

Restaurant Orders Analysis Project (MySQL)

Tools: MySQL

Project Type: SQL Data Analysis / Business Insights

Shashi Yadav

1. Project Objective

The goal of this project is to analyze restaurant order data using MySQL and generate business insights such as:

- Total revenue and order trends
- Top-selling menu items
- Revenue by category and payment method
- Customer spending and ranking

2. Database Schema

Database Name: `restaurant`

Tables Used

`customer`

`menu`

`orders`

Relationship

- `orders.customer_id` → references `customer.customer_id`
- `orders.item_id` → references `menu.item_id`

3. Query

1) Total Revenue

```
SELECT ROUND(SUM(m.price * o.quantity),2) AS total_revenue  
FROM orders o  
JOIN menu m ON o.item_id = m.item_id;
```

```
mysql> SELECT ROUND(SUM(m.price * o.quantity),2) AS total_revenue  
-> FROM orders o  
-> JOIN menu m ON o.item_id = m.item_id;  
+-----+  
| total_revenue |  
+-----+  
|      3950.00 |  
+-----+  
1 row in set (0.01 sec)
```

2) Top 5 Items by Revenue

```
SELECT m.category,  
       ROUND(SUM(m.price * o.quantity),2) AS revenue  
FROM orders o  
JOIN menu m ON o.item_id = m.item_id  
GROUP BY m.category  
ORDER BY revenue DESC;
```

```
mysql> SELECT m.item_name,
->           ROUND(SUM(m.price * o.quantity),2) AS revenue
->     FROM orders o
->   JOIN menu m ON o.item_id = m.item_id
->   GROUP BY m.item_name
->   ORDER BY revenue DESC
->   LIMIT 5;
+-----+-----+
| item_name      | revenue |
+-----+-----+
| Pizza          | 750.00 |
| Chicken Biryani | 720.00 |
| Ice Cream      | 630.00 |
| Cold Coffee    | 550.00 |
| Veg Burger     | 480.00 |
+-----+-----+
5 rows in set (0.00 sec)
```

3) Category-wise Revenue

```
SELECT m.category,
       ROUND(SUM(m.price * o.quantity),2) AS revenue
FROM orders o
JOIN menu m ON o.item_id = m.item_id
GROUP BY m.category
ORDER BY revenue DESC;
```

```
mysql> SELECT m.category,
->           ROUND(SUM(m.price * o.quantity),2) AS revenue
->     FROM orders o
->   JOIN menu m ON o.item_id = m.item_id
->   GROUP BY m.category
->   ORDER BY revenue DESC;
+-----+-----+
| category | revenue |
+-----+-----+
| Fast Food | 1230.00 |
| Main Course | 920.00 |
| Dessert | 630.00 |
| Beverage | 550.00 |
| South Indian | 400.00 |
| Italian | 220.00 |
+-----+
6 rows in set (0.01 sec)
```

4) Customer Ranking (Window Function)

```
SELECT c.customer_name,
       ROUND(SUM(m.price * o.quantity),2) AS total_spent,
       RANK() OVER (ORDER BY SUM(m.price * o.quantity) DESC) AS
spend_rank
FROM orders o
JOIN customer c ON o.customer_id = c.customer_id
JOIN menu m ON o.item_id = m.item_id
GROUP BY c.customer_name;
```

```
mysql> SELECT c.customer_name,
   ->           ROUND(SUM(m.price * o.quantity),2) AS total_spent,
   ->           RANK() OVER (ORDER BY SUM(m.price * o.quantity) DESC) AS spend_rank
k
   -> FROM orders o
   -> JOIN customer c ON o.customer_id = c.customer_id
   -> JOIN menu m ON o.item_id = m.item_id
   -> GROUP BY c.customer_name;
+-----+-----+-----+
| customer_name | total_spent | spend_rank |
+-----+-----+-----+
| Amit          |    760.00  |      1 |
| Priya         |    740.00  |      2 |
| Shashi        |    700.00  |      3 |
| Sneha         |    670.00  |      4 |
| Rahul          |    480.00  |      5 |
| Karthik       |    400.00  |      6 |
| Neha          |    200.00  |      7 |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

5) Order Bill View Output

```
SELECT * FROM order_bill_view ORDER BY order_date, order_id;
```