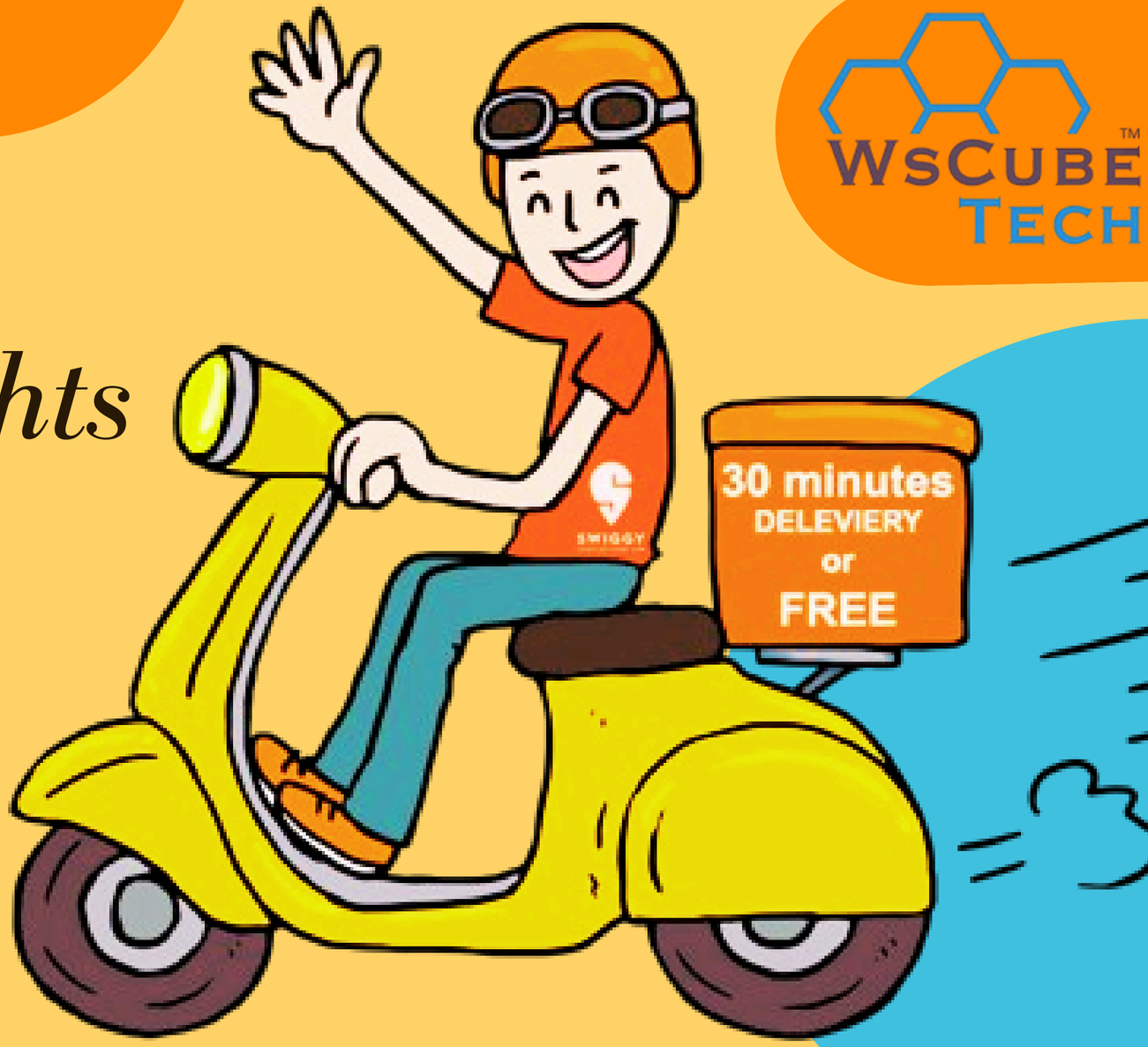
SWIGGY

WsCube Tech™

Strategic Insights for Swiggy

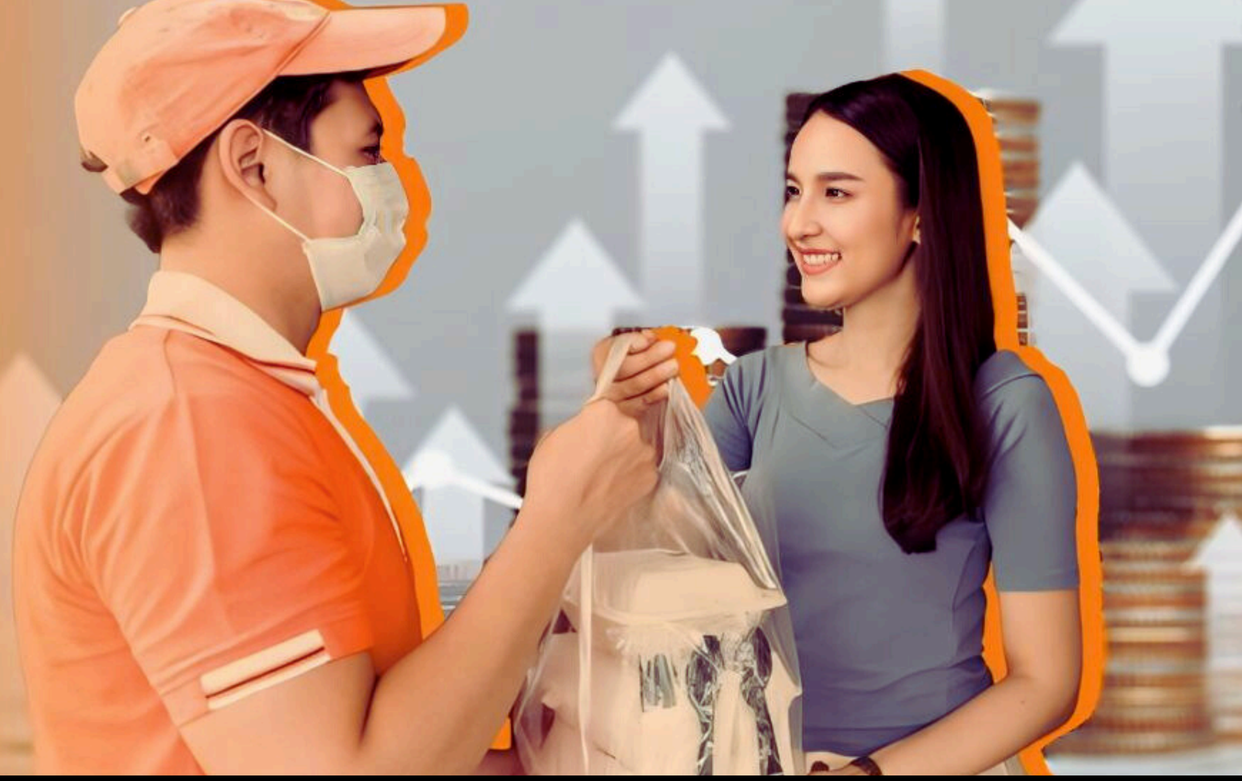


Presented By Keshab Karmakar

SWIGGY

Table of CONTENT

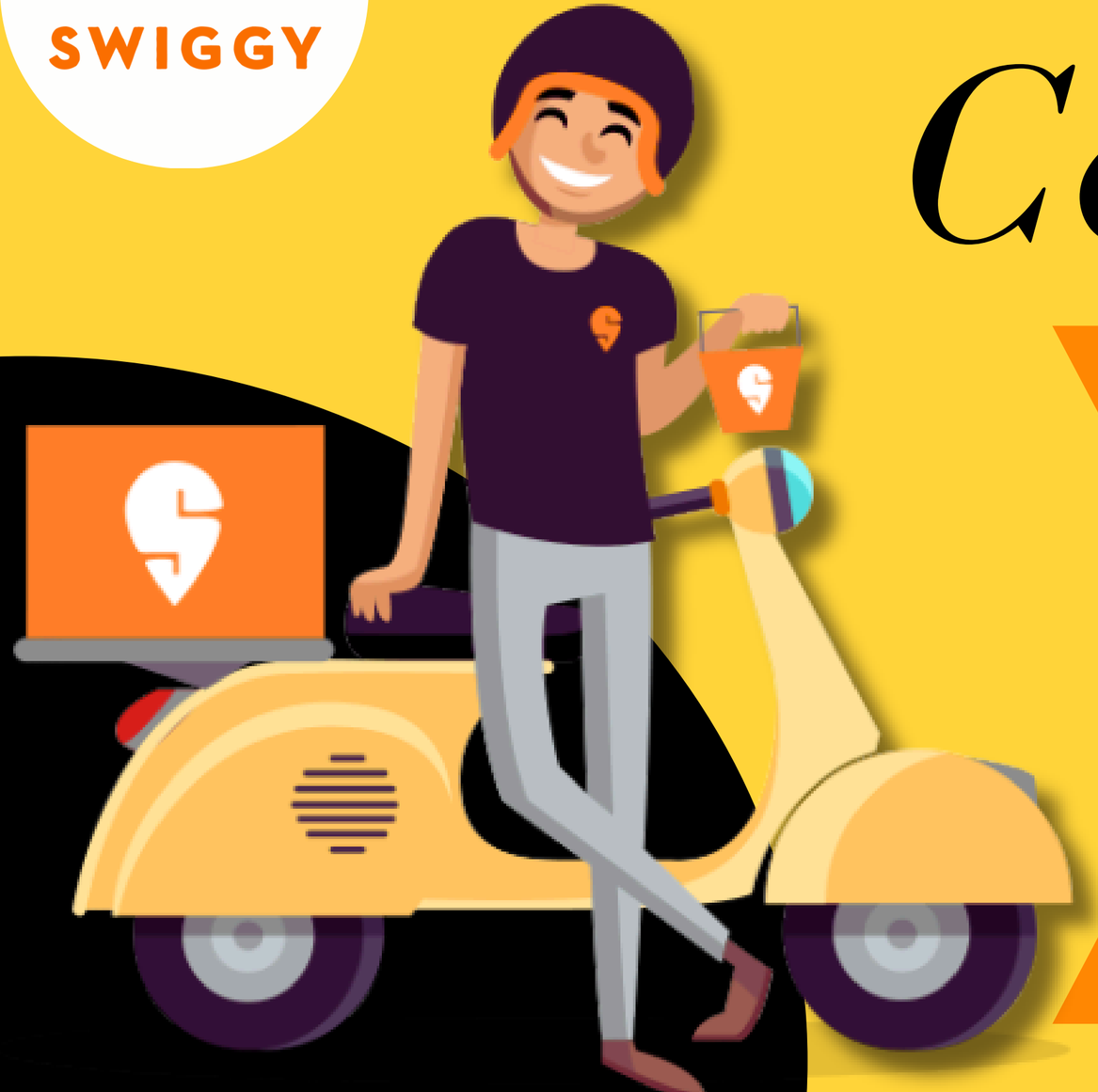
SWIGGY

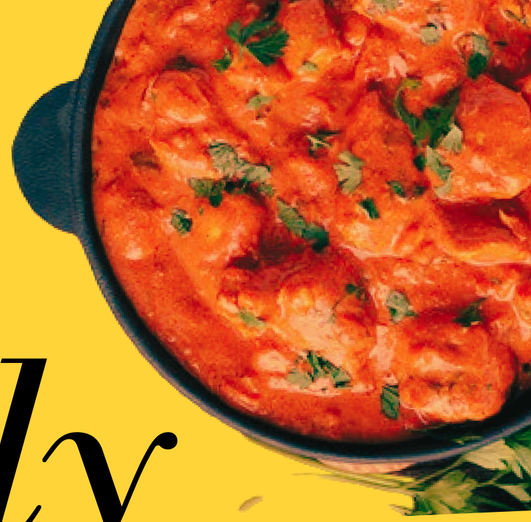


Introduction

