

-: Retail Store Sales Management System :-



❖ Introduction :-

The **Retail Store Sales Management** project is designed to help a small retail business manage and analyse its daily sales operations efficiently.

The goal of this project is to build a structured Sales Management System using **SQL** that can **store, organize** and **retrieve** key business information such as product details, customer records, orders and sales transactions.

In real-world retail environments, handling large amounts of sales data manually can be time-consuming and error-prone.

This project demonstrates how a **relational database** can simplify the process by maintaining **data integrity, reducing redundancy** and allowing quick access to critical insights such as total revenue, best-selling products and customer purchase behaviour.

❖ Objective :-

The objective of this project is to design and implement a structured **Sales Management Database** using **MySQL** that helps a small retail shop efficiently manage its products, customers and sales data.

This project aims to :

- Store and organize retail data in a relational format.
- Retrieve information using SQL queries for analysis.
- Provide insights into sales performance and inventory.
- Practice fundamental SQL concepts like **table creation, joins, primary key, foreign key and group by etc.**

❖ Tools & Technologies Used :-

- **Database :** MySQL (via XAMPP)
- **Data Source :** CSV files (imported using LOAD DATA LOCAL INFILE)
- **Software :** XAMPP Control Panel
- **Languages Used :** SQL

❖ Queries :-

- Create a New Database.
- Create Tables.
- Import CSV Files into MySQL.
- Load Each CSV File.
- Verify Data.
- Top 3 Customers by Total Purchase Amount.
- Total Sales Per Product Category.
- Products That Are Out of Stock.
- Sales Between Two Given Dates.
- Total Number of Orders Placed by Each Customer.
- Average Order Value (AOV) Per Customer.
- Highest-Selling Product.
- Products That Have Never Been Sold.
- Total Revenue Per Month.
- Top 5 Cities by Revenue.
- Customers Who Haven't Placed Any Order.
- Average Quantity Ordered Per Product.
- Total Revenue, Total Products Sold and Total Customers.
- Customer Ranking by Total Spending (With Rank Number).

❖ Solutions :-

- **Create a New Database :**

```
MariaDB [(none)]> Create Database Retail_Store_Sales_Details;
Query OK, 1 row affected (0.007 sec)

MariaDB [(none)]> Use Retail_Store_Sales_Details;
Database changed
MariaDB [Retail_Store_Sales_Details]>
```

Create Database Retail_Store_Sales_Details;
Use Retail_Store_Sales_Details;

- **Create All Four Tables :**

- **Customers :**

```
Create Table Customers( ID INT Primary Key, Customer_Name Varchar(100), Email Varchar(100), City Varchar(100));
```

- **Products :**

```
Create Table Products( Product_ID INT(4) Primary Key, Product_Name Varchar(100), Category Varchar(50), Price Decimal(10,2), Stock INT(4));
```

- **Orders :**

```
Create Table Orders( Order_ID INT Primary Key, Customer_ID INT, Order_Date Date, Total_Amount Decimal(10,2));
```

```
Alter Table Orders Add Foreign Key(Customer_ID) References Customers(ID);
```

- **Order_Details :**

```
Create Table Order_Details( Order_Details_ID INT, Product_Details_ID INT, Quantity INT, Subtotal Decimal(10,2), Foreign Key(Order_Details_ID) References Orders(Order_ID), Foreign Key(Product_Details_ID) References Products(Product_ID));
```

```
MariaDB [Retail_Store_Sales_Details]> Create Table Customers( ID INT Primary Key, Customer_Name Varchar(100), Email Varchar(100), City Varchar(100));
Query OK, 0 rows affected (0.034 sec)

MariaDB [Retail_Store_Sales_Details]> Create Table Products( Product_ID INT(4) Primary Key, Product_Name Varchar(100), Category Varchar(50), Price Decimal(10,2), Stock INT(4));
Query OK, 0 rows affected (0.030 sec)

MariaDB [Retail_Store_Sales_Details]> Create Table Orders( Order_ID INT Primary Key, Customer_ID INT, Order_Date Date, Total_Amount Decimal(10,2));
Query OK, 0 rows affected (0.028 sec)

MariaDB [Retail_Store_Sales_Details]> Alter Table Orders Add Foreign Key(Customer_ID) References Customers(ID);
Query OK, 0 rows affected (0.154 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [Retail_Store_Sales_Details]> Create Table Order_Details( Order_Details_ID INT, Product_Details_ID INT, Quantity INT, Subtotal Decimal(10,2), Foreign Key(Order_Details_ID) References Orders(Order_ID), Foreign Key(Product_Details_ID) References Products(Product_ID));
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '(Product_ID)' at line 1
MariaDB [Retail_Store_Sales_Details]> Create Table Order_Details( Order_Details_ID INT, Product_Details_ID INT, Quantity INT, Subtotal Decimal(10,2), Foreign Key(Order_Details_ID) References Orders(Order_ID), Foreign Key(Product_Details_ID) References Products(Product_ID));
Query OK, 0 rows affected (0.071 sec)

MariaDB [Retail_Store_Sales_Details]> |
```

