**Retail Store data Analysis Document**

1. **BRD**: - A Business Requirement Document (BRD) is a formal document that outlines the business objectives, functional requirements, and specifications for a project or initiative.
2. **Data Cleaning**:- Before jumping on modeling and creating reports check if your data is clean enough.
3. **Data Modeling**:- Create relationships between data tables for better filtering results and more meaningful reports.
4. **RLS**:- Row-Level Security, is a feature in Power BI that allows you to restrict data access at the row level based on user roles and rules defined in the data model.
5. **Gateway**:- A gateway is a tool that acts as a bridge between on-premises data sources and the Power BI service in the cloud. It allows Power BI to refresh and access data from on-premises data sources securely.
6. ﻿**Data Gathering/Requirement**:

1. Sales (folder by year)

2. Categories (Excel)

3. Geography (Excel)

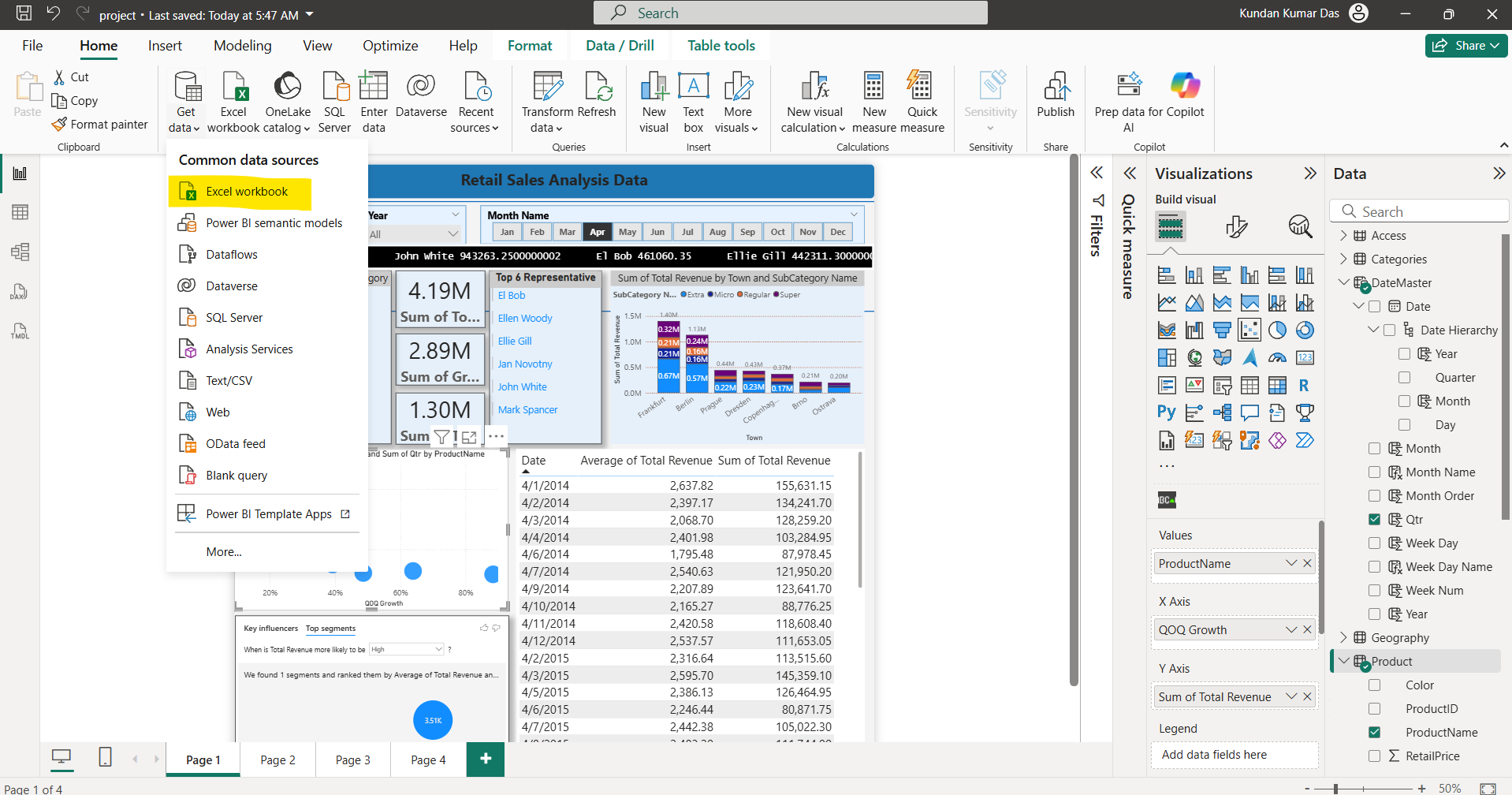
4. Product (CSV/Database)