Swiggy is India’s top food delivery platform, connecting customers with a vast range of restaurants. Since 2014, it has made dining easier with fast delivery, a user-friendly app, and a focus on customer satisfaction.

SWIGGY



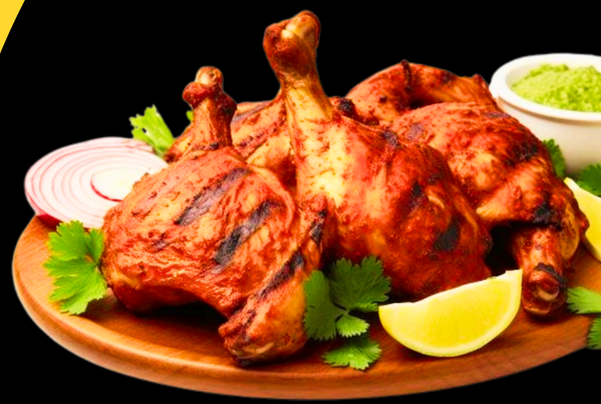


Case Study

- Swiggy, a leading online food delivery platform, aims to gain deeper insights from its extensive SQL dataset. By implementing sophisticated SQL queries, Swiggy can conduct in-depth analyses and make strategic decisions. The following case study explores various SQL queries used to derive meaningful insights from Swiggy's data.

