Axon Retailer Sales Analysis

(Odin School – PowerBI Capstone Project)

Prepared By:

Name: Neha Koti

Batch: DS227B

Student ID: S3554

Contents

Axon Retailer Sales Analysis

Business Problem: Sales Analysis of "Axon company"

Axon is a small company, which is a retailer selling classic cars, is facing issues in managing and analyzing their sales data. We need to manage and analyze their sales data and create sales reports for better decision-making. Given dataset named "Classicmodels" contains 8 Tables. This dataset contains information about Customers, Employees, Products, Order details. We need to use Business Intelligence (BI) tools like Microsoft PowerBI and SQL for this project.

Proposal:

Axon company has given their past three years of data which contains information regarding their orders, sales, customers, employees and offices. We need to analyze this data and retrieve meaningful insights from it. First, we need to load these files into MySQL and using few queries we can retrieve some meaningful insights. Later we load this sql file into PowerBI and do data cleaning and data transformation. And then create interactive Sales dashboards in PowerBI and retrieve meaningful insights and sales reports through which Axon company can make better decision-making.

Tools Required: MySQL, PowerBI

Proposal for Sales Analysis of "Axon company"

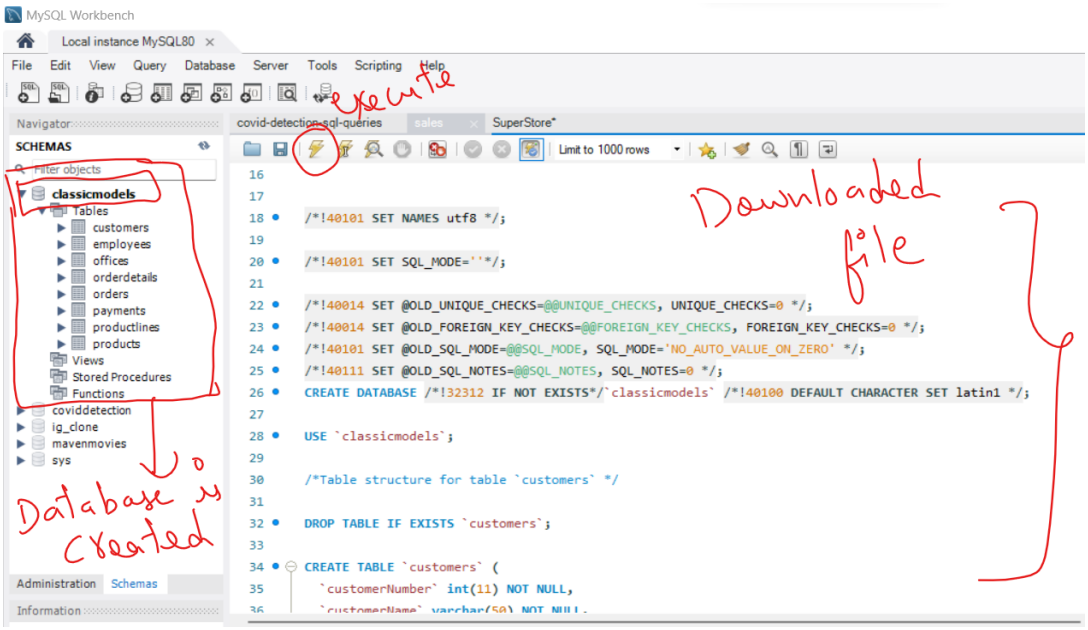
Approach/ Implementation:

Step 1: Creating a new Database in MySQL

We need to first create a Database called "Classicmodels" in MySQL. Download a file from the below link:

https://drive.google.com/file/d/1OB_iGw6vVS5KS7QwiwVChbeTfR4WvUy3/view?usp=share_link

Load this downloaded file into MySQL and execute this file. A new database called "Classicmodels" will be created in MySQL.



Step 2: Data Exploration using sql queries

In this step we try to understand the given data. We have 8 Tables namely:

- **Customers**: stores customer's data (13 columns, **customerNumber** -> primary key)
- **Products**: stores a list of scale model cars (9 columns, **productCode** -> Primary key
- **ProductLines**: stores a list of product line categories (4 columns, **productLine** -> Primary key)
- **Orders:** stores sales orders placed by customers (7 columns, **orderNumber, productCode**-> Primary keys)
- **OrderDetails**: stores sales order line items for each sales order (5 columns, **orderNumber**-> Primary key)
- **Payments:** stores payments made by customers based on their accounts (4 columns, **customerNumber, checkNumber** -> Primary keys)
- **Employees:** stores all employee information as well as the organization structure such as who reports to whom (8 columns, **employeeNumber** -> primary key)
- **Offices:** stores sales office data (9 columns, **officeCode** -> Primary key)

i) SQL Queries

Task 1: Total No. of Unique Customers

```
11      -- Total no.of Unique customers
12 •    SELECT COUNT(DISTINCT customerNumber) AS Unique_Customers FROM customers;
13
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Unique_Customers
122

Task 2: Total No. of Products

```
14      -- Total no.of products
15 •    SELECT COUNT(DISTINCT productCode) AS total_products FROM products;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_products
110

Task 3: Total No. of orders

```
17      -- Total no.of orders
18 •    SELECT COUNT(DISTINCT orderNumber) AS total_orders FROM orders;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_orders
326

Task 4: Total Quantity Ordered

```
23      -- Total Quantity ordered
24 •    SELECT SUM(quantityOrdered) AS Total_Quantity FROM orderdetails;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Total_Quantity
105516

Task 5: Total cost price, selling price, profit

```sql
-- Total profit
SELECT
    selling_price,
    cost_price,
    (selling_price - cost_price) AS Profit
FROM (
    SELECT
        SUM(quantityordered * priceEach) AS selling_price,
        SUM(quantityordered * buyprice) AS cost_price
    FROM orderdetails
    INNER JOIN products ON products.productcode=orderdetails.productCode
)Profit;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

selling_price	cost_price	Profit
9604190.61	5778310.36	3825880.25

Task 6: Total Sales by Product line

```sql
-- Total sales by product line
SELECT productLine,
SUM(quantityordered * priceEach) AS Total_Sales
FROM orderdetails od
JOIN products p ON od.productCode = p.productCode
GROUP BY productLine
ORDER BY Total_Sales DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

productLine	Total_Sales
Classic Cars	3853922.49
Vintage Cars	1797559.63
Motorcycles	1121426.12
Trucks and Buses	1024113.57
Planes	954637.54
Ships	663998.34
Trains	188532.92

Task 7: Total Profit by Product line

```sql
-- Profit by Product Line
SELECT productLine,
SUM(quantityordered * priceEach)-SUM(quantityordered * buyprice) as Profit
FROM orderdetails od
JOIN products p ON od.productCode = p.productCode
GROUP BY productLine
ORDER BY Profit DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

productLine	Profit
Classic Cars	1526212.20
Vintage Cars	737268.33
Motorcycles	469255.30
Trucks and Buses	400553.22
Planes	365960.71
Ships	261289.47
Trains	65341.02

Task 8: Top 5 order count by Year and Month

```sql
-- Top 5 Order count by Year and Month
SELECT LEFT(orderDate, 7) AS Year_Mon,
COUNT(DISTINCT ordl.orderNumber) AS Distinct_Orders
FROM orders ord
JOIN orderdetails ordl ON ord.orderNumber=ordl.orderNumber
GROUP BY Year_Mon
ORDER BY Distinct_Orders DESC
LIMIT 7;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Year_Mon	Distinct_Orders
2004-11	33
2003-11	30
2003-10	18
2005-05	15
2004-12	13
2005-03	13
2004-10	13

Task 9: Top 5 Sales by Year and Month

```sql
-- Top 5 Total Sales by Year and Month
SELECT LEFT(orderDate, 7) AS Year_Mon,
SUM(quantityordered * priceEach) AS Total_Sales
FROM orders ord
JOIN orderdetails ordl ON ord.orderNumber=ordl.orderNumber
GROUP BY Year_Mon
ORDER BY Total_Sales DESC
LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Year_Mon	Total_Sales
2003-11	988025.15
2004-11	979291.98
2003-10	514336.21
2004-10	500233.86
2005-05	441474.94

Task 10: Top 5 customers by Sales

```sql
-- Top 5 Customers by Total Sales
SELECT customerName,
SUM(quantityordered * priceEach) AS Total_Sales
FROM orders ord
JOIN orderdetails ordl ON ord.orderNumber=ordl.orderNumber
JOIN customers c ON ord.customerNumber=c.customerNumber
GROUP BY customerName
ORDER BY Total_Sales DESC
LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customerName	Total_Sales
Euro+ Shopping Channel	820689.54
Mini Gifts Distributors Ltd.	591827.34
Australian Collectors, Co.	180585.07
Muscle Machine Inc	177913.95
La Rochelle Gifts	158573.12

Task 11: Bottom 5 customers by Sales

```
90    SELECT customerName,
91    SUM(quantityordered * priceEach) AS Total_Sales
92    FROM orders ord
93    JOIN orderdetails ordl ON ord.orderNumber=ordl.orderNumber
94    JOIN customers c ON ord.customerNumber=c.customerNumber
95    GROUP BY customerName
96    ORDER BY Total_Sales
97    LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customerName	Total_Sales
Boards & Toys Co.	7918.60
Auto-Moto Classics Inc.	21554.26
Atelier graphique	22314.36
Frau da Collezione	25358.32
Royale Belge	29217.18

Task 12: Top 5 customers by Orders

```
99    -- Top 5 Customers by orders
100   SELECT customerName,
101   count(DISTINCT orderNumber) AS Total_orders
102   FROM orders o
103   JOIN customers c ON o.customerNumber=c.customerNumber
104   GROUP BY customerName
105   ORDER BY Total_orders DESC
106   LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

customerName	Total_orders
Euro+ Shopping Channel	26
Mini Gifts Distributors Ltd.	17
Danish Wholesale Imports	5
Australian Collectors, Co.	5
Down Under Souveniers, Inc	5

Task 13: Bottom 5 customers by Orders

```sql
-- Bottom 5 Customers by orders
SELECT customerName,
count(DISTINCT orderNumber) AS Total_orders
FROM orders o
JOIN customers c ON o.customerNumber=c.customerNumber
GROUP BY customerName
ORDER BY Total_orders
LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customerName	Total_orders
Bavarian Collectables Imports, Co.	1
Amica Models & Co.	2
Vida Sport, Ltd	2
Norway Gifts By Mail, Co.	2
Mini Classics	2

Task 14: Top 5 products by Sales

```sql
-- Top 5 Products by Total Sales
SELECT productName,
SUM(quantityordered * priceEach) AS Total_Sales
FROM orders ord
JOIN orderdetails ordl ON ord.orderNumber=ordl.orderNumber
JOIN customers c ON ord.customerNumber=c.customerNumber
JOIN products p ON ordl.productCode=p.productCode
GROUP BY productName
ORDER BY Total_Sales DESC
LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

productName	Total_Sales
1992 Ferrari 360 Spider red	276839.98
2001 Ferrari Enzo	190755.86
1952 Alpine Renault 1300	190017.96
2003 Harley-Davidson Eagle Drag Bike	170686.00
1968 Ford Mustang	161531.48

Task 15: Bottom 5 products by Sales

```
128     -- Bottom 5 Products by Total Sales
129 •   SELECT productName,
130     SUM(quantityordered * priceEach) AS Total_Sales
131     FROM orders ord
132     JOIN orderdetails ordl ON ord.orderNumber=ordl.orderNumber
133     JOIN customers c ON ord.customerNumber=c.customerNumber
134     JOIN products p ON ordl.productCode=p.productCode
135     GROUP BY productName
136     ORDER BY Total_Sales
137     LIMIT 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

productName	Total_Sales
1939 Chevrolet Deluxe Coupe	28052.94
1936 Mercedes Benz 500k Roadster	29763.39
1982 Lamborghini Diablo	30972.87
1958 Chevy Corvette Limited Edition	31627.96
1982 Ducati 996 R	33268.76

Task 16: Top 10 Employees by order count

