Team Members :
Abhi Mehta 221080001
Aayush Khirraiyya 221080037
Branch: SY IT 2022-26
COURSE INSTRUCTOR : V.B. Nikam Sir

Assignment-03

Aim- 1.SQL-DDL to create tables and constraints for the Logical Schema you designed for your ER Data Model

2. SQL-DML to manipulate data for the your ER Data Model

3.SQL-DML should explore Complex queries, Aggregate functions like Avg, group, sort etc to manipulate data for the your ER Data Model

4.SQL-DML should explore Nested queries, Join Queries, along with the complex queries, Aggregate functions etc to manipulate data for the your ER Data Model

Theory:

Entity-Relationship (ER) Data Model:

The Entity-Relationship (ER) model is a conceptual data model used to describe the relationships between entities in a database. Entities are represented as tables, attributes as columns, and relationships as links between tables. This model helps visualize and design the structure of a database system.

SQL-DML (Data Manipulation Language):

SQL-DML is a subset of SQL (Structured Query Language) used for manipulating data within a database. It consists of commands for querying, updating, deleting, and inserting data into tables. SQL-DML operations are crucial for managing data in relational database management systems (RDBMS).

Key Components of SQL-DML:

1. SELECT Statement:
   - Retrieves data from one or more tables in the database.
   - Allows specifying columns to retrieve, conditions for filtering rows, and ordering of the result set.

2. INSERT Statement:
   - Adds new rows of data into a table.
   - Allows specifying values for each column in the table.

3. UPDATE Statement:
   - Modifies existing data in a table.
   - Allows specifying which columns to update and the new values.

4. DELETE Statement:
   - Removes rows from a table based on specified conditions.
   - Allows selectively deleting rows from the table.

5. Aggregate Functions:
   - Perform calculations on sets of values and return a single result.
   - Common aggregate functions include SUM, AVG, COUNT, MAX, and MIN.

6. Grouping and Sorting:
   - GROUP BY clause is used to group rows that have the same values into summary rows.
   - ORDER BY clause is used to sort the result set based on one or more columns.

7. Joins:
   - Combine rows from two or more tables based on related columns between them.
   - Common types of joins include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.

<u>Application in ER Data Model:</u>

- SQL-DML manipulates data stored in tables representing entities, attributes,

and relationships in an ER data model.

- It allows querying specific information, updating existing data, deleting unnecessary records, and inserting new data into tables.

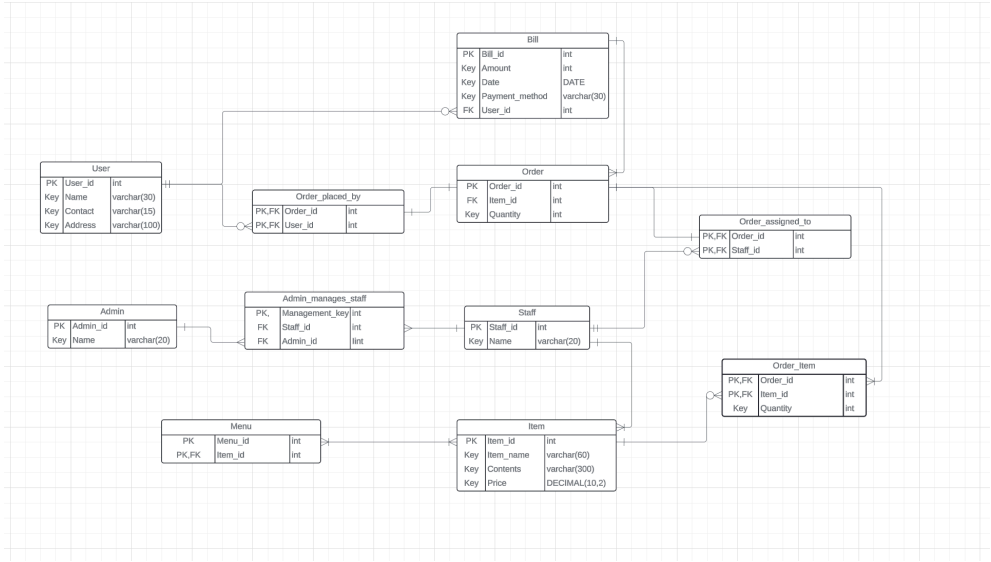
- Complex queries, aggregate functions, grouping, sorting, nested queries, and join operations help extract meaningful insights from the data stored in the Database.

- By applying SQL-DML commands effectively, users can interact with the database to perform various operations based on the requirements of their application or business logic.

In summary, SQL-DML plays a critical role in managing and manipulating data within the context of an Entity-Relationship data model. It provides the necessary tools and commands to query, update, and maintain the integrity of data stored in relational databases.

1.SQL-DDL to create tables and constraints for the Logical Schema you designed for your ER Data Model

Logical Schema:



Tables and constraints for the Logical Schema designed for ER Data Model :

User:

```
CREATE TABLE User (
   User_id INT PRIMARY KEY,
   Name VARCHAR(30),
   Contact VARCHAR(15),
   Address VARCHAR(100),
   KEY idx_user_details (Name, Contact, Address)
);
```

```
MariaDB [canteen_management]> describe User;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| User_id | int(11)     | NO   | PRI | NULL    |       |
| Name    | varchar(30) | YES  | MUL | NULL    |       |
| Contact | varchar(15) | YES  |     | NULL    |       |
| Address | varchar(700)| YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
4 rows in set (0.024 sec)
```

Order:

```
CREATE TABLE Order (
   Order_id INT PRIMARY KEY,
   Item_id INT,
   Quantity INT,
   KEY idx_quantity (Quantity)
);
```

```
MariaDB [canteen_management]> describe `Order`;
+----------+---------+------+-----+---------+-------+
| Field    | Type    | Null | Key | Default | Extra |
+----------+---------+------+-----+---------+-------+
| Order_id | int(11) | NO   | PRI | NULL    |       |
| Item_id  | int(11) | YES  |     | NULL    |       |
| Quantity | int(11) | YES  | MUL | NULL    |       |
+----------+---------+------+-----+---------+-------+
3 rows in set (0.023 sec)
```

Order_placed_by:

```
CREATE TABLE Order_placed_by (
   Order_id INT,
   User_id INT,
   PRIMARY KEY (Order_id, User_id),
   FOREIGN KEY (Order_id) REFERENCES Order(Order_id),
   FOREIGN KEY (User_id) REFERENCES User(User_id)
);
```

```
MariaDB [canteen_management]> describe Order_placed_by;
+----------+---------+------+-----+---------+-------+
| Field    | Type    | Null | Key | Default | Extra |
+----------+---------+------+-----+---------+-------+
| Order_id | int(11) | NO   | PRI | NULL    |       |
| User_id  | int(11) | NO   | PRI | NULL    |       |
+----------+---------+------+-----+---------+-------+
2 rows in set (0.019 sec)
```

Order_assigned_to:

```
CREATE TABLE Order_assigned_to (
   Order_id INT,
   Staff_id INT,
   PRIMARY KEY (Order_id, Staff_id),
   FOREIGN KEY (Order_id) REFERENCES Order(Order_id),
   FOREIGN KEY (Staff_id) REFERENCES Staff(Staff_id)
);
```

```
MariaDB [canteen_management]> describe Order_assigned_to;
+----------+---------+------+-----+---------+-------+
| Field    | Type    | Null | Key | Default | Extra |
+----------+---------+------+-----+---------+-------+
| Order_id | int(11) | NO   | PRI | NULL    |       |
| Staff_id | int(11) | NO   | PRI | NULL    |       |
+----------+---------+------+-----+---------+-------+
2 rows in set (0.023 sec)
```

Bill:

```
CREATE TABLE Bill (
   Bill_id INT PRIMARY KEY,
   Amount INT,
   Date DATE,
   Payment_method VARCHAR(30),
   User_id INT,
   FOREIGN KEY (User_id) REFERENCES User(User_id),
   KEY idx_bill_details (Amount, Date, Payment_method)
);
```

```
MariaDB [canteen_management]> describe Bill;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| Bill_id        | int(11)     | NO   | PRI | NULL    |       |
| Amount         | int(11)     | YES  | MUL | NULL    |       |
| Date           | date        | YES  |     | NULL    |       |
| Payment_method | varchar(30) | YES  |     | NULL    |       |
| User_id        | int(11)     | YES  | MUL | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
5 rows in set (0.026 sec)
```

