Swiggy Case Study :

1. Find customers who have never ordered.

2. Average Price/dish.

3. Find the top restaurant in terms of the number of orders for a given month .

4. restaurants with monthly sales greater than x for

5. Show all orders with orders details for a particular customer in a date range (suppose for user “ankit”. What he have order between 10 june and 10 july and from where he have ordered)?

6. Find restaurants with max repeated customers.

7. Customer - favorite food.

8. Month over month revenue growth of swiggy.

9 month over month revenue for a particular restaurant?

10 Most Paired products

Database Overview:

➔ Tables in my database:

1. User
2. Delivery_partner
3. Food
4. Menu
5. Order_details
6. Orders
7. Restaurant

1. Users:

user_id	name	email	password
1	Nitish	nitish@gmail.com	p252h
2	Khushboo	khushboo@gmail.com	hxn9b
3	Vartika	vartika@gmail.com	9hu7j
4	Ankit	ankit@gmail.com	lkko3
5	Neha	neha@gmail.com	3i7qm
6	Anupama	anupama@gmail.com	46rdw2
7	Rishabh	rishabh@gmail.com	4sw123
NULL	NULL	NULL	NULL

2. Delivery_partner:

partner_id	partner_name
1	Suresh
2	Amit
3	Lokesh
4	Kartik
5	Gyandeep
NULL	NULL

3. Food:

f_id	f_name	type
1	Non-veg Pizza	Non-veg
2	Veg Pizza	Veg
3	Choco Lava cake	Veg
4	Chicken Wings	Non-veg
5	Chicken Popcorn	Non-veg
6	Rice Meal	Veg
7	Roti meal	Veg
8	Masala Dosa	Veg
9	Rava Idli	Veg
10	Schezwan Noodles	Veg
11	Veg Manchurian	Veg
NULL	NULL	NULL

4. Menu:

menu_id	r_id	f_id	price
1	1	1	450.00
2	1	2	400.00
3	1	3	100.00
4	2	3	115.00
5	2	4	230.00
6	2	5	300.00
7	3	3	80.00
8	3	6	160.00
9	3	7	140.00
10	4	6	230.00
11	4	8	180.00
12	4	9	120.00
13	5	6	250.00
14	5	10	220.00
15	5	11	180.00
NULL	NULL	NULL	NULL

5. Order_details:

id	order_id	f_id
1	1001	1
2	1001	3
3	1002	4
4	1002	3
5	1003	6
6	1003	3
7	1004	6
8	1004	3
9	1005	7
10	1005	3
11	1006	1
12	1006	2
13	1006	3
14	1007	4
15	1007	3
16	1008	6
17	1008	3
18	1009	8
19	1009	9
20	1010	10
21	1010	11
22	1010	6
23	1011	1
24	1012	8
25	1013	4
26	1014	4
27	1015	4
28	1016	8

6 Orders:

order_id	user_id	r_id	amount	date	partner_id	delivery_time	delivery_rating	restaurant_rating
1001	1	1	550.00	2022-05-10	1	25	5	3
1002	1	2	415.00	2022-05-26	1	19	5	2
1003	1	3	240.00	2022-06-15	5	29	4	NULL
1004	1	3	240.00	2022-06-29	4	42	3	5
1005	1	3	220.00	2022-07-10	1	58	1	4
1006	2	1	950.00	2022-06-10	2	16	5	NULL
1007	2	2	530.00	2022-06-23	3	60	1	5
1008	2	3	240.00	2022-07-07	5	33	4	5
1009	2	4	300.00	2022-07-17	4	41	1	NULL
1010	2	5	650.00	2022-07-31	1	67	1	4
1011	3	1	450.00	2022-05-10	2	25	3	1
1012	3	4	180.00	2022-05-20	5	33	4	1
1013	3	2	230.00	2022-05-30	4	45	3	NULL
1014	3	2	230.00	2022-06-11	2	55	1	2
1015	3	2	230.00	2022-06-22	3	21	5	NULL
1016	4	4	300.00	2022-05-15	3	31	5	5
1017	4	4	300.00	2022-05-30	1	50	1	NULL
1018	4	4	400.00	2022-06-15	2	40	3	5
1019	4	5	400.00	2022-06-30	1	70	2	4
1020	4	5	400.00	2022-07-15	3	26	5	3
1021	5	1	550.00	2022-07-01	5	22	2	NULL
1022	5	1	550.00	2022-07-08	1	34	5	1
1023	5	2	645.00	2022-07-15	4	38	5	1
1024	5	2	645.00	2022-07-21	2	58	2	1
1025	5	2	645.00	2022-07-28	2	44	4	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

8. Restaurant:

r_id	r_name	cuisine
1	dominos	Italian
2	kfc	American
3	box8	North Indian
4	Dosa Plaza	South Indian
5	China Town	Chinese
NULL	NULL	NULL

Questions :

Q1 . Find the customer who have never ordered ?

```
208     -- Q1  Find the customer who have never ordered
209 •   select user_id, name from swiggy.users
210     where user_id
211     not in ( select  user_id from orders);
```

Result Grid | Filter Rows: | Edit:

user_id	name
6	Anupama
7	Rishabh
NULL	NULL

Q2. Avg Price/Dish ?

```
213 •   select f.f_name , avg(price) as "avg_price"
214     from menu m
215     join food f
216     on f.f_id = m.f_id
217     group by m.f_id;
```

Result Grid | Filter Rows: | Export:

f_name	avg_price
Non-veg Pizza	450.000000
Veg Pizza	400.000000
Choco Lava cake	98.333333
Chicken Wings	230.000000
Chicken Popcorn	300.000000
Rice Meal	213.333333
Roti meal	140.000000
Masala Dosa	180.000000
Rava Idli	120.000000
Schezwan Noodles	220.000000
Veg Manchurian	180.000000

