



# Food DELIVERY

SWIGGY PROJECT

PRESENTED BY  
HARDIK JAIN

Order Now



**IN THIS PROJECT I SOLVE SOME  
QUESTIONS BY USING SQL**



**swiggy**

# QUESTION 1



```
1 #Question 1
2 #Display all customers who live in 'Delhi'
3
4 #Solution 1
5 • SELECT
6     *
7 FROM
8     customers
9 WHERE
10    city = 'Delhi';
```

# QUESTION 2



```
1 #Question 2
2 #Find the average rating of
3 #all restaurants in 'Mumbai'.
4
5 #Solution 2
6 • SELECT
7     city, AVG(rating)
8 FROM
9     restaurants
10 WHERE
11     city = 'mumbai'
12 GROUP BY city;
```

# QUESTION 3



```
1 #Question 3
2 #List all customers who have placed at least one order
3
4 #Solution 3
5 • SELECT DISTINCT
6     customers.name
7 FROM
8     customers
9     INNER JOIN
10    orders ON customers.customer_id = orders.customer_id;
```

# QUESTION 4



```
1 #Question 4
2 #Display the total number of orders placed by each customer
3
4 #Solution 4
5 • SELECT
6     customers.name, COUNT(orders.order_id)
7 FROM
8     customers
9     LEFT JOIN
10    orders ON customers.customer_id = orders.customer_id
11   GROUP BY customers.name;
```

# QUESTION 5



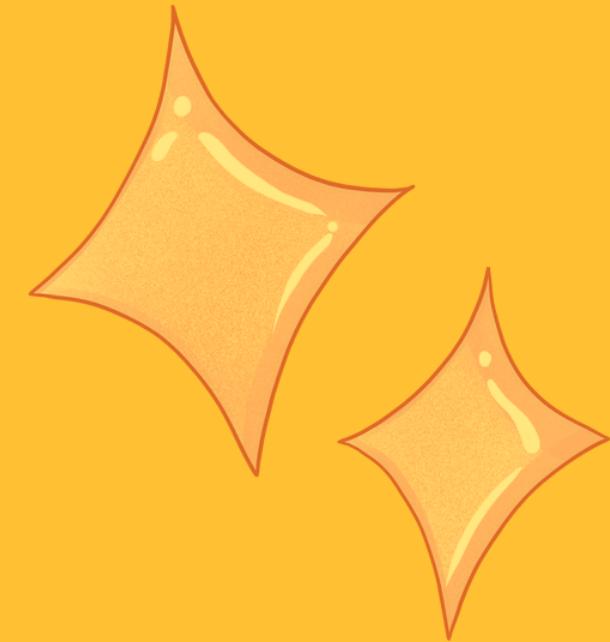
```
1 #Question 5
2 #Find the total revenue
3 #generated by each restaurant.
4
5 #Solution 5
6 • SELECT
7     restaurants.name, SUM(orders.total_amount)
8 FROM
9     restaurants
10    LEFT JOIN
11    orders ON orders.restaurant_id = restaurants.restaurant_id
12    GROUP BY restaurants.name;
```

# QUESTION 6



```
1 #Question 6
2 #Find the top 5 restaurants with
3 #the highest average rating
4
5 #Solution 6
6 • SELECT
7     name, rating
8 FROM
9     restaurants
10 ORDER BY rating DESC
11 LIMIT 5;
```

# QUESTION 7



```
1 #Question 7
2 #Display all customers who have never placed an order.
3
4
5 #Solution 7
6 • SELECT DISTINCT
7     customers.name
8 FROM
9     customers
10    LEFT JOIN
11        orders ON customers.customer_id = orders.customer_id
12 WHERE
13     orders.order_id IS NULL;
```

# QUESTION 8



```
1 #Question 8
2 #Find the number of orders
3 #placed by each customer in 'Mumbai'
4
5 #Solution 8
6 • SELECT
7     customers.name, COUNT(orders.order_id)
8 FROM
9     customers
10    LEFT JOIN
11    orders ON customers.customer_id = orders.customer_id
12 WHERE
13     customers.city = 'mumbai'
14 GROUP BY customers.name;
```

# QUESTION 9



```
1 #Question 9
2 #Display all orders placed in the last 30 days
3
4 #Solution 9
5 • SELECT
6     *
7 FROM
8     orders
9 WHERE
10    order_date >= DATE_SUB(NOW(), INTERVAL 30 DAY);
```

# QUESTION 10



```
1 #Question 10
2 #List all delivery partners who
3 #have completed more than 1 delivery
4
5 #Solution 10
6 • select deliverypartners.name, count(orderdelivery.order_id)
7   from deliverypartners join orderdelivery
8     on deliverypartners.partner_id = orderdelivery.partner_id
9   join deliveryupdates
10    on deliveryupdates.order_id = orderdelivery.order_delivery_id
11   where deliveryupdates.status <> "failed"
12   group by deliverypartners.name
13   having count(orderdelivery.order_id) > 1;
```

# QUESTION 11



```
1 #Question 11
2 #Find the customers who have placed
3 #orders on exactly three different days.
4
5 #Solution 11
6 • SELECT
7     customers.name
8 FROM
9     customers
10    INNER JOIN
11        orders ON customers.customer_id = orders.customer_id
12 GROUP BY customers.name
13 HAVING COUNT(DISTINCT orders.order_date) = 3;
```

# QUESTION 12



```
1 #Question 12
2 #Find the delivery partner who has
3 #worked with the most different customers
4
5 #Solution 12
6 • SELECT
7     deliverypartners.partner_id,
8     deliverypartners.name,
9     COUNT(DISTINCT orders.customer_id) diff_customers
10    FROM
11        deliverypartners
12            INNER JOIN
13                orderdelivery ON deliverypartners.partner_id = orderdelivery.partner_id
14            JOIN
15                orders ON orderdelivery.order_id = orders.order_id
16    GROUP BY deliverypartners.partner_id , deliverypartners.name
17    ORDER BY diff_customers DESC
18    LIMIT 1;
```

# QUESTION 13



```
1 #Question 13
2 #Identify customers who have the same
3 #city and have placed orders at the same restaurants, but on different dates.
4
5 #Solution 13
6 • SELECT DISTINCT c1.name AS customer1, c2.name AS customer2, c1.city,
7 r.name AS restaurant, o1.order_date AS order_date1, o2.order_date AS order_date
8 FROM Customers c1
9 JOIN Orders o1 ON c1.customer_id = o1.customer_id
10 JOIN Orders o2 ON o1.restaurant_id = o2.restaurant_id
11 JOIN Customers c2 ON c1.city = c2.city
12             AND c1.customer_id <> c2.customer_id
13             AND o2.customer_id = c2.customer_id
14 JOIN Restaurants r ON o1.restaurant_id = r.restaurant_id
15 WHERE o1.order_date <> o2.order_date
16 ORDER BY c1.city, r.name, o1.order date;
```

**IF YOU GUYS LIKED THE PROJECT  
PLEASE DO FOLLOW ME ON  
LINKED IN**

**CLICK HERE**