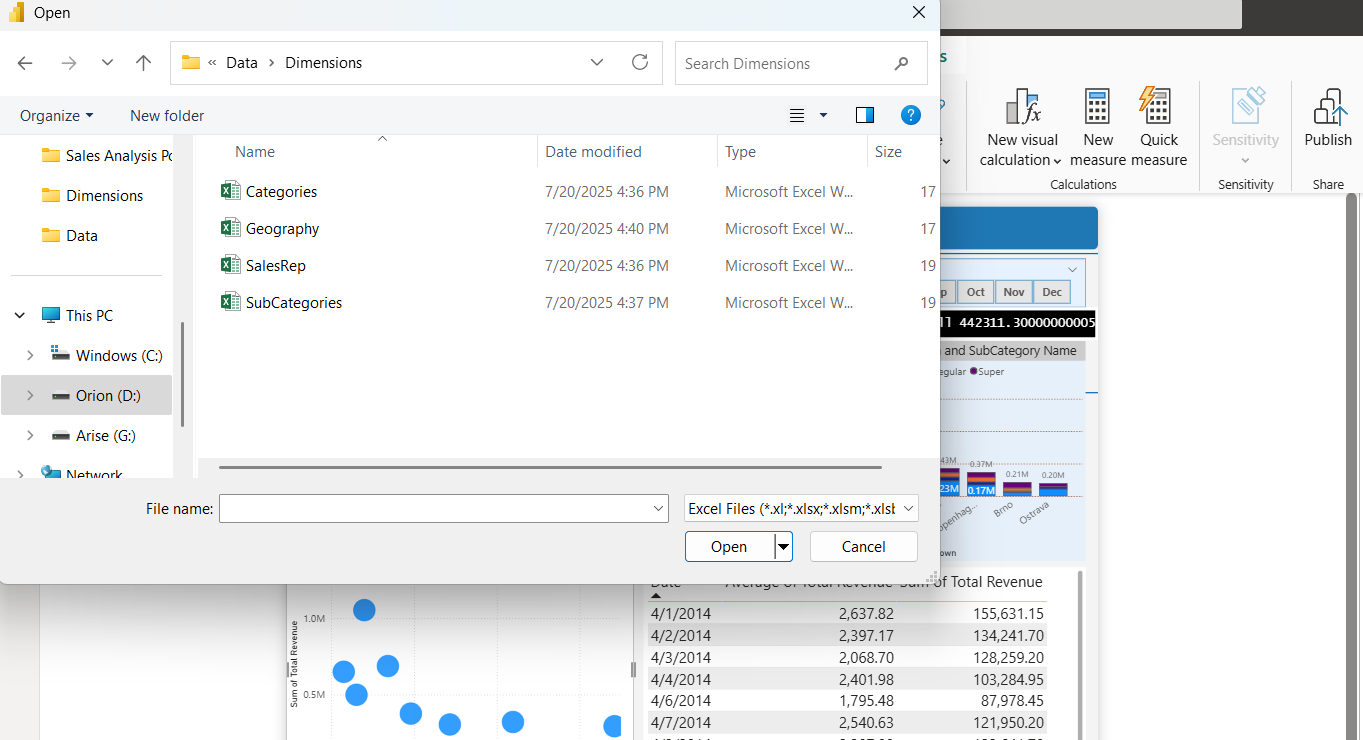
5. SalesRep (Excel)

