Swiggy

SQL Database Analysis

Presented By Manish Kumar

Introduction

- Swiggy is India’s most popular food delivery platform.
- It connects customers, restaurants, and delivery partners.
- This project uses SQL to analyze Swiggy’s dataset.
- Main focus: customer behavior, restaurant performance, delivery partner efficiency.
- The insights help in strategic decision-making and business growth.

Dataset Overview

- Dataset shared through Google Drive.Includes multiple interconnected tables:
- **Customers:** Customer details and city.
- **Orders:** Order transactions and dates.
- **Restaurants:** Restaurant details and ratings.
- **Delivery Partners:** Partner assignments and deliveries.
- **Revenue & Ratings:** Restaurant earnings and performance.
- Designed to answer 13 SQL-based business queries.

Q1. List all products with a discounted price below ₹500

```
Select  *

From customers

Where city  =  'delhi' ;
```

customer_id	name	email	phone_number	city	address
2	Rohini Verma	rohini.verma@yahoo.com	9823456789	Delhi	B-23, Saket
5	Manish Kumar	NULL	9834567890	Delhi	D-45, Lajpat Nagar
18	Sonali Mishra	NULL	9878345678	Delhi	N-54, Karol Bagh
NULL	NULL	NULL	NULL	NULL	NULL

RESULT

Q2. FIND THE AVERAGE RATING OF ALL RESTAURANTS IN 'MUMBAI' .

```
Select  city, ROUND(AVG(rating), 2)

From restaurants

Where city =  'mumbai'

GROUP BY city;
```

	city	ROUND(AVG(rating), 2)
▶	Mumbai	4.30

RESULT

Q3. LIST ALL CUSTOMERS WHO HAVE PLACED AT LEAST ONE ORDER.

```
Select customers.name, COUNT(orders.customer_id)

From customers

JOIN

Orders ON customers.customer_id = orders.customer_id

GROUP BY customers.customer_id

HAVING COUNT(orders.order_id >= 1);
```

name	COUNT(orders.customer_id)
Amit Sharma	2
Rohini Verma	3
Rajesh Gupta	3
Sneha Mehta	2
Manish Kumar	4
Priya Singh	3
Vikas Reddy	3
Anjali Patel	3
Suresh Nair	1
Kavita Deshmukh	2
Vivek Bhatt	2
Meera Joshi	2
Pankaj Jain	2
Nidhi Saxena	3
Ashok Kumar	3
Deepa Rao	2
Karan Kapoor	1

RESULT

Q4. DISPLAY THE TOTAL NUMBER OF ORDERS PLACED BY EACH CUSTOMER.

```
Select  customers.name, COUNT(orders.customer_id)

From customers

LEFT JOIN

Orders ON customers.customer_id = orders.customer_id

GROUP BY customers.name;
```

name	COUNT(orders.customer_id)
Amit Sharma	2
Rohini Verma	3
Rajesh Gupta	3
Sneha Mehta	2
Manish Kumar	4
Priya Singh	3
Vikas Reddy	3
Anjali Patel	3
Suresh Nair	1
Kavita Deshmukh	2
Vivek Bhatt	2
Meera Joshi	2
Pankaj Jain	2
Nidhi Saxena	3
Ashok Kumar	3
Deepa Rao	2
Karan Kapoor	1

RESULT

Q5. FIND THE TOTAL REVENUE GENERATED BY EACH RESTAURANT.

```
Select  restaurants.name, SUM(orders.total_amount)

From restaurants

LEFT JOIN

Orders ON restaurants.restaurant_id = orders.restaurant_id

GROUP BY restaurants.name;
```

name	SUM(orders.total_amount)
Spice of India	1100.00
Tandoori Flames	1200.00
Biryani House	5300.00
Curry Pot	3200.00
Taste of Punjab	600.00
Royal Biryani	650.00
Coastal Delight	2100.00
Veggie Delight	1600.00
Gujarat Express	2550.00
Andhra Spice	4050.00
Punjabi Tadka	900.00
Flavours of Ben...	4050.00
South Treat	2950.00
The Great India...	1600.00
Rajasthani Rasoi	2100.00
Kerala Kitchen	950.00
Chaat Junction	2150.00

RESULT

Q6. FIND THE TOP 5 RESTAURANTS WITH THE HIGHEST AVERAGE RATING.

```
Select  name, rating

From restaurants

OrRDER BY Rating DESC

LIMIT 5;
```

name	rating
Biryani House	4.80
Paradise Biryani	4.80
Lucknowi Nawabi	4.70
Royal Biryani	4.70
Flavours of Bengal	4.60

RESULT

Q7. DISPLAY ALL CUSTOMERS WHO HAVE NEVER PLACED AN ORDER.

```
Select  DISTINCT customers.name , Orders.order_id

From customers

LEFT JOIN

Orders ON customers.customer_id = Orders.customer_id

Where Orders.order_id IS NULL;
```

name	order_id
Sonal Kaur	NULL
Vivek Malhotra	NULL
Divya Iyer	NULL
Rakesh Yadav	NULL
Mona Sharma	NULL
Sudha Pillai	NULL
Gaurav Khanna	NULL