Order_Item:

```
CREATE TABLE Order_Item (
   Order_id INT,
   Item_id INT,
   Quantity INT,
   PRIMARY KEY (Order_id, Item_id),
   FOREIGN KEY (Order_id) REFERENCES Order(Order_id),
   FOREIGN KEY (Item_id) REFERENCES Item(Item_id),
   KEY idx_quantity (Quantity)
);
```

```
MariaDB [canteen_management]> describe Order_Item;
+----------+---------+------+-----+---------+-------+
| Field    | Type    | Null | Key | Default | Extra |
+----------+---------+------+-----+---------+-------+
| Order_id | int(11) | NO   | PRI | NULL    |       |
| Item_id  | int(11) | NO   | PRI | NULL    |       |
| Quantity | int(11) | YES  | MUL | NULL    |       |
+----------+---------+------+-----+---------+-------+
3 rows in set (0.025 sec)
```

Item:

```
CREATE TABLE Item (
    Item_id INT PRIMARY KEY,
    Item_name VARCHAR(60),
    Contents VARCHAR(300),
    Price DECIMAL(10, 2),
    KEY idx_item_details (Item_name, Contents, Price)
);
```

```
MariaDB [canteen_management]> describe Item;
+-----------+---------------+------+-----+---------+-------+
| Field     | Type          | Null | Key | Default | Extra |
+-----------+---------------+------+-----+---------+-------+
| Item_id   | int(11)       | NO   | PRI | NULL    |       |
| Item_name | varchar(60)   | YES  | MUL | NULL    |       |
| Contents  | varchar(300)  | YES  |     | NULL    |       |
| Price     | decimal(10,2) | YES  |     | NULL    |       |
+-----------+---------------+------+-----+---------+-------+
4 rows in set (0.021 sec)
```

Menu:

```
CREATE TABLE Menu (
    Menu_id INT,
    Item_id INT,
    PRIMARY KEY (Menu_id, Item_id),
    FOREIGN KEY (Item_id) REFERENCES Item(Item_id)
);
```

```
MariaDB [canteen_management]> describe Menu;
+---------+---------+------+-----+---------+-------+
| Field   | Type    | Null | Key | Default | Extra |
+---------+---------+------+-----+---------+-------+
| Menu_id | int(11) | NO   | PRI | NULL    |       |
| Item_id | int(11) | NO   | PRI | NULL    |       |
+---------+---------+------+-----+---------+-------+
2 rows in set (0.023 sec)
```

Staff:

```
CREATE TABLE Staff (
   Staff_id INT PRIMARY KEY,
   Name VARCHAR(20),
   KEY idx_name (Name)
);
```

```
MariaDB [canteen_management]> describe Staff;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| Staff_id   | int(11)     | NO   | PRI | NULL    |       |
| Staff_Name | varchar(20) | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.024 sec)
```

Admin:

```
CREATE TABLE Admin (
   Admin_id INT PRIMARY KEY,
   Name VARCHAR(20),
   KEY idx_name (Name)
);
```

```
MariaDB [canteen_management]> describe Admin;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| Admin_id   | int(11)     | NO   | PRI | NULL    |       |
| Admin_Name | varchar(20) | YES  | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.022 sec)
```

Admin_manages_staff:

```
CREATE TABLE Admin_manages_staff (
   Management_key INT AUTO_INCREMENT PRIMARY KEY,
   Staff_id INT,
   Admin_id INT,
   FOREIGN KEY (Staff_id) REFERENCES Staff(Staff_id),
   FOREIGN KEY (Admin_id) REFERENCES Admin(Admin_id)
);
```

```
MariaDB [canteen_management]> describe Admin_manages_staff;
+----------------+---------+------+-----+---------+----------------+
| Field          | Type    | Null | Key | Default | Extra          |
+----------------+---------+------+-----+---------+----------------+
| Management_key | int(11) | NO   | PRI | NULL    | auto_increment |
| Staff_id       | int(11) | YES  | MUL | NULL    |                |
| Admin_id       | int(11) | YES  | MUL | NULL    |                |
+----------------+---------+------+-----+---------+----------------+
3 rows in set (0.023 sec)
```

2. SQL-DML to manipulate data for the your ER Data Model

1.Insert a new menu item:

```
MariaDB [canteen_management]> INSERT INTO Menu (Menu_id, Item_id)
    -> VALUES (2211, 44);
Query OK, 1 row affected (0.003 sec)
```

```
|    2210 |      34 |
|    2210 |      35 |
|    2210 |      36 |
|    2210 |      37 |
|    2210 |      38 |
|    2210 |      39 |
|    2210 |      40 |
|    2210 |      43 |
|    2211 |      44 |
+---------+---------+
42 rows in set (0.001 sec)
```

2.Insert a new order item:

```
MariaDB [canteen_management]> INSERT INTO Order_Item (Order_id, Item_id, Quantity)
    -> VALUES (241, 4, 2);
Query OK, 1 row affected (0.003 sec)
```

```
|      241 |       4 |        2 |
|      288 |      11 |        2 |
|      672 |      28 |        2 |
|      939 |      32 |        2 |
|      323 |      37 |        3 |
|      362 |      40 |        3 |
|      664 |      36 |        3 |
|      700 |      20 |        3 |
|      909 |      30 |        3 |
+----------+---------+----------+
41 rows in set (0.001 sec)
```

3.Insert a new order:

```
MariaDB [canteen_management]> INSERT INTO `Order` (Order_id,Item_id, Quantity)
    -> VALUES (2211,4,2);
Query OK, 1 row affected (0.003 sec)
```

```
|      878 |      33 |        1 |
|      879 |      31 |        1 |
|      909 |      30 |        3 |
|      939 |      32 |        2 |
|     2211 |       4 |        2 |
+----------+---------+----------+
41 rows in set (0.001 sec)
```

4.Insert a new item:

```
MariaDB [canteen_management]> INSERT INTO Item (Item_id,Item_name, Contents, Price)
    -> VALUES (66,'Samosa Chaat', 'Samosa topped with chutney and spices', 45.75);
Query OK, 1 row affected (0.004 sec)
```

```
|      66 | Samosa Chaat          | Samosa topped with chutney and spices
                                                                                                   | 45.75 |
|       2 | Samosa Pav            | Samosa pav is a fusion street food where a samosa is sandwiched between a pav bread roll, typically served with
 chutneys and sometimes garnished with onions and cilantro.                                        | 14.00 |
|       4 | Samosa Plate          | A samosa plate typically consists of crispy triangular pastry filled with spiced potatoes and peas, served alon
gside tangy tamarind chutney and minty coriander chutney.                                          | 24.00 |
```

5.Insert a new user:

```
MariaDB [canteen_management]> INSERT INTO User (User_id,Name, Contact, Address)
    -> VALUES (44,'Michael Brown', '5678901234', '789 Oak Ave, City');
Query OK, 1 row affected (0.004 sec)
```

```
|      45 | Jane Smith    | 555-1234   | 123 Main St, Anytown USA                                        |
|      39 | Labdhi        | 7620489209 |        Dunhill, Pali Hill Shop 3, Dr Ambedkar Rd, Khar West            |
|      13 | Maithili      | 7620489263 | Saki Vihar Rd, Murarji Nagar, Mayur Nagar, Passpoli, Powai, Mumbai |
|      14 | manas         | 7620489264 | Sahar Airport Road, Andheri - Kurla Rd, near Mumbai International Airport |
|      44 | Michael Brown | 5678901234 | 789 Oak Ave, City                                               |
|      35 | Moksh         | 7620489205 | Shop No. 1, Edward Apartment, Off Link Road, Evershine Nagar, Malad West |
|      15 | Mrunmayi      | 7620489265 | 462, Senapati Bapat Marg, Lower Parel, Mumbai, Maharashtra 400013, India |
|      16 | Nishit        | 7620489266 | Plot No.34, 21, MIDC Central Rd, near Akruti Center Point        |
|      17 | Pavan         | 7620489267 | Oberoi Garden City, International Business Park, Yashodham, Goregaon |
```