```
143 •   SELECT CONCAT(e.firstName, ' ', e.lastName) AS Employee_Name,
144     COUNT(ordl.orderNumber) AS Order_Count
145     FROM employees e
146     JOIN customers c ON e.employeeNumber=c.employeeNumber
147     JOIN orders ord ON c.customerNumber=ord.customerNumber
148     JOIN orderdetails ordl ON ord.orderNumber=ordl.orderNumber
149     GROUP BY Employee_Name
150     ORDER BY Order_Count DESC
151     LIMIT 10;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch row

Employee_Name	Order_Count
Larry Bott	236
Barry Jones	220
George Vanauf	211
Andy Fixter	185
Peter Marsh	185
Loui Bondur	177
Steve Patterson	152

Task 17: Total Sales, Quantity ordered, cost price ordered by each country

```
153     -- Total Sales, Actual Sales, Quantity ordered by Country
154 •   SELECT country,
155     COUNT(o.orderNumber) AS Quantity_Ordered,
156     SUM(quantityordered * priceEach) AS Total_Sales,
157     SUM(quantityordered * buyprice) AS Actual_Sales
158     FROM orderdetails o
159     JOIN products p ON o.productCode=p.productCode
160     JOIN orders ord ON o.orderNumber=ord.orderNumber
161     JOIN customers c ON ord.customerNumber=c.customerNumber
162     GROUP BY country
163     ORDER BY Quantity_Ordered DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

country	Quantity_Ordered	Total_Sales	Actual_Sales
USA	1004	3273280.05	1964464.46
Spain	342	1099389.09	659384.55
France	314	1007374.02	594357.90
Australia	185	562582.59	340375.41
New Zealand	149	476847.01	287340.43
UK	144	436947.44	263983.18
Italy	121	360616.81	216156.95
Finland	92	295149.35	177910.02

Step 3: Loading schema from MySQL to POWERBI

Open PowerBI Desktop and follow the below steps:

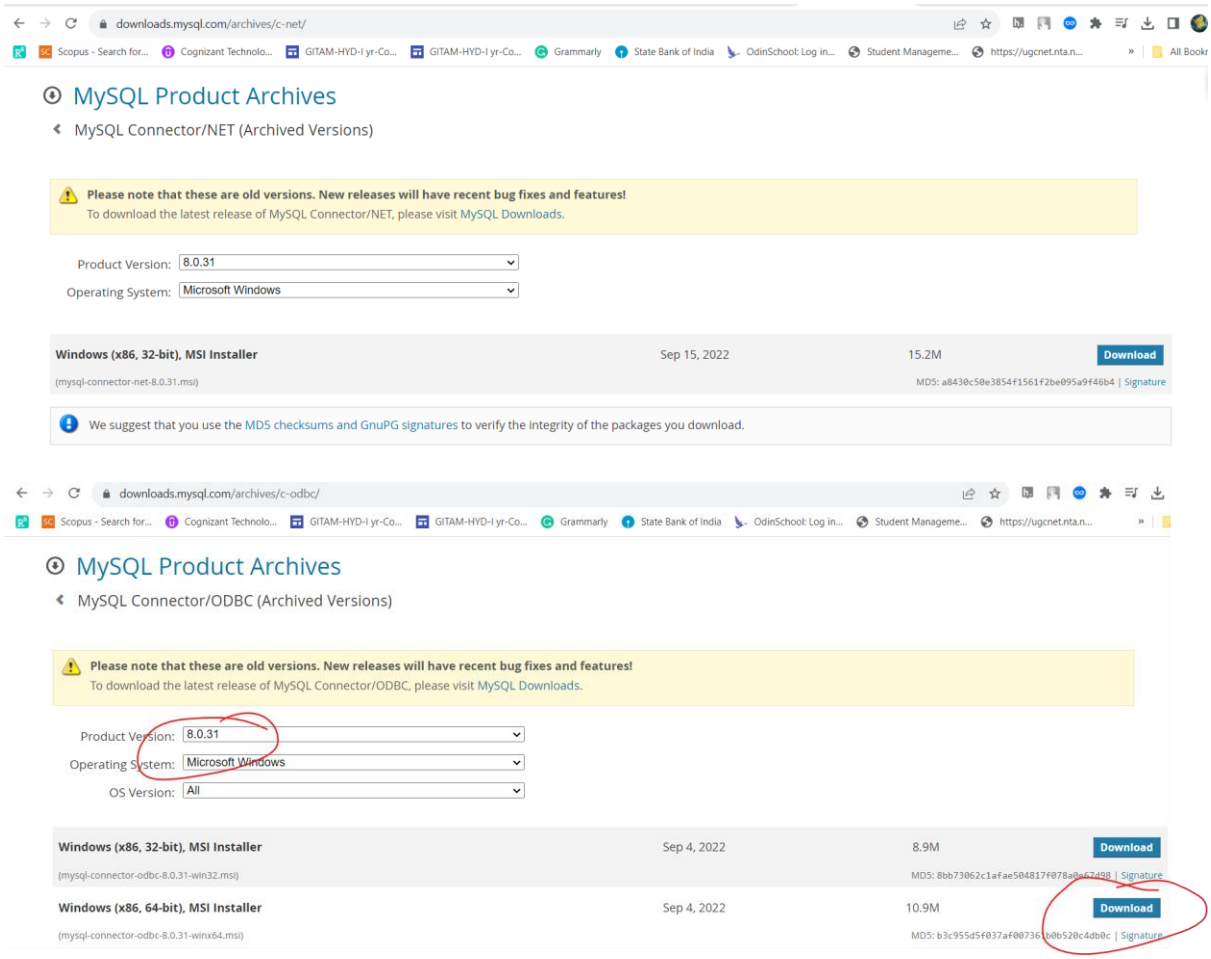
Powerbi ribbon -> Home --> get data -> more -> mysql database

For first time it gives error box like “connectors are missing install them”

Go to below links and install 8.0.31 version. ALWAYS .net connector and ODBC connector should be of SAME VERSION ONLY. Then only it works.

https://downloads.mysql.com/archives/c-net/

https://downloads.mysql.com/archives/c-odbc/

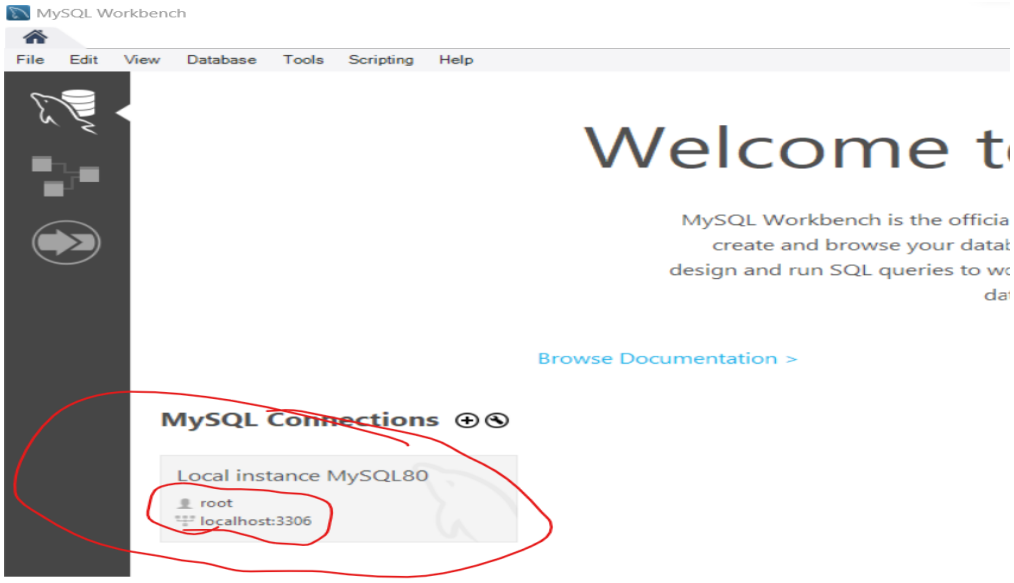