6. SubCategories (Excel)

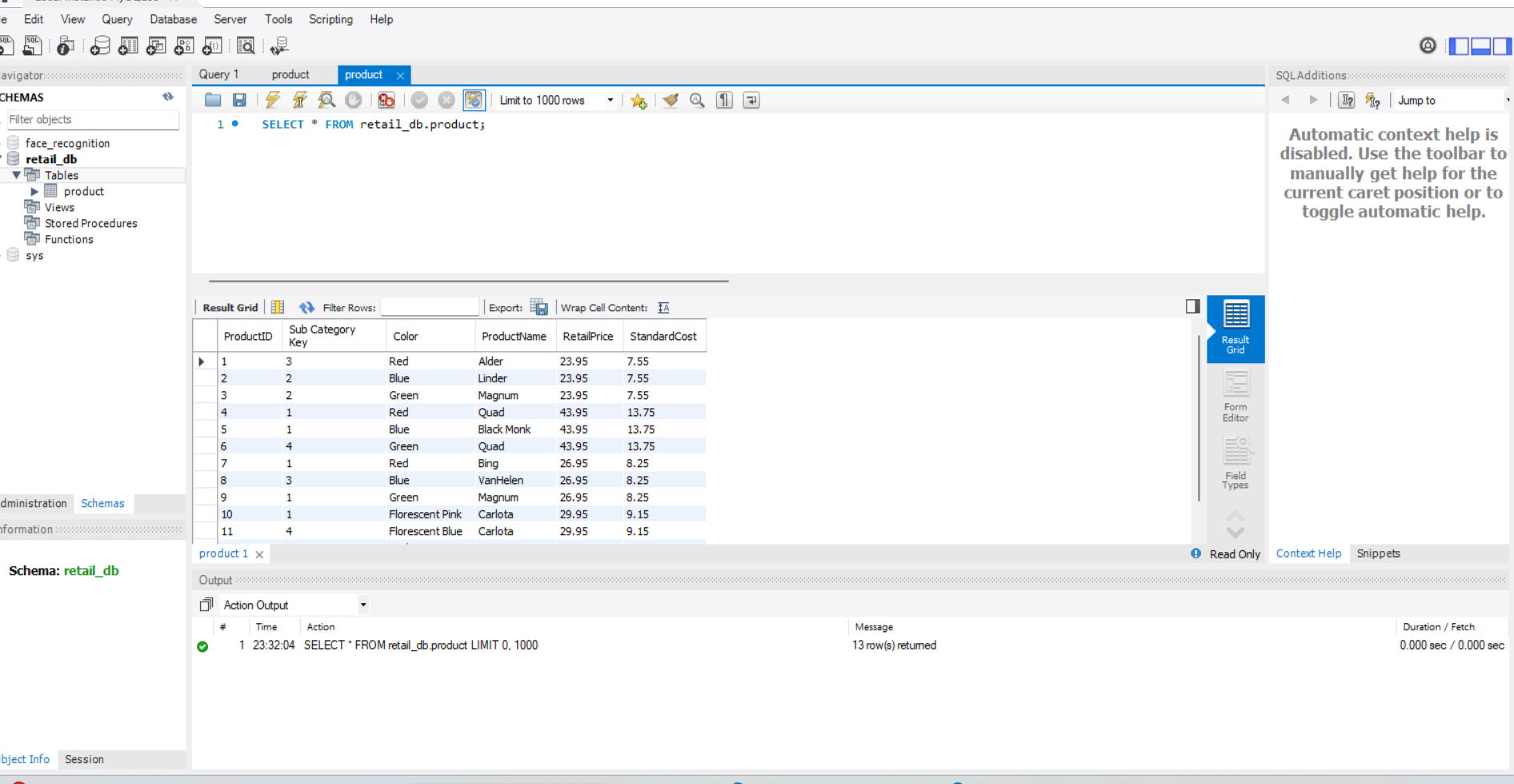
**Task 1.1**:

* For our project, we need to collect data from different sources like Excel, CSV, and MySQL Database. If the client gives us the data, that’s fine; if not, it’s our job to gather it. Everything mentioned in the BRD must be implemented. As per the BRD, the first table we need to bring is the **Category** table, which comes from an Excel file and will be used as a **Dimension Table** in our project.





Similarly, load the Geography, SalesRep, and SubCategories tables. According to the BRD document, the Product data can be taken either from a CSV file or from the database. In our case, we will use the MySQL Database to load the Product table, where all the product-related data is stored.



Open Power BI Desktop and choose ‘Get Data,’ then select **MySQL Database**. It will ask for two things: the **Server name** and the **Database name** where your data is stored. You can find both easily in MySQL Workbench. In our case, the Server name is **localhost/ID**. After this, you need to connect Power BI Desktop to the MySQL instance by providing the **Username** (usually ‘root’) and **Password** for the MySQL database.