- **Import CSV Files into MySQL :**

```
Move the downloaded files into this path : C:\xampp\mysql\data
```

- **Enable Local File Loading :**

```
MariaDB [Retail_Store_Sales_Details]> Set Global local_infile = 1;
Query OK, 0 rows affected (0.009 sec)

MariaDB [Retail_Store_Sales_Details]> Show Variables Like "local_infile";
+-----+----+
| Variable_name | Value |
+-----+----+
| local_infile | ON   |
+-----+----+
1 row in set (0.051 sec)
```

By default, MySQL blocks local file imports, enable it like this :

```
Set Global local_infile = 1;
```

```
Show Variables Like "local_infile";
```

The first command enables the **LOCAL capability** for the **LOAD DATA INFILE** statement in MySQL. Allow data to be loaded from local files on the client computer.
0 → Disabled (default in some MySQL setups)
1 → Enabled (allows loading local files)

The second command is used to check the **current setting or status** of a specific system variable — in this case, `local_infile` — in MySQL or MariaDB.
`local_infile` is a **server variable** that controls whether MySQL allows you to load data from local files.

- **Load Each CSV Fils :**

- **Customers :**

```
Load Data Local Infile "C:\\xampp\\mysql\\data\\customers.csv" Into Table Customers
Fields Terminated By "," Enclosed By "" Lines Terminated By "\\n"
Ignore 1 Rows (ID, Customer_Name, Email, City);
```

- **Products :**

```
Load Data Local Infile "C:\\xampp\\mysql\\data\\products.csv" Into Table Products
Fields Terminated By "," Enclosed By "" Lines Terminated By "\\n"
Ignore 1 Rows (Product_ID, Product_Name, Category, Price, Stock);
```

o Orders :

```
Load Data Local Infile "C:\\xampp\\mysql\\data\\orders.csv" Into Table Orders  
Fields Terminated By ","  
Enclosed By "" Lines Terminated By "\\n" Ignore 1 Rows (Order_ID, Customer_ID,  
Order_Date, Total_Amount);
```

o Order_Details :

```
Load Data Local Infile "C:\\xampp\\mysql\\data\\order_details.csv" Into Table  
Order_Details Fields Terminated By "," Enclosed By ""  
Lines Terminated By "\\n" Ignore 1 Rows (Order_Details_ID, Product_Details_ID,  
Quantity, Subtotal);
```

```
MariaDB [Retail_Store_Sales_Details]> Load Data Local Infile "C:\\xampp\\mysql\\data\\customers.csv" Into Table Customers Fields Terminated By "," Enclosed  
By "" Lines Terminated By "\\n" Ignore 1 Rows (ID, Customer_Name, Email, City);  
Query OK, 0 rows affected, 100 warnings (0.001 sec)  
Records: 100 Deleted: 0 Skipped: 100 Warnings: 100  
MariaDB [Retail_Store_Sales_Details]> Load Data Local Infile "C:\\xampp\\mysql\\data\\products.csv" Into Table Products Fields Terminated By "," Enclosed By ""  
Lines Terminated By "\\n" Ignore 1 Rows (Product_ID, Product_Name, Category, Price, Stock);  
Query OK, 100 rows affected (0.026 sec)  
Records: 100 Deleted: 0 Skipped: 0 Warnings: 0  
MariaDB [Retail_Store_Sales_Details]> Load Data Local Infile "C:\\xampp\\mysql\\data\\orders.csv" Into Table Orders Fields Terminated By "," Enclosed By ""  
Lines Terminated By "\\n" Ignore 1 Rows (Order_ID, Customer_ID, Order_Date, Total_Amount);  
Query OK, 100 rows affected (0.039 sec)  
Records: 100 Deleted: 0 Skipped: 0 Warnings: 0  
MariaDB [Retail_Store_Sales_Details]> Load Data Local Infile "C:\\xampp\\mysql\\data\\order_details.csv" Into Table Order_Details Fields Terminated By "," Enclosed  
By "" Lines Terminated By "\\n" Ignore 1 Rows (Order_Details_ID, Product_Details_ID, Quantity, Subtotal);  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near  
'Enclosed By "' Lines Terminated By "\\n" Ignore 1 Rows (Order_Details_ID, Prod... at line 1  
MariaDB [Retail_Store_Sales_Details]>  
MariaDB [Retail_Store_Sales_Details]> Load Data Local Infile "C:\\xampp\\mysql\\data\\orders.csv" Into Table Order_Details Fields Terminated By "," Enclosed  
By "" Lines Terminated By "\\n" Ignore 1 Rows (Order_Details_ID, Product_Details_ID, Quantity, Subtotal);  
Query OK, 100 rows affected, 100 warnings (0.049 sec)  
Records: 100 Deleted: 0 Skipped: 0 Warnings: 100
```

LOAD DATA LOCAL INFILE : It is used to **import data** from a CSV or text file directly into a MySQL or MariaDB table.