SWIGGY

Display all customers who live in 'Delhi'



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

This query identifies all customers residing in Delhi, allowing Swiggy to focus on region-specific marketing strategies.

SWIGGY

Find the average Rating of all restaurants in 'Mumbai'

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

This helps Swiggy assess the overall customer satisfaction in Mumbai and identify areas for improvement

SWIGGY

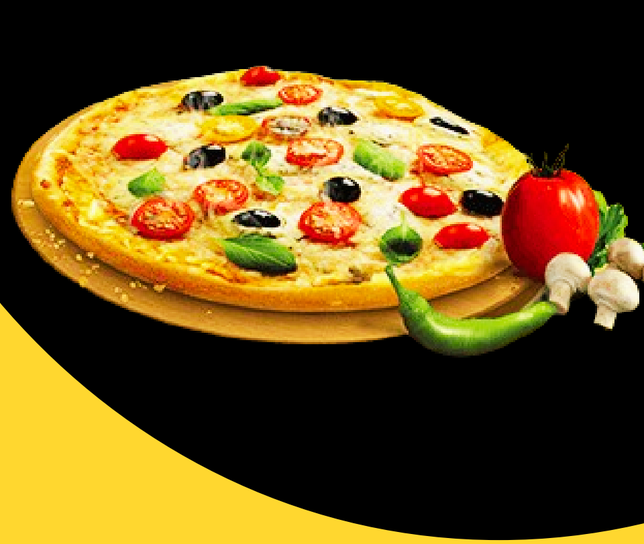
List all customers who have placed at least one order


```sql
SELECT
    distinct customers.name
FROM
    customers
        INNER JOIN
    orders ON customers.customer_id = orders.customer_id;
```

This query lists all active customers, helping Swiggy identify its core user base

SWIGGY

Display the Total Number of Orders Placed by Each Customer



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

By understanding customer order frequency, Swiggy can personalize promotions and rewards

SWIGGY

Find the Total Revenue Generated by Each Restaurant

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

This query allows Swiggy to identify top-performing restaurants and potential partners for exclusive deals

SWIGGY

Find the Top 5 Restaurants with the Highest Average Rating

```sql
SELECT
    restaurants.name, restaurants.rating
FROM
    restaurants
ORDER BY rating DESC
LIMIT 5;
```

Highlighting top-rated restaurants helps Swiggy enhance customer experience by promoting quality dining options

SWIGGY

Display All Customers Who Have Never Placed an Order

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

Identifying inactive customers enables Swiggy to target them with re-engagement campaigns

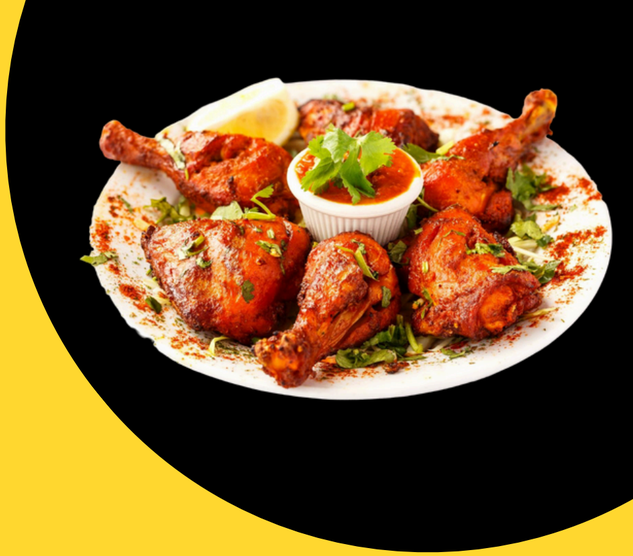
SWIGGY

Find the Number of OrdersPlaced by Each Customer in 'Mumbai'


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

This helps Swiggy understand customer activity within a specific city for localized marketing efforts