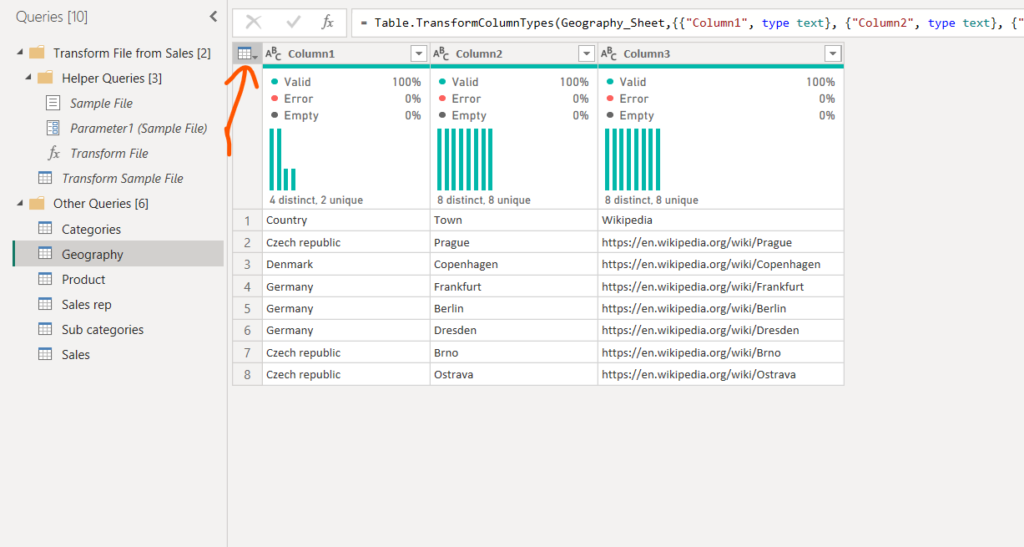
2. **Data Modeling:**

**Task 2.1:**

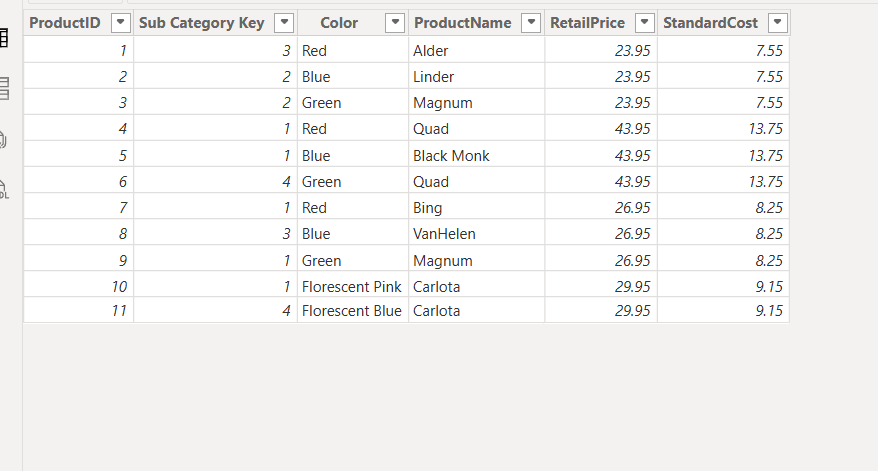
After importing all the data, we need to verify its accuracy by performing transformations in Power Query Editor. what to do in step by step:

* **Categories table:** The columns CategoryKey and CategoryName look correct, so no changes are needed.
* **Geography table:** The column names are currently in the first row, so we need to fix this.
* **Product table:** The ProductID column has duplicate records, but since dimension tables must be unique, we need to remove duplicates here.
* **SubCategory and SalesRep tables:** Both are dimension tables; however, the SalesRep table requires some corrections.

In Power Query Editor, open the Geography table and apply the **‘Use First Row as Header’** option to fix the column names.



In the **Sales** table, I remove unnecessary columns like **‘Source.Name’**. Next, I split the **Location** column into two separate columns—one for **Country** and one for **Town**—using the semicolon (;) as the delimiter.

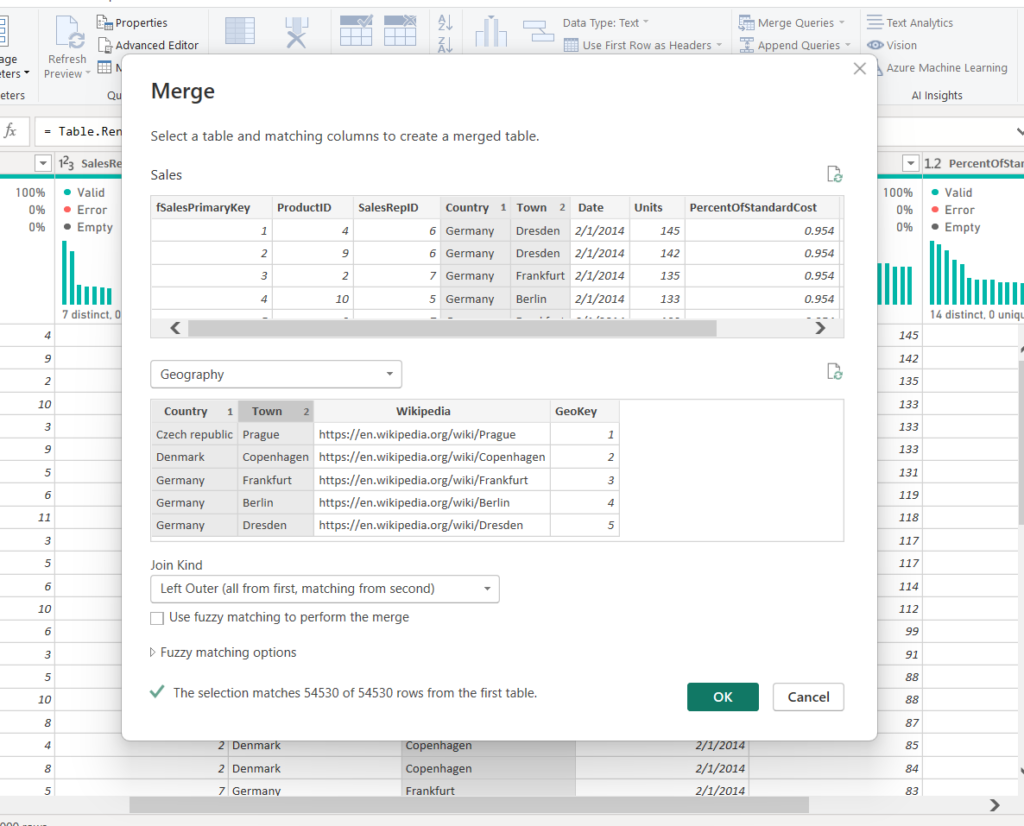


For the **Product** table, I remove duplicate records from the **ProductID** column by applying the **Remove Duplicates** option to ensure all IDs are unique in the dimension table.

**Task 2.2**

Create unique key (**GeoKey**) in Sales and Geography tables.

