SWIGGY

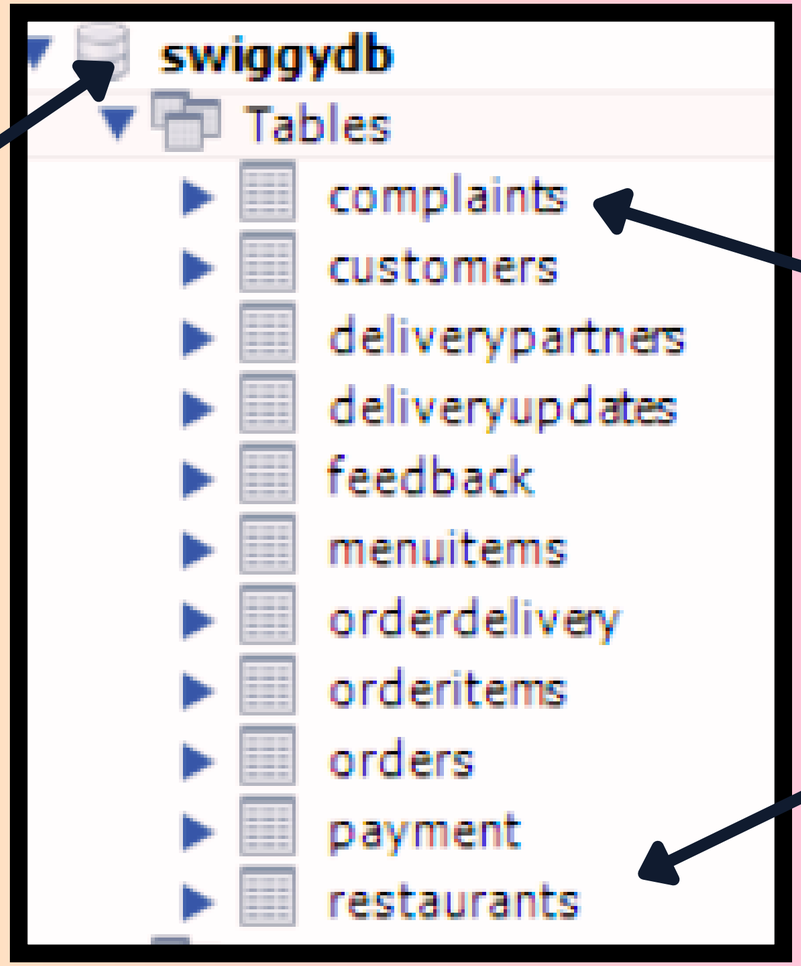
SQL DATA ANALYSIS

OVERVIEW

THIS SWIGGY SQL ANALYSIS PROVIDES A BRIEF OVERVIEW OF ORDER, CUSTOMER, RESTAURANT, AND DELIVERY DATA TO UNCOVER KEY BUSINESS INSIGHTS. IT HIGHLIGHTS TOP-PERFORMING RESTAURANTS, CUSTOMER ORDERING PATTERNS, DELIVERY EFFICIENCY, AND REVENUE TRENDS. THE FINDINGS SUPPORT DATA-DRIVEN DECISIONS TO IMPROVE CUSTOMER EXPERIENCE, OPTIMIZE OPERATIONS, AND BOOST GROWTH.

ABOUT DATASET

Database name
"swiggydb"



Database contains
11 "Tables"

01 Display all customers who live in 'Delhi'

INPUT QUERY

```sql
SELECT customer_id, name, city

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

OUTPUT QUERY

customer_id	name	city
2	Rohini Verma	Delhi
5	Manish Kumar	Delhi
18	Sonali Mishra	Delhi
NULL	NULL	NULL

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

INPUT QUERY

```sql
SELECT
    city,
    ROUND(AVG(COALESCE(rating, 0)), 2) AS avg_rating
FROM
    restaurants
WHERE
    city = 'Mumbai'
GROUP BY city;
```

OUTPUT QUERY

	city	avg_rating
▶	Mumbai	3.23

03 List all customers who have placed at least one order.

INPUT QUERY

```sql
SELECT
    c.customer_id, c.name,
    COUNT(o.order_id) AS total_order
FROM
    customers AS c
JOIN
    orders AS o
ON c.customer_id = o.customer_id
GROUP BY c.customer_id , c.name
HAVING total_order > 0
ORDER BY total_order DESC;
```

OUTPUT QUERY

customer_id	name	total_order
5	Manish Kumar	4
2	Rohini Verma	3
3	Rajesh Gupta	3
6	Priya Singh	3
7	Vikas Reddy	3
8	Anjali Patel	3
14	Nidhi Saxena	3
15	Ashok Kumar	3
18	Sonali Mishra	3
1	Amit Sharma	2
4	Sneha Mehta	2

04 Display the total number of orders placed by each customer

INPUT QUERY

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

OUTPUT QUERY

name	total_orders
Manish Kumar	4
Rohini Verma	3
Rajesh Gupta	3
Priya Singh	3
Vikas Reddy	3
Anjali Patel	3
Nidhi Saxena	3
Ashok Kumar	3
Sonali Mishra	3
Amit Sharma	2
Sneha Mehta	2
Kavita Desh...	2
Vivek Bhatt	2

05 Find the total revenue generated by each restaurant.

INPUT QUERY

```sql
SELECT
    r.name AS restaurant_name,
    SUM(o.total_amount) AS total_revenue
FROM
    restaurants AS r

LEFT JOIN
    orders AS o
ON r.restaurant_id = o.restaurant_id
GROUP BY r.name;
```

OUTPUT QUERY

restaurant_name	total_revenue
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 Bengal	4050.00

06 Find the top 5 restaurants with the highest average rating.

INPUT QUERY