SWIGGY



Display All Orders Placed in the Last 30 Days

```sql
SELECT
    *
FROM
    orders
WHERE
    DATEDIFF(NOW(), order_date) <= 30;
```

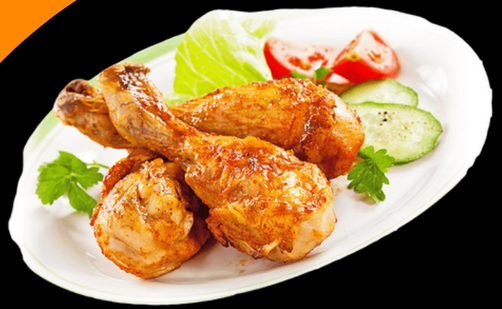
Analyzing recent orders allows Swiggy to track current trends and make timely business decisions

SWIGGY

List All Delivery Partners Who Have Completed More Than 1 Delivery

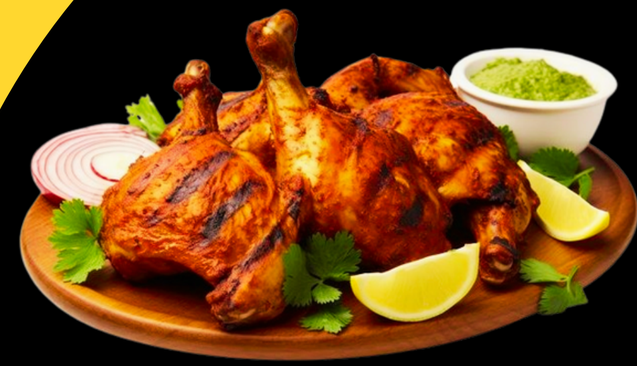
```sql
SELECT
    deliverypartners.name,
    COUNT(orderdelivery.partner_id) AS total_delibery
FROM
    deliverypartners
        JOIN
    orderdelivery ON deliverypartners.partner_id = orderdelivery.partner_id
        JOIN
    deliveryupdates ON orderdelivery.order_id = deliveryupdates.order_id
WHERE
    deliveryupdates.status <> 'Failed'
GROUP BY deliverypartners.name
HAVING total_delibery > 1;
```

Understanding delivery partner activity helps Swiggy optimize its logistics network



SWIGGY

Find the Customers Who Have Placed Orders on Exactly Three Different Days

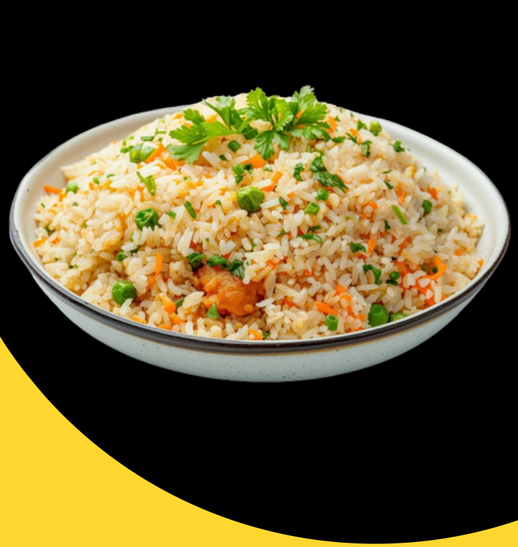


```sql
SELECT
    customers.name, COUNT(DISTINCT orders.order_date)
FROM
    customers
        JOIN
    orders ON customers.customer_id = orders.customer_id
GROUP BY customers.customer_id
HAVING COUNT(DISTINCT orders.order_date) = 3;
```

Identifying customers with specific ordering patterns can help Swiggy tailor promotions for consistent engagement

