SQL PROJECT

FAASOS DATA ANALYSIS

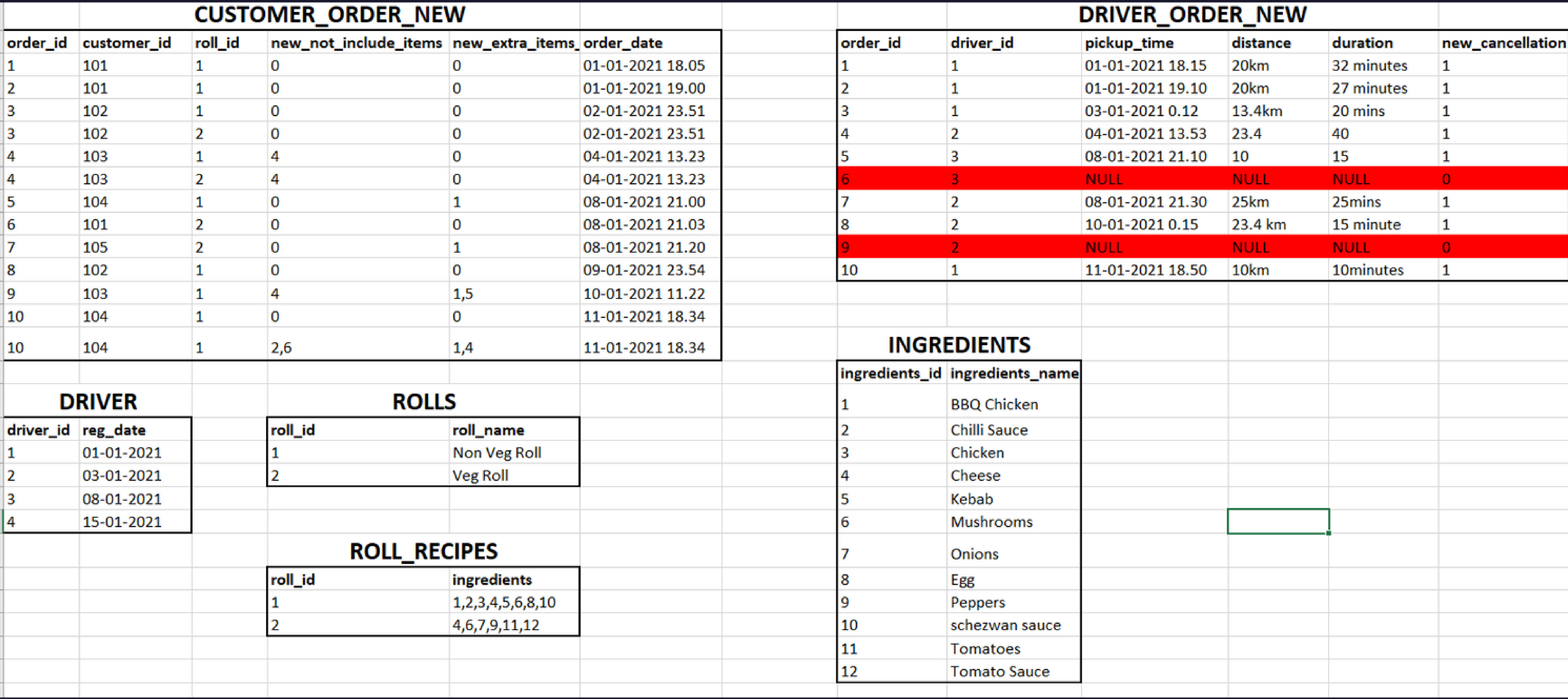
LinkedIn

GitHub

OBJECTIVE

This project's objective is to use data-driven insights derived from FAASOS delivery data to improve **operational efficiency, customer satisfaction, and business performance**. By analysing various aspects of the delivery process, such as driver performance, order customization, delivery times, and customer preferences, the project intends to provide actionable recommendations for optimising delivery services, enhancing order accuracy, and adapting the customer experience to meet changing demands.

