**Power Query Data Extraction and Transformation**

1. **Data Extraction:**
   1. Connecting to the MySQL Database:

* Connected Power BI to the MySQL database containing the “classicmodels” schema.
* Selected relevant tables: customers, employees, offices, orderdetails , orders, payments, productlines, and products.
  1. Steps followed:
* I opened Power BI Desktop and navigated to **Home > Get Data > More…**.
* I selected **MySQL Database** as the data source.
* I entered the server’s name as 127.0.0.1 and database details as “classicmodels” to connect to the classicmodels database.
* I selected and loaded the required tables (customers, employees, offices, orderdetails , orders, payments, productlines, and products) into Power Query.

1. **Data Transformation:**

Once the data was extracted, I performed the following transformations in Power Query to prepare the data for analysis.

2.1 Merging Tables:

Combined data from related tables to create a dataset for analysis.

* Merged payments with customers:

Merged the payments table with the customers table using **customerNumber** to get **salesRepEmployeeNumber** in the payments table.A screenshot of a computer

Description automatically generated

* Merged payments with employees:

Merged the updated payments table with employees using **salesRepEmployeeNumber** to get **officeCode** field from the employees table.

A screenshot of a computer

Description automatically generated

* Merge payments with offices:

Merged the updated payments table using officeCode with the offices table to get country from the offices table.

A screenshot of a computer

Description automatically generated

2.2. Calculating Total Revenue

Calculated the total revenue for each order and aggregated it by office, product line, and region.

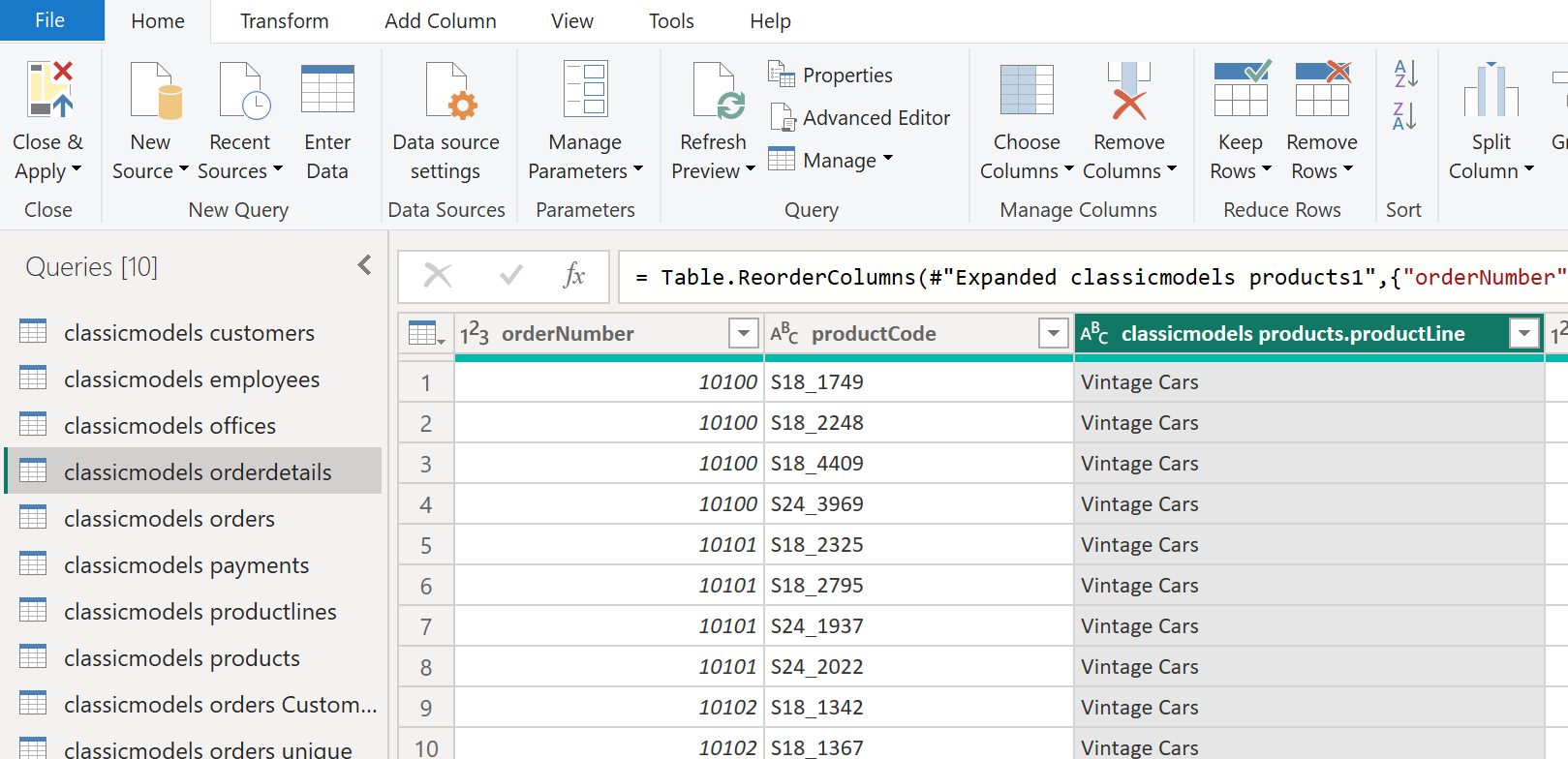
* Created a custom column in the orderdetails table to calculate revenue per order line:

Formula: Total Revenue(Sales) = quantityOrdered \* priceEach.

A screenshot of a computer

Description automatically generated

* Merged orderdetails with products using productCode to get productLine.



* Created report for revenue by offices (country) and revenue by productlines (productLine) using measure Shippedorders

Shippedorders = calculate([TotalRevenue(Sales)],

          'classicmodels orders'[status]="Shipped")

Please Note: The variable “amount” in the table “payments” provides the revenue data for the orders that have been completed. This variable needs to be used to ensure that only completed orders are considered in revenue calculations. However, the table “payments” doesn’t have order number that provides the relationship between amount paid and the product line. Hence, the orders with “status” as “Shipped” have been considered as completed orders for the above report after ensuring that the shipped orders revenue is approximately matches as the total amount paid.

2.3. Grouping and Aggregating Data

1. Summarized data to analyse office performance.
   * Group by Office:

Created a table “classicmodels orders unique” by duplicating the orders table and grouped the data by officeCode to calculate:

Total Orders: Count of rows.

Unique Customers: Count of distinct customerNumber.A screenshot of a computer

Description automatically generated

1. Summarized data to analyse Customer segment purchase frequency.
   * Created a table “classicmodels Purchase Frequency” by duplicating the orders table.
   * Created year, Month columns by using Date functions on orderDate.
   * Created Order Quarter Continuous by using Date.StartOfQuarter([orderDate]).
   * Grouped the data by customerNumber, year, Month, to calculate:
   * Purchase count: Count of rows.
   * Merged the table with customers table using customerNumber to get country.

A screenshot of a computer

Description automatically generated

1. **Closing and Applying Changes**

After completing the data transformations, all changes were applied by selecting Close & Apply in Power Query. This loaded the prepared data into Power BI for visualization and analysis.