```sql
WITH avg_rat AS
(SELECT restaurant_id,name,
round(avg(coalesce(rating,0)),2)AS avg_rating
FROM restaurants
GROUP BY restaurant_id,name),
ranked AS
(SELECT  *,
dense_rank() over(order by avg_rating desc) AS top_rank
FROM avg_rat)
SELECT *
FROM ranked
WHERE top_rank <=5;
```

OUTPUT QUERY

restaurant_id	name	avg_rating	top_rank
3	Biryani House	4.80	1
22	Paradise Biryani	4.80	1
33	Biryani House	4.80	1
52	Paradise Biryani	4.80	1
30	Lucknowi Nawabi	4.70	2
60	Lucknowi Nawabi	4.70	2
6	Royal Biryani	4.70	2
36	Royal Biryani	4.70	2
12	Flavours of Bengal	4.60	3
42	Flavours of Bengal	4.60	3
19	Awadhi Zaika	4.60	3
49	Awadhi Zaika	4.60	3
1	Spice of India	4.50	4

07 Display all customers who have never placed an order.

INPUT QUERY

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

OUTPUT QUERY

customer_id	name	order_id
24	Sonal Kaur	NULL
25	Vivek Malhotra	NULL
26	Divya Iyer	NULL
27	Rakesh Yadav	NULL
28	Mona Sharma	NULL
29	Sudha Pillai	NULL
30	Gaurav Khanna	NULL

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

INPUT QUERY

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

OUTPUT QUERY

name	city	COUNT(o.order_id)
Amit Sharma	Mumbai	2
Rajesh Gupta	Mumbai	3
Arjun Desai	Mumbai	2
Ravi Singh	Mumbai	2

09

Display all orders placed in the last 15 days

INPUT QUERY

```sql
SELECT
    order_id, date(order_date), status
FROM
    orders
WHERE
    order_date BETWEEN (SELECT
            MAX(order_date)
        FROM
            orders) - INTERVAL 15 DAY AND
            (SELECT
            MAX(order_date)
        FROM
            orders);
```

OUTPUT QUERY

order_id	date(order_date)	status
1	2024-08-01	Completed
2	2024-08-02	Completed
3	2024-08-04	Cancelled
4	2024-08-01	Completed
5	2024-08-03	Completed
6	2024-08-06	Processing
7	2024-08-03	Completed
8	2024-08-08	Completed
9	2024-08-02	Completed
10	2024-08-09	Cancelled
11	2024-08-01	Completed
12	2024-08-04	Completed
13	2024-08-05	Completed
14	2024-08-06	Processing
15	2024-08-10	Completed
16	2024-08-01	Completed
17	2024-08-11	Cancelled

10 List all delivery partners who have completed more than 1 delivery.

INPUT QUERY

```sql
SELECT
    dp.partner_id,
    dp.name,
    o.status,
    COUNT(o.order_id) AS delievery_count
FROM
    deliverypartners AS dp
JOIN
    orderdelivery AS od
ON dp.partner_id = od.partner_id
JOIN
    orders AS o
ON od.order_id = o.order_id
WHERE
    o.status = 'completed'
GROUP BY dp.partner_id,dp.name , o.status
HAVING delievery_count > 1;
```

OUTPUT QUERY

partner_id	name	status	delievery_count
4	Suresh Reddy	Completed	4
5	Anita Desai	Completed	4
6	Rajesh Gupta	Completed	2
3	Priya Patel	Completed	3
1	Amit Sharma	Completed	2
7	Sonia Agarwal	Completed	3
2	Ravi Kumar	Completed	3
8	Vikram Singh	Completed	2
13	Mohit Saini	Completed	2

11

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

INPUT QUERY

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

OUTPUT QUERY

customer_id	name	unique_days
2	Rohini Verma	3
6	Priya Singh	3
8	Anjali Patel	3
14	Nidhi Saxena	3
15	Ashok Kumar	3
18	Sonali Mishra	3

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

INPUT QUERY

```sql
SELECT
    dp.name,
    COUNT(DISTINCT od.order_id) AS diff_customer
FROM
    deliverypartners AS dp
JOIN
    orderdelivery od
ON dp.partner_id = od.partner_id
GROUP BY dp.name
ORDER BY diff_customer DESC
LIMIT 1;
```

OUTPUT QUERY

name	diff_customer
Suresh Reddy	6

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

INPUT QUERY

```sql
SELECT DISTINCT
    c1.name AS cust1,
    c2.name AS cust2,
    c1.city,
    restaurants.restaurant_id,
    restaurants.name,
    DATE(o1.order_date) AS order_cust1,
    DATE(o2.order_date) AS order_cust2
FROM
    customers AS c1
JOIN
    orders AS o1 ON c1.customer_id = o1.customer_id
JOIN
    customers AS c2 ON c1.city = c2.city
JOIN
    orders AS o2 ON o2.customer_id = c2.customer_id
JOIN
    restaurants ON restaurants.restaurant_id = o2.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;
```

OUTPUT QUERY

cust1	cust2	city	restaurant_id	name	order_cust1	order_cust2
Manish Kumar	Sonali Mishra	Delhi	3	Biryani House	2024-08-04	2024-08-05
Sonali Mishra	Manish Kumar	Delhi	3	Biryani House	2024-08-05	2024-08-04
Sonali Mishra	Manish Kumar	Delhi	3	Biryani House	2024-08-05	2024-08-07
Arjun Desai	Ravi Singh	Mumbai	8	Veggie Delight	2024-08-03	2024-08-09
Manish Kumar	Sonali Mishra	Delhi	3	Biryani House	2024-08-07	2024-08-05
Ravi Singh	Arjun Desai	Mumbai	8	Veggie Delight	2024-08-09	2024-08-03

THANK YOU

JASWINDER KAUR