INTO TABLE Order_Details : **Store the data** from this CSV file into the table named Order_Details inside the current MySQL database.

FIELDS TERMINATED BY "," : This part tells MariaDB how to **separate columns or fields** in each row of your CSV or text file. Each value (field) in your file is separated by a comma (,).

ENCLOSED BY "" : The **text values** in this file are **wrapped** in double quotes — remove those quotes and take the data inside them.

LINES TERMINATED BY "\\n" : Each row of data in your file ends with a newline Character.

IGNORE 1 ROWS () : This part tells MariaDB to skip the first line of your CSV file while importing. Because most CSV files have a header row — column names etc, not actual data.

- Verify Data :

```
Select Count(*) From Customers;
```

```
Select Count(*) From Products;
```

```
Select Count(*) From Orders;
```

```
Select Count(*) From Order_Details;
```

```
MariaDB [Retail_Store_Sales_Details]> Select Count(*) From Customers;
+-----+
| Count(*) |
+-----+
|      100 |
+-----+
1 row in set (0.002 sec)

MariaDB [Retail_Store_Sales_Details]> Select Count(*) From Products;
+-----+
| Count(*) |
+-----+
|      100 |
+-----+
1 row in set (0.001 sec)

MariaDB [Retail_Store_Sales_Details]> Select Count(*) From Orders;
+-----+
| Count(*) |
+-----+
|      100 |
+-----+
1 row in set (0.001 sec)

MariaDB [Retail_Store_Sales_Details]> Select Count(*) From Order_Details;
+-----+
| Count(*) |
+-----+
|      100 |
+-----+
1 row in set (0.000 sec)
```

```
Select * From Customers Limit 5;
```

```
Select * From Products Limit 5;
```

```
Select * From Orders Limit 5;
```

```
Select * From Order_Details Limit 5;
```

```
MariaDB [Retail_Store_Sales_Details]> Select * From Customers Limit 5;
+----+-----+-----+-----+
| ID | Customer_Name | Email           | City    |
+----+-----+-----+-----+
| 1  | Faiyaz Seth   | aurorachirag@dey.com | Sagar   |
| 2  | Mannat Guha  | tiyer@rout.com     | Sri Ganganagar |
| 3  | Raunak Trivedi | tiyabaral@shere.org | Tiruppur |
| 4  | Purab Issac   | esachdev@ram.net   | New Delhi |
| 5  | Yasmin Chaudry | akarsh04@gera.com  | Davanagere |
+----+-----+-----+-----+
5 rows in set (0.017 sec)
```

```
MariaDB [Retail_Store_Sales_Details]> Select * From Products Limit 5;
```

Product_ID	Product_Name	Category	Price	Stock
1	Unde Groceries	Groceries	1014.58	3
2	Fugit Groceries	Groceries	1164.86	143
3	Quod Household	Household	572.67	149
4	Vitae Snacks	Snacks	1407.30	80
5	Natus Snacks	Snacks	1167.84	81

```
5 rows in set (0.001 sec)
```

```

MariaDB [Retail_Store_Sales_Details]> Select * From Orders Limit 5;
+-----+-----+-----+-----+
| Order_ID | Customer_ID | Order_Date | Total_Amount |
+-----+-----+-----+-----+
|      1   |       32    | 2024-11-03 |     6855.27 |
|      2   |       80    | 2025-09-22 |     5214.21 |
|      3   |       20    | 2025-03-19 |     7036.50 |
|      4   |       71    | 2025-05-09 |     1938.80 |
|      5   |        1   | 2025-08-25 |     540.67  |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)

MariaDB [Retail_Store_Sales_Details]> Select * From Order_Details Limit 5;
+-----+-----+-----+-----+
| Order_Details_ID | Product_Details_ID | Quantity | Subtotal |
+-----+-----+-----+-----+
|          1        |           32       |    2024   |   6855.27 |
|          2        |           80       |    2025   |   5214.21 |
|          3        |           20       |    2025   |   7036.50 |
|          4        |           71       |    2025   |   1938.80 |
|          5        |            1      |    2025   |   540.67  |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)

```

- **Top 3 customers by total purchase amount :**

```

Select C.Customer_Name As C_Name, Sum(O.Total_Amount) AS Total_Purchase
From Customers As C Join Orders As O On C.ID = O.Customer_ID
Group By C.Customer_Name Order By Total_Purchase Desc Limit 3;

```

```

MariaDB [Retail_Store_Sales_Details]> Select C.Customer_Name, Sum(O.Total_Amount) As Total_Purchase From Customers As C Join Orders As O On C.ID = O.Customer_ID Group By C.Customer_Name Order By Total_Purchase Desc Limit 3;
+-----+-----+
| Customer_Name | Total_Purchase |
+-----+-----+
| Yasmin Gopal |     37799.31 |
| Lakshit Sarkar |    31031.64 |
| Pari Ahluwalia |  28077.46 |
+-----+-----+
3 rows in set (0.084 sec)