RESULT

Q8. FIND THE NUMBER OF ORDERS PLACED BY EACH CUSTOMER IN 'MUMBAI'

```
Select  customers.name, COUNT(orders.customer_id)

From customers

LEFT JOIN

Orders ON customers.customer_id = Orders.customer_id

Where city = 'Mumbai'

GROUP BY customers.name;
```

name	COUNT(orders.customer_id)
Amit Sharma	2
Rajesh Gupta	3
Arjun Desai	2
Ravi Singh	2

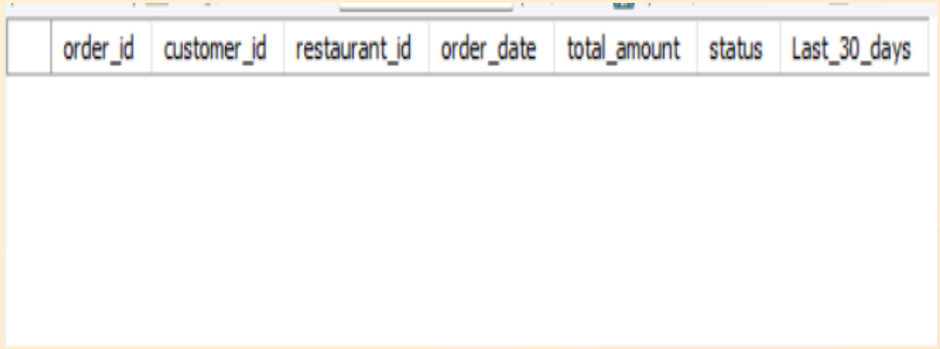
RESULT

Q9. DISPLAY ALL ORDERS PLACED IN THE LAST 30 DAYS.

```
Select  *, DATEDIFF(NOW(),  order_date) AS Last_30_days

From Orders

Where DATEDIFF(NOW(), order_date) <=  30;
```



	order_id	customer_id	restaurant_id	order_date	total_amount	status	Last_30_days

RESULT

Q10. LIST ALL DELIVERY PARTNERS WHO HAVE COMPLETED MORE THAN 1 DELIVERY.

```
Select  deliverypartners.name, COUNT(orderdelivery.order_id)

From deliverypartners

JOIN

Orderdelivery ON deliverypartners.partner_id = orderdelivery.partner_id

GROUP BY deliverypartners.name

HAVING COUNT(orderdelivery.order_id)  >  1;
```

name	COUNT(orderdelivery.order_id)
Suresh Reddy	6
Anita Desai	4
Ravi Kumar	5
Rajesh Gupta	4
Priya Patel	3
Sneha Iyer	2
Amit Sharma	2
Sonia Agarwal	3
Reena Rao	2
Vikram Singh	2
Mohit Saini	2
Ritika Sharma	2

RESULT

Q11. FIND THE CUSTOMERS WHO HAVE PLACED ORDERS ON EXACTLY THREE DIFFERENT DAYS.

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

name	COUNT(DISTINCT orders.order_date)
Anjali Patel	3
Ashok Kumar	3
Nidhi Saxena	3
Priya Singh	3
Rohini Verma	3
Sonali Mishra	3

RESULT

Q12. FIND THE DELIVERY PARTNER WHO HAS WORKED WITH THE MOST DIFFERENT CUSTOMERS.

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

	name	partner_id	customer_ids
▶	Suresh Reddy	4	6

RESULT

Q13. IDENTIFY CUSTOMERS WHO HAVE THE SAME CITY AND HAVE PLACED ORDERS AT THE SAME RESTAURANTS, BUT ON DIFFERENT DATES.

```
SELECT  o1.customer_id AS customer1,  o2.customer_id AS customer2,
        c1.city, o1.restaurant_id
FROM swiggydb.orders o1
JOIN
   swiggydb.orders o2 ON o1.restaurant_id = o2.restaurant_id
      AND o1.order_date <> o2.order_date
      AND o1.customer_id <> o2.customer_id
JOIN
   swiggydb.customers c1 ON o1.customer_id = c1.customer_id
JOIN
   swiggydb.customers c2 ON o2.customer_id = c2.customer_id
WHERE c1.city = c2.city
ORDER BY   c1.city, o1.restaurant_id;
```

customer1	customer2	city	restaurant_id
5	18	Delhi	3
18	5	Delhi	3
18	5	Delhi	3
5	18	Delhi	3
19	23	Mumbai	8
23	19	Mumbai	8

RESULT

Conclusion

- **SQL queries reveal customer, restaurant, and delivery insights.**
- **Helps Swiggy in**:

1. Customer segmentation
2. Restaurant performance analysis
3. Delivery partner optimization

- **Supports better decision-making and growth strategies.**

THANK YOU