6.Delete from order item:

```
MariaDB [canteen_management]> DELETE FROM Order_Item
    -> WHERE Order_id = 221 AND Item_id = 4;
Query OK, 0 rows affected (0.004 sec)
```

7.Delete from User:

```
MariaDB [canteen_management]> DELETE FROM User
    -> WHERE User_id = 44;
Query OK, 1 row affected (0.003 sec)
```

8.Delete from Item:

```
MariaDB [canteen_management]> DELETE FROM Item
    -> WHERE Item_id = 44;
Query OK, 0 rows affected (0.001 sec)
```

9.Delete from Menu:

```
MariaDB [canteen_management]> DELETE FROM Menu
    -> WHERE Menu_id = 2211 AND Item_id = 44;
Query OK, 1 row affected (0.003 sec)
```

10.Delete from Order:

```
MariaDB [canteen_management]> DELETE FROM `Order`
    -> WHERE Order_id =2211;
Query OK, 1 row affected (0.002 sec)
```

11.Update Bill:

```
MariaDB [canteen_management]> UPDATE Bill
    -> SET Payment_method = 'Credit Card'
    -> WHERE Bill_id = 2200;
Query OK, 1 row affected (0.002 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

12.Update Staffl:

```
MariaDB [canteen_management]> UPDATE Staff
    -> SET Staff_Name = 'Joginder'
    -> WHERE Staff_id = 7891;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

13.Update User:

```
MariaDB [canteen_management]> UPDATE User
    -> SET Contact = '9876543210'
    -> WHERE User_id = 1;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

14.Update Item:

```
MariaDB [canteen_management]> UPDATE Item
    -> SET Contents = 'Spicy potato patty in bread bun'
    -> WHERE Item_id = 1;
Query OK, 1 row affected (0.004 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

15.Update Admin:

```
MariaDB [canteen_management]> UPDATE Admin
    -> SET Admin_Name = 'Ranvir'
    -> WHERE Admin_id=80037;
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

3.SQL-DML should explore Complex queries, Aggregate functions like Avg, group, sort etc to manipulate data for the your ER Data Model

1.Calculate the total amount spent by each user:

```
MariaDB [canteen_management]> SELECT User_id, SUM(Amount) AS Total_Amount_Spent
    -> FROM Bill
    -> GROUP BY User_id;
+---------+--------------------+
| User_id | Total_Amount_Spent |
+---------+--------------------+
|       1 |                  5 |
|       2 |                  8 |
|       3 |                 45 |
|       4 |                 10 |
|       5 |                 10 |
|       6 |                 20 |
|       7 |                 10 |
|       8 |                 20 |
|       9 |                 10 |
|      10 |                 20 |
|      11 |                 40 |
|      12 |                 55 |
|      13 |                 38 |
|      14 |                 40 |
|      15 |                 10 |
|      16 |                 20 |
|      17 |                 10 |
|      18 |                 15 |
|      19 |                 10 |
|      20 |                 30 |
|      21 |                 30 |
|      22 |                 20 |
|      23 |                 25 |
|      24 |                 10 |
|      25 |                 10 |
|      26 |                 20 |
|      27 |                 20 |
|      28 |                 40 |
|      29 |                 28 |
|      30 |                 60 |
|      31 |                 20 |
|      32 |                 20 |
|      33 |                 20 |
|      34 |                 20 |
|      35 |                 10 |
|      36 |                 90 |
|      37 |                 80 |
|      38 |                 10 |
|      39 |                 30 |
|      40 |                 40 |
+---------+--------------------+
40 rows in set (0.000 sec)
```

2.Find the average price of items in the menu:

```
MariaDB [canteen_management]> SELECT AVG(Price) AS Average_Item_Price
    -> FROM Item;
+--------------------+
| Average_Item_Price |
+--------------------+
|          32.450000 |
+--------------------+
1 row in set (0.001 sec)
```

3. Retrieve the top 5 highest-priced items:

```
MariaDB [canteen_management]> SELECT Item_name, Price
    -> FROM Item
    -> ORDER BY Price DESC
    -> LIMIT 5;
+----------------------+-------+
| Item_name            | Price |
+----------------------+-------+
| Lunch                | 64.00 |
| Veg Schezwan Rice    | 55.00 |
| Veg Schezwan Noodles | 55.00 |
| Chinese Dosa         | 53.00 |
| Veg Chesse Sandwich  | 50.00 |
+----------------------+-------+
5 rows in set (0.001 sec)
```

4.List all orders along with the user's name and contact information:

```
MariaDB [canteen_management]> SELECT o.Order_id, u.Name AS User_Name, u.Contact
    -> FROM `Order` o
    -> JOIN Order_placed_by opb ON o.Order_id = opb.Order_id
    -> JOIN User u ON opb.User_id = u.User_id;
+----------+-----------+------------+
| Order_id | User_Name | Contact    |
+----------+-----------+------------+
|       13 | Toro      | 7719865554 |
|       20 | Achal     | 7719865559 |
|       23 | manas     | 7620489264 |
|       59 | Amrut     | 7719865553 |
|       67 | Abhijit   | 7719865558 |
|       74 | Maithili  | 7620489263 |
|      123 | Aditi     | 7719865552 |
|      215 | Abhi      | 7719865557 |
|      234 | Gaurav    | 7620489262 |
|      241 | Aayush    | 7719865551 |
|      255 | Aastha    | 7719865556 |
|      288 | Dhruv     | 7620489261 |
|      323 | Chaitali  | 7620489207 |
|      324 | Aaditya   | 7719865555 |
|      331 | Mrunmayi  | 7620489265 |
|      362 | Yuvraj    | 7620489210 |
|      444 | Anam      | 7719865550 |
|      523 | Labdhi    | 7620489209 |
|      652 | Pranjal   | 7620489208 |
|      661 | Sahil     | 7620489202 |
|      663 | Sachin    | 7620489201 |
|      664 | Soham     | 7620489206 |
|      670 | Vansh     | 7620489209 |
|      672 | Tanmay    | 7620489208 |
|      694 | Moksh     | 7620489205 |
|      700 | Riddhi    | 7620489260 |
|      703 | Sumit     | 7620489207 |
|      704 | Anish     | 7620489204 |
|      710 | Nishit    | 7620489266 |
|      713 | Raj       | 7620489269 |
|      717 | Purvi     | 7620489268 |
|      721 | Pavan     | 7620489267 |
|      731 | Shubham   | 7620489206 |
|      800 | Shravani  | 7620489205 |
|      844 | Sarvesh   | 7620489203 |
|      849 | Shon      | 7620489204 |
|      878 | Ruchi     | 7620489203 |
|      879 | Vedika    | 7620489211 |
|      909 | Yash      | 7620489200 |
|      939 | Vaibhavi  | 7620489202 |
+----------+-----------+------------+
```

5.Find the total quantity of each item ordered:

```
MariaDB [canteen_management]> SELECT Item_id, SUM(Quantity) AS Total_Quantity_Ordered
    -> FROM Order_Item
    -> GROUP BY Item_id;
+---------+-------------------------+
| Item_id | Total_Quantity_Ordered |
+---------+-------------------------+
|       1 |                       1 |
|       2 |                       2 |
|       3 |                       1 |
|       4 |                       1 |
|       5 |                       1 |
|       6 |                       1 |
|       7 |                       1 |
|       8 |                       1 |
|       9 |                       1 |
|      10 |                       1 |
|      11 |                       2 |
|      12 |                       1 |
|      13 |                       2 |
|      14 |                       1 |
|      15 |                       1 |
|      16 |                       1 |
|      17 |                       1 |
|      18 |                       1 |
|      19 |                       1 |
|      20 |                       3 |
|      21 |                       1 |
|      22 |                       1 |
|      23 |                       1 |
|      24 |                       1 |
|      25 |                       1 |
|      26 |                       1 |
|      27 |                       1 |
|      28 |                       2 |
|      29 |                       1 |
|      30 |                       3 |
|      31 |                       1 |
|      32 |                       2 |
|      33 |                       1 |
|      34 |                       1 |
|      35 |                       1 |
|      36 |                       3 |
|      37 |                       3 |
|      38 |                       1 |
|      39 |                       1 |
|      40 |                       3 |
+---------+-------------------------+
40 rows in set (0.001 sec)
```

6.Retrieve the average quantity of items ordered per order:

```
MariaDB [canteen_management]> SELECT AVG(Quantity) AS Average_Quantity_Per_Order
    -> FROM Order_Item;
+----------------------------+
| Average_Quantity_Per_Order |
+----------------------------+
|                     1.3750 |
+----------------------------+
1 row in set (0.000 sec)
```

7.List all bills sorted by date in descending order:

```
MariaDB [canteen_management]> SELECT * FROM Bill ORDER BY Date DESC;
+---------+--------+------------+----------------+---------+
| Bill_id | Amount | Date       | Payment_method | User_id |
+---------+--------+------------+----------------+---------+
|    2239 |     40 | 2024-04-05 | Cash           |      40 |
|    2228 |     28 | 2024-04-05 | Cash           |      29 |
|    2227 |     40 | 2024-04-05 | Cash           |      28 |
|    2226 |     20 | 2024-04-05 | Cash           |      27 |
|    2225 |     20 | 2024-04-05 | Cash           |      26 |
|    2224 |     10 | 2024-04-05 | Cash           |      25 |
|    2223 |     10 | 2024-04-05 | Cash           |      24 |
|    2222 |     25 | 2024-04-05 | Cash           |      23 |
|    2221 |     20 | 2024-04-05 | Cash           |      22 |
|    2229 |     60 | 2024-04-05 | Cash           |      30 |
|    2230 |     20 | 2024-04-05 | Cash           |      31 |
|    2238 |     30 | 2024-04-05 | Cash           |      39 |
|    2237 |     10 | 2024-04-05 | Cash           |      38 |
|    2236 |     80 | 2024-04-05 | Cash           |      37 |
|    2235 |     90 | 2024-04-05 | Cash           |      36 |
|    2234 |     10 | 2024-04-05 | Cash           |      35 |
|    2233 |     20 | 2024-04-05 | Cash           |      34 |
|    2232 |     20 | 2024-04-05 | Cash           |      33 |
|    2231 |     20 | 2024-04-05 | Cash           |      32 |
|    2220 |     30 | 2024-04-05 | Cash           |      21 |
|    2219 |     30 | 2024-04-05 | Cash           |      20 |
|    2218 |     10 | 2024-04-05 | Cash           |      19 |
|    2207 |     20 | 2024-04-05 | Cash           |       8 |
|    2206 |     10 | 2024-04-05 | Cash           |       7 |
|    2205 |     20 | 2024-04-05 | Cash           |       6 |
|    2204 |     10 | 2024-04-05 | Cash           |       5 |
|    2203 |     10 | 2024-04-05 | Cash           |       4 |
|    2202 |     45 | 2024-04-05 | Cash           |       3 |
|    2201 |      8 | 2024-04-05 | Cash           |       2 |
|    2200 |      5 | 2024-04-05 | Cash           |       1 |
|    2208 |     10 | 2024-04-05 | Cash           |       9 |
|    2209 |     20 | 2024-04-05 | Cash           |      10 |
|    2217 |     15 | 2024-04-05 | Cash           |      18 |
|    2216 |     10 | 2024-04-05 | Cash           |      17 |
|    2215 |     20 | 2024-04-05 | Cash           |      16 |
|    2214 |     10 | 2024-04-05 | Cash           |      15 |
|    2213 |     40 | 2024-04-05 | Cash           |      14 |
|    2212 |     38 | 2024-04-05 | Cash           |      13 |
|    2211 |      5 | 2024-04-05 | Cash           |      12 |
|    2210 |     40 | 2024-04-05 | Cash           |      11 |
|     201 |     50 | 2023-04-10 | Cash           |      12 |
+---------+--------+------------+----------------+---------+
41 rows in set (0.000 sec)
```

8.Find the total number of orders placed by each user:

```
MariaDB [canteen_management]> SELECT u.User_id, COUNT(o.Order_id) AS Total_Orders_Placed
    -> FROM User u
    -> LEFT JOIN Order_placed_by opb ON u.User_id = opb.User_id
    -> LEFT JOIN `Order` o ON opb.Order_id = o.Order_id
    -> GROUP BY u.User_id;
+---------+---------------------+
| User_id | Total_Orders_Placed |
+---------+---------------------+
|       1 |                   1 |
|       2 |                   1 |
|       3 |                   1 |
|       4 |                   1 |
|       5 |                   1 |
|       6 |                   1 |
|       7 |                   1 |
|       8 |                   1 |
|       9 |                   1 |
|      10 |                   1 |
|      11 |                   1 |
|      12 |                   1 |
|      13 |                   1 |
|      14 |                   1 |
|      15 |                   1 |
|      16 |                   1 |
|      17 |                   1 |
|      18 |                   1 |
|      19 |                   1 |
|      20 |                   1 |
|      21 |                   1 |
|      22 |                   1 |
|      23 |                   1 |
|      24 |                   1 |
|      25 |                   1 |
|      26 |                   1 |
|      27 |                   1 |
|      28 |                   1 |
|      29 |                   1 |
|      30 |                   1 |
|      31 |                   1 |
|      32 |                   1 |
|      33 |                   1 |
|      34 |                   1 |
|      35 |                   1 |
|      36 |                   1 |
|      37 |                   1 |
|      38 |                   1 |
|      39 |                   1 |
|      40 |                   1 |
|      45 |                   0 |
+---------+---------------------+
41 rows in set (0.002 sec)
```

9.Retrieve the top 3 highest-spending users:

```
MariaDB [canteen_management]> SELECT u.User_id, u.Name, SUM(b.Amount) AS Total_Amount_Spent
    -> FROM User u
    -> LEFT JOIN Bill b ON u.User_id = b.User_id
    -> GROUP BY u.User_id
    -> ORDER BY Total_Amount_Spent DESC
    -> LIMIT 3;
+---------+---------+--------------------+
| User_id | Name    | Total_Amount_Spent |
+---------+---------+--------------------+
|      36 | Soham   |                 90 |
|      37 | Chaitali |                80 |
|      30 | Yash    |                 60 |
+---------+---------+--------------------+
3 rows in set (0.001 sec)
```

10.List all orders along with the item details and quantities:

```
MariaDB [canteen_management]> SELECT o.Order_id, i.Item_name, oi.Quantity
    -> FROM `Order` o
    -> JOIN Order_Item oi ON o.Order_id = oi.Order_id
    -> JOIN Item i ON oi.Item_id = i.Item_id;
+----------+------------------------+----------+
| Order_id | Item_name              | Quantity |
+----------+------------------------+----------+
|       13 | Samosa Plate           |        1 |
|       20 | Medu Wada              |        1 |
|       23 | Samosa Usal            |        1 |
|       59 | Vada Plate             |        1 |
|       67 | Upma Poha              |        1 |
|      215 | Upma                   |        1 |
|      234 | Vada Usal              |        1 |
|      241 | Vada Pav               |        1 |
|      255 | Idli-Sambhar           |        1 |
|      324 | Poha                   |        1 |
|      331 | Veg Sandwich           |        1 |
|      444 | Batata Wada            |        1 |
|      523 | Chass                  |        1 |
|      652 | Coffe                  |        1 |
|      661 | Veg Hakka Noddles      |        1 |
|      663 | Veg Schezwan Rice      |        1 |
|      670 | Mysore Sada Dosa       |        1 |
|      694 | Chinese Dosa           |        1 |
|      703 | Butter Masala Dosa     |        1 |
|      704 | Masala Uttappa         |        1 |
|      710 | Veg Chesse Sandwich    |        1 |
|      713 | Cheese Sandwich        |        1 |
|      717 | Mumbai Cheese Sandwich |        1 |
|      721 | Mumbai Sandwich        |        1 |
|      731 | Butter Sada Dosa       |        1 |
|      800 | Masala Dosa            |        1 |
|      844 | Veg Schezwan Noodles   |        1 |
|      849 | Sada Dosa              |        1 |
|      878 | Onion Uttappa          |        1 |
|      879 | Schezwan Masala Dosa   |        1 |
|       74 | Ragada Samosa          |        2 |
|      123 | Samosa Pav             |        2 |
|      288 | Misal Pav              |        2 |
|      672 | Mysore Masala Dosa     |        2 |
|      939 | Sada Uttappa           |        2 |
|      323 | Tea                    |        3 |
|      362 | Lassi                  |        3 |
|      664 | Lunch                  |        3 |
|      700 | Veg Fried Rice         |        3 |
|      909 | Schezwan Sada Dosa     |        3 |
+----------+------------------------+----------+
```

11.Calculate the average price per bill:

```
MariaDB [canteen_management]> SELECT AVG(Amount) AS Average_Bill_Amount
    -> FROM Bill;
+----------------------+
| Average_Bill_Amount |
+----------------------+
|              25.0976 |
+----------------------+
1 row in set (0.000 sec)
```

12.Retrieve the names of all staff members managed by admins whose name starts with 'A':

```
MariaDB [canteen_management]> SELECT s.Staff_Name AS Name
    -> FROM Staff s
    -> JOIN Admin_manages_staff ams ON s.Staff_id = ams.Staff_id
    -> JOIN Admin a ON ams.Admin_id = a.Admin_id
    -> WHERE s.Staff_Name LIKE 'A%';
+------+
| Name |
+------+
| Arun |
| Arun |
| Arun |
| Arun |
| Arun |
| Arun |
| Arun |
| Arun |
+------+
8 rows in set (0.003 sec)
```

13.List all bills along with the payment method, sorted by amount in ascending order:

```
MariaDB [canteen_management]> SELECT * FROM Bill ORDER BY Amount ASC;
+---------+--------+------------+----------------+---------+
| Bill_id | Amount | Date       | Payment_method | User_id |
+---------+--------+------------+----------------+---------+
|    2200 |      5 | 2024-04-05 | Cash           |       1 |
|    2211 |      5 | 2024-04-05 | Cash           |      12 |
|    2201 |      8 | 2024-04-05 | Cash           |       2 |
|    2204 |     10 | 2024-04-05 | Cash           |       5 |
|    2234 |     10 | 2024-04-05 | Cash           |      35 |
|    2216 |     10 | 2024-04-05 | Cash           |      17 |
|    2208 |     10 | 2024-04-05 | Cash           |       9 |
|    2218 |     10 | 2024-04-05 | Cash           |      19 |
|    2206 |     10 | 2024-04-05 | Cash           |       7 |
|    2223 |     10 | 2024-04-05 | Cash           |      24 |
|    2224 |     10 | 2024-04-05 | Cash           |      25 |
|    2203 |     10 | 2024-04-05 | Cash           |       4 |
|    2237 |     10 | 2024-04-05 | Cash           |      38 |
|    2214 |     10 | 2024-04-05 | Cash           |      15 |
|    2217 |     15 | 2024-04-05 | Cash           |      18 |
|    2225 |     20 | 2024-04-05 | Cash           |      26 |
|    2226 |     20 | 2024-04-05 | Cash           |      27 |
|    2230 |     20 | 2024-04-05 | Cash           |      31 |
|    2231 |     20 | 2024-04-05 | Cash           |      32 |
|    2232 |     20 | 2024-04-05 | Cash           |      33 |
|    2233 |     20 | 2024-04-05 | Cash           |      34 |
|    2221 |     20 | 2024-04-05 | Cash           |      22 |
|    2207 |     20 | 2024-04-05 | Cash           |       8 |
|    2209 |     20 | 2024-04-05 | Cash           |      10 |
|    2205 |     20 | 2024-04-05 | Cash           |       6 |
|    2215 |     20 | 2024-04-05 | Cash           |      16 |
|    2222 |     25 | 2024-04-05 | Cash           |      23 |
|    2228 |     28 | 2024-04-05 | Cash           |      29 |
|    2238 |     30 | 2024-04-05 | Cash           |      39 |
|    2219 |     30 | 2024-04-05 | Cash           |      20 |
|    2220 |     30 | 2024-04-05 | Cash           |      21 |
|    2212 |     38 | 2024-04-05 | Cash           |      13 |
|    2239 |     40 | 2024-04-05 | Cash           |      40 |
|    2210 |     40 | 2024-04-05 | Cash           |      11 |
|    2227 |     40 | 2024-04-05 | Cash           |      28 |
|    2213 |     40 | 2024-04-05 | Cash           |      14 |
|    2202 |     45 | 2024-04-05 | Cash           |       3 |
|     201 |     50 | 2023-04-10 | Cash           |      12 |
|    2229 |     60 | 2024-04-05 | Cash           |      30 |
|    2236 |     80 | 2024-04-05 | Cash           |      37 |
|    2235 |     90 | 2024-04-05 | Cash           |      36 |
+---------+--------+------------+----------------+---------+
41 rows in set (0.000 sec)
```

14.Find the total number of orders assigned to each staff member:

```
MariaDB [canteen_management]> SELECT Staff_id, COUNT(Order_id) AS Total_Orders_Assigned
    -> FROM Order_assigned_to
    -> GROUP BY Staff_id;
+----------+-----------------------+
| Staff_id | Total_Orders_Assigned |
+----------+-----------------------+
|     7891 |                     8 |
|     7892 |                     6 |
|     7893 |                     7 |
|     7894 |                    11 |
|     7895 |                     8 |
+----------+-----------------------+
5 rows in set (0.000 sec)
```

15.Calculate the total revenue generated by each menu item:

```
MariaDB [canteen_management]> SELECT i.Item_name,
   ->       SUM(oi.Quantity * i.Price) AS Total_Revenue
   -> FROM Order_Item oi
   -> JOIN Item i ON oi.Item_id = i.Item_id
   -> GROUP BY i.Item_id, i.Item_name;
+----------------------+--------------+
| Item_name            | Total_Revenue |
+----------------------+--------------+
| Vada Pav             |        14.00 |
| Samosa Pav           |        28.00 |
| Vada Plate           |        22.00 |
| Samosa Plate         |        24.00 |
| Poha                 |        24.00 |
| Idli-Sambhar         |        24.00 |
| Upma                 |        25.00 |
| Upma Poha            |        28.00 |
| Medu Wada            |        26.00 |
| Batata Wada          |        29.00 |
| Misal Pav            |        62.00 |
| Vada Usal            |        29.00 |
| Ragada Samosa        |        70.00 |
| Samosa Usal          |        35.00 |
| Veg Sandwich         |        33.00 |
| Veg Chesse Sandwich  |        50.00 |
| Mumbai Sandwich      |        33.00 |
| Mumbai Cheese Sandwich |        42.00 |
| Cheese Sandwich      |        42.00 |
| Veg Fried Rice       |       150.00 |
| Veg Schezwan Rice    |        55.00 |
| Veg Hakka Noddles    |        50.00 |
| Veg Schezwan Noodles |        55.00 |
| Sada Dosa            |        23.00 |
| Masala Dosa          |        33.00 |
| Butter Sada Dosa     |        29.00 |
| Butter Masala Dosa   |        39.00 |
| Mysore Masala Dosa   |        58.00 |
| Mysore Sada Dosa     |        35.00 |
| Schezwan Sada Dosa   |        93.00 |
| Schezwan Masala Dosa |        35.00 |
| Sada Uttappa         |        48.00 |
| Onion Uttappa        |        33.00 |
| Masala Uttappa       |        33.00 |
| Chinese Dosa         |        53.00 |
| Lunch                |       192.00 |
| Tea                  |        33.00 |
| Coffe                |        15.00 |
| Chass                |        13.00 |
| Lassi                |        84.00 |
+----------------------+--------------+
```

4.SQL-DML should explore Nested queries, Join Queries, along with the complex queries, Aggregate functions etc to manipulate data for the your ER Data Model

1.Retrieve the names of users who have placed orders totaling more than the average amount spent by all users:

```
MariaDB [canteen_management]> SELECT Name
    -> FROM User
    -> WHERE User_id IN (
    ->     SELECT User_id
    ->     FROM Bill
    ->     GROUP BY User_id
    ->     HAVING SUM(Amount) > (
    ->         SELECT AVG(Total_Amount)
    ->         FROM (
    ->             SELECT User_id, SUM(Amount) AS Total_Amount
    ->             FROM Bill
    ->             GROUP BY User_id
    ->         ) AS User_Total
    ->     )
    -> );
+----------+
| Name     |
+----------+
| Amrut    |
| Chaitali |
| Dhruv    |
| Gaurav   |
| Labdhi   |
| Maithili |
| manas    |
| Riddhi   |
| Sachin   |
| Soham    |
| Tanmay   |
| Vansh    |
| Yash     |
| Yuvraj   |
+----------+
14 rows in set (0.003 sec)
```

2.Find the total number of orders placed by each user, sorted by the number of orders in descending order:

```
MariaDB [canteen_management]> SELECT u.User_id, u.Name, COUNT(o.Order_id) AS Total_Orders_Placed
    -> FROM User u
    -> LEFT JOIN Order_placed_by opb ON u.User_id = opb.User_id
    -> LEFT JOIN `Order` o ON opb.Order_id = o.Order_id
    -> GROUP BY u.User_id
    -> ORDER BY Total_Orders_Placed DESC;
+---------+------------+----------------------+
| User_id | Name       | Total_Orders_Placed  |
+---------+------------+----------------------+
|      18 | Purvi      |                    1 |
|      35 | Moksh      |                    1 |
|       3 | Amrut      |                    1 |
|      20 | Riddhi     |                    1 |
|      37 | Chaitali   |                    1 |
|       5 | Aaditya    |                    1 |
|      22 | Sahil      |                    1 |
|      39 | Labdhi     |                    1 |
|       7 | Abhi       |                    1 |
|      24 | Shon       |                    1 |
|       9 | Achal      |                    1 |
|      26 | Shubham    |                    1 |
|      11 | Dhruv      |                    1 |
|      28 | Tanmay     |                    1 |
|      13 | Maithili   |                    1 |
|      30 | Yash       |                    1 |
|      15 | Mrunmayi   |                    1 |
|      32 | Vaibhavi   |                    1 |
|      17 | Pavan      |                    1 |
|      34 | Anish      |                    1 |
|       2 | Aditi      |                    1 |
|      19 | Raj        |                    1 |
|      36 | Soham      |                    1 |
|       4 | Toro       |                    1 |
|      21 | Sachin     |                    1 |
|      38 | Pranjal    |                    1 |
|       6 | Aastha     |                    1 |
|      23 | Sarvesh    |                    1 |
|      40 | Yuvraj     |                    1 |
|       8 | Abhijit    |                    1 |
|      25 | Shravani   |                    1 |
|      10 | Anam       |                    1 |
|      27 | Sumit      |                    1 |
|      12 | Gaurav     |                    1 |
|      29 | Vansh      |                    1 |
|      14 | manas      |                    1 |
|      31 | Vedika     |                    1 |
|      16 | Nishit     |                    1 |
|      33 | Ruchi      |                    1 |
|       1 | Aayush     |                    1 |
|      45 | Jane Smith |                    0 |
+---------+------------+----------------------+
41 rows in set (0.001 sec)
```

3.Retrieve the names of all users who have not placed any orders yet:

```
MariaDB [canteen_management]> SELECT Name
    -> FROM User
    -> WHERE User_id NOT IN (
    ->     SELECT DISTINCT User_id
    ->     FROM `Order`
    -> );
Empty set (0.001 sec)
```

4.Calculate the total revenue generated from all orders placed:

```
MariaDB [canteen_management]> SELECT SUM(oi.Quantity * i.Price) AS Total_Revenue
    -> FROM Order_Item oi
    -> JOIN Item i ON oi.Item_id = i.Item_id;
+---------------+
| Total_Revenue |
+---------------+
|       1799.00 |
+---------------+
1 row in set (0.001 sec)
```

5.List all bills along with the corresponding user's name and address:

```
MariaDB [canteen_management]> SELECT b.Bill_id, u.Name AS User_Name, u.Address, b.Amount, b.Date, b.Payment_method
    -> FROM Bill b
    -> JOIN User u ON b.User_id = u.User_id;
+---------+-----------+-----------------------------------------------------------------------------+--------+------------+----------------+
| Bill_id | User_Name | Address                                                                     | Amount | Date       | Payment_method |
+---------+-----------+-----------------------------------------------------------------------------+--------+------------+----------------+
|     201 | Gaurav    | opp. Domestic Airport, Navpada, Vile Parle East, Vile Parle, Mumbai         |     50 | 2023-04-10 | Cash           |
|    2200 | Aayush    | Apollo Bandar, Colaba, Mumbai                                               |      5 | 2024-04-05 | Cash           |
|    2201 | Aditi     | Dr E Moses Rd, Gandhi Nagar, Upper Worli, Mumbai                            |      8 | 2024-04-05 | Cash           |
|    2202 | Amrut     | Juhu Tara Rd, Uditi Tarang Housing Colony, Juhu Tara, Juhu, Mumbai          |     45 | 2024-04-05 | Cash           |
|    2203 | Toro      | Sahar Airport Rd, Ashok Nagar, Andheri East, Mumbai                         |     10 | 2024-04-05 | Cash           |
|    2204 | Aaditya   | Balraj Sahani Marg, Juhu, Mumbai                                            |     10 | 2024-04-05 | Cash           |
|    2205 | Aastha    | Sakinaka Junction, Andheri - Kurla Rd, Andheri East, Mumbai                 |     20 | 2024-04-05 | Cash           |
|    2206 | Abhi      | Juhu Beach, Juhu, Mumbai, Maharashtra                                       |     10 | 2024-04-05 | Cash           |
|    2207 | Abhijit   | Opposite Mittal Industrial Estate, Andheri Kurla Rd, Andheri East, Mumbai   |     20 | 2024-04-05 | Cash           |
|    2208 | Achal     | Andheri - Kurla Rd, J B Nagar, Andheri East, Mumbai                         |     10 | 2024-04-05 | Cash           |
|    2209 | Anam      | Chhatrapati Shivaji International Airport, IA Project Rd, Navpada           |     20 | 2024-04-05 | Cash           |
|    2210 | Dhruv     | Apollo Bandar, Colaba, Mumbai, Maharashtra 400001, India                    |     40 | 2024-04-05 | Cash           |
|    2211 | Gaurav    | opp. Domestic Airport, Navpada, Vile Parle East, Vile Parle, Mumbai         |      5 | 2024-04-05 | Cash           |
|    2212 | Maithili  | Saki Vihar Rd, Murarji Nagar, Mayur Nagar, Passpoli, Powai, Mumbai          |     38 | 2024-04-05 | Cash           |
|    2213 | manas     | Sahar Airport Road, Andheri - Kurla Rd, near Mumbai International Airport   |     40 | 2024-04-05 | Cash           |
|    2214 | Mrunmayi  | 462, Senapati Bapat Marg, Lower Parel, Mumbai, Maharashtra 400013, India    |     10 | 2024-04-05 | Cash           |
|    2215 | Nishit    | Plot No.34, 21, MIDC Central Rd, near Akruti Center Point                   |     20 | 2024-04-05 | Cash           |
|    2216 | Pavan     | Oberoi Garden City, International Business Park, Yashodham, Goregaon        |     10 | 2024-04-05 | Cash           |
|    2217 | Purvi     | Nariman Point, Mumbai, Maharashtra 400021, India                            |     15 | 2024-04-05 | Cash           |
|    2218 | Raj       | X - 22, MIDC Central Rd, Hanuman Nagar                                      |     10 | 2024-04-05 | Cash           |
|    2219 | Riddhi    | Cambridge School, Karishma Chambers, Off, Sahar Rd                          |     30 | 2024-04-05 | Cash           |
|    2220 | Sachin    | 19A, Sunshine Building, Opp: Domino's Pizza, 1st Cross Road                 |     30 | 2024-04-05 | Cash           |
|    2221 | Sahil     | 333, Dr Ambedkar Rd, Pali Pathar, Khar West                                 |     20 | 2024-04-05 | Cash           |
|    2222 | Sarvesh   | 14/A, SV Rd, next to Khar Masjid, Khar, Khar Danda                          |     25 | 2024-04-05 | Cash           |
|    2223 | Shon      | 6, 1st floor, Puran Niwas Bldg. Opp Radio C                                 |     10 | 2024-04-05 | Cash           |
|    2224 | Shravani  | Sagar Fortune, Ground Floor, First Floor,Water Field Road                    |     10 | 2024-04-05 | Cash           |
|    2225 | Shubham   | 505 , Mastermind-1 , Royal Palm Estate,, Aarey Colony                       |     20 | 2024-04-05 | Cash           |
|    2226 | Sumit     | India House No.2, Near Indian Bank Pedder Road                              |     20 | 2024-04-05 | Cash           |
|    2227 | Tanmay    | Unit 39, Madhu Corporate Park, Pandurang Budhkar Marg, Worli, Mumbai        |     40 | 2024-04-05 | Cash           |
|    2228 | Vansh     | Shop No. 7, 15th Rd, Kamal Kunj, Bandra West, Mumbai                        |     28 | 2024-04-05 | Cash           |
|    2229 | Yash      | P-57, 23rd Rd, Bandra West, Mumbai, Maharashtra 400050, India               |     60 | 2024-04-05 | Cash           |
|    2230 | Vedika    | Bandra Kurla Complex, Grand Hyatt Shopping Plaza                            |     20 | 2024-04-05 | Cash           |
|    2231 | Vaibhavi  | 201, Monterossa, 2nd Floor, 90 Feet Road, Pant Nagar                        |     20 | 2024-04-05 | Cash           |
|    2232 | Ruchi     | Marine Mansion, Shop No.110, 1st Marine Street, Anandilal Podar Marg        |     20 | 2024-04-05 | Cash           |
|    2233 | Anish     | Shop No 18/A Kamdhenu Shopping Centre Lokhandwala                           |     20 | 2024-04-05 | Cash           |
|    2234 | Moksh     | Shop No. 1, Edward Apartment, Off Link Road, Evershine Nagar, Malad West    |     10 | 2024-04-05 | Cash           |
|    2235 | Soham     | Shop No 4, Parvati Bhavan Opp Dr Sanap Clinic, GD Ambedkar Marg             |     90 | 2024-04-05 | Cash           |
|    2236 | Chaitali  | Shop No.2 Jai Ambe garage, K.N.G Marg, Nr Bank of India, Chembur Naka       |     80 | 2024-04-05 | Cash           |
|    2237 | Pranjal   | Shop No. 3, Maaz Centre, Opposite The Shop Pali Naka                        |     10 | 2024-04-05 | Cash           |
|    2238 | Labdhi    |         Dunhill, Pali Hill Shop 3, Dr Ambedkar Rd, Khar West                |     30 | 2024-04-05 | Cash           |
|    2239 | Yuvraj    | Rustom & Co, Arsiwala Building, 3rd Ground floor Wodehouse Rd               |     40 | 2024-04-05 | Cash           |
+---------+-----------+-----------------------------------------------------------------------------+--------+------------+----------------+
41 rows in set (0.000 sec)
```

6.Retrieve the names of users who have spent more than Rs.10 in total:

```
MariaDB [canteen_management]> SELECT Name
    -> FROM User
    -> WHERE User_id IN (
    ->     SELECT User_id
    ->     FROM Bill
    ->     GROUP BY User_id
    ->     HAVING SUM(Amount) > 10
    -> );
+----------+
| Name     |
+----------+
| Aastha   |
| Abhijit  |
| Amrut    |
| Anam     |
| Anish    |
| Chaitali |
| Dhruv    |
| Gaurav   |
| Labdhi   |
| Maithili |
| manas    |
| Nishit   |
| Purvi    |
| Riddhi   |
| Ruchi    |
| Sachin   |
| Sahil    |
| Sarvesh  |
| Shubham  |
| Soham    |
| Sumit    |
| Tanmay   |
| Vaibhavi |
| Vansh    |
| Vedika   |
| Yash     |
| Yuvraj   |
+----------+
27 rows in set (0.000 sec)
```

7.List all items along with the total quantity ordered for each item, sorted by quantity in descending order:

```
MariaDB [canteen_management]> SELECT i.Item_name, SUM(oi.Quantity) AS Total_Quantity_Ordered
    -> FROM Order_Item oi
    -> JOIN Item i ON oi.Item_id = i.Item_id
    -> GROUP BY i.Item_id
    -> ORDER BY Total_Quantity_Ordered DESC;
+-----------------------+-------------------------+
| Item_name             | Total_Quantity_Ordered |
+-----------------------+-------------------------+
| Lunch                 |                       3 |
| Tea                   |                       3 |
| Lassi                 |                       3 |
| Veg Fried Rice        |                       3 |
| Schezwan Sada Dosa    |                       3 |
| Samosa Pav            |                       2 |
| Misal Pav             |                       2 |
| Ragada Samosa         |                       2 |
| Mysore Masala Dosa    |                       2 |
| Sada Uttappa          |                       2 |
| Masala Uttappa        |                       1 |
| Chinese Dosa          |                       1 |
| Vada Plate            |                       1 |
| Samosa Plate          |                       1 |
| Poha                  |                       1 |
| Coffe                 |                       1 |
| Idli-Sambhar          |                       1 |
| Chass                 |                       1 |
| Upma                  |                       1 |
| Upma Poha             |                       1 |
| Medu Wada             |                       1 |
| Batata Wada           |                       1 |
| Vada Usal             |                       1 |
| Samosa Usal           |                       1 |
| Veg Sandwich          |                       1 |
| Veg Chesse Sandwich   |                       1 |
| Mumbai Sandwich       |                       1 |
| Mumbai Cheese Sandwich |                       1 |
| Cheese Sandwich       |                       1 |
| Veg Schezwan Rice     |                       1 |
| Veg Hakka Noddles     |                       1 |
| Veg Schezwan Noodles  |                       1 |
| Sada Dosa             |                       1 |
| Masala Dosa           |                       1 |
| Butter Sada Dosa      |                       1 |
| Butter Masala Dosa    |                       1 |
| Mysore Sada Dosa      |                       1 |
| Schezwan Masala Dosa  |                       1 |
| Onion Uttappa         |                       1 |
| Vada Pav              |                       1 |
+-----------------------+-------------------------+
40 rows in set (0.001 sec)
```

8.Retrieve the names of staff members who have not been assigned any orders yet:

```
MariaDB [canteen_management]> SELECT Staff_Name
    -> FROM Staff
    -> WHERE Staff_id NOT IN (
    ->     SELECT DISTINCT Staff_id
    ->     FROM Order_assigned_to
    -> );
Empty set (0.000 sec)
```

9.Calculate the average price per item in the menu:

```
MariaDB [canteen_management]> SELECT AVG(Price) AS Average_Item_Price
    -> FROM Item;
+--------------------+
| Average_Item_Price |
+--------------------+
|          32.450000 |
+--------------------+
1 row in set (0.000 sec)
```

10.List all orders along with the item details and quantities, sorted by order ID in ascending order:

```
MariaDB [canteen_management]> SELECT o.Order_id, i.Item_name, oi.Quantity
    -> FROM `Order` o
    -> JOIN Order_Item oi ON o.Order_id = oi.Order_id
    -> JOIN Item i ON oi.Item_id = i.Item_id
    -> ORDER BY o.Order_id ASC;
+----------+-----------------------+----------+
| Order_id | Item_name             | Quantity |
+----------+-----------------------+----------+
|       13 | Samosa Plate          |        1 |
|       20 | Medu Wada             |        1 |
|       23 | Samosa Usal           |        1 |
|       59 | Vada Plate            |        1 |
|       67 | Upma Poha             |        1 |
|       74 | Ragada Samosa         |        2 |
|      123 | Samosa Pav            |        2 |
|      215 | Upma                  |        1 |
|      234 | Vada Usal             |        1 |
|      241 | Vada Pav              |        1 |
|      255 | Idli-Sambhar          |        1 |
|      288 | Misal Pav             |        2 |
|      323 | Tea                   |        3 |
|      324 | Poha                  |        1 |
|      331 | Veg Sandwich          |        1 |
|      362 | Lassi                 |        3 |
|      444 | Batata Wada           |        1 |
|      523 | Chass                 |        1 |
|      652 | Coffe                 |        1 |
|      661 | Veg Hakka Noddles     |        1 |
|      663 | Veg Schezwan Rice     |        1 |
|      664 | Lunch                 |        3 |
|      670 | Mysore Sada Dosa      |        1 |
|      672 | Mysore Masala Dosa    |        2 |
|      694 | Chinese Dosa          |        1 |
|      700 | Veg Fried Rice        |        3 |
|      703 | Butter Masala Dosa    |        1 |
|      704 | Masala Uttappa        |        1 |
|      710 | Veg Chesse Sandwich   |        1 |
|      713 | Cheese Sandwich       |        1 |
|      717 | Mumbai Cheese Sandwich |        1 |
|      721 | Mumbai Sandwich       |        1 |
|      731 | Butter Sada Dosa      |        1 |
|      800 | Masala Dosa           |        1 |
|      844 | Veg Schezwan Noodles  |        1 |
|      849 | Sada Dosa             |        1 |
|      878 | Onion Uttappa         |        1 |
|      879 | Schezwan Masala Dosa  |        1 |
|      909 | Schezwan Sada Dosa    |        3 |
|      939 | Sada Uttappa          |        2 |
+----------+-----------------------+----------+
```

11.Retrieve the names of all users who have placed orders for items containing 'coffee' in the contents:

```
MariaDB [canteen_management]> SELECT DISTINCT u.Name
    -> FROM User u
    -> JOIN Order_placed_by opb ON u.User_id = opb.User_id
    -> JOIN `Order` o ON opb.Order_id = o.Order_id
    -> JOIN Order_Item oi ON o.Order_id = oi.Order_id
    -> JOIN Item i ON oi.Item_id = i.Item_id
    -> WHERE i.Contents LIKE '%coffee%';
+---------+
| Name    |
+---------+
| Pranjal |
+---------+
1 row in set (0.001 sec)
```

12.List all orders placed by users who have '10' in their contact number:

```
MariaDB [canteen_management]> SELECT o.Order_id, u.Name AS User_Name, u.Contact
    -> FROM `Order` o
    -> JOIN Order_placed_by opb ON o.Order_id = opb.Order_id
    -> JOIN User u ON opb.User_id = u.User_id
    -> WHERE u.Contact LIKE '%10';
+----------+-----------+------------+
| Order_id | User_Name | Contact    |
+----------+-----------+------------+
|      362 | Yuvraj    | 7620489210 |
+----------+-----------+------------+
1 row in set (0.000 sec)
```

13.Retrieve the names of users who have placed orders for items with a price higher than the average price of all items:

```
MariaDB [canteen_management]> SELECT DISTINCT u.Name
    -> FROM User u
    -> JOIN Order_placed_by opb ON u.User_id = opb.User_id
    -> JOIN `Order` o ON opb.Order_id = o.Order_id
    -> JOIN Order_Item oi ON o.Order_id = oi.Order_id
    -> JOIN Item i ON oi.Item_id = i.Item_id
    -> WHERE i.Price > (SELECT AVG(Price) FROM Item);
+----------+
| Name     |
+----------+
| Maithili |
| manas    |
| Mrunmayi |
| Nishit   |
| Pavan    |
| Purvi    |
| Raj      |
| Riddhi   |
| Sachin   |
| Sahil    |
| Sarvesh  |
| Shravani |
| Sumit    |
| Vansh    |
| Vedika   |
| Ruchi    |
| Anish    |
| Moksh    |
| Soham    |
+----------+
19 rows in set (0.001 sec)
```

14.Find the total number of orders assigned to each staff member, sorted by staff ID in ascending order:

```
MariaDB [canteen_management]> SELECT Staff_id, COUNT(Order_id) AS Total_Orders_Assigned
    -> FROM Order_assigned_to
    -> GROUP BY Staff_id
    -> ORDER BY Staff_id ASC;
+----------+-----------------------+
| Staff_id | Total_Orders_Assigned |
+----------+-----------------------+
|     7891 |                     8 |
|     7892 |                     6 |
|     7893 |                     7 |
|     7894 |                    11 |
|     7895 |                     8 |
+----------+-----------------------+
5 rows in set (0.000 sec)
```

15.Retrieve the names of users who have placed orders for items with a price higher than the average price of items containing 'coffee' in their contents:

```
MariaDB [canteen_management]> SELECT DISTINCT u.Name
    -> FROM User u
    -> JOIN Order_placed_by opb ON u.User_id = opb.User_id
    -> JOIN `Order` o ON opb.Order_id = o.Order_id
    -> JOIN Order_Item oi ON o.Order_id = oi.Order_id
    -> JOIN Item i ON oi.Item_id = i.Item_id
    -> WHERE i.Price > (
    ->     SELECT AVG(Price)
    ->     FROM Item
    ->     WHERE Contents LIKE '%coffee%'
    -> );
+----------+
| Name     |
+----------+
| Amrut    |
| Toro     |
| Aaditya  |
| Aastha   |
| Abhi     |
| Abhijit  |
| Achal    |
| Anam     |
| Dhruv    |
| Gaurav   |
| Maithili |
| manas    |
| Mrunmayi |
| Nishit   |
| Pavan    |
| Purvi    |
| Raj      |
| Riddhi   |
| Sachin   |
| Sahil    |
| Sarvesh  |
| Shon     |
| Shravani |
| Shubham  |
| Sumit    |
| Tanmay   |
| Vansh    |
| Yash     |
| Vedika   |
| Vaibhavi |
| Ruchi    |
| Anish    |
| Moksh    |
| Soham    |
| Yuvraj   |
+----------+
35 rows in set (0.001 sec)
```

Conclusion:

Entity-Relationship Model Representation:

- SQL allows us to create tables that directly correspond to entities in our ER model. For example, tables like Customer, Order, and Product represent entities, while OrderItem represents a relationship between Order and Product.

Defining Relationships:

- SQL constraints such as foreign keys establish relationships between tables, ensuring referential integrity. This means that we can't insert data into the OrderItem table unless the corresponding OrderID and ProductID exist in the Order and Product tables, respectively.

Efficient Data Analysis:

Retrieving Data:

- SQL's SELECT statement allows us to retrieve data based on specific criteria, such as retrieving all orders made by a particular customer or all orders that exceed a certain total amount.

Aggregating Data:

- SQL's aggregate functions like AVG() and SUM() enable us to perform calculations on groups of data. For instance, we can calculate the average order amount or the total amount spent by each customer.

Analyzing Data with Complex Queries:

- SQL supports complex queries involving JOINs, subqueries, and conditional logic. This allows us to perform advanced analysis, such as retrieving order details along with customer information or finding orders where the total amount exceeds the average.

Data Manipulation with DML:

- SQL's DML statements like INSERT, UPDATE, and DELETE facilitate the manipulation of data. For instance, we can insert new customers, update order details, or delete outdated records.

SQL's capabilities to help us to efficiently manage and analyze complex datasets according to the logical schema of our ER model. By leveraging SQL's features for defining relationships, retrieving data, aggregating information, and performing complex queries, we can gain valuable insights into our data and make informed decisions. Whether it's analyzing customer behavior, tracking sales trends, or optimizing inventory management, SQL provides the tools necessary to extract meaningful information from our data model.