```

- **Total sales per product category :**

```

Select P.Category, Sum(OD.Subtotal) As Total_Sales From Products As P
Join Order_Details As OD On P.Product_ID = OD.Product_Details_ID Group By
P.category Order By Total_Sales Desc;

```

```

MariaDB [Retail_Store_Sales_Details]> Select P.Category, Sum(OD.Subtotal) As Total_Sales From Products As P Join Order_Details As OD On P.Product_ID = OD.Product_Details_ID Group By P.Category Order By Total_Sales Desc;
+-----+-----+
| Category | Total_Sales |
+-----+-----+
| Snacks   | 227282.01 |
| Personal Care | 190775.18 |
| Groceries | 152773.08 |
| Household | 62685.13  |
| Beverages | 44315.35  |
+-----+-----+
5 rows in set (0.001 sec)

```

- Products that are out of stock :**

```
Select Product_Name, category, Stock From Products Where Stock = 0;
```

```
MariaDB [Retail_Store_Sales_Details]> Select Product_Name, Stock From Products Where Stock = 0;
Empty set (0.009 sec)
```

- Sales between two given dates :**

```
Select O.Order_Id, C.Customer_Name, O.Order_Date, O.Total_Amount From Orders
As O Join Customers As C On O.Customer_ID = C.ID Where Order_Date Between
"2024-11-01" And "2024-11-31" Order By O.Order_Date;
```

```
MariaDB [Retail_Store_Sales_Details]> Select O.Order_ID, O.Order_Date, O.Total_Amount, C.Customer_Name From Orders As O Join Customers As C On O.Customer_ID
= C.ID Where Order_Date Between "2024-11-01" And "2024-11-31" Order By Order_Date Desc;
+-----+-----+-----+-----+
| Order_ID | Order_Date | Total_Amount | Customer_Name |
+-----+-----+-----+-----+
| 23 | 2024-11-30 | 11985.36 | Tiya Dyal |
| 36 | 2024-11-24 | 6230.12 | Misha Karpe |
| 57 | 2024-11-21 | 9388.22 | Faiyaz Seth |
| 53 | 2024-11-18 | 3073.14 | Mannat Guha |
| 14 | 2024-11-16 | 5759.76 | Kabir Chauhan |
| 31 | 2024-11-08 | 4927.89 | Prerak Madan |
| 1 | 2024-11-03 | 6855.27 | Vaibhav Rajan |
+-----+-----+-----+-----+
7 rows in set (0.001 sec)
```

- Total number of orders placed by each customer :**

```
Select C.Customer_Name, Count(O.Order_ID) AS Total_Orders From Customers AS C
Left Join Orders AS O On C.ID = O.Customer_ID Group By C.Customer_Name
Order By Total_Orders Desc;
```

```
MariaDB [Retail_Store_Sales_Details]> Select C.Customer_Name, Count(O.Order_ID) AS Total_Orders From Customers As C Left Join Orders As O On C.ID = O.Customer_ID Group By C.Customer_Name Order By Total_Orders Desc;
+-----+-----+
| Customer_Name | Total_Orders |
+-----+-----+
| Lakshit Sarkar | 4 |
| Mannat Guha | 3 |
| Pari Ahluwalia | 3 |
| Sana Bhatt | 3 |
| Saksham Chaudhary | 3 |
| Yasmine Gopal | 3 |
| Siya Anand | 3 |
| Kabir Chauhan | 3 |
| Divyansh Kala | 3 |
| Darshit Bhattacharyya | 2 |
| Ayesha Soman | 2 |
| Nakul Suresh | 2 |
| Tiya Dyal | 2 |
| Shreyas Tandon | 2 |
| Riya Bandi | 2 |
| Saksham Sabharwal | 2 |
| Arshaan Grewal | 2 |
| Pari Gopal | 2 |
| Khushi Mangal | 2 |
| Taimur Gera | 2 |
| Arnav Sura | 2 |
| Faiyaz Seth | 2 |
| Dishani Jaggi | 2 |
| Shlok Badami | 2 |
| Nirvi Sankaran | 2 |
| Vaibhav Rajan | 2 |
| Vanya Roy | 2 |
| Ishita Agami | 2 |
| Bhavin Bal | 2 |
| Dharmajan Krishna | 1 |
| Samarth Chakrabarti | 1 |
| Charvi Sule | 1 |
| Anvi Ramesh | 1 |
+-----+-----+
```