A GLIMPSE OF OUR FAASOS DATA



CUSTOMER_ORDER_NEW

order_id	customer_id	roll_id	new_not_include_items	new_extra_items_	order_date
1	101	1	0	0	01-01-2021 18.05
2	101	1	0	0	01-01-2021 19.00
3	102	1	0	0	02-01-2021 23.51
3	102	2	0	0	02-01-2021 23.51
4	103	1	4	0	04-01-2021 13.23
4	103	2	4	0	04-01-2021 13.23
5	104	1	0	1	08-01-2021 21.00
6	101	2	0	0	08-01-2021 21.03
7	105	2	0	1	08-01-2021 21.20
8	102	1	0	0	09-01-2021 23.54
9	103	1	4	1,5	10-01-2021 11.22
10	104	1	0	0	11-01-2021 18.34
10	104	1	2,6	1,4	11-01-2021 18.34

DRIVER_ORDER_NEW

order_id	driver_id	pickup_time	distance	duration	new_cancellation
1	1	01-01-2021 18.15	20km	32 minutes	1
2	1	01-01-2021 19.10	20km	27 minutes	1
3	1	03-01-2021 0.12	13.4km	20 mins	1
4	2	04-01-2021 13.53	23.4	40	1
5	3	08-01-2021 21.10	10	15	1
6	3	NULL	NULL	NULL	0
7	2	08-01-2021 21.30	25km	25mins	1
8	2	10-01-2021 0.15	23.4 km	15 minute	1
9	2	NULL	NULL	NULL	0
10	1	11-01-2021 18.50	10km	10minutes	1

DRIVER

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

ROLLS

roll_id	roll_name
1	Non Veg Roll
2	Veg Roll

ROLL_RECIPES

roll_id	ingredients
1	1,2,3,4,5,6,8,10
2	4,6,7,9,11,12

INGREDIENTS

ingredients_id	ingredients_name
1	BBQ Chicken
2	Chilli Sauce
3	Chicken
4	Cheese
5	Kebab
6	Mushrooms
7	Onions
8	Egg
9	Peppers
10	schezwan sauce
11	Tomatoes
12	Tomato Sauce

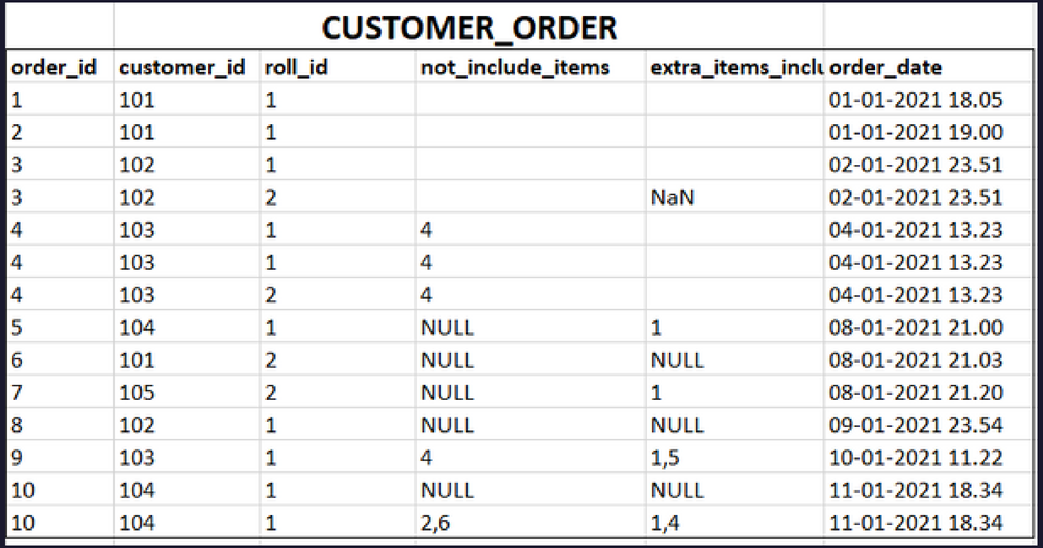
Cleaning and Transforming Tables

```
drop table if exists customer_order_new;
CREATE TABLE customer_order_new AS
(
 SELECT DISTINCT order_id,customer_id,roll_id,
 CASE WHEN not_include_items = '  ' OR not_include_items IS NULL
THEN '0' ELSE not_include_items END AS new_not_include_items,
  CASE WHEN extra_items_included IN ('  '|'NaN') OR
extra_items_included IS NULL THEN '0' ELSE extra_items_included
END AS new_extra_items_incuded,
  order_date
  FROM customer_orders
);
```

