Case Study: Swiggy's Restaurant Performance Analysis

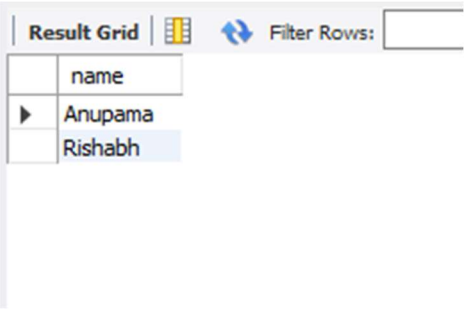
- Swiggy manages 5 restaurants, and all analyses focus on these restaurants' operations and performance.
- Key metrics include identifying top-performing restaurants, analyzing monthly sales, customer preferences, and loyalty trends.
- The database comprises 6 tables, enabling detailed insights into orders, customers, food items, and revenue metrics.
- Each query is designed to drive strategic decisions and optimize restaurant performance.

1. FIND CUSTOMERS WHO HAVE NEVER ORDERED

SELECT name

FROM users

WHERE user_id NOT IN (SELECT user_id FROM orders);



name
Anupama
Rishabh

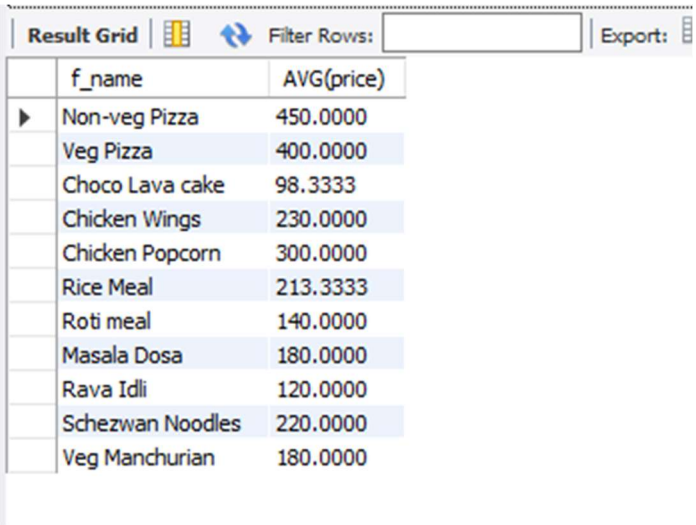
2. AVERAGE PRICE PER DISH

SELECT f.f_name, AVG(price)

FROM menu m

JOIN food f ON f.f_id = m.f_id

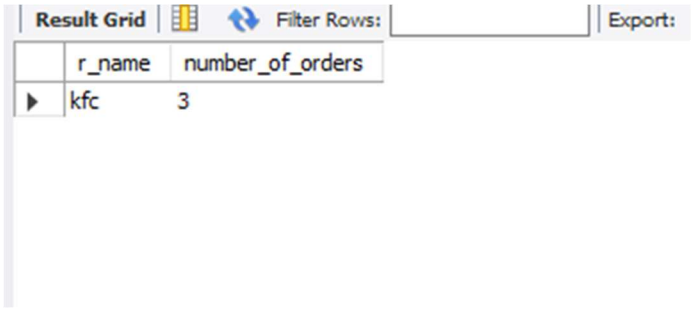
GROUP BY m.f_id;



f_name	AVG(price)
Non-veg Pizza	450.0000
Veg Pizza	400.0000
Choco Lava cake	98.3333
Chicken Wings	230.0000
Chicken Popcorn	300.0000
Rice Meal	213.3333
Roti meal	140.0000
Masala Dosa	180.0000
Rava Idli	120.0000
Schezwan Noodles	220.0000
Veg Manchurian	180.0000

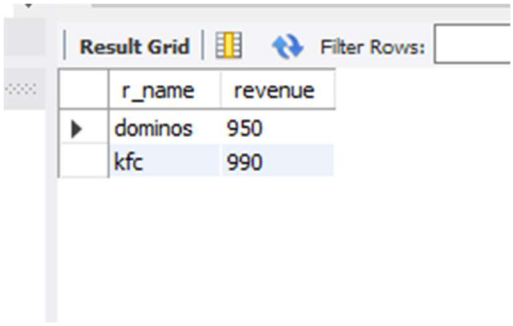
3. FIND TOP RESTAURANT IN TERMS OF NUMBER OF ORDERS FOR A GIVEN MONTH (Lets say July)