Rhea Viswanathan	1
Dhruv Cherian	1
Elakshi Chakrabarti	1
Kabir Krish	1
Tarini Bhatt	1
Divij Anand	1
Mahika Kanda	1
Anika Mann	1
Nitara Sen	1
Neelofar Chaudhari	1
Armaan Iyengar	1
Misha Karpe	1
Zara Tara	1
Madhup Mangal	1
Prerak Madan	1
Taran Soman	1
Veer Bhatia	1
Kimaya Sule	1
Alisha Sibal	1
Dishani Wali	1
Kaira Jha	1
Advik Wali	1
Alia Ram	1
Ishaan Sama	1
Yasmin Chakrabarti	1
Yasmin Chaudry	1
Divyansh Bahri	1
Nitya Badal	1
Anahi Saraf	0
Anay Kurian	0
Diya Tak	0
Tejas Ramachandran	0
Krish Dutta	0
Zeeshan Seshadri	0
Ivana Krish	0
Purab Luthra	0
Reyansh Bhatnagar	0
Ahana Sarin	0
Shanaya Kothari	0
Vivaan Bera	0
Purab Issac	0
Shalv Chokshi	0
Darshit Sahota	0
Yuvraj Bhatnagar	0
Urvi Madan	0
Parinaaz De	0
Elakshi Chauhan	0
Renee Raju	0
Misha Chowdhury	0
Rania Dani	0
Vardaniya Kumar	0
Raunak Trivedi	0
Emir Sami	0
Bhavin Malhotra	0
Vivaan Vohra	0
Vritika De	0
Sara Rege	0
Keya Magar	0
Vritika Jaggi	0
Krish Dugar	0
Nakul Saini	0
Vardaniya Dewan	0
Adira Ben	0
Vritika Viswanathan	0
Siya Venkatesh	0
Rohan Deshmukh	0

+-----+
99 rows in set (0.005 sec)

- **Average order value (AOV) per customer :**

```
Select C.Customer_Name, O.Customer_ID, Round(Avg(O.Total_Amount),2) As Average_Order_Value From Customers As C Join Orders As O On C.ID = O.Customer_ID Group By C.Customer_Name, O.Customer_ID Order By Average_Order_Value Desc;
```

Customer_Name	Customer_ID	Average_Order_Value
Divyansh Bahlri	45	17378.91
Rhea Viswanathan	15	12822.57
Tiya Dyal	58	13682.12
Advik Wali	48	13654.31
Mahika Kanda	65	12735.37
Yasmin Gopal	70	12599.77
Taran Soman	54	12291.85
Arhaan Grewal	52	11185.41
Dishani Jaggi	22	11159.34
Yasmin Chaudry	5	10821.92
Yasmin Chakrabarti	85	10861.66
Pari Ahluwalia	29	9359.15
Iainira Dua	87	9377.71
Jasbir Gopal	60	9163.83
Nitya Badal	53	9076.28
Khushi Mangal	81	8985.25
Nitara Sen	9	8888.42
Ishaan Sama	69	8370.68
Kimaya Sule	19	8357.93
Mannat Guha	2	7931.66
Siya Anand	94	7867.95
Lakshit Sarkar	92	7757.91
Hazel Goswami	29	7711.75
Dhruv Cherian	23	7480.72
Riya Bandi	55	7355.74
Sana Bhatt	49	7070.60
Vaibhav Rajan	32	6928.95
Dishani Wali	72	6498.75
Saksham Chaudhary	57	6435.12
Divij Anand	36	6426.58
Misha Karpe	86	6230.12
Bhavin Bal	61	6087.18
Alisha Sibal	27	5508.65
Alia Ram	64	5287.61
Kaira Jha	88	5214.21
Kabir Krish	39	5180.53
Kabir Chauhan	99	5038.56
Shlok Badami	91	5011.44
Zara Tara	6	4985.88
Faiyaz Seth	1	4964.45
Prerak Madan	38	4927.69
Shayak Tandon	71	4836.98
Vanya Roy	13	4592.28
Divyansh Kali	51	4059.39
Ananya Kara	97	3861.81
Madhupur Mangal	14	3686.50
Nakul Surresh	98	3534.24
Elakshi Chakrabarti	31	3421.72
Darsshit Bhattacharyya	66	3157.81
Tarini Bhatt	12	3037.16
Saksham Sabharwal	68	3033.86
Mirvi Sankaran	96	2971.18
Anika Mann	73	2647.41
Ayesha Soman	82	2638.34
Armaan Iyengar	78	2345.51
Charvi Sule	50	2139.34
Veer Bhatia	62	1379.31
Anvi Ramesh	87	1338.87
Neelofar Chaudhari	100	1058.76
Dhammajan Krishna	79	628.82
Samarth Chakrabarti	18	473.52

61 rows in set (0.002 sec)

- **Highest-selling product :**

```
Select P.Product_Name, Sum(OD.Quantity) As Total_Sell From Products As P
Join Order_Details As OD On P.Product_ID = OD.Product_Details_ID Group By
Product_Name Order By Total_Sell Desc Limit 1;
```

```
MariaDB [Retail_Store_Sales_Details]> Select P.Product_Name, Sum(OD.Quantity) As Total_Sell From Products As P Join Order_Details As OD On P.Product_ID = OD.Product_Details_ID Group By P.Product_Name Order By Total_Sell Desc Limit 1;
+-----+-----+
| Product_Name | Total_Sell |
+-----+-----+
| A Snacks     |      8100 |
+-----+-----+
1 row in set (0.002 sec)
```