Q3. Find the top restaurant in terms of number of orders for a given month

```
227     -- Q3 find the top restaurant in terms of number of orders for a given month
228 •   select r.r_name ,count(*) as "Month"
229     from orders o
230     join restaurant r on o.r_id = r.r_id
231     where monthname(date) like "june"      -- just change the name of the month for diff month
232     group by o.r_id
233     order by count(*) desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

r_name	Month
kfc	3
box8	2
dominos	1
Dosa Plaza	1
China Town	1

Q4 Restaurant with montly sales > x

```
237     -- Q4 restaurant with montly sales > x
238 •   select r.r_name,o.r_id,sum(amount) as "revenue"
239     from orders o
240     join restaurant r
241     on r.r_id = o.r_id
242     where monthname(date) like "June"
243     group by  o.r_id
244     having revenue >500;
```

Result Grid | Filter Rows: | Export: | Wrap Cell

r_name	r_id	revenue
dominos	1	950.00
kfc	2	990.00

Q5 Show all orders with orders details for a particular customer in a date range (suppose for user ankit what he have order between 10 june and 10 july and from where he have ordered)?

```sql
select o.order_id,r.r_name,f.f_name from orders o
join restaurant r
on o.r_id = r.r_id
join order_details od
on od.order_id= o.order_id
join food f
on f.f_id = od.f_id
where o.user_id = (select u.user_id from users u where u.name like "ankit")
and date between "2022-06-10" and "2022-07-10";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_id	r_name	f_name
1018	Dosa Plaza	Schezwan Noodles
1018	Dosa Plaza	Veg Manchurian
1019	China Town	Schezwan Noodles
1019	China Town	Veg Manchurian

Q6 find restaurant with max repeated customers

```sql
select r.r_name, count(*) as "loyal_customer"
from(
        select r_id,user_id,count(*) as "visits"
        from orders
        group by r_id,user_id
    having visits>1
    )t
join restaurant r
on r.r_id = t.r_id
group by t.r_id
order by loyal_customer desc limit 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Co

r_name	loyal_customer
kfc	2

Q7 Every Customers Favourite Food

```sql
-- Q7 Customer Favourite Food
with temp as (
    select o.user_id,od.f_id,count(*) as "Frequency" from orders o
    join order_details od
    ON o.order_id=od.order_id
    group by o.user_id,od.f_id
)
select u.name,f.f_name,t1.frequency from
temp t1
join users u
on u.user_id=t1.user_id
join food f
on f.f_id=t1.f_id
where t1.Frequency = (
select max(Frequency)
from temp t2
where t2.user_id=t1.user_id
)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	f_name	frequency
Nitish	Choco Lava cake	5
Khushboo	Choco Lava cake	3
Vartika	Chicken Wings	3
Ankit	Schezwan Noodles	3
Ankit	Veg Manchurian	3
Neha	Choco Lava cake	5

Q8. month over month revenue

```sql
-- Q8. month over omth revenue
select month ,((revenue -prev)/prev)*100 from (
    with sales as
    (
        select  monthname(date) as "month", sum(amount) as "revenue"
        from orders
        group by month
    )
select month,revenue,LAG(revenue,1) OVER (order by revenue) as prev from sales
)t;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

month	((revenue -prev)/prev)*100
May	NULL
June	32.783505
July	50.465839

Q9 month over month revenue for a particular restaurant?

```sql
-- Q9 month over month revenue for a particular restaurant
WITH sales AS (
    SELECT YEAR(o.date) AS yr,MONTH(o.date) AS mon,MONTHNAME(o.date) AS month,SUM(o.amount) AS revenue,r.r_name
    FROM orders o
    JOIN restaurant r ON r.r_id = o.r_id
    WHERE o.r_id = 1
    GROUP BY YEAR(o.date), MONTH(o.date), MONTHNAME(o.date), r.r_name
)
SELECT yr,month,r_name,revenue,
    LAG(revenue,1) OVER (ORDER BY yr, mon) AS prev,
    ROUND(((revenue - LAG(revenue,1) OVER (ORDER BY yr, mon))
          / LAG(revenue,1) OVER (ORDER BY yr, mon)) * 100, 2) AS mom_growth_pct
FROM sales;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

yr	month	r_name	revenue	prev	mom_growth_pct
2022	May	dominos	1000.00	NULL	NULL
2022	June	dominos	950.00	1000.00	-5.00
2022	July	dominos	1100.00	950.00	15.79

Q10 Most Paired products

```sql
-- 10 Most Paired products
SELECT
    f1.f_name AS product1,
    f2.f_name AS product2,
    COUNT(*) AS pair_count
FROM order_details od1
JOIN order_details od2
    ON od1.order_id = od2.order_id
   AND od1.f_id < od2.f_id    -- avoid duplicate and self-pairing
JOIN food f1 ON f1.f_id = od1.f_id
JOIN food f2 ON f2.f_id = od2.f_id
GROUP BY f1.f_name, f2.f_name
ORDER BY pair_count DESC
LIMIT 10;   -- Top 10 most paired products
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

product1	product2	pair_count
Choco Lava cake	Chicken Wings	5
Non-veg Pizza	Choco Lava cake	4
Schezwan Noodles	Veg Manchurian	4
Choco Lava cake	Rice Meal	3
Choco Lava cake	Chicken Popcorn	3
Chicken Wings	Chicken Popcorn	3
Masala Dosa	Rava Idli	3
Non-veg Pizza	Veg Pizza	1
Veg Pizza	Choco Lava cake	1
Choco Lava cake	Roti meal	1

Result 34