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

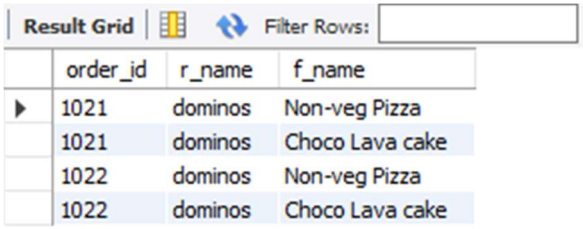


4. RESTAURANTS WITH MONTHLY SALES > x (Lets say 500)

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



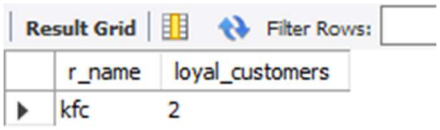
5. SHOW ALL ORDERS WITH ORDER DETAILS FOR A PARTICULAR CUSTOMER IN A PARTICULAR DATE RANGE



order_id	r_name	f_name
1021	dominos	Non-veg Pizza
1021	dominos	Choco Lava cake
1022	dominos	Non-veg Pizza
1022	dominos	Choco Lava cake

```
SELECT o.order_id, r.r_name, f.f_name
FROM orders o
JOIN restaurants r ON r.r_id = 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 'Neha')
AND (date > '2022-06-10' AND date < '2022-07-10');
```

6. FIND RESTAURANTS WITH MAX REPEATED CUSTOMERS

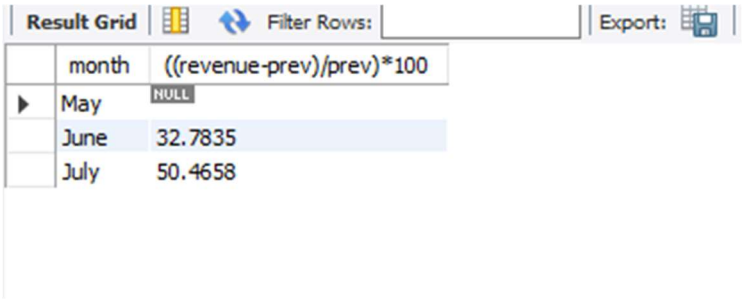


r_name	loyal_customers
kfc	2

```
SELECT r.r_name,COUNT(*) AS 'loyal_customers'
FROM (
        SELECT r_id, user_id, COUNT(*) AS 'visits'
   FROM orders
   GROUP BY r_id, user_id
   HAVING visits>1
        ) t
JOIN restaurants r
ON r.r_id = t.r_id
GROUP BY t.r_id
ORDER BY loyal_customers DESC LIMIT 1;
```

7. MONTH OVER MONTH REVENUE GROWTH OF SWIGGY

```
SELECT month, ((revenue-prev)/prev)*100 FROM
        (
  WITH sales AS
                (
                SELECT MONTHNAME(date) AS 'month', SUM(amount) AS 'revenue'
                FROM orders
                GROUP BY month
                ORDER BY MONTH(date)
    )
        SELECT month, revenue, LAG(revenue,1) OVER(ORDER BY revenue) AS prev FROM sales
  ) t
```

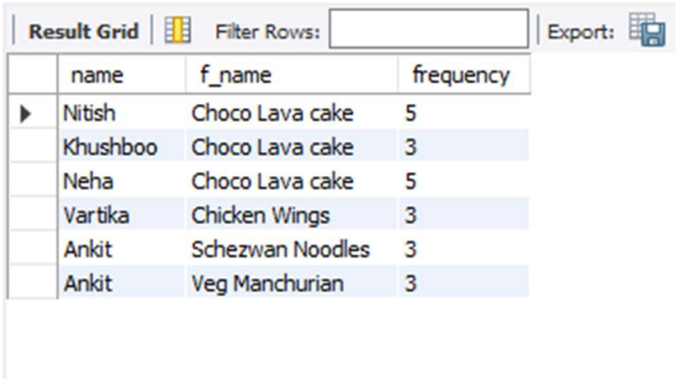


Result Grid | Filter Rows: | Export:

month	((revenue-prev)/prev)*100
May	NULL
June	32.7835
July	50.4658

8. 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:

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