INSTALL THIS OLDER VERSION. Close powerbi refresh your system and open powerbi again.

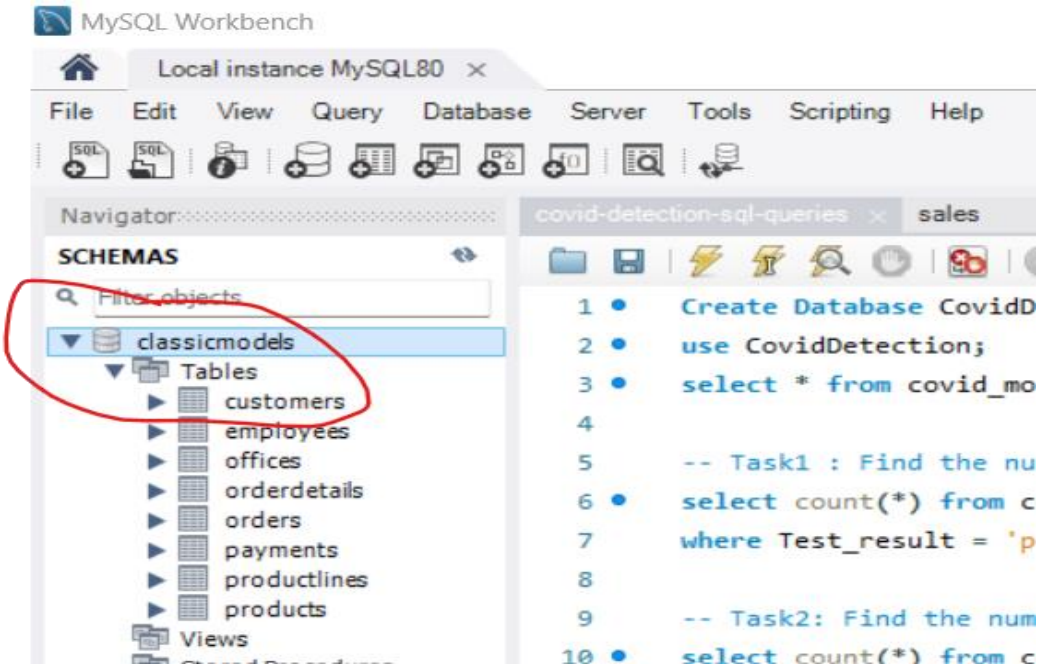
Home → get data → more → mysql database → connect

We get a new pop up box as below:

Now open mysql workbench. It is installed in localhost:3306



And we have the following databases:

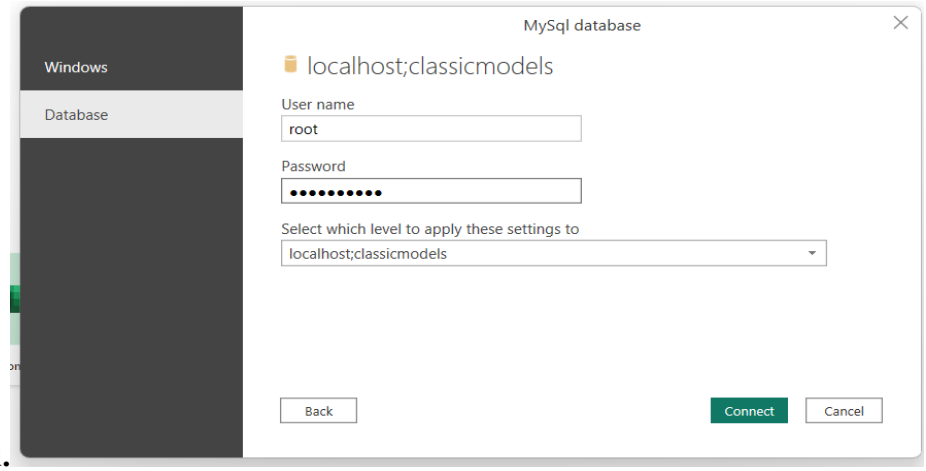


We want to load “classicmodels” database into powerbi.

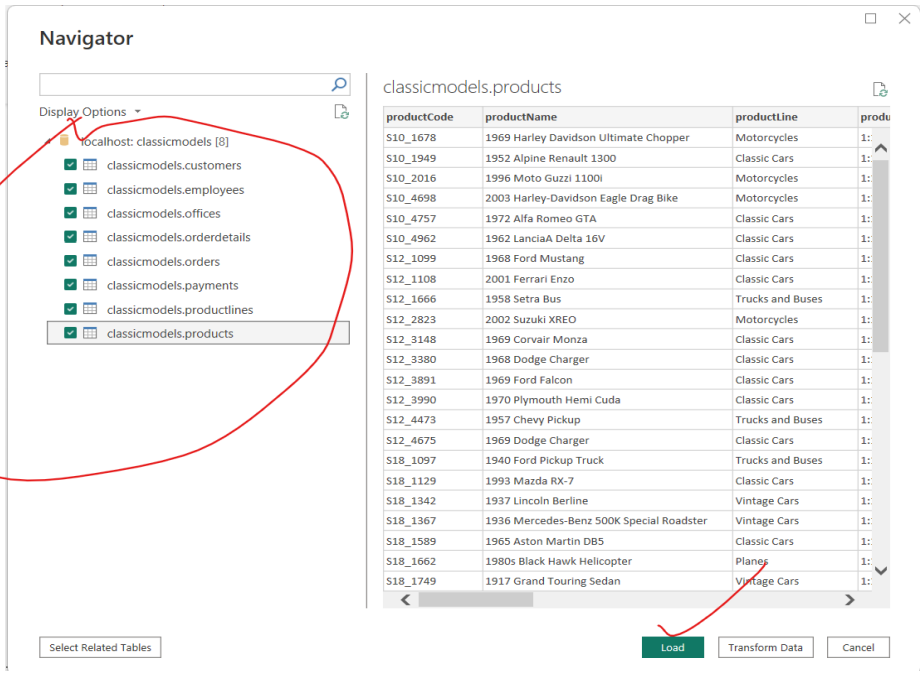
So now goto powerbi and give below details and click ok.

Or server: 127.00.1:3306, Database: “schema name” can also be given. Both works well.

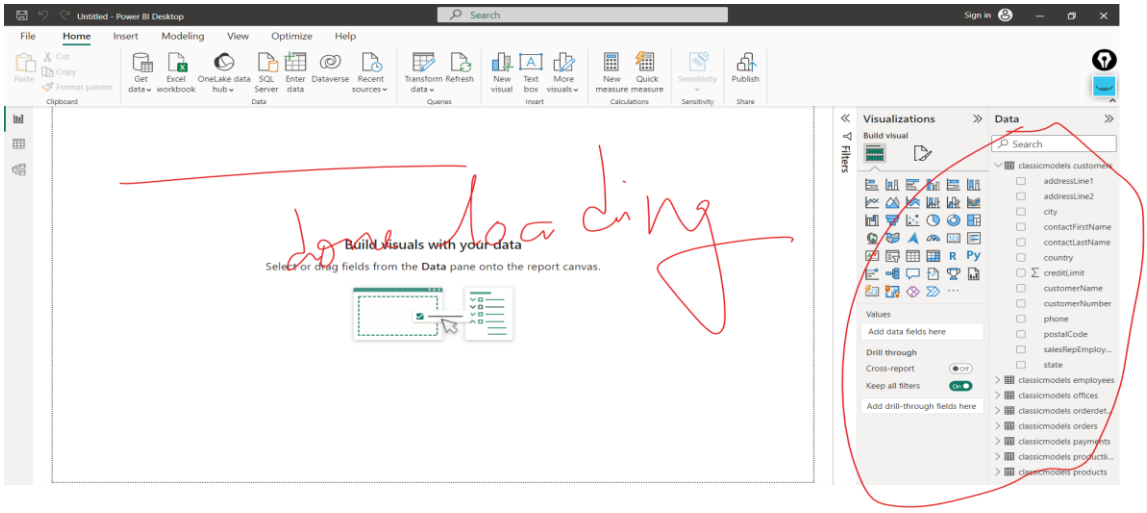
Give username password of your mysql which you have given during installation if asked.



Or else it opens a page like below. Tick mark/choose tables which you want to load into powerbi from MySQL database named “classicmodels”.



‘classicmodels’ schema is now loaded into powerbi.

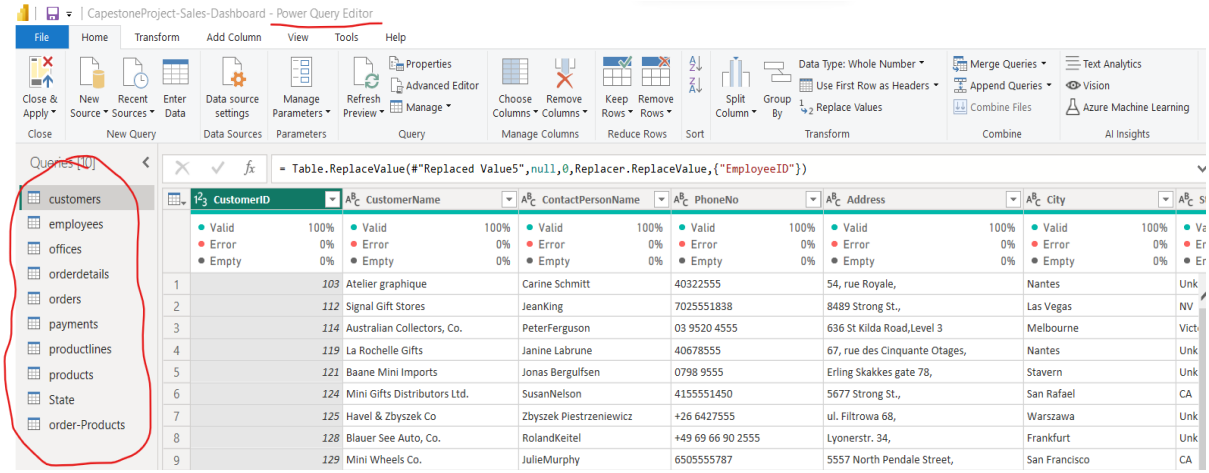


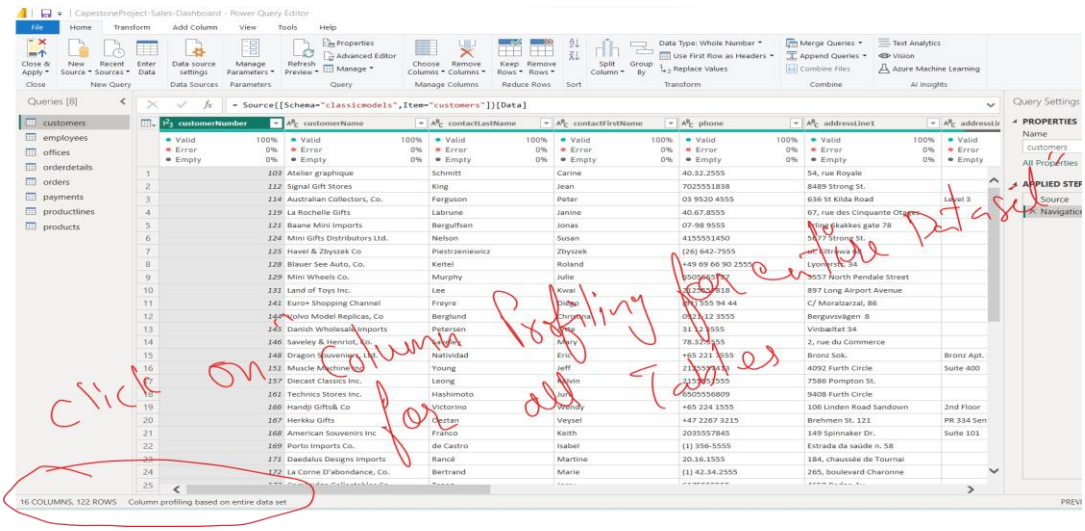
Step 4: Data Cleaning

We use power Query editor in POWERBI Desktop for Data Cleaning.

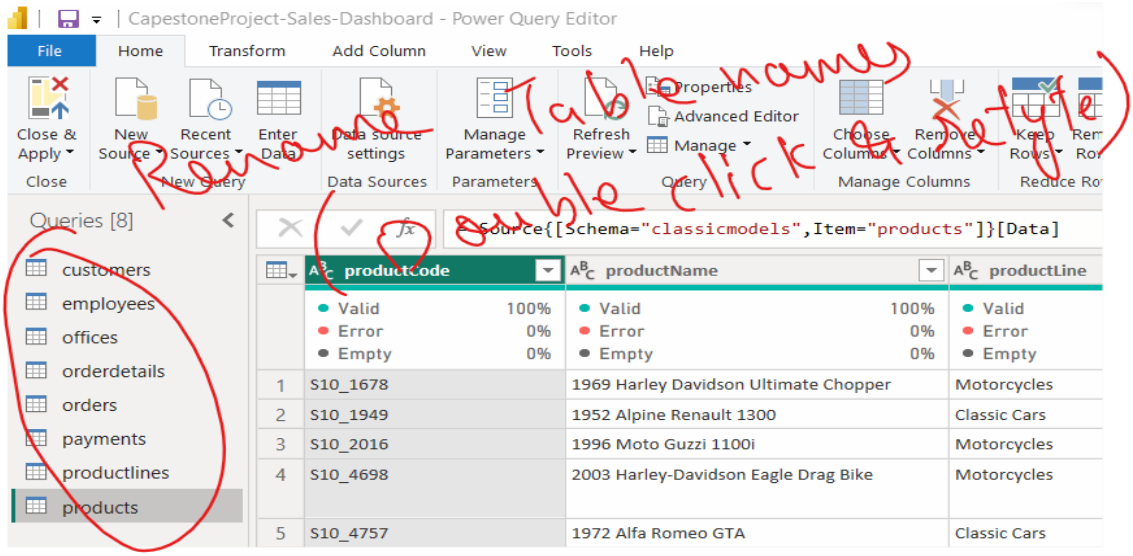
Home → Transform Data

A new power query editor window opens as below:





i) **Renaming Column Names:** We have 8 tables and each table has few columns. We can rename column names according to our requirement or business problem statement. Go to each table double click on the column which you want to rename and type the new name and press enter. Column name will be renamed.



Like this rename all the necessary columns from all tables. The new column names are:

- ✓ **Customers:** CustomerID, CustomerName, ContactFirstName, ContactLastName, PhoneNo, Address, City, State, PostalCode, Country, EmployeeID, CreditLimit
- ✓ **Employees:** EmployeeID, lastName, FirstName, Extension, Email, OfficeCode, MagerID, JobTitle.
- ✓ **Offices:** OfficeCode, City, PhoneNo, Address, State, Country, PostalCode, Territory
- ✓ **Orderdetails:** OrderID, ProductID, QuantityOrdered, UnitPrice, OrderLineNumber
- ✓ **Orders:** OrderID, OrderDate, RequiredDate, ShippedDate, OrderStatus, Comments, CustomerID