- **Products that have never been sold :**

```
Select P.Product_Name, OD.Quantity From Products As P Left Join Order_Details As
OD On P.Product_ID = OD.Product_Details_ID Where OD.Product_Details_ID Is Null;
```

```
MariaDB [Retail_Store_Sales_Details]> Select P.Product_Name, OD.Quantity From Products As P Left Join Order_Details As OD On P.Product_ID = OD.Product_Details_ID Where OD.Product_Details_ID Is Null;
+-----+-----+
| Product_Name | Quantity |
+-----+-----+
| Hic Groceries |    NULL |
| Accusamus Snacks |    NULL |
| Quasi Beverages |    NULL |
+-----+-----+
3 rows in set (0.001 sec)
```

- **Total revenue per month :**

```
Select Date_Format(Order_Date, "%y-%m") As Month, Sum(Total_Amount) As
Total_Revenue From Orders Group By Month Order By Month;
```

```
MariaDB [Retail_Store_Sales_Details]> Select Date_Format(Order_Date, "%y-%m") As Month, Sum(Total_Amount) As Total_Revenue From Orders Group By Month Order
By Month;
+-----+-----+
| Month | Total_Revenue |
+-----+-----+
| 24-10 |      4736.00 |
| 24-11 |      48218.96 |
| 24-12 |      84466.45 |
| 25-01 |      52912.68 |
| 25-02 |      53973.54 |
| 25-03 |      66979.98 |
| 25-04 |      106739.31 |
| 25-05 |      23801.40 |
| 25-06 |      27166.47 |
| 25-07 |      53132.78 |
| 25-08 |      39720.21 |
| 25-09 |      51517.77 |
| 25-10 |      64465.28 |
+-----+-----+
13 rows in set (0.112 sec)
```

- **Top 5 Cities By Revenue :**

```
Select C.City, Sum(O.Total_Amount) As Total_Sales From Customers As C
Join Orders As O On C.ID = O.Customer_ID Group By C.City
Order By Total_Sales Desc Limit 5;
```

```
MariaDB [Retail_Store_Sales_Details]> Select C.City, Sum(O.Total_Amount) As Total_Sales From Customers As C Join Orders As O ON C.ID = O.Customer_ID Group By C.City Order By Total_Sales Desc Limit 5;
+-----+-----+
| City | Total_Sales |
+-----+-----+
| Mehsana | 51453.62 |
| Gaya | 35722.17 |
| Kurnool | 31236.35 |
| Navi Mumbai | 31031.64 |
| Ambarnath | 28077.46 |
+-----+-----+
5 rows in set (0.040 sec)
```

- **Customers who haven't placed any order :**

```
Select C.Customer_Name From Customers As C Left Join Orders As O
On C.ID = O.Customer_ID Where O.Customer_ID Is Null;
```

```
MariaDB [Retail_Store_Sales_Details]> Select C.Customer_Name, O.Total_Amount From Customers As C Left Join Orders As O On C.ID = O.Customer_ID Where O.Customer_ID Is Null;
+-----+-----+
| Customer_Name | Total_Amount |
+-----+-----+
| Raunak Trivedi | NULL |
| Purab Isaac | NULL |
| Purab Luthra | NULL |
| Keya Magar | NULL |
| Anay Kurian | NULL |
| Emir Sami | NULL |
| Vritika Jaggi | NULL |
| Yuvraj Bhatnagar | NULL |
| Siya Venkatesh | NULL |
| Krish Dugar | NULL |
| Umar Khan | NULL |
| Diya Tak | NULL |
| Shalv Chokshi | NULL |
| Parinaz De | NULL |
| Tejas Ramachandran | NULL |
| Bhavin Malhotra | NULL |
| Rohan Deshmukh | NULL |
| Elakshi Chauhan | NULL |
| Nakul Saini | NULL |
| Neelofar Chaudhari | NULL |
| Krish Dutta | NULL |
| Vivaan Vohra | NULL |
| Darshit Sahota | NULL |
| Renee Raju | NULL |
| Reyansh Bhatnagar | NULL |
| Vardaniya Dewan | NULL |
| Vritika De | NULL |
| Ahana Sarin | NULL |
| Misha Chowdhury | NULL |
| Rania Dani | NULL |
| Shanaya Kothari | NULL |
| Adira Ben | NULL |
| Zeeshan Seshadri | NULL |
| Vardaniya Kumar | NULL |
| Vivaan Bera | NULL |
| Sara Rege | NULL |
| Anahi Saraf | NULL |
| Vritika Viswanathan | NULL |
| Ivana Krish | NULL |
+-----+-----+
39 rows in set (0.025 sec)
```