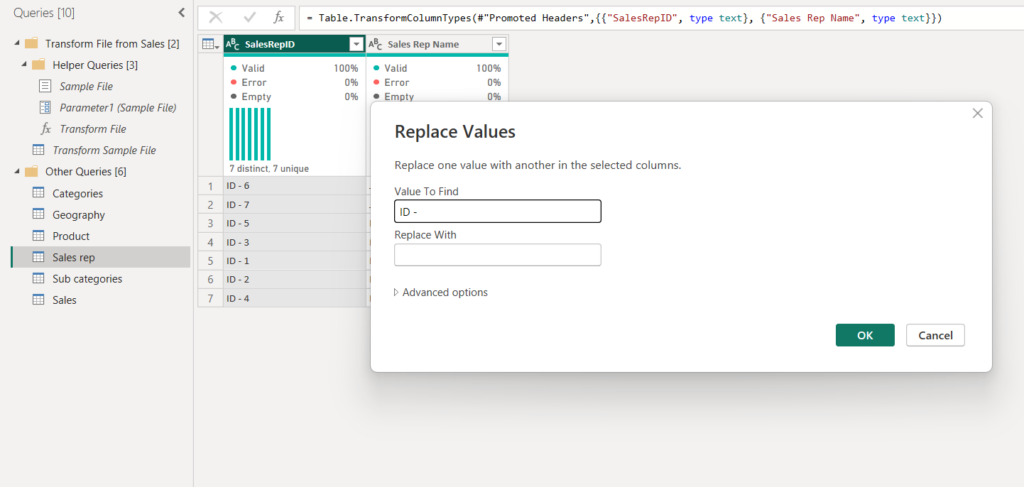
We will add index column in Geography table and merge it with Sales table for Country and Town columns. So by doing this the categorical data in Sales table we can remove and just make use of index values. After all this changes just click Close and Apply button.



**Task 2.3:**

The Dimensional queries SalesRep and Sub category need additional treatment. Some ID columns have the following format:

|  |
| --- |
| SalesRepID |
| ID – 6 |
| ID – 7 |
| ID – 5 |
| ID – 3 |
| ID – 1 |
| ID – 2 |
| ID – 4 |



Using the same ‘Use First Row as Header’ option to the SalesRep table in Power Query Editor. Also, check the SalesRepID column in the Sales table—it contains simple IDs like 1, 2, 3, etc., but in the SalesRep table, some IDs are formatted as ‘ID – 1’. We need to clean and standardize these values so they match correctly. Then go to the **SalesRep** table and select the **SalesRepID** column. Then I right-click and choose the **‘Replace’** option to replace **‘ID -’** with a blank value. I do the same for the **CategoryKey** column in the **SubCategory** table.

**Task 2.4:**

Creating a new table here using DAX query. Name this table as **DateMaster**table.

DateMaster = CALENDAR(FIRSTDATE(Sales[Date]), LASTDATE(Sales[Date]))

If you commit the above query, you will get a new table. now in this table we can perform time intelligence functions and find out sales by year, sales by quarter, sales by month, sales by week, and many more things.

In this same **DateMaster**table add a new column and name it as ‘Year’.

Year = Year(DateMaster[Date])

Similarly create another column for month.

Month = month(DateMaster[Date])

Month Name = FORMAT(DateMaster[Date], "MMM")

Similarly create another column for quarter.