- ✓ **Payments:** CustomerID, CheckNumber, paymentDate, Amount
- ✓ **Productlines:** ProductLine, textDescription, htmlDescription, image
- ✓ **Products:** ProductID, ProductName, ProductLine, ProductScale, ProductVendor, ProductDescription, QuantityInStock, Costprice, MSRP(Sellingprice)
- ✓ **State:** OfficeCode, city, phoneNo, Address, State, country, PostalCode, Territory

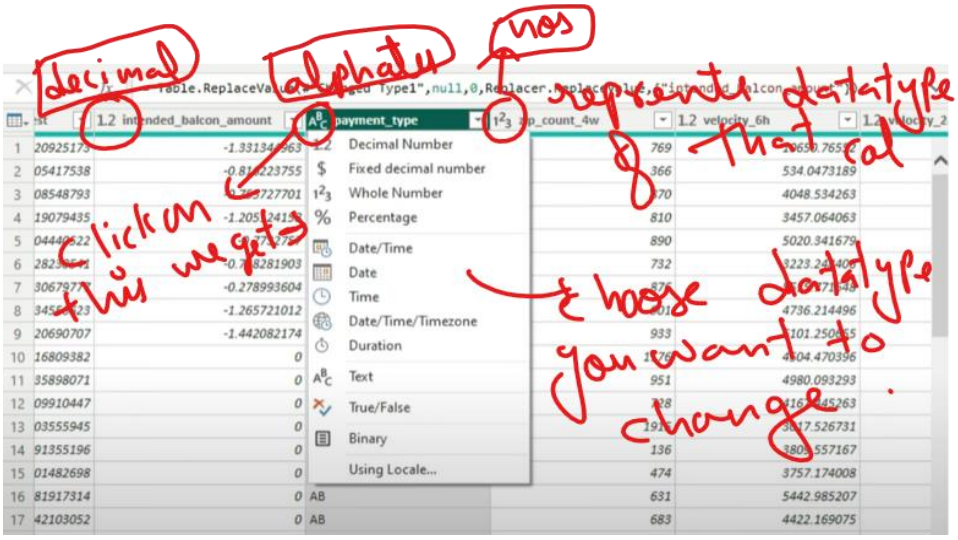
ii) **Removing Columns:** If we have any unnecessary columns we can remove them from the tables. Select any column right click and select “remove”.

From Productlines table we removed htmlDescrption and image columns as they are not needed for any sales analysis.

iii) **Merging Columns:** We can merge/ combine two columns if needed. Select two columns which you want to merge then right click → merge columns.
Give new column name and any separator if needed and click ok.
In customers Table: ContactFirstName, ContactLastName we merged into “ContactpersonName”.

In Employees table: lastname, Firstname columns are merged together as "EmployeeName" column.

iv) **Changing datatypes:** double click on the datatype in column name and choose the new datatype.

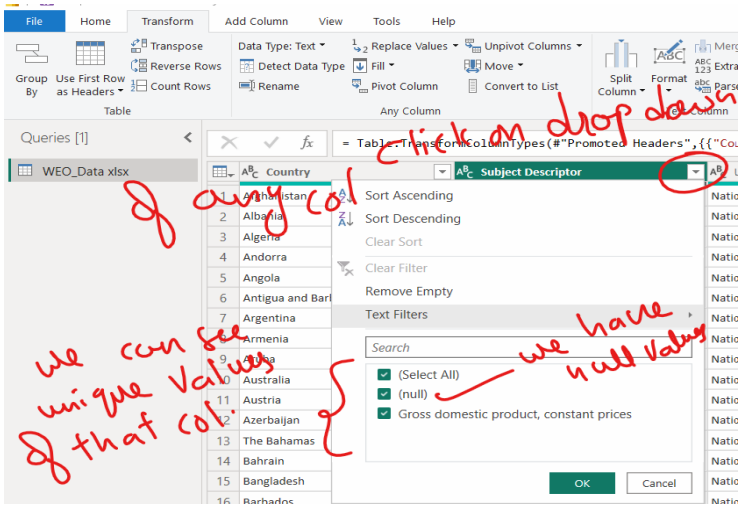


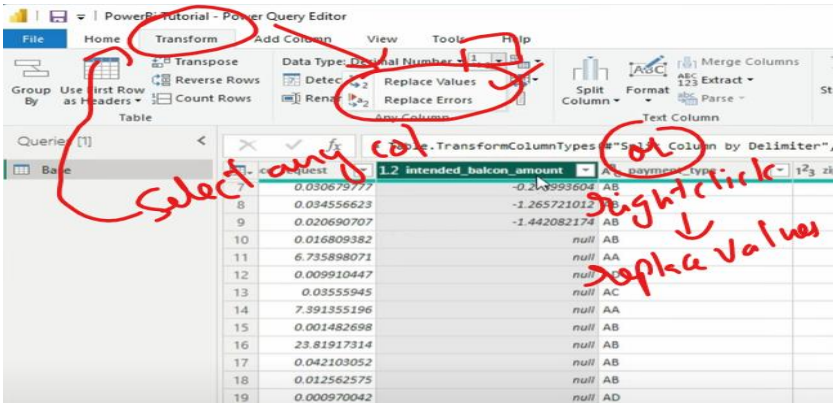
For date column always choose date datatype in powerbi format.

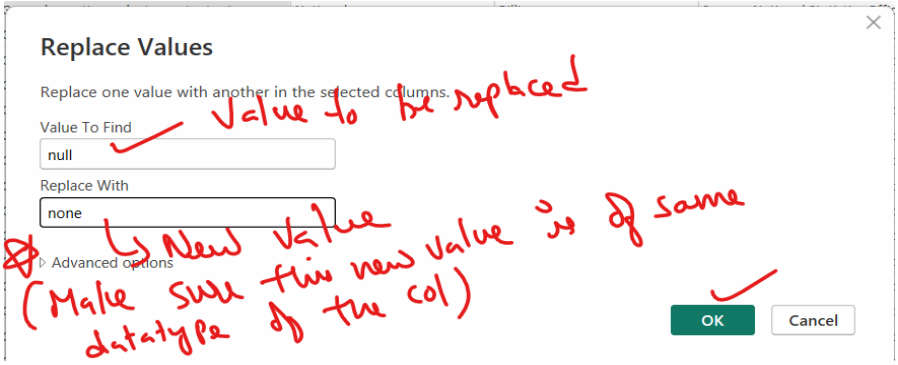
In customers table, "phoneNo" column is changed to text datatype.

v) **Replacing null & wrong values:**

select any column Transform →Replace values or right click on selected column → replace values





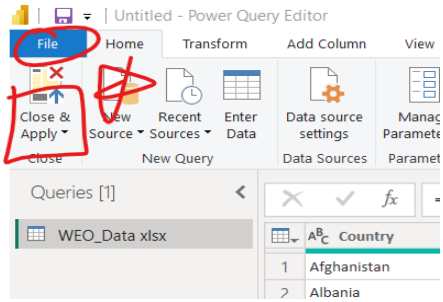


Check all unnecessary values like null, n/a etc and replace them appropriately according to the column data.

- In customers table: PhoneNo column has different number styles. All "(" are replaced with "+" and all ")" replaced with space.All "-" values are replaced with "" empty space. State and PostalCode columns have null values they are replaced with "unknown". EmployeeId column has null values replaced with "0".
- In Employees table: ManagerID column has null values replaced with "0".
