SIMZ ROLLS

PROJECT BY RANVEER SURKUNTWAR

SKILLS USED : STRUCTURED QUERYLANGUAGE (SQL)

CONCEPTS USED - JOINS ,CTE , WINDOW FUNCTION,DATETIME FUNCTION, STRING FUNCTION, SUBQUERY

Welcome to the Simz Rolls! In this exploration, we dive into the vast realm of data to uncover valuable insights for Simz Rolls. Through meticulous analysis, we aim to reveal patterns, trends and opportunities that will enhance understanding and contribute to informed decision-making in the dynamic world of selling delicious rolls. This project employs SQL skills to dissect and analyse data to uncover trends and patterns in the data provided. Join us on this data-driven journey as we unravel the story behind Simz Rolls and its diverse offerings.

Tabels

```
SELECT * FROM rolls_information.customer_orders;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_id	customer_id	roll_id	not_include_items	extra_items_included	order_date
1	101	1			2021-01-01 18:05:02
2	101	1			2021-01-01 19:00:52
3	102	1			2021-01-02 23:51:23
3	102	2		NaN	2021-01-02 23:51:23
4	103	1	4		2021-01-04 13:23:46
4	103	1	4		2021-01-04 13:23:46
4	103	2	4		2021-01-04 13:23:46
5	104	1	NULL	1	2021-01-08 21:00:29
6	101	2	NULL	NULL	2021-01-08 21:03:13
7	105	2	NULL	1	2021-01-08 21:20:29
8	102	1	NULL	NULL	2021-01-09 23:54:33
9	103	1	4	1,5	2021-01-10 11:22:59
10	104	1	NULL	NULL	2021-01-11 18:34:49
10	104	1	2,6	1,4	2021-01-11 18:34:49

```
SELECT * FROM rolls_information.driver_order;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_id	driver_id	pickup_time	distance	duration	cancellation
1	1	2021-01-01 18:15:34	20km	32 minutes	
2	1	2021-01-01 19:10:54	20km	27 minutes	
3	1	2021-01-03 00:12:37	13.4km	20 mins	NaN
4	2	2021-01-04 13:53:03	23.4	40	NaN
5	3	2021-01-08 21:10:57	10	15	NaN
6	3	NULL	NULL	NULL	Cancellation
7	2	2020-01-08 21:30:45	25km	25mins	NULL
8	2	2020-01-10 00:15:02	23.4 km	15 minute	NULL
9	2	NULL	NULL	NULL	Customer Cancellation
10	1	2020-01-11 18:50:20	10km	10minutes	NULL

```
SELECT * FROM rolls_information.driver;
```

Result Grid | Filter Rows:

driver_id	reg_date
1	2021-01-01
2	2021-01-03
3	2021-01-08
4	2021-01-15

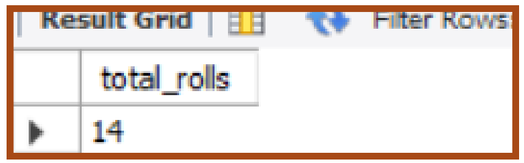
```
SELECT * FROM rolls_information.rolls;
```

Result Grid | Filt

roll_id	roll_name
1	Non Veg Roll
2	Veg Roll

1.How many rolls were ordered ?

```sql
SELECT
    COUNT(roll_id) AS total_rolls
FROM
    customer_orders;
```



2. How many customers ordered rolls?

```sql
SELECT
    COUNT(DISTINCT order_id) AS total_orders
FROM
    customer_orders;
```

Result Grid | Filter Rows:

	total_orders
▶	10

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

```sql
SELECT
    do.driver_id, COUNT(do.order_id) AS total_orders
FROM
    driver_order AS do
WHERE
    do.cancellation NOT IN ('cancellation' , 'Customer Cancellation')
GROUP BY do.driver_id;
```

Result Grid | Filter Rows:

driver_id	total_orders
1	3
2	1
3	1

4. How many each type of rolls were delivered?

```sql
with total_delivers as (
    select order_id, r.roll_name, r.roll_id, count(order_id) as total_orders
    from rolls r
    join customer_orders co on r.roll_id = co.roll_id
    group by roll_name, roll_id, order_id
)
select roll_name, roll_id, sum(total_orders) as total_orders
from total_delivers
where order_id not in (
    select order_id
    from driver_order
    where cancellation in ('cancellation', 'Customer Cancellation')
)
group by roll_name, roll_id;
```

Result Grid | Filter Rows:

roll_name	roll_id	total_orders
Non Veg Roll	1	9
Veg Roll	2	3

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

```sql
SELECT
    co.customer_id,
    r.roll_name,
    COUNT(co.order_id) AS total_orders
FROM
    customer_orders AS co
        JOIN
    rolls AS r ON r.roll_id = co.roll_id
GROUP BY co.customer_id , r.roll_name
order by customer_id ;
```