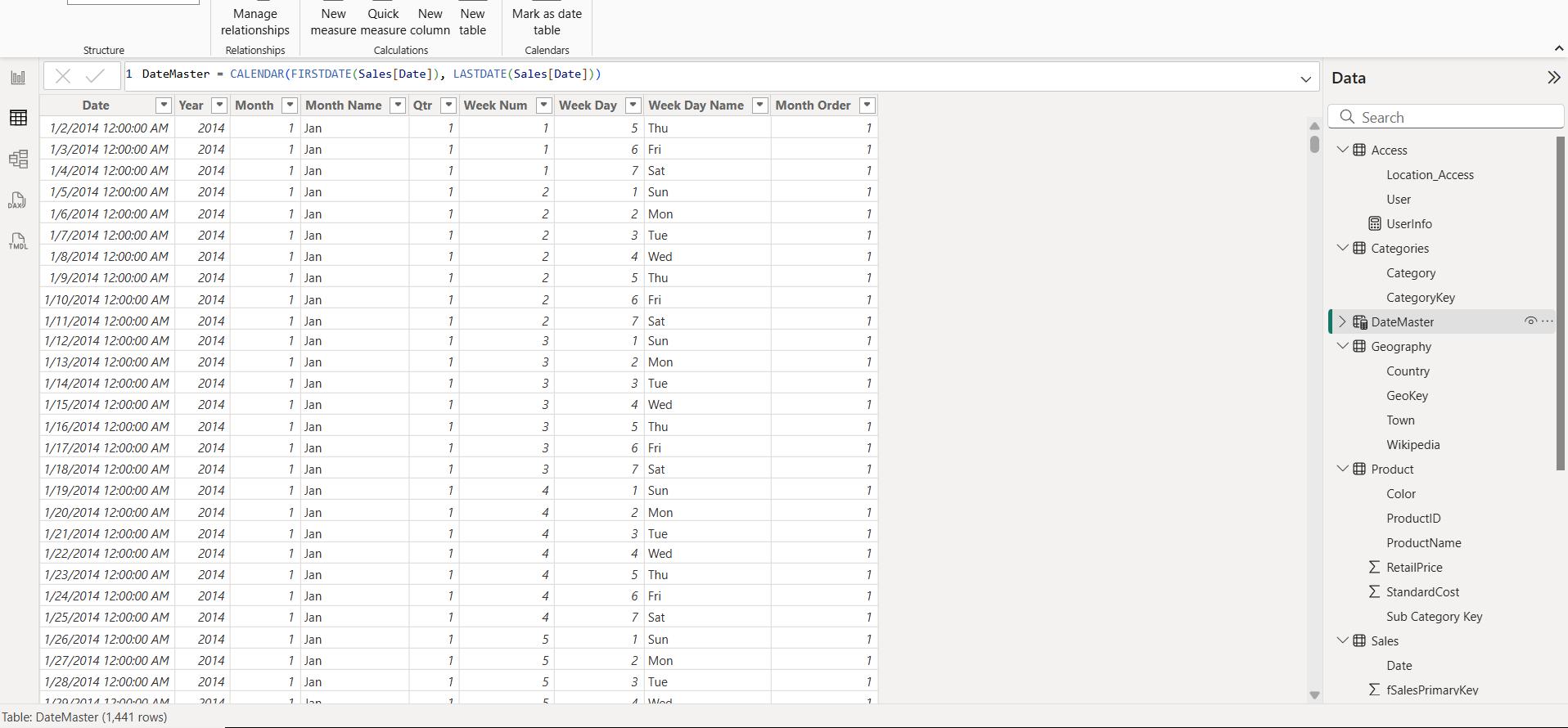
Qtr = QUARTER(DateMaster[Date])

Similarly create another column for week number and week day.

Week Num = WEEKNUM(DateMaster[Date])

Week Day = WEEKDAY(DateMaster[Date])

Week Day Name = FORMAT(DateMaster[Date], "DDD")