- Average quantity ordered per product :**

```
Select P.Product_Name, Round(Avg(OD.Quantity),2) AS Avg_Quantity From Products
As P Join Order_Details As OD On P.Product_ID = OD.Product_Details_ID
Group By P.Product_Name Order By Avg_Quantity;
```

MariaDB [Retail_Store_Sales_Details]> Select P.Product_Name, Round(Avg(OD.Quantity),2) AS Avg_Quantity From Products As P Join Order_Details As OD On P.Product_ID = OD.Product_Details_ID Group By P.Product_Name Order By Avg_Quantity;

Product_Name	Avg_Quantity
Cum Personal Care	1.00
Omnis Groceries	1.00
Delectus Personal Care	1.25
Iusto Groceries	1.50
Quam Personal Care	2.00
Quo Snacks	2.20
Dolore Household	2.25
Labore Groceries	2.25
Ullentiti Groceries	2.25
Uta Beverages	2.50
Voluptate Snacks	2.67
Sint Groceries	2.80
Temporibus Groceries	2.80
Alias Beverages	2.80
Quod Household	3.00
Magnam Personal Care	3.00
Placeat Snacks	3.00
Harum Groceries	3.00
Assumenda Groceries	3.25
Expedita Beverages	3.25
Minus Groceries	3.33
Est Personal Care	3.40
Facilis Household	3.50
Earum Beverages	3.50
Id Personal Care	3.50
Vitae Snacks	3.50
Ullorum Personal Care	3.67
Ullus Personal Care	3.67
Ipsa Beverages	3.71
Ex Beverages	3.71
Omnis Household	3.75
Exercitationem Groceries	3.75
Quisquam Beverages	4.00
Alias Snacks	227.11
Iste Groceries	227.22
Ipsam Personal Care	339.50
Tempora Snacks	339.67
Culpa Snacks	340.67
Distinctio Groceries	340.67
Impedit Personal Care	407.00
Istan Snacks	407.00
Sunt Household	408.00
Dolor Personal Care	452.22
Necessitatibus Snacks	507.50
Mollitia Household	508.00
Perferendis Snacks	508.00
Inde Personal Care	510.00
Vitae Household	510.00
Esse Snacks	675.67
Ex Groceries	676.00
Error Groceries	676.00
Minus Beverages	676.33
Eveniet Personal Care	676.33
Iusto Snacks	676.67
Aut Personal Care	677.00
Quaerat Snacks	677.00
Sint Beverages	677.00
Uta Beverages	677.33
Numen Snacks	677.33
Ex Snacks	678.00
Dolores Household	678.00
Vel Groceries	678.17
Velit Groceries	769.63
Deleniti Personal Care	769.50
Aliquid Household	810.00
Dicimus Snacks	811.00
Nemo Household	811.00
Facere Snacks	868.86
Pariatur Personal Care	869.86
Voltum Groceries	869.86
Ullus Personal Care	1013.17
Occaecati Snacks	1013.25
Dolores Personal Care	1013.50
Fugit Snacks	1013.50
Provident Snacks	1013.67
Voluptatis Snacks	1013.67
Dolore Groceries	1014.50
Voluptatem Personal Care	1014.50
Et Beverages	1014.50
Ullus Personal Care	1014.50
Fugit Groceries	1216.40
Unde Groceries	1350.00
A Snacks	1350.33
Unde Snacks	1350.67
Doloribus Household	1351.33
Itaque Personal Care	1351.33
Nemo Personal Care	1351.33
Ullus Personal Care	1352.00
Ilo Household	2025.00
Inventore Groceries	2025.00
Voluntas Snacks	2025.00
Fugiat Snacks	2025.00

93 rows in set (0.013 sec)

- Total Revenue, Total Products Sold and Total Customers :**

```
Select(Select Sum(Total_Amount) From Orders) As Total_Revenue,
(Select Count(Distinct Product_Details_ID) From Order_Details) As
Total_Product_Sold, (Select Count(Distinct Customer_ID) From Orders) As
Total_Customers;
```

```
MariaDB [Retail_Store_Sales_Details]> Select (Select Sum(Total_Amount) From Orders) As Total_Revenue, (Select Count(Distinct Product_Details_ID) From Order_Details) As Total_Products_Sold, (Select Count(Distinct Customer_ID) From Orders) As Total_Customers;
+-----+-----+-----+
| Total_Revenue | Total_Products_Sold | Total_Customers |
+-----+-----+-----+
|      677830.75 |              97 |             61 |
+-----+-----+-----+
1 row in set (0.040 sec)
```

