SWIGGY SQL CASE STUDY :

By Akarshan Kapoor

Tables:

1. food
2. menu
3. orders
4. order details
5. restaurant
6. users

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

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

order_details

order_id	user_id	r_id	amount	date
1001	1	1	550	2022-05-10
1002	1	2	415	2022-05-26
1003	1	3	240	2022-06-15
1004	1	3	240	2022-06-29
1005	1	3	220	2022-07-10
1006	2	1	950	2022-06-10
1007	2	2	530	2022-06-23
1008	2	3	240	2022-07-07
1009	2	4	300	2022-07-17
1010	2	5	650	2022-07-31
1011	3	1	450	2022-05-10

orders

restaurant

r_id	r_name	cuisine	rating
1	dominos	Italian	4.1
2	kfc	American	3.9
3	box8	North Indian	4.3
4	Dosa Plaza	South Indian	4.6
5	China Town	Chinese	4.0

menu_id	r_id	f_id	price
1	1	1	450
2	1	2	400
3	1	3	100
4	2	3	115
5	2	4	230
6	2	5	300
7	3	3	80
8	3	6	160
9	3	7	140
10	4	6	230

menu

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

food

Questions & Solutions:

1. Find the customers who never ordered

```
SELECT name FROM users
  WHERE user_id NOT IN ( SELECT user_id FROM order );
```

2. Average price per dish

```
SELECT f.food_name, AVG(price) AS 'Avg_price'
  FROM menu m
  JOIN food f
   ON m.food_id = f.food_id
  GROUP BY food_id;
```

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

```
SELECT r.r_name, COUNT(*) AS 'month'
  FROM orders o
  JOIN restaurants r
   ON o.r_id = r.r_id
  WHERE MONTHNAME(date) LIKE 'June'  #june_month
  GROUP BY o.r_id
  ORDER BY COUNT(*) DESC
  LIMIT 1;
```

4. Restaurants with monthly sales > x for

```
SELECT r.r_name, SUM(amount) AS 'revenue'
   FROM orders o
   JOIN restaurant r
    ON o.r_id = r.r_id
   WHERE MONTHNAME(date) LIKE 'JUNE'
   GROUP BY r_id
   HAVING revenue >500;
```

5. Show all orders with the order details for a particular customer in a particular date range

```
SELECT o.oorr_id,r.r_id
   FROM orders o
   JOIN rstuarants r
```

```
    ON r.rid = o.r_id
   JOIN order_details od
    ON o.order_id = od.order_id
   JOIN food f
    ON f.f_id = od.f_id
   WHERE user_id = ( SELECT user_id FROM users WHERE name LIKE
'Ankit')
   AND ( date > '2206-10' AND dat < '2022-07-10');
```

6. Find the restaurants with max repeated customers.

```
SELECT r.r_name, COUNT(*) AS 'loyal_customers'
   FROM (
           SELECT r_id, user_id, CONT() AS 'visits'
           FROM orders
           GROUP BY r_id, user_id
           HAVING visits>1
        ) t
   JOIN resturant r
    ON .r_id =t.r_id

   GROUP BY t.r_id
   ORDER BY loyal_customers DESC LIMIT 1;
```

7. Month over revenue growth of swiggy.

```
SELECT month, ((revenue - prev)/pev)*100
   FROM (

   WITH sales AS
   (
     SELECT *, MONTHNAME(date) AS 'month', SUM(amount) AS 'revenue'
     FROM orders
     GROUP BY MOTHNAME(date)
     ORDER BY MONTH(date)
   )

   SELECT month, revenue, LAG(revenue, 1) OVER ( ORDE BY revenue) AS
prev
   FROM sales
          ) t
```

8. Customers --> favorite food

```
WITH temp AS
   (
    SELECT o.user_id, od.f_id, COUT(*) AS 'frequency'
    FROM orders o
```

```
  JOIN order_detals od
   ON o.order_id = od.order_id
  GROUP BY o.user_id, od.f_id
 )

SELECT u.name, od.f_id, 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 2
                        WHERE t2.user_id = t1.user_id )
```
