**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

)