- In offices table: State column null values are replaced with "Unknown".
- In orders table: Comments column null values are replaced with "Unknown"
- In State table: State column null values are replaced with "Unknown".

In Data cleaning process we can even split columns, append columns and filter columns. But these steps are not needed for the given Data.

After all cleaning is done make sure you click on power query editors File -> close and apply. Or else changes done will not be reflected in powerbi desktop.



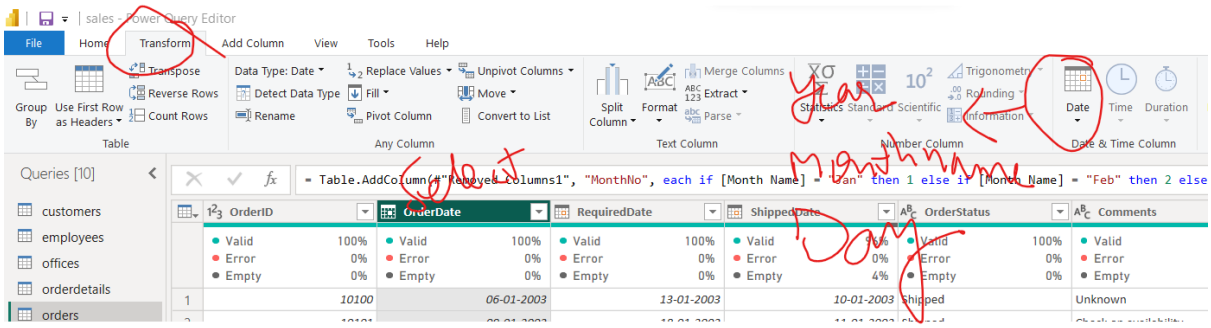
Step 5: Data Transformation

We need to create few measures and new columns using power query editor and dax formulas for further analysis of given data.

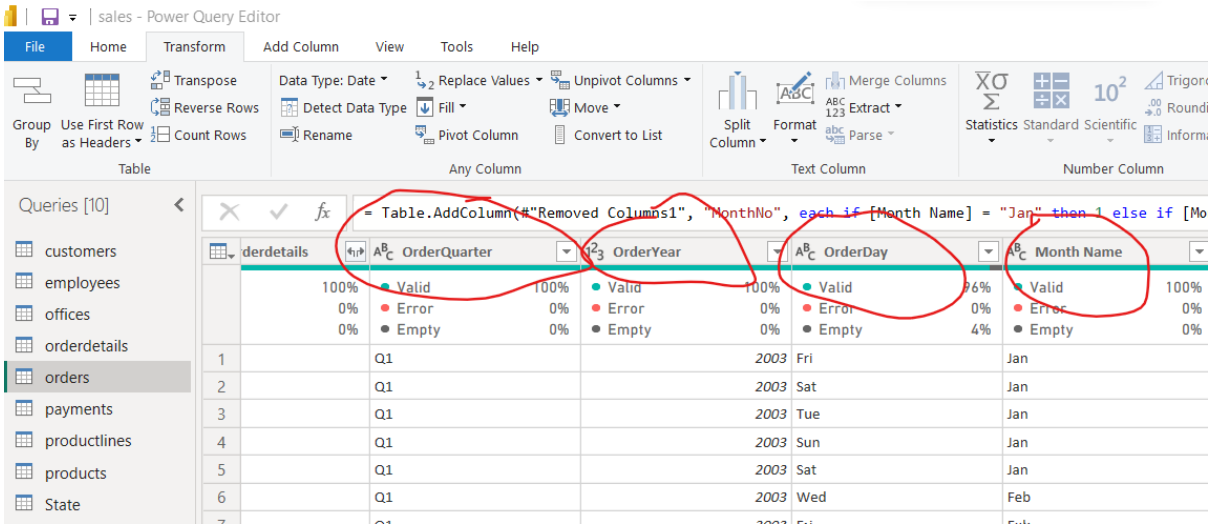
i) Creating new columns

- In "orders table" we have 'OrderDate' column. We need to extract day, month and year from this column separately. To do that follow as below:

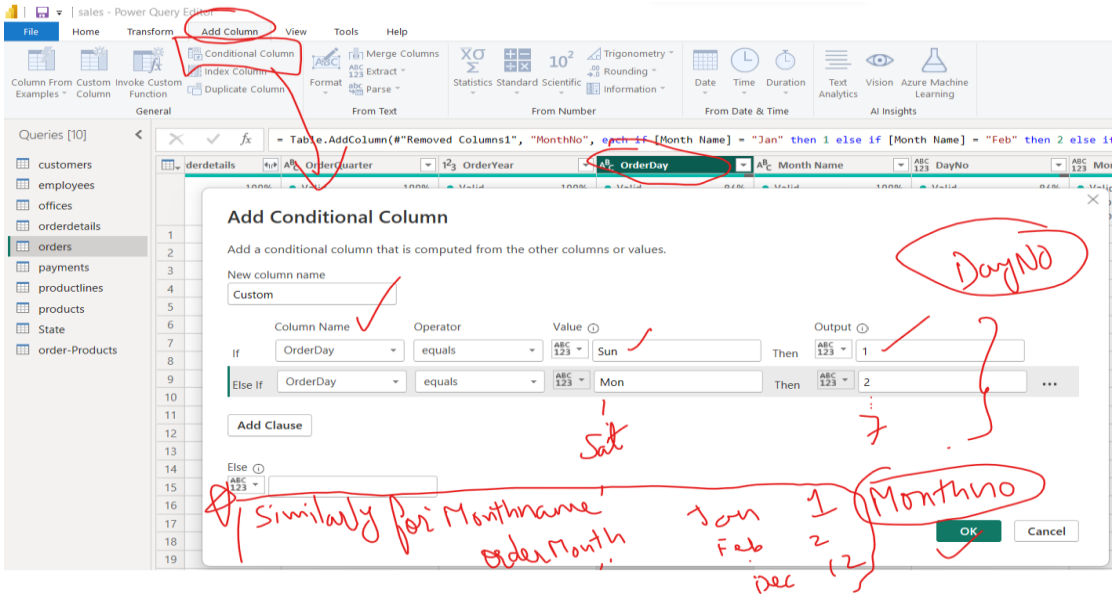
In power query editor: select OrderDate column → Transform → Date → Day/Month Name/ Year/Quarter

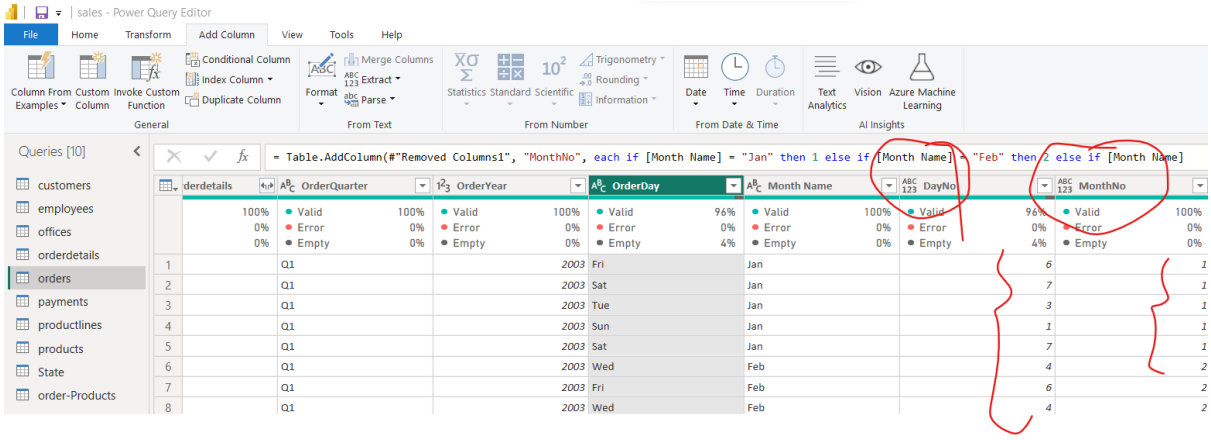


New columns will be generated after this as below. We replaced 1 as Q1, 2 as Q2, 3 as Q3 and 4 as Q4 in OrderQuarter column. Similarly, OrderDay and Month name are replaced as three letter days and month names.

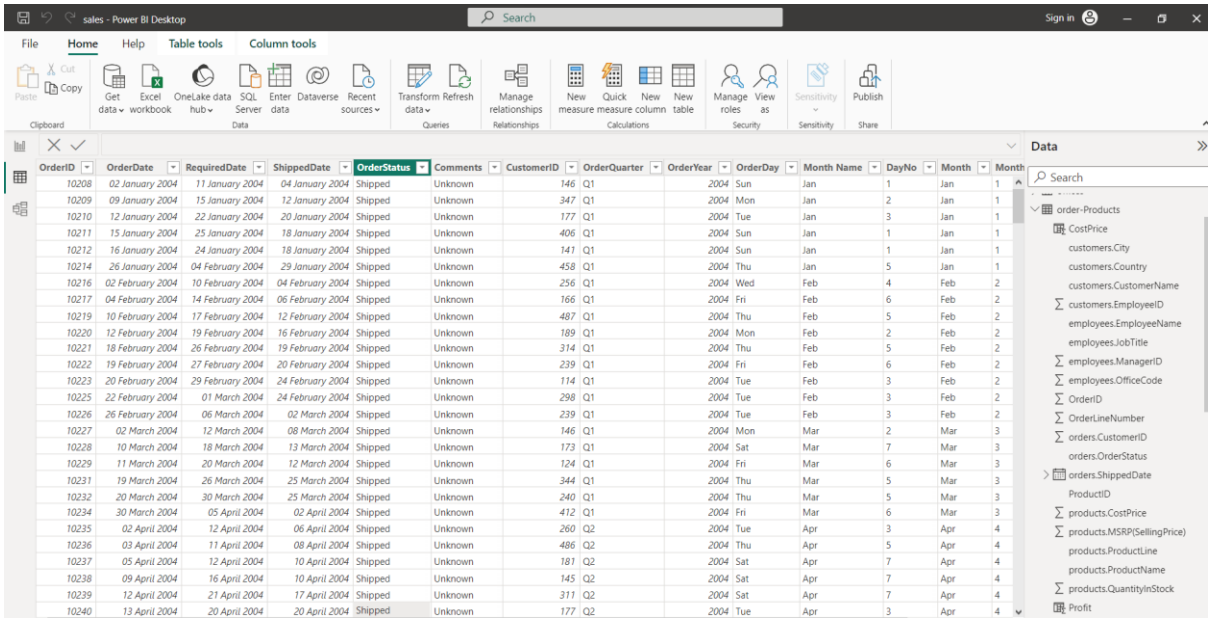


If we try to plot graphs for these above columns we will not get them in sorted order like Sun, Mon,.. etc or Jan, Feb etc. To make them in sorted order we need to create new columns as below:

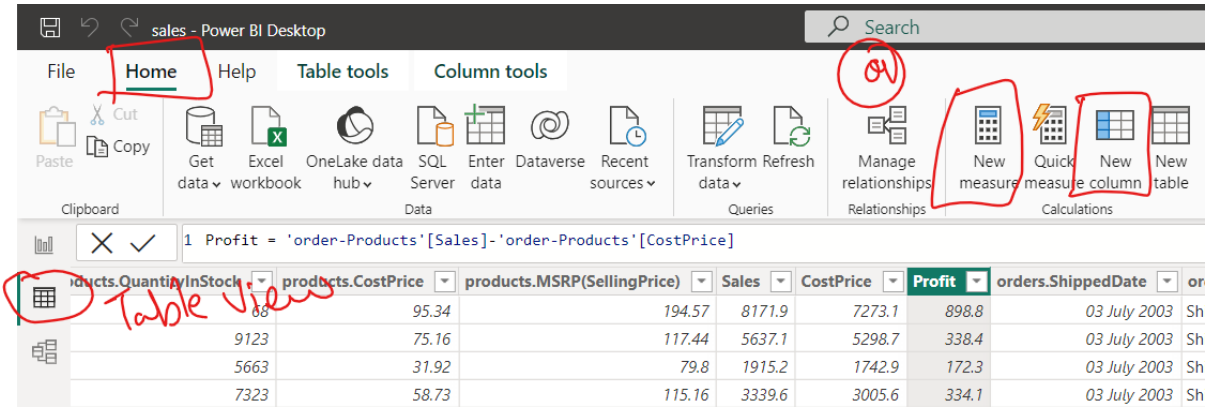




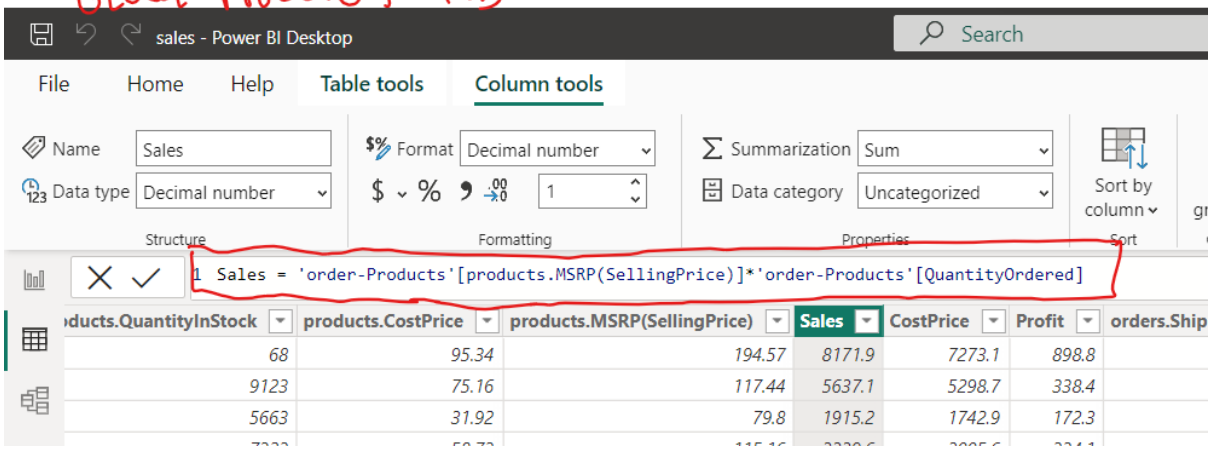
- Next, we created a new table **"order-products"** by merging orders, products, employees and customers tables together. Only required columns are kept in this new merged table.

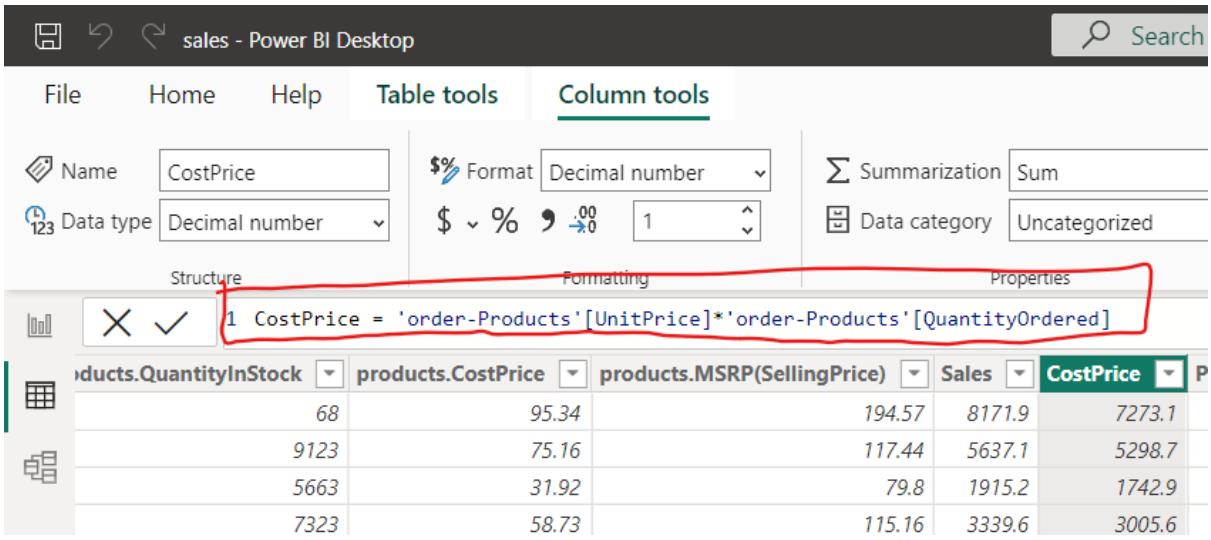


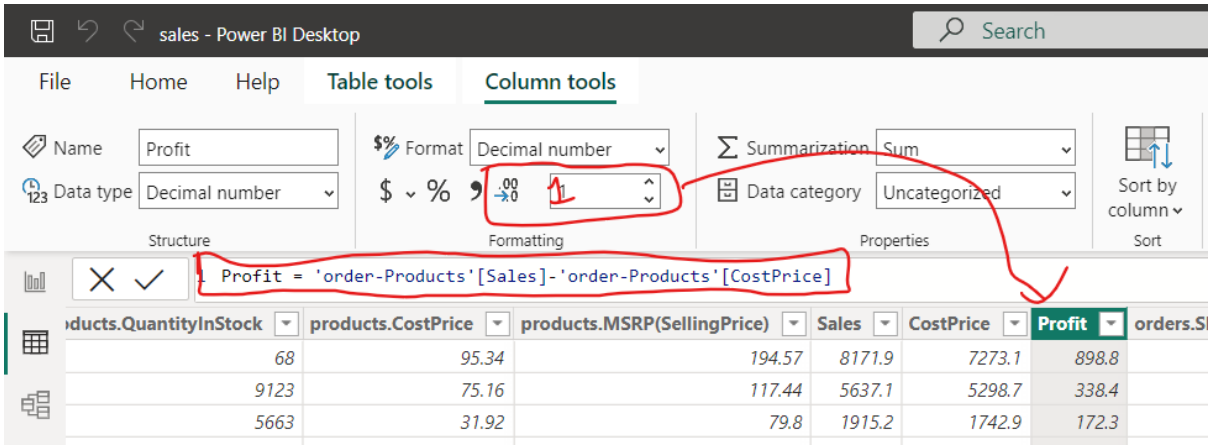
- **In "Order-Products" table we created few new columns using below Dax formulas:**



"order-Products Table"







ii) Creating New Measures:

To create a new measure do as below:

Table view → Home → New Measure:

```
1  Total Customers = DISTINCTCOUNT(customers[CustomerID])
```

```
Total Orders = DISTINCTCOUNT(orderdetails[OrderID])
```

```
Total products = DISTINCTCOUNT(products[ProductID])
```

```
Total Profit = SUM('order-Products'[Profit])
```

```
Total Sales = SUM('order-Products'[Sales])
```

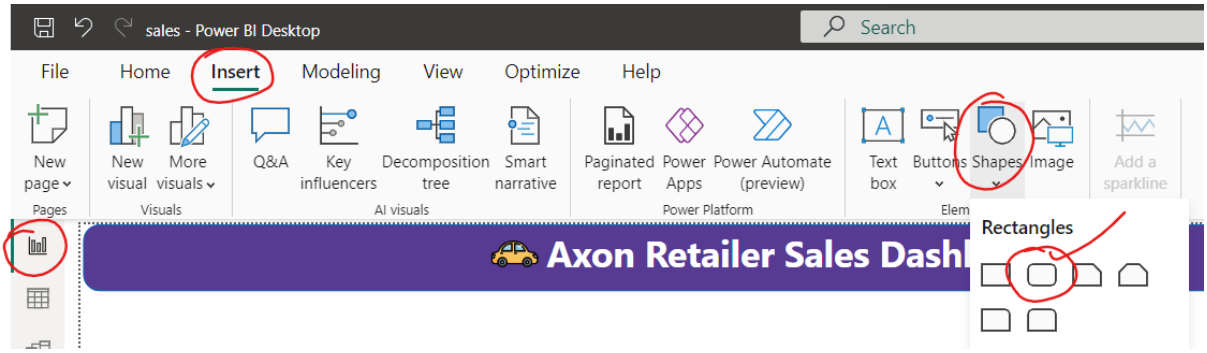
```
Total Revenue = SUM(payments[Amount])
```

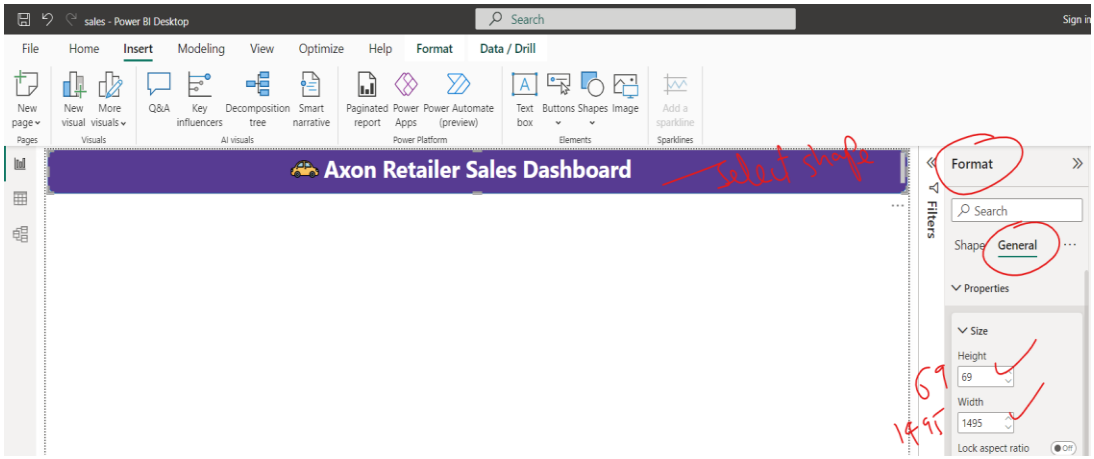
These new measures and columns will be used for creating interactive dashboards.

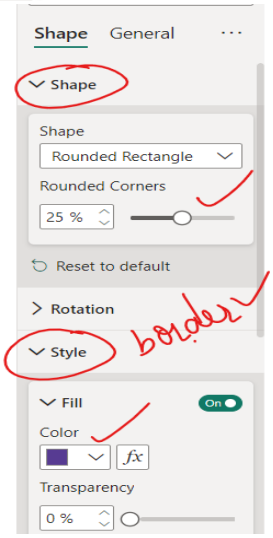
Step 6: Dashboard Creation

Creating Title bar:

Go to report view → insert → shapes → select rounded rectangle

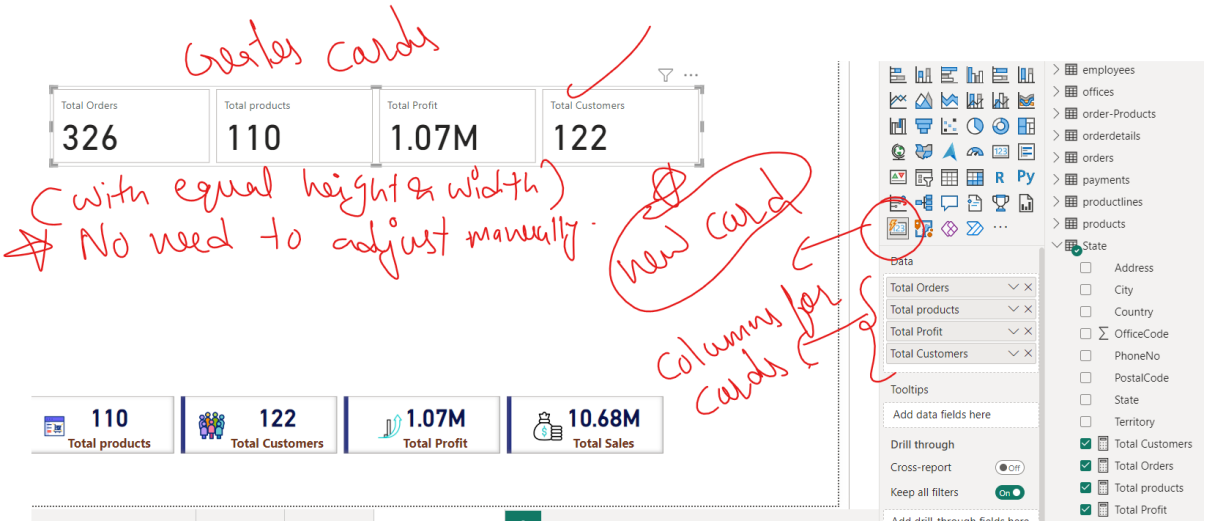




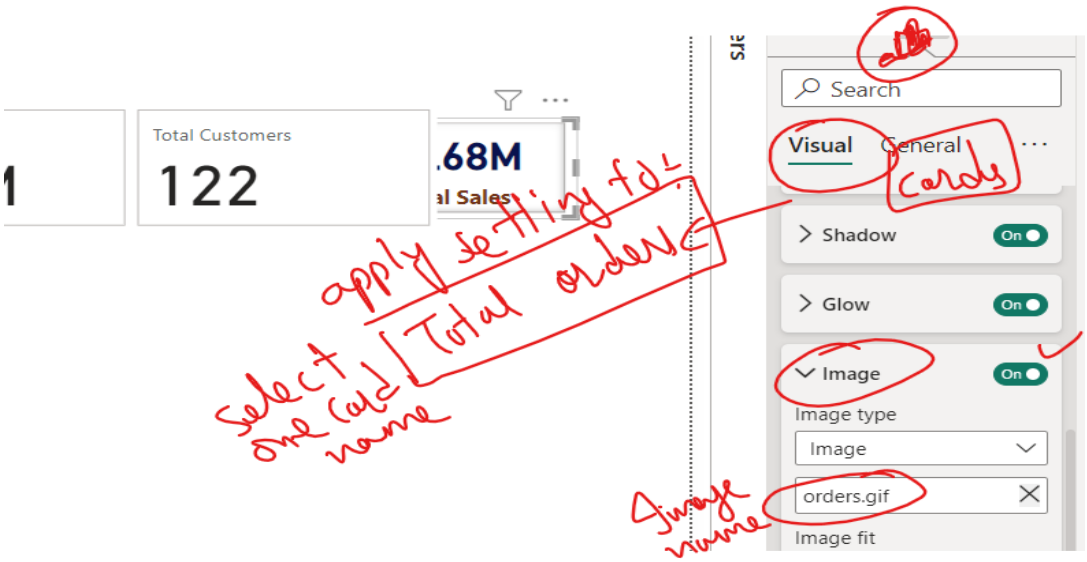


similarly insert a text box inside this shape and type title name.Insert slicers in title bar too (country, year slicers)

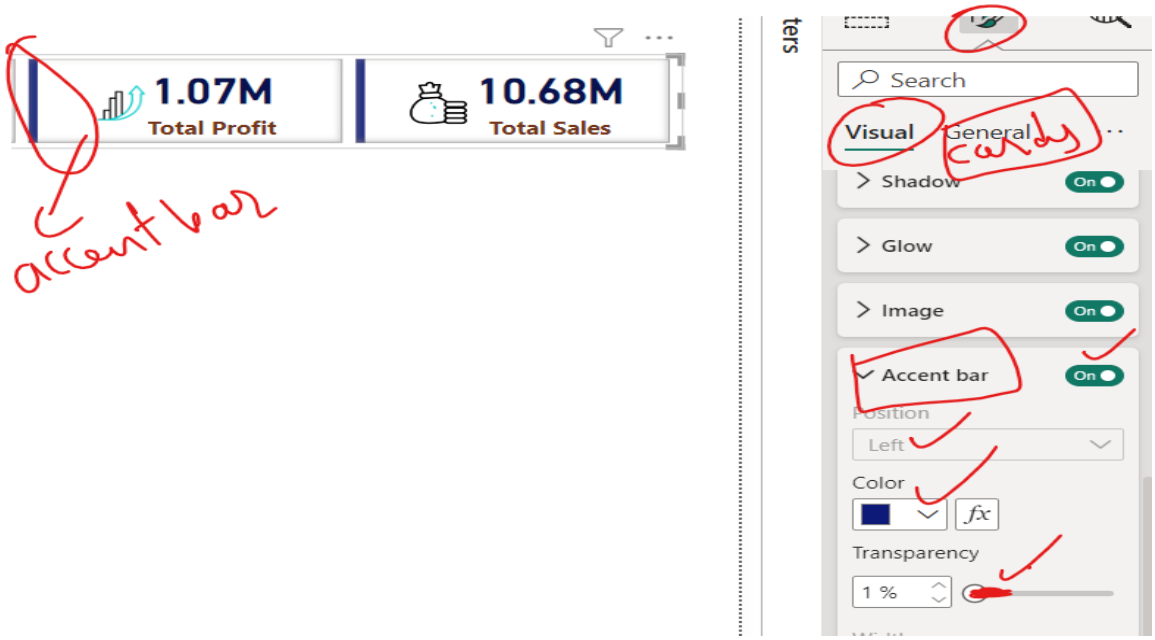
Creating cards:



Later format their color tittle color etc. We can insert images in cards as below:

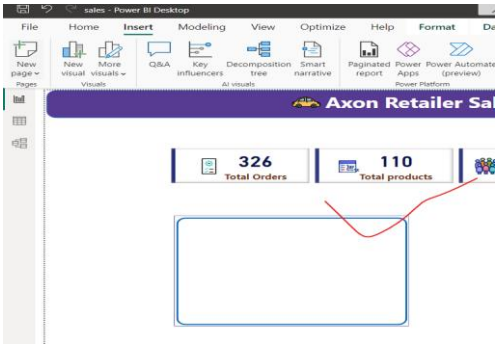


We can even insert accent bars as below in cards:

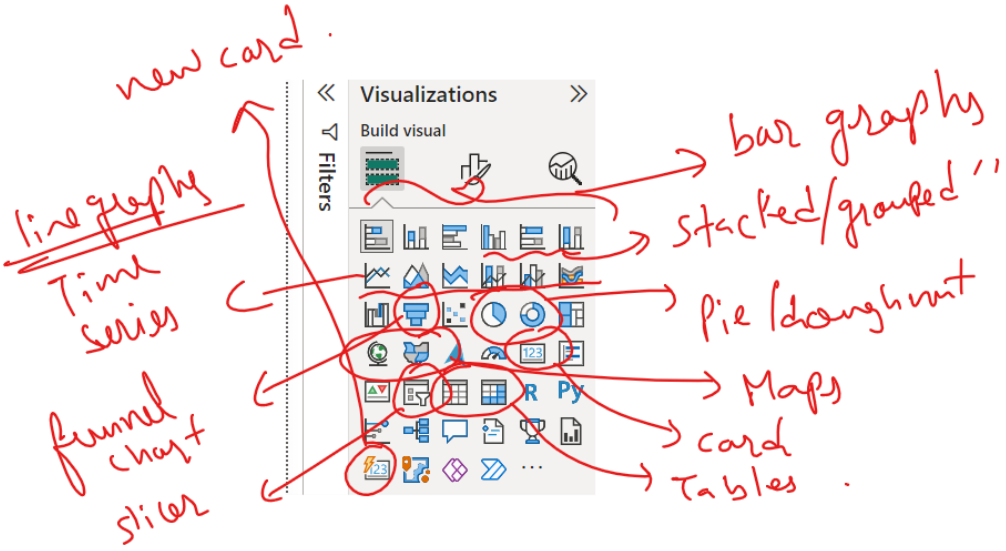


Next goto insert → shapes → rectangle:

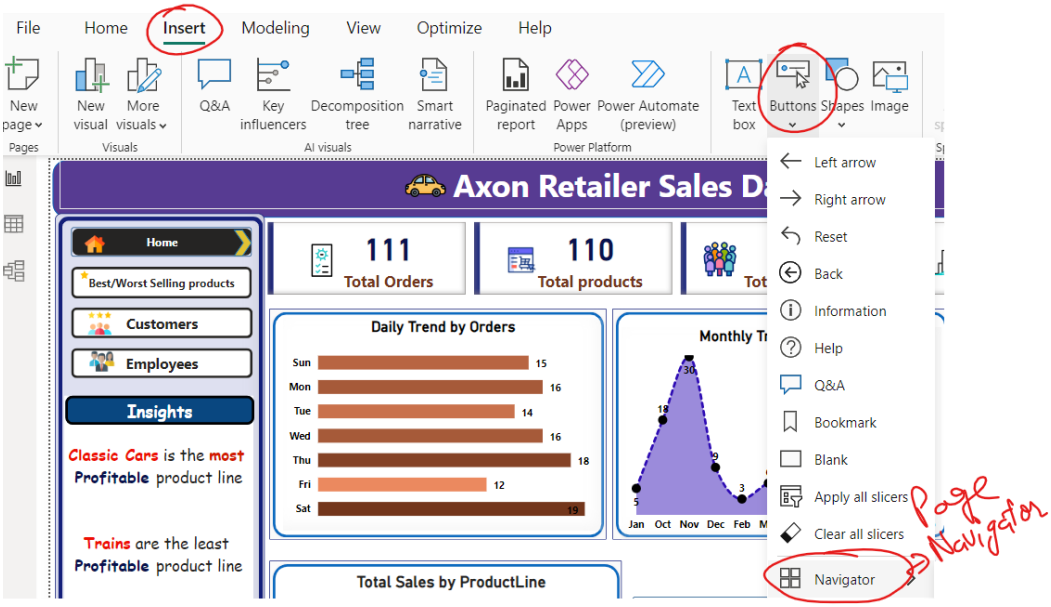
Shape: rounded rectangle, rounded corners: 6%, style: color: white, border:color-> blue, width:3