SWIGGY

Find the Delivery Partner Who Has Worked with the Most Different Customers

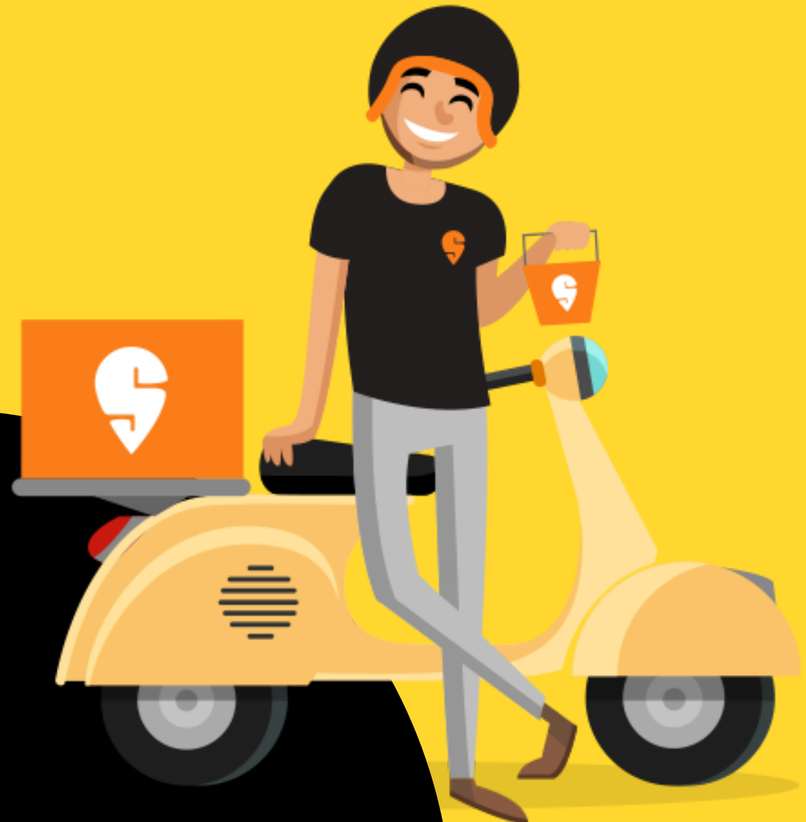


```sql
SELECT
    deliverypartners.name,
    COUNT(DISTINCT customers.customer_id) AS A
FROM
    deliverypartners
        JOIN
    orderdelivery ON deliverypartners.partner_id = orderdelivery.partner_id
        JOIN
    orders ON orderdelivery.order_id = orders.order_id
        JOIN
    customers ON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

Recognizing top-performing delivery partners can help Swiggy reward and retain key contributors

SWIGGY

Identify Customers Who Have the Same City and Have Placed Orders at the Same Restaurants, but on Different Dates

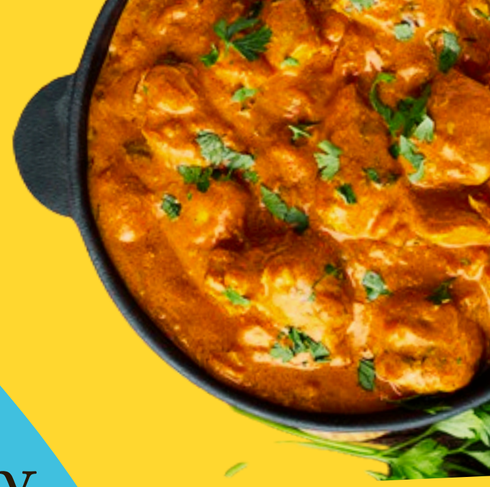


```sql
SELECT DISTINCT
    c1.name customer1,
    c2.name AS customer2,
    restaurants.name AS rest,
    c1.city,
    o1.order_date AS order_date1,
    o2.order_date AS order_date2
FROM
    customers AS c1
        JOIN
    orders AS o1 ON c1.customer_id = o1.customer_id
        JOIN
    orders AS o2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