Result Grid | Filter Rows:

customer_id	roll_name	total_orders
101	Non Veg Roll	2
101	Veg Roll	1
102	Non Veg Roll	2
102	Veg Roll	1
103	Non Veg Roll	3
103	Veg Roll	1
104	Non Veg Roll	3
105	Veg Roll	1

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

```sql
with max_order as ( select order_id,count(order_id) as total_orders
from  customer_orders
group by order_id
order by total_orders desc limit 1 )
select * from max_order
where order_id not in (select order_id from driver_order where cancellation in ('cancellation', 'Customer Cancellation'));
```

Result Grid | Filter Rows:

order_id	total_orders
4	3

7. How many rolls were delivered which had both inclusions and extras?

```sql
SELECT
    COUNT(roll_id) AS total_rolls_delivered
FROM
    customer_orders
WHERE
    not_include_items
        OR extra_items_included > 0
        AND order_id NOT IN (SELECT
            order_id
        FROM
            driver_order
        WHERE
            cancellation IN ('cancellation' , 'Customer Cancellation'));
```

Result Grid | Filter Rows:

total_rolls_delivered
7

8. What was the total number of rolls ordered each hour?

```sql
SELECT
    CONCAT(EXTRACT(HOUR FROM order_date),
            ' to ',
            EXTRACT(HOUR FROM order_date) + 1) AS hour_interval,
    COUNT(order_id) AS total_orders_per_hour
FROM
    customer_orders
GROUP BY hour_interval
ORDER BY hour_interval;
```

Result Grid | Filter Rows:

hour_interval	total_orders_per_hour
11 to 12	1
13 to 14	3
18 to 19	3
19 to 20	1
21 to 22	3
23 to 24	3

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

```sql
SELECT
    DAYNAME(order_date) AS day_of_week,
    COUNT( distinct order_id) AS total_orders_of_day
FROM
    customer_orders
GROUP BY day_of_week
ORDER BY day_of_week;
```

Result Grid | Filter Rows:

day_of_week	total_orders_of_day
Friday	5
Monday	2
Saturday	2
Sunday	1

10. What was the average distance travelled for each customer?

```sql
SELECT
    customer_id, ROUND(AVG(distance), 2) AS avg_distance_km
FROM
    driver_order do
        JOIN
    customer_orders co ON do.order_id = co.order_id
GROUP BY customer_id;
```

Result Grid | Filter Rows:

customer_id	avg_distance_km
101	20
102	16.73
103	23.4
104	10
105	25

11. What is the average time taken to deliver an order?

```sql
WITH cte AS (
    SELECT
        COUNT(order_id) AS total_order,
        SUM(duration) AS total_duration
    FROM
        driver_order
    WHERE
        duration IS NOT NULL
)
SELECT
    CAST(total_duration AS float ) / total_order AS Average_duration_for_an_Delivery
FROM
    cte;
```

Result Grid | Filter Rows:

Average_duration_for_an_Delivery
23

12. What is the difference between the longest and the shortest delivery time for all orders?

```
SELECT
    CONCAT(MAX(duration), ' ', 'Minutes') AS longest_time,
    MIN(duration) AS shortest_time,
    CONCAT(MAX(duration) - MIN(duration),
            ' ',
            'Minutes') AS diff_delivery_time
FROM
    driver_order;
```

Result Grid | Filter Rows:

longest_time	shortest_time	diff_delivery_time
40 Minutes	10minutes	30 Minutes

13.What is the average speed for each driver each delivery?

```sql
SELECT
    order_id,
    driver_id,
    distance,
    duration,
    ROUND(CONCAT((distance* 1000) / (duration * 60),2), ' m/s') AS 'speed in (m/s)'
FROM (
    SELECT
        order_id,
        driver_id,
        CAST((distance) AS FLOAT) AS distance,
        CAST(LEFT(duration, 2) AS FLOAT) AS duration
    FROM
        driver_order
    WHERE
        distance IS NOT NULL
) AS a;
```

Result Grid | Filter Rows: | Export:

order_id	driver_id	distance	duration	speed in (m/s)
1	1	20	32	10
2	1	20	27	12
3	1	13.4	20	11
4	2	23.4	40	10
5	3	10	15	11
7	2	25	25	17
8	2	23.4	15	26
10	1	10	10	17

14. What is cancellation percentage for each driver?

```sql
WITH cte AS (
    SELECT
        driver_id,
        COUNT(driver_id) AS total_orders,
        SUM(CASE WHEN cancellation IN ('cancellation', 'Customer Cancellation') THEN 1 ELSE 0 END) AS total_cancellations
    FROM
        driver_order
    GROUP BY
        driver_id
)
SELECT
    driver_id,
    CONCAT(ROUND((total_cancellations * 100.0) / total_orders, 2), '%') AS cancellation_percentage
FROM
    cte;
```

Result Grid | Filter Rows:

driver_id	cancellation_percentage
1	0.00%
2	25.00%
3	50.00%