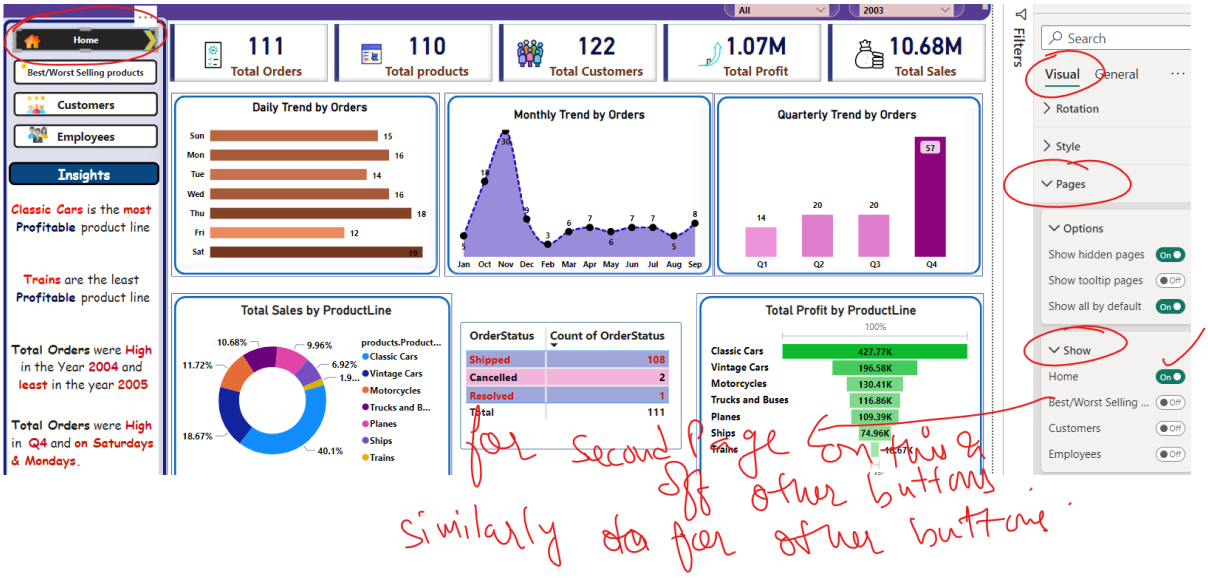
Now create graphs which are needed according to the requirement.



Creation of Navigation buttons

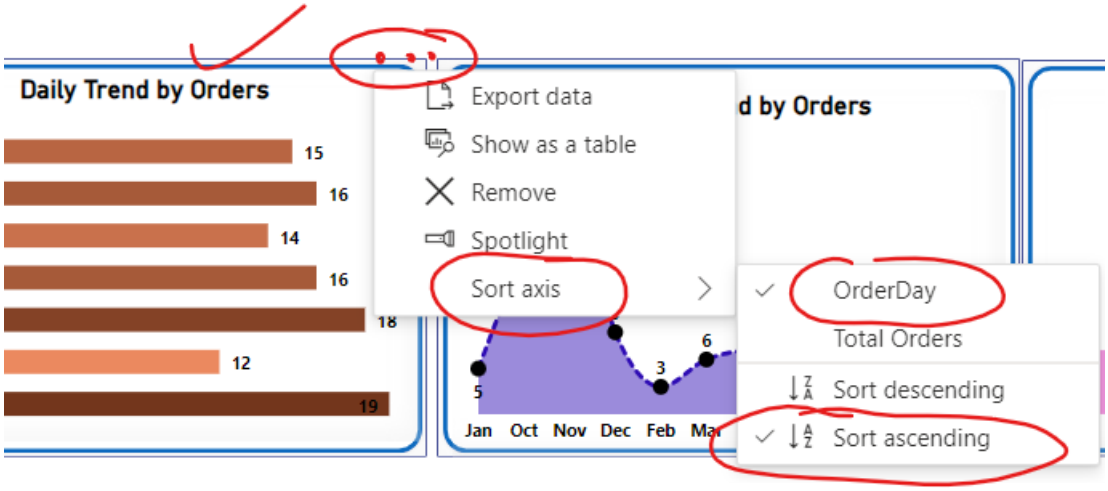


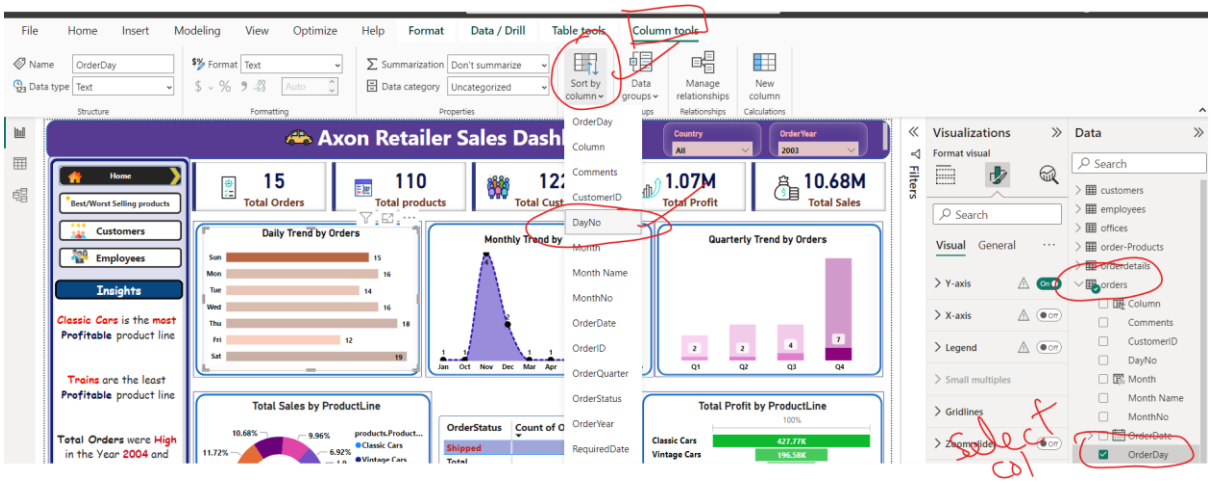
Say if we have 4 pages then 4 buttons are created automatically horizontally. Say now if we want these buttons vertically then delete 3 other page buttons.



*****We need to first press the button and press ctrl key in keyboard then only button navigation works in PowerBI Desktop. For PowerBI service directly if we press buttons it works.**

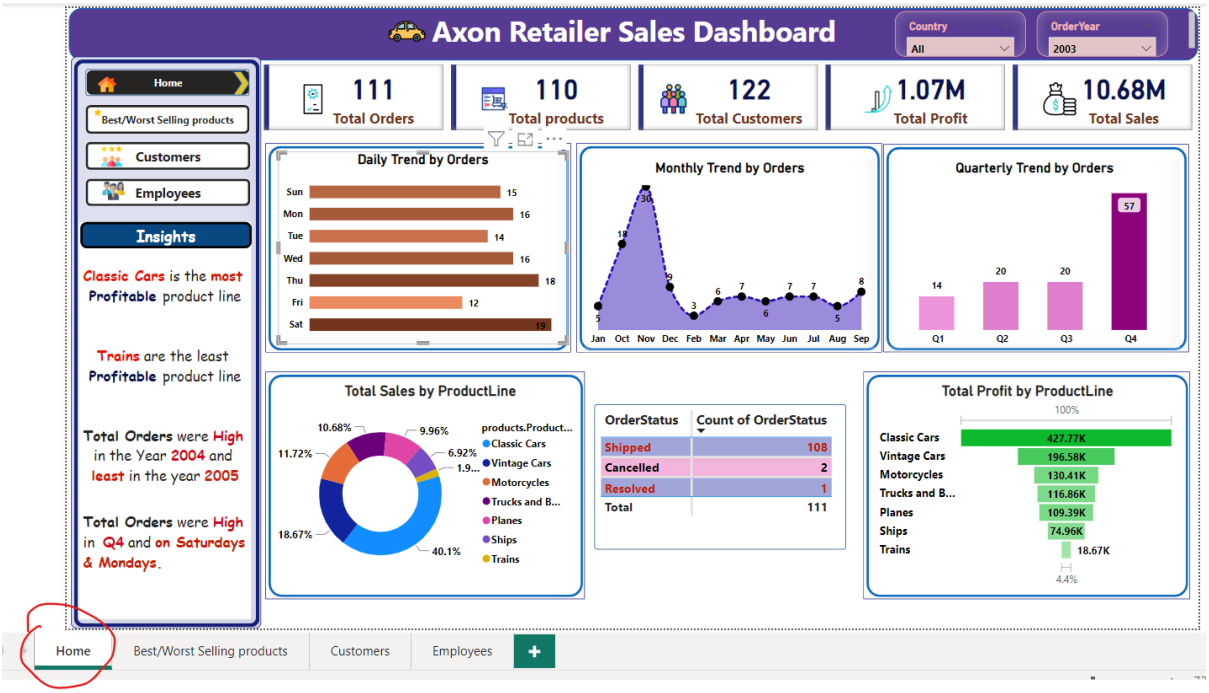
- **For graphs to be in ascending order for day wise, month wise graphs do as below:**

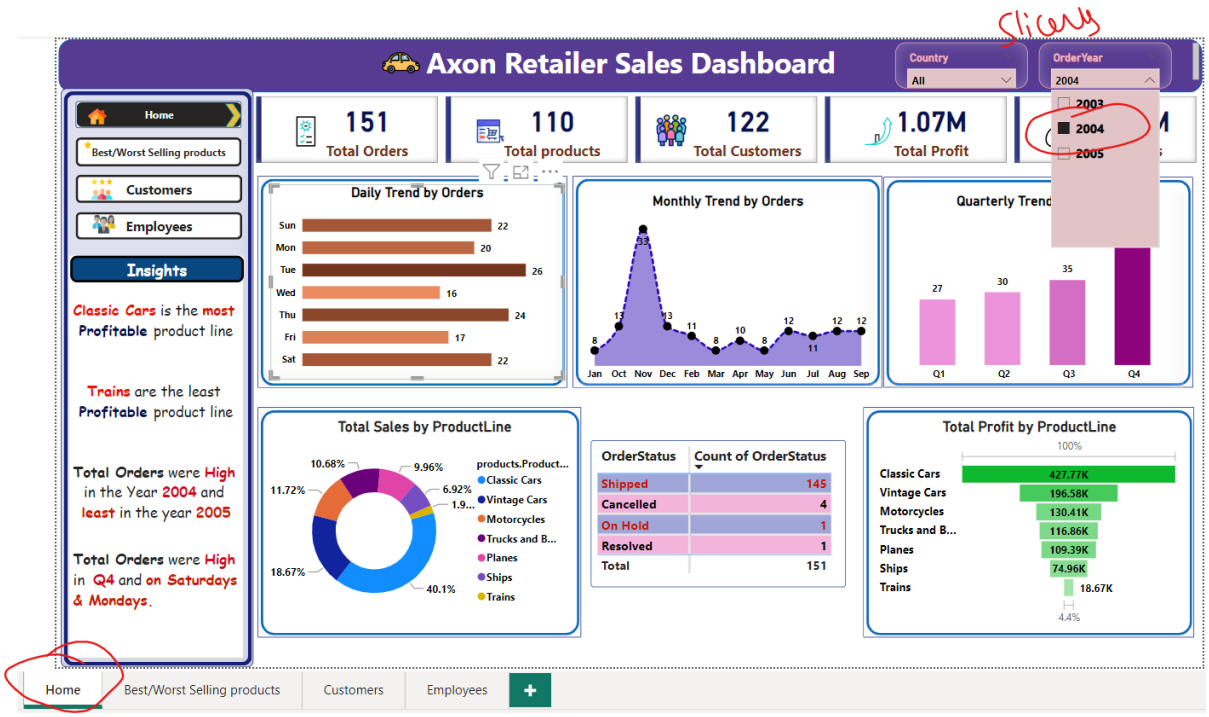




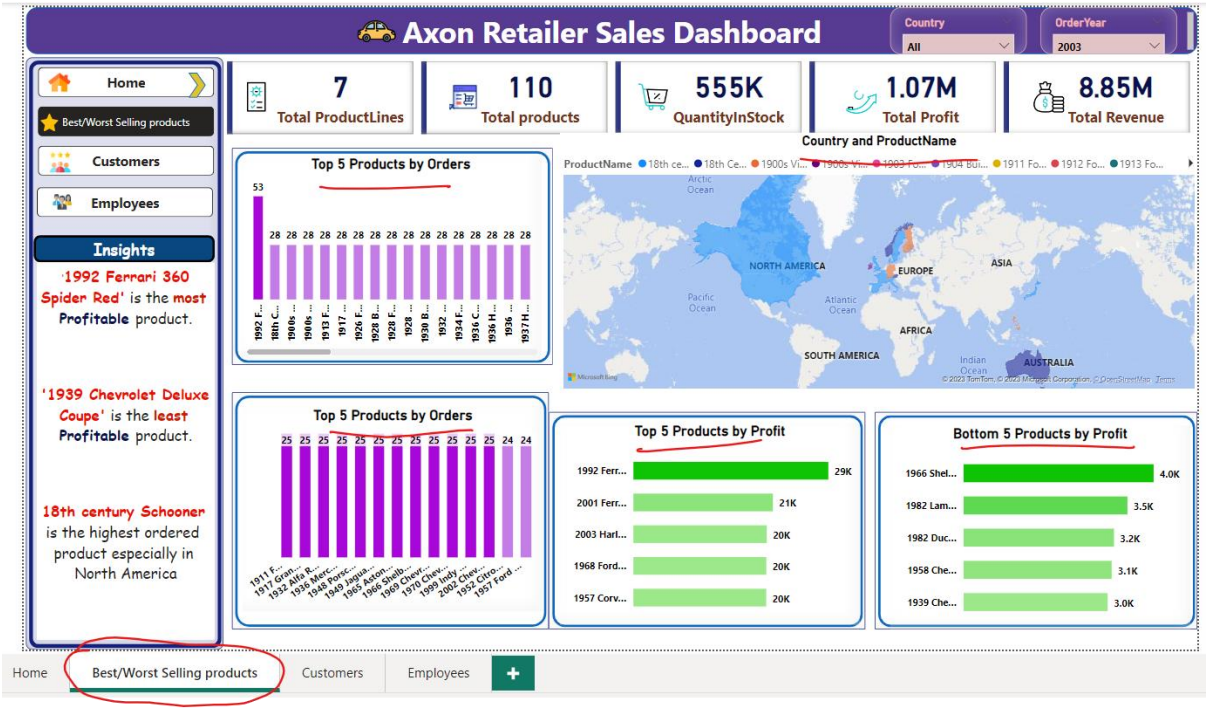
We created 4 pages namely: Home, Best/Worst Selling Products, Customers and Employees pages. Each page has graphs and tables accordingly.

Home page:

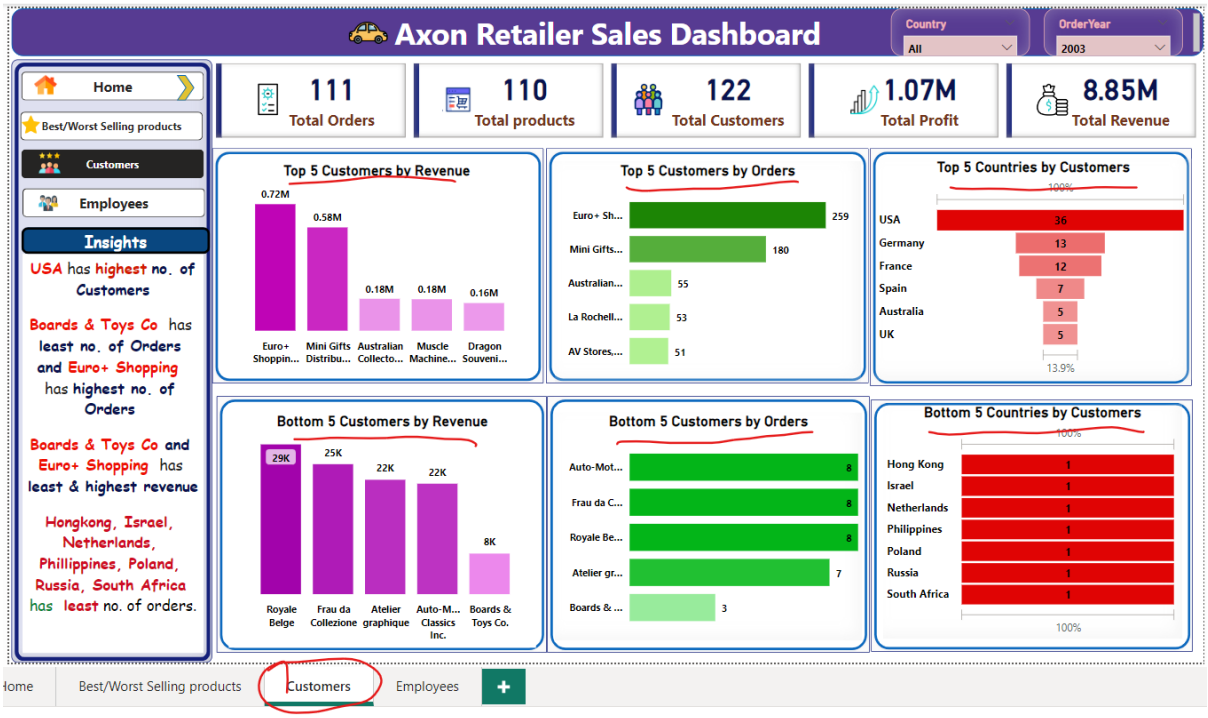




Best/Worst Selling Products page:

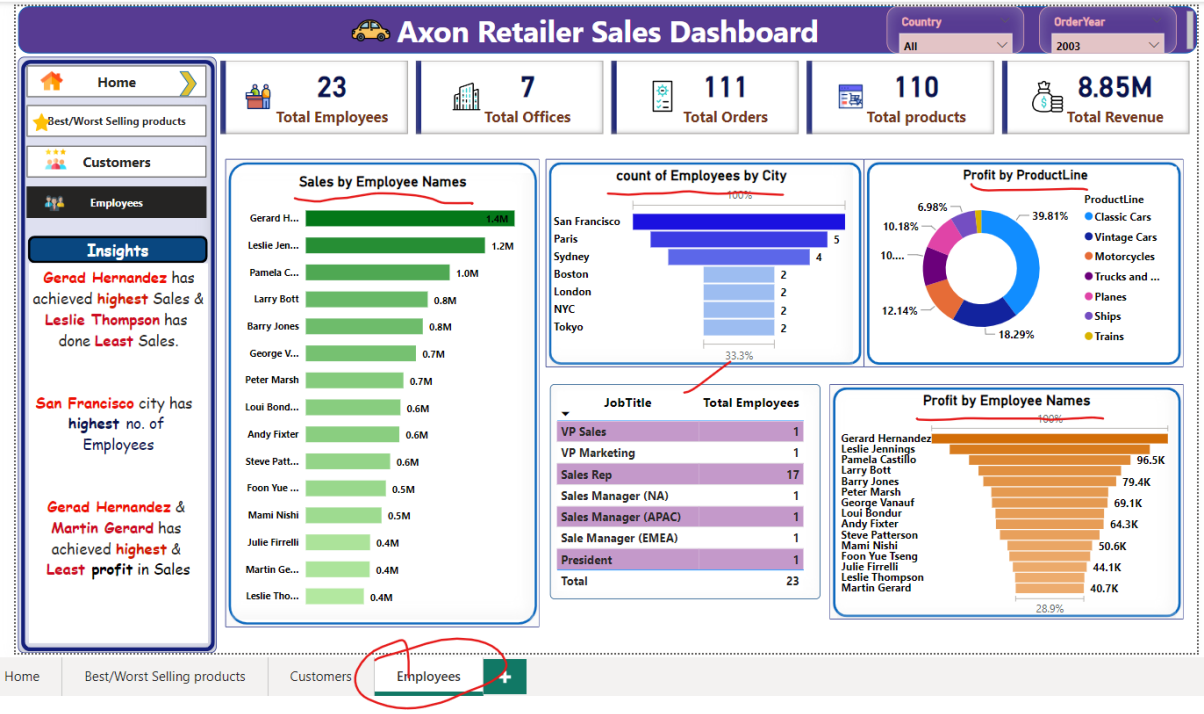


Customers page:



Employees Page:

When any button is selected it is black in color.



Insights:

The following are some of the insights from the above dashboards:

- **Classic Cars** is the **most Profitable** product line
- **Trains** are the least **Profitable** product line