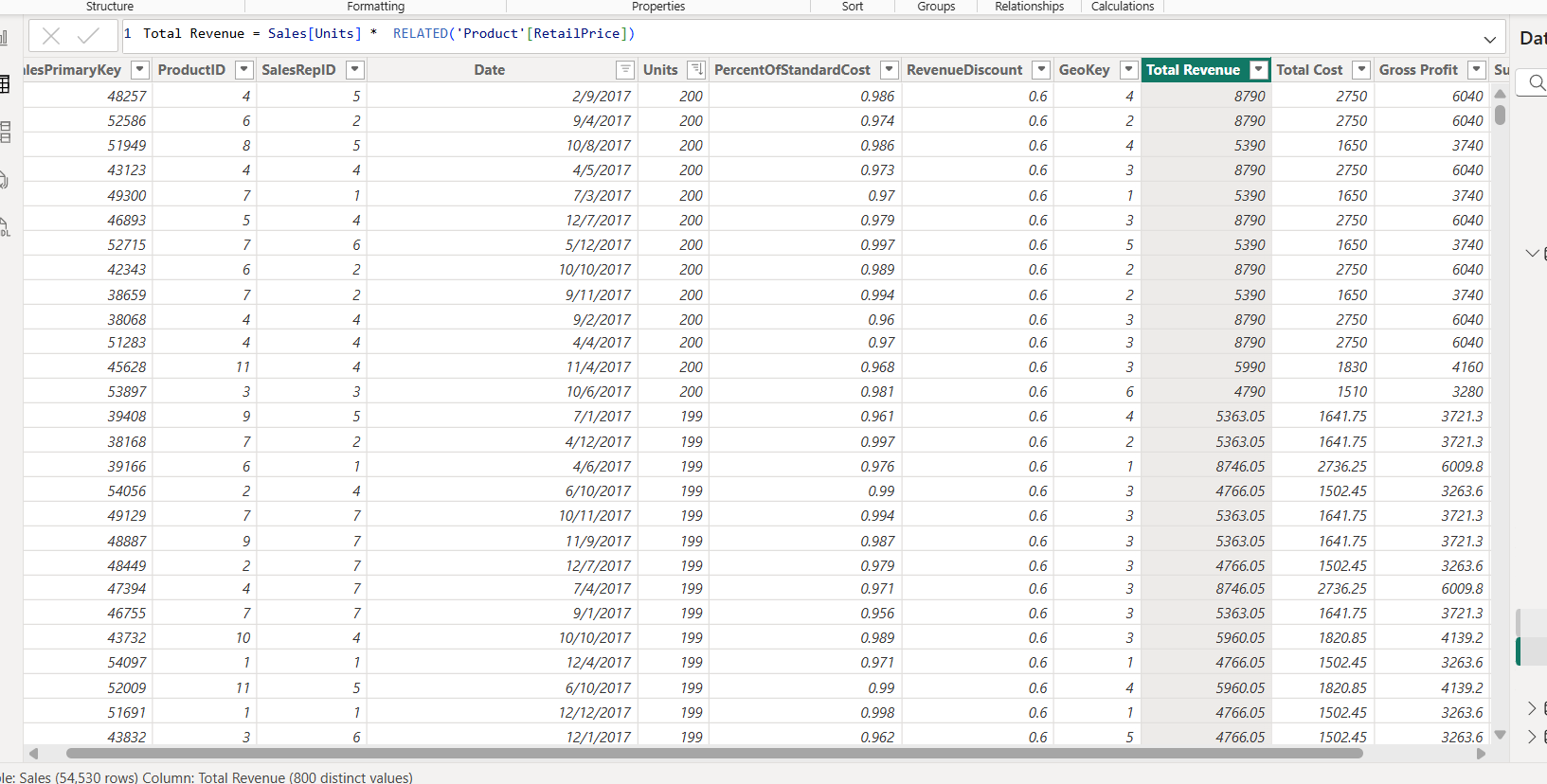
**3. DAX calculations**

**Task 3.1:**

Calculate **Total Revenue** in Sales table, using the Product's Retail Price, and multiplying it by the Units.

Total Revenue column in same Sales table.

Total Revenue = Sales[Units] \* RELATED(Product[RetailPrice])



**Task 3.2:**

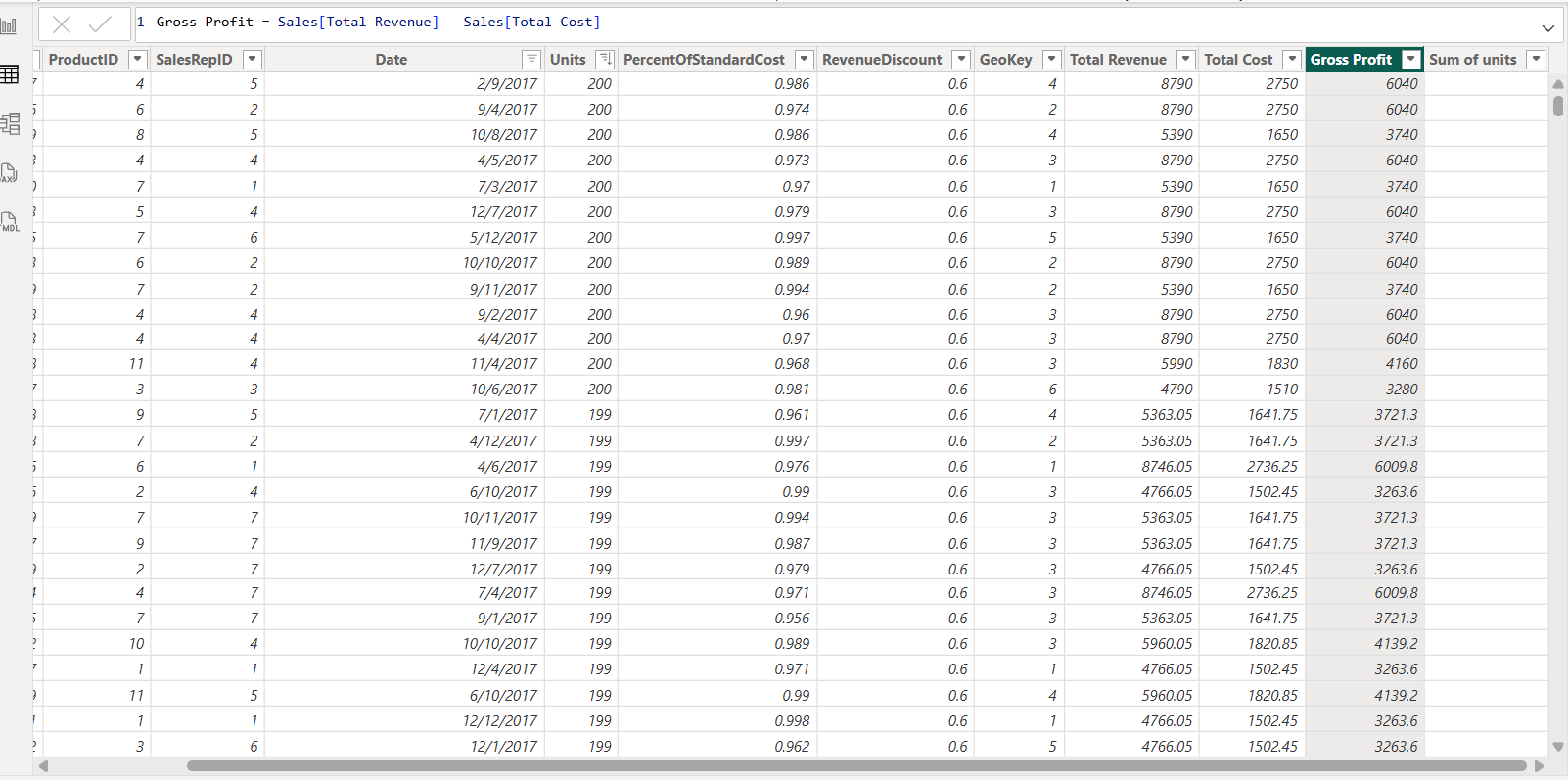
Calculate **Total Cost** in Sales table, using the Product's Standard Cost, and multiplying it by the Units.

