SWIGGY

SQL SALES ANALYSIS

BY: POORNIMA SAXENA

ABOUT SWIGGY

Swiggy is an online food ordering & delivery company that operates in india , was founded in 2014. It is based in banglore India ,was founded by Nandan Reddy, Sriharsha Majety, and Rahul Jaimini. it operates in 580 indian cities .

Swiggy partners with hundreds of thousands uses machine learning (ML) technology and processes terabytes of data every day. sands of restaurants,

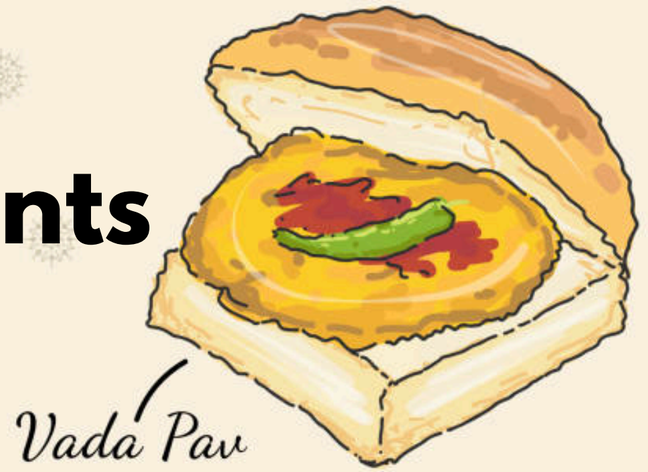
PROBLEM STATEMENT

Swiggy seeks insights from its SQL dataset. Implement sophisticated SQL queries with intricate joins for in-depth analysis and strategic decision-making.

Display all customers who live in 'Delhi'

```sql
SELECT
    *
FROM
    customers
WHERE
    city = 'Delhi';
```

Find the average rating of all restaurants in 'Mumbai'.

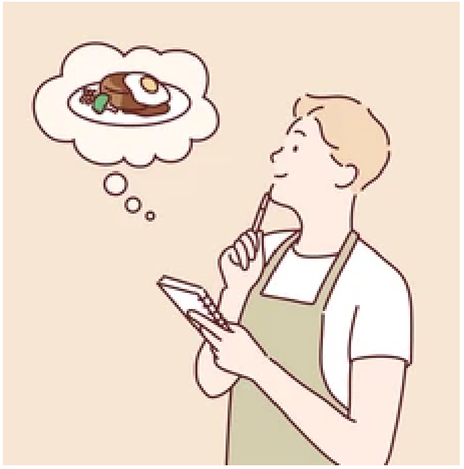



```sql
SELECT
    ROUND(AVG(rating), 2) AS avg_rating
FROM
    restaurants
WHERE
    city = 'mumbai';
```

List all customers who have placed at least one order.

```sql
SELECT
    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) total_order
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)
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) 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.

```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;
```

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


```sql
SELECT
    customers.customer_id,
    customers.name,
    COUNT(orders.order_id)
FROM
    customers
        LEFT JOIN
    orders ON orders.customer_id = customers.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


```
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 = 'Deliverd'
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 c1. name as customer, c2. name as customer2,
 c1.city as city1, c2.city as c2,restaurants.name
 from customers as c1
 join orders as 01
 on C1.customer_id = 01.customer_id
 join orders as 02
 on 02.restaurant_id = 02.restaurant_id
 join customers as c2
 on c1.city = c2.city and
 c1.name <> c2.name and
 o2.customer_id = c2.customer_id
 join restaurants
 on 01. restaurant_id = restaurants.restaurant_id
 where 01.orderdate <> 02.orderdate;
```

THANK YOU

Poornima Saxena