- **Total Orders** were **High** in the Year **2004** and **least** in the year **2005**
- **Total Orders** were **High** in **Q4** and **on Saturdays & Mondays.**
- '**1992 Ferrari 360 Spider Red**' is the **most Profitable** product.
- '**1939 Chevrolet Deluxe Coupe**' is the **least Profitable product**
- **USA** has **highest no. of Customers**
- **Boards & Toys Co** has **least no. of Orders and Euro+ Shopping** has **highest no. of Orders**
- **Boards & Toys Co and Euro+ Shopping** has **least & highest revenue**
- **Hongkong, Israel, Netherlands, Philippines, Poland, Russia, South Africa** has **least** no. of orders.
- **18th century Schooner** is the highest ordered product especially in North America
- **Gerad Hernandez** has achieved **highest** Sales & **Leslie Thompson** has done **Least** Sales.
- **San Francisco** city has **highest** no. of Employees
- **Gerad Hernandez** & **Martin Gerard** has achieved **highest** & **Least profit** in Sales

Suggestions for Sales improvement:

- ✓ **Target New customers** by marketing campaigns in countries with less Sales like Hongkong, Israel, Netherlands, Philippines, Poland, Russia, South Africa.
- ✓ **Train and reward employees** to improve their performance.
- ✓ **Cut costs** by less advertising in North America region as orders are more there.