Total Cost = Sales[Units] \* RELATED(Product[StandardCost])

**Task 3.3:**

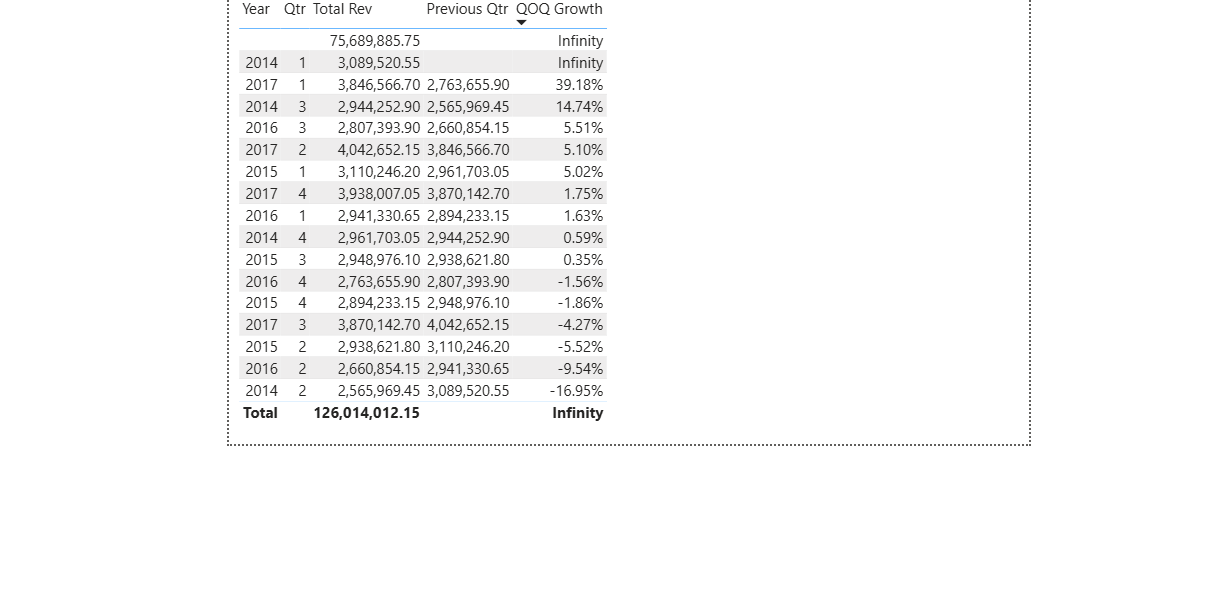
Calculate **Gross Profit** in Sales: Total Revenue – Total Cost

 Gross Profit = Sales[Total Revenue] - Sales[Total Cost]



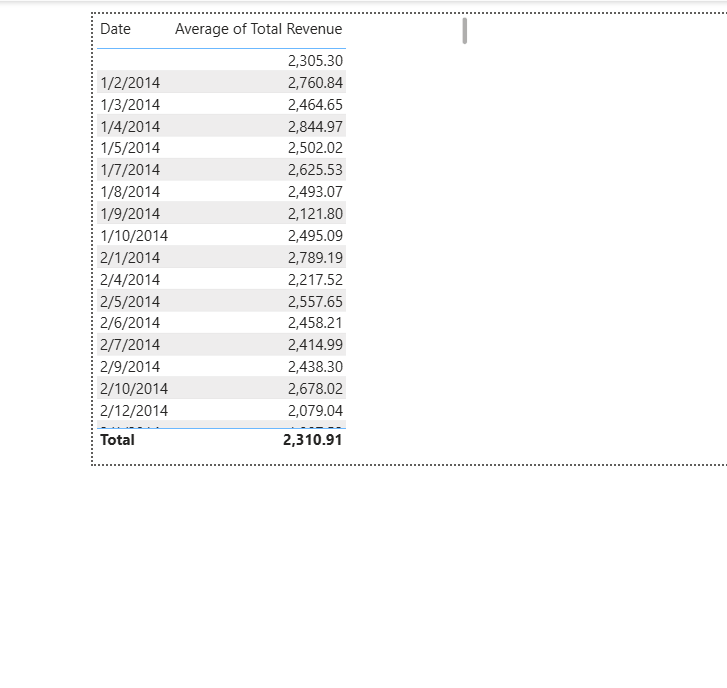
**Task 3.4:**

Calculate a **QoQ growth Change%** measure that could benefit us in decision making.



**Task 3.5:**

Calculate a measure for the average sum of **Total Revenue** per day based on the Dates of actual Sales.



4. Dashboard insights

