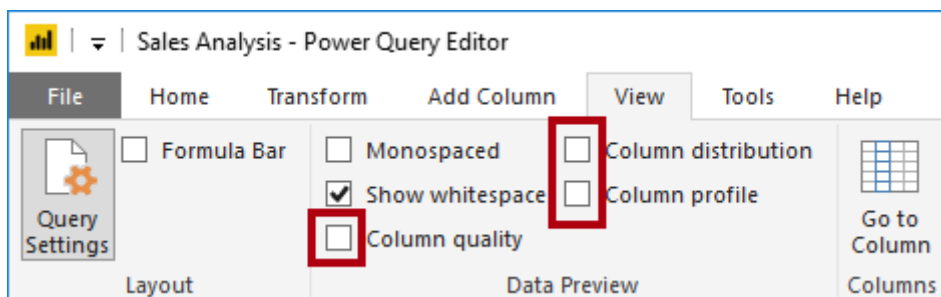


The **ColorFormats** CSV file contains one row per product color. Each row records the HEX codes to format background and font colors. You'll integrate this data with the **DimProduct** query data in the **Load Data in Power BI Desktop** lab.

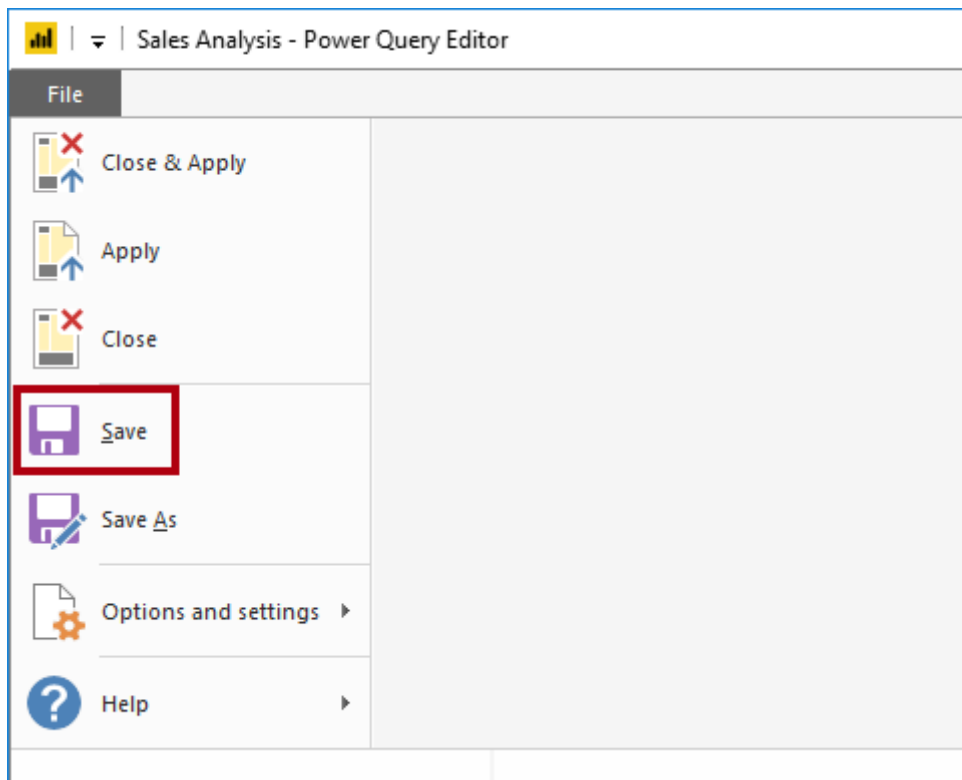
## Task 7: Finish up

In this task you will complete the lab.

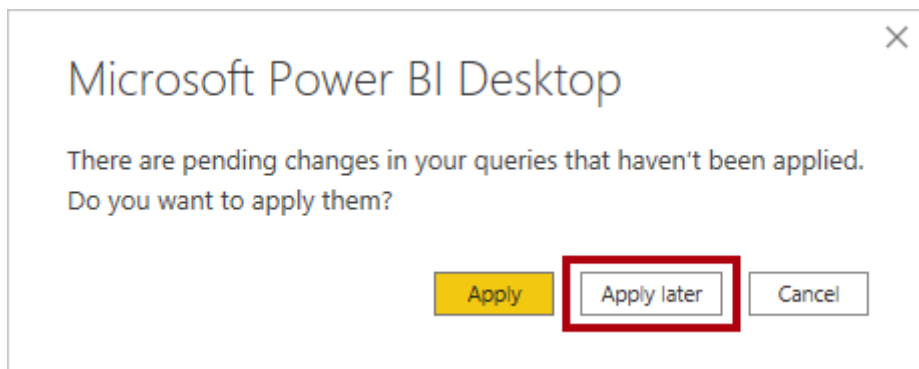
1. On the **View** ribbon tab, from inside the **Data Preview** group, uncheck the three data preview options that were previously enabled in this lab:
  - Column quality
  - Column distribution
  - Column profile



2. To save the Power BI Desktop file, in the **Power Query Editor** window, on the **File** backstage view, select **Save**.



3. When prompted to apply the queries, click **Apply Later**.



Applying the queries will load their data to the data model. You're not ready to do that, as there are many transformations that must be applied first.

4. If you intend to start the next lab, leave Power BI Desktop open.

You'll apply various transformations to the queries and then apply the queries to load them to the data model in the **Load Data in Power BI Desktop** lab.