RDBMS TASK 5

Food Booking Application

1. Retrieve the names and locations of restaurants with a rating of 4.5 or higher.

```
SELECT c.first_name, c.last_name, COUNT(o.order_id) AS total_orders
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
group BY c.phone
```

---

2. Find the total number of orders placed by each customer.

```
SELECT  Customers.FirstName,  Customers.LastName,COUNT(Orders.OrderID)  AS
TotalOrders
FROM Customers
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID
GROUP BY Customers.CustomerID, Customers.FirstName, Customers.LastName;
```

3.List all restaurants offering "Italian" cuisine in "Mumbai".

```
SELECT product_name, SUM(od.quantity*unit_price) AS total_sum
from products p
JOIN orderdetails od ON p.product_id = od.product_id
GROUP BY p.product_name
```

---

4. Calculate the total revenue generated by each restaurant from completed orders.

```
SELECT c.first_name, c.last_name,o.order_date
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
WHERE o.order_date >= CURDATE() - INTERVAL 30 DAY
ORDER BY o.order_date DESC
```

---

5. Retrieve the most recent order placed by each customer.

```
SELECT c.first_name, c.last_name, sum(o.total_amount ) AS total_amount
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.first_name, c.last_name
```

6. List customers who have not placed any orders yet.

```
SELECT P.category, SUM(OD.quantity)
FROM products P
JOIN orderdetails OD ON P.product_id = OD.product_id
GROUP BY P.category
```

7. Identify the most reviewed restaurants.

```
SELECT OD.order_id
FROM orderdetails OD
JOIN orders o  ON OD.order_id = o.order_id
WHERE o.order_status = 'pending'
GROUP BY OD.order_id
```

8. Find the most preferred payment method.

```
SELECT SUM(o.total_amount) AS total_order_amount,
COUNT(o.order_id) AS number_of_orders,
SUM(o.total_amount) / COUNT(o.order_id) AS average_order_value
FROM Orders o
```

9. List the top 5 restaurants by total revenue.

```
SELECT DISTINCT c.first_name, c.last_name, sum(o.total_amount) AS
total_amount_spent FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id
ORDER BY total_amount_spent DESC
 LIMIT 5
```

10. Show the details of all cancelled orders along with the customer's and restaurant's names.

```
SELECT p.product_id, p.product_name FROM products p
LEFT JOIN orderdetails od ON p.product_id = od.product_id
WHERE od.order_id IS NULL;
```