Restaurant online sales data analysis using SQL

Objective: To perform analysis on restaurant dataset by extracting various types of business insights

This Dataset contains 6 tables:

1. **Customer_orders**: contains 6 columns
    - Order_id
    - Customer_id
    - Roll_id
    - Not_include_items
    - Extra_included_items
    - Order_data
2. **Driver:** contains 2 columns
    - Driver_id
    - Reg_date

3. **Driver_order:** contains 6 columns
    - Order_id
    - Driver_id
    - Pickup_time
    - Distance
    - Duration
    - Cancellation
4. **Ingredients:** contains 2 columns
    - Ingredient_id
    - Ingredient_name
5. **Rolls:** contains 2 columns
    - Roll_id
    - Roll_name
6. **Rolls_recipes:** contains 2 columns
    - Roll_id
    - Ingredients

Research Questions:

1. How many rolls were ordered ?

- SQL Query:
- select count(roll_id) as rolls_ordered from customer_orders;
- Output:

Rolls_ordered
14

- Using this insight, restaurants get to know the total number of rolls sold.

2. How many unique customers who ordered from this restaurant ?

- SQL Query:
- select count(distinct customer_id) as no_of_unique_customers from customer_orders;
- Output:

no_of_unique_customers
5

- Using this insight, restaurants get to know their total number of customer

3. How many successful orders were delivered by each driver ?

- SQL Query:
- Clean the data using case statement:

- select *, case when cancellation in ('cancellation', 'customer cancellation') then 'c' else 'nc' end as order_cancel_details from driver_order;
- Use this table as CTE using WITH Clause:
- WITH clean_driver_order_CTE AS (select *, case when cancellation in ('cancellation', 'customer cancellation') then 'c' else 'nc' end as order_cancel_details from driver_order)

  select driver_id, count(distinct order_id) as successful_delivery from clean_driver_order_CTE where order_cancel_details in ('nc') group by driver_id;
- Output:

driver_id	successful_delivery
1	4
2	3
3	1

- Through this insight, restaurants get to know which driver is delivering more orders, so they can rate their drivers accordingly and they can pay the drivers using this data.

4. How many of each type of rolls were delivered ?

- SQL Query:
- WITH clean_driver_order_CTE AS (select *, case when cancellation in ('cancellation', 'customer cancellation') then 'c' else 'nc' end as order_cancel_details from driver_order)

  select roll_id, count(roll_id) from customer_orders where order_id in (select order_id from clean_driver_order_CTE where order_cancel_details='nc') group by roll_id;

- Output:

roll_id	No. of rolls ordered
1	9
2	3

- Through this insight, restaurants get to know sale of each types of rolls. Using this data, restaurants can focus on increasing sales of that roll type which was ordered lesser no. of times.

5. How many veg and non veg rolls were ordered by each customer ?

- SQL Query:
- Method 1 using multiple groupby:
- select a.*,b.roll_name from (select customer_id, roll_id, count(roll_id) from customer_orders group by customer_id,roll_id) as a join rolls as b on a.roll_id=b.roll_id;
- Output:

customer_id	roll_id	Rolls ordered	roll_name
101	1	2	Non Veg Roll
102	1	2	Non Veg Roll
103	1	3	Non Veg Roll
104	1	3	Non Veg Roll
102	2	1	Veg Roll

103	2	1	Veg Roll
101	2	1	Veg Roll
105	2	1	Veg Roll

- Method 2 using Window function:
- select a.*,b.roll_name from (select customer_id, roll_id, count(roll_id) over(partition by customer_id order by roll_id) as total_rolls_ordered from customer_orders) as a join rolls as b on a.roll_id=b.roll_id;
- Output:

customer_id	roll_id	total_rolls_ordered	roll_name
101	1	2	Non Veg Roll
101	1	2	Non Veg Roll
102	1	2	Non Veg Roll
102	1	2	Non Veg Roll

103	1	3	Non Veg Roll
103	1	3	Non Veg Roll
103	1	3	Non Veg Roll
104	1	3	Non Veg Roll
104	1	3	Non Veg Roll
104	1	3	Non Veg Roll
101	2	3	Veg Roll
102	2	3	Veg Roll
103	2	4	Veg Roll
105	2	1	Veg Roll

- Through this insight, restaurants get to know the choice or preference of each customer. Using this insight, they can advertise accordingly for each customer that is target ads.

6. What was the maximum number of rolls delivered in a single order ?

- SQL Query:
- WITH clean_driver_order_CTE AS (select *, case when cancellation in ('cancellation', 'customer cancellation') then 'c' else 'nc' end as order_cancel_details from driver_order),

  customer_orders_CTE AS (select * from customer_orders where order_id in (select order_id from clean_driver_order_CTE where order_cancel_details='nc')),

  count_CTE AS (select order_id, count(roll_id) as cnt from customer_orders_CTE group by order_id)

  select *, rank() over(order by cnt desc) as rnk from count_CTE;

- Output:

order_id	cnt	rnk
4	3	1

3	2	2
10	2	2
1	1	4
2	1	4
5	1	4
7	1	4
8	1	4

- Through this insight, restaurants get to know which customers are ordering the most number of rolls. They can also use this for ease in handling rolls during delivery.

7. What was the total number of rolls ordered for each hour of the day ?

- SQL Query:
- select hours_bracket, count(hours_bracket) from (select *, concat(cast(hour(order_date) as char), '-' , cast(hour(order_date) + 1 as char)) as hours_bracket from customer_orders) as a group by hours_bracket order by hours_bracket;

- Output:

hours_bracket	count(hours_bracket)
11-12	1
13-14	3
18-19	3
19-20	1
21-22	3
23-24	3

- Using this insight, restaurants get to know in which hours they got the most orders from customers. They can use this for pricing strategy. Increase Price in peak hours and lowering the price during off peak hours.

8. What was the number of orders for each day of the week ?

- SQL Query:
- select dayofweek, count(distinct order_id) from (select *, dayname(order_date) as dayofweek

from customer_orders) as a group by dayofweek;

- Output:

dayofweek	count(distinct order_id)
Friday	5
Monday	2
Saturday	2
Sunday	1

- Through this insight, restaurants get to know on which day, customers are ordering the most.

9. What was the average time in minutes it took for each driver to arrive at the restaurant to pick up the order ?

- SQL Query:
- select driver_id, sum(diff)/count(order_id) as average from

  (select *, row_number() over(partition by order_id order by diff) as rnk from

```
(select a.*, b.driver_id, b.pickup_time, timestampdiff(minute,a.order_date,b.pickup_time)

as diff from customer_orders as a join driver_order as b on a.order_id=b.order_id where b.pickup_time is not null) as a) as b where rnk=1 group by driver_id;
```

- Output:

driver_id	average
1	14
2	19.67
3	10

- Through this insight, restaurants get to know which drivers are more punctual and active towards their delivery.

10. What is the successful delivery percentage for each driver ?

- SQL query:
- select driver_id,sum(cancel_order) as successful_delivery, count(driver_id) as total_delivery,

```
(sum(cancel_order)/count(driver_id))*100 as
successful_Delivery_percent from

(select driver_id, case when lower(cancellation)
like "%cancel%" then 0 else 1 end as
cancel_order from driver_order) as a group by
driver_id;
```

-

driver_id	successful_delivery	total_delivery	successful_Delivery_percent
1	4	4	100
2	3	4	75
3	1	2	50

- Through this insight, restaurants get to know the successful delivery percentage of drivers.