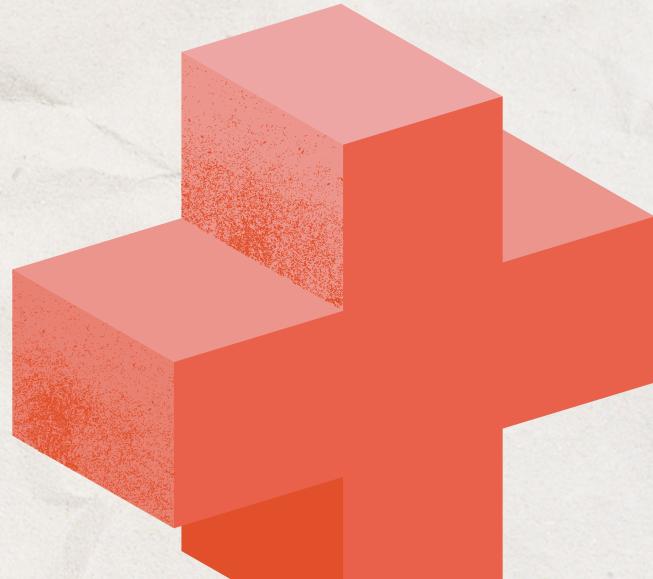


Analyzing Swiggy Data with → SQL

Azam Mustufa

August 25, 2024



About Swiggy



Swiggy is an Indian online food ordering and delivery giant. Founded in 2014, it connects customers with a vast network of restaurants, offering diverse cuisines and convenient delivery options. Headquartered in Bangalore, Swiggy operates in over 500 Indian cities. Beyond food delivery, Swiggy has expanded its services to include grocery delivery through Instamart and same-day package delivery via Genie. It has become an integral part of India's food tech landscape, revolutionizing how people order and experience food.

Display all customers who live in 'Delhi'.

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

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

```
SELECT
    AVG(rating) as avg_rating
FROM
    restaurants
WHERE
    city = 'Mumbai'; 102ms
```

List all customers who have placed at least one order.

```
SELECT
    customers.customer_id,
    customers.name,
    COUNT(orders.order_id) AS order_count
FROM
    customers
    INNER JOIN orders ON customers.customer_id = orders.customer_id
GROUP BY
    customers.customer_id,
    customers.name; 71ms
```

Display the total number of orders placed by each customer.

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

Find the total revenue generated by each restaurant.

```
SELECT
    orders.restaurant_id,
    restaurants.name,
    SUM(orders.total_amount) AS revenue
FROM
    orders
    LEFT JOIN restaurants ON orders.restaurant_id = restaurants.restaurant_id
GROUP BY
    orders.restaurant_id, restaurants.name
ORDER BY revenue DESC;  69ms
```

Find the top 5 restaurants with the highest average rating.

```
SELECT
    name,
    rating
FROM
    restaurants
WHERE
    rating IS NOT NULL
ORDER BY
    rating DESC
LIMIT 5;| 1.3s
```

Display all customers who have never placed an order.

```
SELECT  
    customers.customer_id,  
    customers.name  
FROM  
    customers  
    LEFT JOIN orders ON customers.customer_id = orders.customer_id  
WHERE  
    orders.order_id IS NULL; | 72ms
```

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

```
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
WHERE
    customers.city = 'Mumbai'
GROUP BY
    customers.customer_id,
    customers.name; 68ms
```

Display all orders placed in the last 30 days.

```
SELECT
  *
FROM
  orders
WHERE
  order_date >= CURRENT_DATE - INTERVAL '30 days';
```

List all delivery partners who have completed more than 1 delivery

```
SELECT
    deliverypartners.partner_id,
    deliverypartners.name,
    COUNT(orderdelivery.order_delivery_id) AS orders_delivered
FROM
    deliverypartners
    INNER JOIN orderdelivery ON orderdelivery.partner_id = deliverypartners.partner_id
GROUP BY
    deliverypartners.partner_id,
    deliverypartners.name
HAVING
    COUNT(orderdelivery.order_delivery_id) > 1; 70ms
```

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

```
SELECT
    customers.customer_id,
    customers.name
FROM
    customers
    INNER 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.

```
SELECT
    deliverypartners.partner_id,
    deliverypartners.name,
    COUNT(DISTINCT orders.customer_id) AS unique_customers
FROM
    deliverypartners
    INNER JOIN orderdelivery ON orderdelivery.partner_id = deliverypartners.partner_id
    INNER JOIN orders ON orderdelivery.order_id = orders.order_id
GROUP BY
    deliverypartners.partner_id,
    deliverypartners.name;
```

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

```
SELECT
    c1.customer_id,
    c1.name AS customer_name,
    c1.city AS customer_city,
    r.name AS restaurant_name
FROM
    customers c1
    JOIN orders o1 ON c1.customer_id = o1.customer_id
    JOIN restaurants r ON r.restaurant_id = o1.restaurant_id
WHERE
    EXISTS (
        SELECT
            1
        FROM
            customers c2
            JOIN orders o2 ON c2.customer_id = o2.customer_id
        WHERE
            c1.city = c2.city
            AND c1.customer_id <> c2.customer_id
            AND o1.restaurant_id = o2.restaurant_id
            AND o1.order_date <> o2.order_date
    )
GROUP BY
    c1.customer_id,
    c1.name,
    c1.city,
    r.name;
```



Thank You.

