WsCube Tech™
System For Satisfaction

SWIGGY

ADVANCED SQL PROJECT

PRESENTED BY : Manik Gupta








SWIGGY

About swiggy

Founded in India in 2014, SWIGGY is a prominent food delivery platform that has revolutionized dining with its user-friendly app, extensive restaurant options, and quick delivery services. In addition to food, it has expanded into grocery deliveries, adapting to the competitive market with continuous innovation.

Display all customers who live in 'Delhi'

```sql
SELECT
    customer_id, name, city
FROM
    customers
WHERE
    city = 'Delhi';
```

Find the average rating of all restaurants in 'Mumbai'

```sql
SELECT
    ROUND(AVG(rating), 2)
FROM
    restaurants
WHERE
    city = 'Mumbai';
```

List all customers who have placed at least one order.

```
SELECT DISTINCT
    customers.customer_id, customers.name
FROM
    customers
        INNER JOIN
    orders ON customers.customer_id = orders.customer_id;
```








Display the total number of orders placed by each customer.

```sql
SELECT
    customers.customer_id,
    customers.name,
    COUNT(orders.order_id) AS total_orders
FROM
    customers
        LEFT JOIN
    orders ON customers.customer_id = orders.customer_id
GROUP BY customers.customer_id , customers.name;
```

Find the total revenue generated by each restaurant.

```sql
SELECT
    restaurants.restaurant_id,
    restaurants.name,
    COALESCE(SUM(orders.total_amount), 0) AS revenue
FROM
    restaurants
        LEFT JOIN
    orders ON restaurants.restaurant_id = orders.restaurant_id
GROUP BY restaurants.restaurant_id , restaurants.name;
```








Find the top 5 restaurants with the highest average rating.

```sql
SELECT
    restaurant_id, name, AVG(rating) AS avg_rating
FROM
    restaurants
GROUP BY restaurant_id , name
ORDER BY avg_rating DESC
LIMIT 5;
```

Display all customers who have never placed an order.

Method-1

```sql
SELECT
    customers.customer_id, customers.name
FROM
    customers
        LEFT JOIN
    orders ON customers.customer_id = orders.customer_id
WHERE
    orders.customer_id IS NULL;
```

Method-2

```sql
SELECT
    customers.customer_id,
    customers.name,
    COUNT(orders.order_id) AS number_of_orders
FROM
    customers
        LEFT JOIN
    orders ON customers.customer_id = orders.customer_id
GROUP BY customers.customer_id , customers.name
HAVING number_of_orders = 0;
```

Find the number of orders placed by each customer in 'Mumbai'.

```
SELECT
    customers.customer_id,
    customers.name,
    COUNT(orders.order_id)
FROM
    customers
        LEFT JOIN
    orders ON customers.customer_id = orders.customer_id
WHERE
    customers.city = 'Mumbai'
GROUP BY customers.customer_id , customers.name;
```

Display all orders placed in the last 30 days.

```sql
SELECT
    *
FROM
    orders
WHERE
    order_date >= CURDATE() - INTERVAL 30 DAY;
```








List all delivery partners who have completed more than 1 delivery

```sql
SELECT
    deliverypartners.partner_id,
    deliverypartners.name,
    COUNT(deliveryupdates.order_id)
FROM
    deliverypartners
        JOIN
    orderdelivery ON deliverypartners.partner_id = orderdelivery.partner_id
        JOIN
    deliveryupdates ON orderdelivery.order_delivery_id = deliveryupdates.delivery_id
WHERE
    deliveryupdates.status = 'Delivered'
GROUP BY deliverypartners.partner_id , deliverypartners.name;
```








Find the customers who have placed orders on exactly three different days.

```sql
SELECT
    customers.customer_id,
    customers.name,
    COUNT(orders.order_date)
FROM
    customers
        JOIN
    orders ON customers.customer_id = orders.customer_id
GROUP BY customers.customer_id , customers.name
HAVING COUNT(DISTINCT orders.order_date) = 3;
```








Find the delivery partner who has worked with the most different customers.

```sql
SELECT
    deliverypartners.partner_id,
    deliverypartners.name,
    COUNT(DISTINCT orders.customer_id) customer_count
FROM
    deliverypartners
        JOIN
    orderdelivery ON deliverypartners.partner_id = orderdelivery.partner_id
        JOIN
    orders ON orderdelivery.order_id = orders.order_id
GROUP BY deliverypartners.partner_id , deliverypartners.name
ORDER BY customer_count DESC
LIMIT 1;
```








Identify customers who have the same city and have placed orders at the same restaurants, but on different dates.

```sql
SELECT DISTINCT
    c1.name AS customer1,
    c2.name AS customer2,
    c1.city,
    o1.restaurant_id,
    r.name,
    DATE(o1.order_date) AS order_date1,
    DATE(o2.order_date) AS order_date2
FROM
    customers c1
        JOIN
    orders o1 ON c1.customer_id = o1.customer_id
        JOIN
    customers c2 ON c1.city = c2.city
        JOIN
    orders o2 ON c2.customer_id = o2.customer_id
        JOIN
    restaurants r ON r.restaurant_id = o1.restaurant_id
WHERE
    o1.restaurant_id = o2.restaurant_id
        AND DATE(o1.order_date) <> DATE(o2.order_date)
        AND c1.customer_id <> c2.customer_id
ORDER BY c1.city , o1.restaurant_id , order_date1;
```

Thank You!

www.linkedin.com/in/isthatmanik