- Customer Ranking By Total Spending (With Rank Number) :**

```
Select C.Customer_Name, Sum(O.Total_Amount) As Total_Spending,
Rank() Over(Order By Sum(O.Total_Amount) Desc) As Rank_Number
From Customers As C Join Orders As O ON C.ID = O.Customer_ID
GROUP BY C.Customer_Name Order By Total_Spending Desc;
```

```
MariaDB [Retail_Store_Sales_Details]> Select C.Customer_Name, Sum(O.Total_Amount) As Total_Spending, Rank() Over(Order By Sum(O.Total_Amount) Desc) As Rank_Number
From Customers As C Join Orders As O On C.ID = O.Customer_ID Group BY C.Customer_Name Order By Total_Spending Desc;
+-----+-----+-----+
| Customer_Name | Total_Spending | Rank_Number |
+-----+-----+-----+
| Yashmin Gopal |      37799.31 |          1 |
| Lakshit Sarkar |      31031.64 |          2 |
| Pari Ahivalia |      28061.46 |          3 |
| Riya Dyal |      23561.24 |          4 |
| Neelam Guha |      23790.81 |          5 |
| Siya Anand |      23602.86 |          6 |
| Arshaan Grewal |      22370.81 |          7 |
| Dishani Jaggi |      22318.67 |          8 |
| Sanchi Bhatt |      21211.88 |          9 |
| Saksham Chaudhary |      19385.37 |         10 |
| Taimur Gera |      18714.42 |         11 |
| Pari Gopal |      18327.65 |         12 |
| Khushi Mangal |      17970.58 |         13 |
| Divyansh Bahri |      17378.01 |         14 |
| Hazel Goswami |      16423.49 |         15 |
| Kabir Bhambhani |      14711.98 |         16 |
| Riya Bandi |      13657.89 |         17 |
| Ishchay Rajan |      13822.57 |         18 |
| Rhea Viswanathan |      13654.31 |         19 |
| Advik Wali |      12735.37 |         20 |
| Mahika Kanda |      12291.85 |         21 |
| Taran Soman |      12174.19 |         22 |
| Bhavin Bal |      12088.56 |         23 |
| Divyansh Kala |      10821.92 |         24 |
| Yasinin Chaudry |      10961.00 |         25 |
| Yasinin Chakrabarti |      10922.87 |         26 |
| Shlok Badami |      9928.89 |         27 |
| Faizay Seth |      9675.75 |         28 |
| Shakay Tandon |      9101.39 |         29 |
| Vanya Roy |      9076.28 |         30 |
| Nitya Madal |      8888.42 |         31 |
| Nitita Sen |      8370.68 |         32 |
| Ishaan Sama |      8357.93 |         33 |
| Kimaya Sule |      8357.93 |         34 |
| Arnav Sura |      7734.02 |         35 |
| Dhruv Cherian |      7480.77 |         36 |
| Nakul Dureesh |      7668.48 |         37 |
| Dishaan Wali |      6000.75 |         38 |
| Darjeet Anand |      4226.58 |         39 |
| Darshit Bhattacharyya |      6314.01 |         40 |
| Mishu Kapre |      6320.32 |         41 |
| Sakham Sabharwal |      6067.72 |         42 |
| Nirvi Sankaran |      5942.19 |         43 |
| Alisha Sibal |      5500.65 |         44 |
| Alia Ram |      5287.61 |         45 |
| Ayesha Soman |      5276.67 |         46 |
| Kaira Jha |      5214.21 |         47 |
| Kabir Krish |      5180.53 |         48 |
| Zara Tara |      4985.08 |         49 |
| Preerak Madan |      4927.09 |         50 |
| Madhup Mangal |      3686.58 |         51 |
| Elakshi Chakrabarti |      3421.72 |         52 |
| Tarini Bhatt |      3037.16 |         53 |
| Anika Mann |      2647.41 |         54 |
| Armaan Iyengar |      2345.51 |         55 |
| Charvi Sule |      2139.34 |         56 |
| Veer Bhatia |      1379.31 |         57 |
| Anvi Ramesh |      1338.87 |         58 |
| Neelofar Chaudhari |      1058.76 |         59 |
| Dharmajan Krishna |      628.02 |         60 |
| Samarth Chakrabarti |      473.52 |         61 |
+-----+-----+-----+
61 rows in set (0.001 sec)
```

❖ Acknowledgement :-

I would like to express my heartfelt gratitude for the support and guidance I received during the completion of my project, "**Retail Store Sales Management System**".

I am thankful to my instructor for providing valuable guidance and resources that helped me understand SQL concepts such as **database design, relationships and data analysis**.

I would also like to acknowledge the use of learning materials, online documentation and tools like MySQL and XAMPP, which played an important role in building and testing this database system.

❖ Difficulties Faced:-

During the development of the *Retail Store Sales Management System* project, I faced a few challenges :

- **Understanding Database Relationships :**
Setting up correct foreign keys and ensuring proper relationships between tables such as Orders and Order_Details required careful planning.
- **Handling CSV Imports in MySQL :**
Loading external CSV files into MySQL using the LOAD DATA INFILE command was challenging initially, especially due to file path issues and permissions.
- **Writing Multi-table SQL Queries :**
Some complex queries involving joins and aggregation (category-wise sales, top customers) took time to understand and write correctly.