SWIGGY CASE STUDY USING SQL

Q. Find the customers who never ordered

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

Q. Average price/dish

```
SELECT food.f_name AS food_name, AVG(menu.price)
AS avg_price FROM food
JOIN menu ON
food.f_id = menu.f_id
GROUP BY food.f_id, food.f_name;
```

Q. Find the top restaurant in terms of number of orders for a given month (June month)

```
SELECT  restaurants.r_name, COUNT(*) as total from
restaurants
INNER JOIN orders ON
orders.r_id = restaurants.r_id
WHERE orders.date BETWEEN '2022-06-01' AND '2022-06-30'
```

```
GROUP BY restaurants.r_id, restaurants.r_name
ORDER BY 2 DESC LIMIT 1;
```

Q. Restaurants with month sales > 500

```
SELECT  restaurants.r_name from restaurants
INNER JOIN orders ON
orders.r_id = restaurants.r_id
WHERE orders.date BETWEEN '2022-06-01' AND '2022-06-30'
GROUP BY restaurants.r_id, restaurants.r_name
HAVING SUM(orders.amount) > 500;
```

Q. Show all orders with order details for a particular customer in a particular date range (Ankit - 15th May to 15th June)

```
SELECT users.name, food.f_name FROM users
INNER JOIN orders
ON users.user_id = orders.user_id
INNER JOIN order_details
ON orders.order_id = order_details.order_id
INNER JOIN food
ON order_details.f_id = food.f_id
WHERE (users.name = 'Nitish') AND (orders.date
BETWEEN '2022-06-11' AND '2022-07-09')
```

```
ORDER BY orders.date;
```

Q. Month over month revenue growth of Swiggy

```
WITH sales AS

(
SELECT SUM(orders.amount) AS total_amt_generated,
MONTHNAME(orders.date) as monthly_reveneue
FROM orders
INNER JOIN restaurants
ON orders.r_id = restaurants.r_id
GROUP BY MONTHNAME(orders.date)
ORDER BY 2 DESC
)

SELECT monthly_reveneue, total_amt_generated,
LAG(total_amt_generated, 1) OVER (ORDER BY
total_amt_generated) AS previous_amt,
ROUND(((total_amt_generated -
LAG(total_amt_generated, 1) OVER (ORDER BY
total_amt_generated)) / (LAG(total_amt_generated, 1)
OVER (ORDER BY total_amt_generated) ) ) * 100,1) AS
perc_gain
FROM sales;
```

Q. Find each customers favourite food

```
WITH temp AS
(
SELECT o.user_id, od.f_id, COUNT(*) as frequency FROM
orders o
INNER JOIN order_details od
ON o.order_id = od.order_id
GROUP BY o.user_id, od.f_id
)

SELECT users.name, food.f_name FROM temp t1
INNER JOIN users
ON users.user_id = t1.user_id
INNER JOIN food
ON food.f_id = t1.f_id
 WHERE t1.frequency = (
SELECT MAX(frequency) FROM temp t2 WHERE
t2.user_id = t1.user_id
)
```