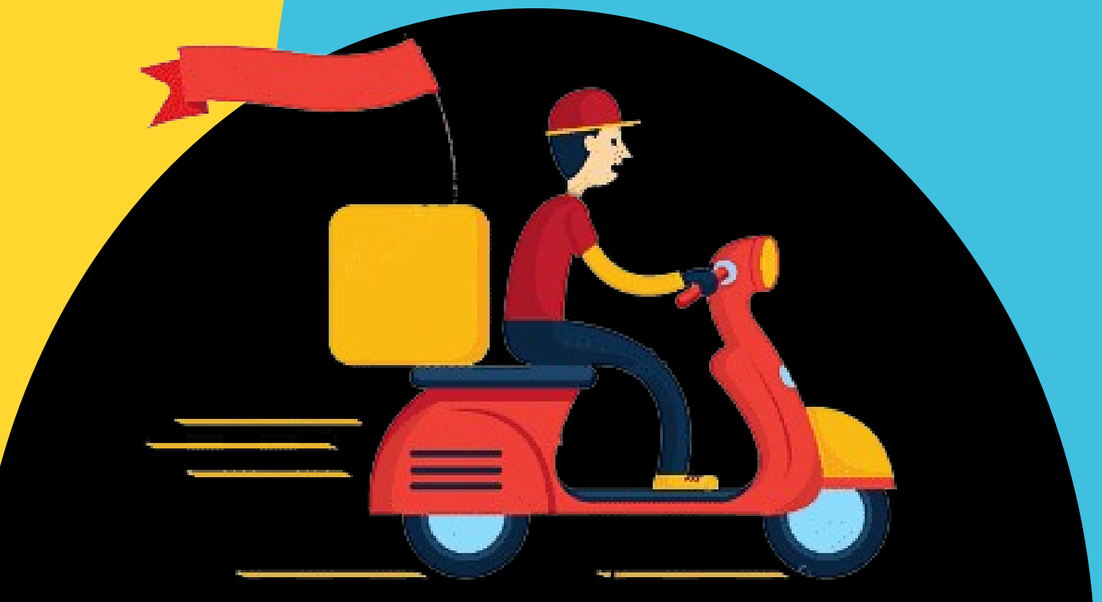
This query identifies patterns of shared dining preferences among customers, enabling Swiggy to explore social or group-based marketing strategies

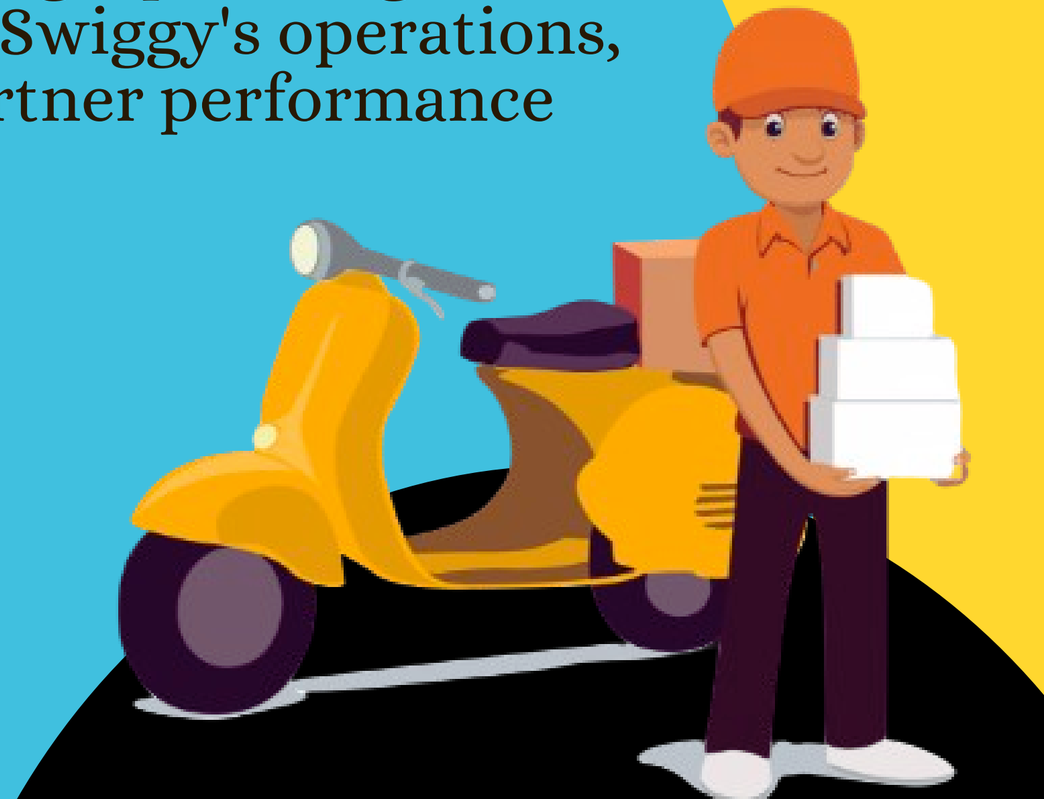
SWIGGY

Conclusion



By leveraging these sophisticated SQL queries, Swiggy can extract valuable insights from its dataset, driving informed decision-making and strategic planning. These analyses support various aspects of Swiggy's operations, from customer engagement to partner performance optimization.





Thank You

For more updates follow me on in