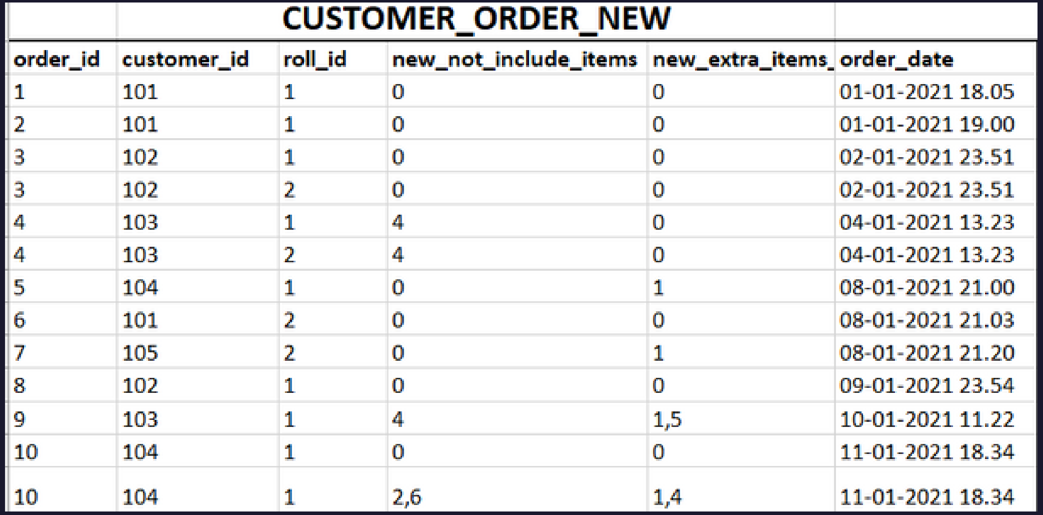
--> Creating a customer_order_new table

--> Making all 'empty' and 'null' values to '0' in two columns

--> Removing duplicates rows from data

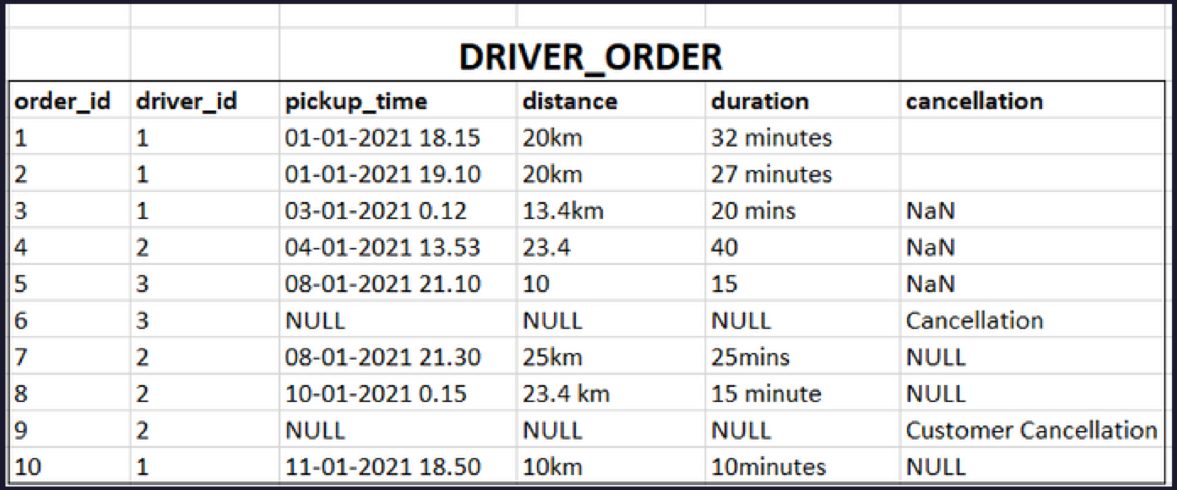


		CUSTOMER_ORDER			
order_id	customer_id	roll_id	not_include_items	extra_items_inclu	order_date
1	101	1			01-01-2021 18.05
2	101	1			01-01-2021 19.00
3	102	1			02-01-2021 23.51
3	102	2		NaN	02-01-2021 23.51
4	103	1	4		04-01-2021 13.23
4	103	1	4		04-01-2021 13.23
4	103	2	4		04-01-2021 13.23
5	104	1	NULL	1	08-01-2021 21.00
6	101	2	NULL	NULL	08-01-2021 21.03
7	105	2	NULL	1	08-01-2021 21.20
8	102	1	NULL	NULL	09-01-2021 23.54
9	103	1	4	1,5	10-01-2021 11.22
10	104	1	NULL	NULL	11-01-2021 18.34
10	104	1	2,6	1,4	11-01-2021 18.34



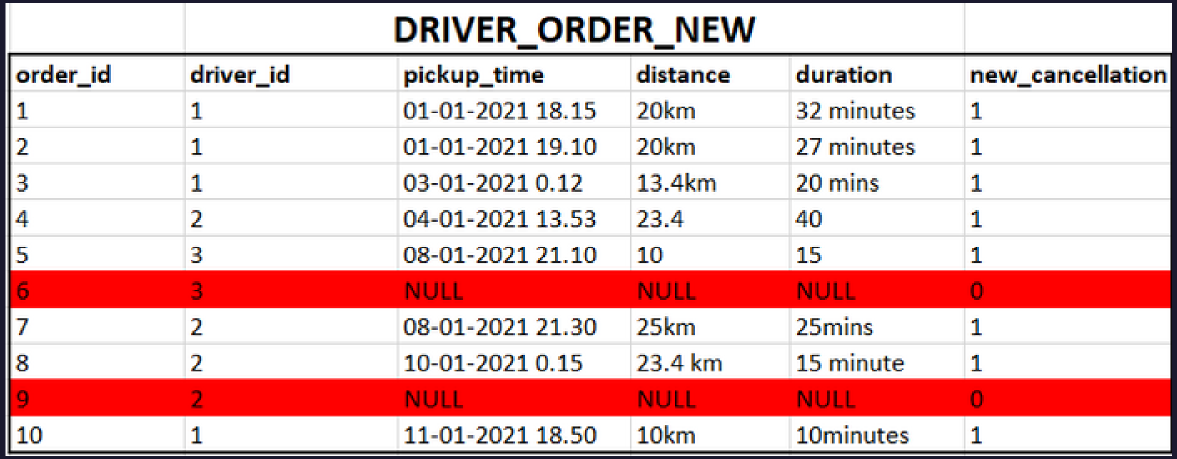
		CUSTOMER_ORDER_NEW			
order_id	customer_id	roll_id	new_not_include_items	new_extra_items_	order_date
1	101	1	0	0	01-01-2021 18.05
2	101	1	0	0	01-01-2021 19.00
3	102	1	0	0	02-01-2021 23.51
3	102	2	0	0	02-01-2021 23.51
4	103	1	4	0	04-01-2021 13.23
4	103	2	4	0	04-01-2021 13.23
5	104	1	0	1	08-01-2021 21.00
6	101	2	0	0	08-01-2021 21.03
7	105	2	0	1	08-01-2021 21.20
8	102	1	0	0	09-01-2021 23.54
9	103	1	4	1,5	10-01-2021 11.22
10	104	1	0	0	11-01-2021 18.34
10	104	1	2,6	1,4	11-01-2021 18.34

```
drop table if exists driver_order_new;
CREATE TABLE driver_order_new AS
(
SELECT
order_id,driver_id,pickup_time,distance,duration,
CASE WHEN cancellation IN ('Cancellation','Customer
Cancellation') THEN '0' ELSE 1 END AS new_cancellation
FROM driver_order
);
```



DRIVER_ORDER

order_id	driver_id	pickup_time	distance	duration	cancellation
1	1	01-01-2021 18.15	20km	32 minutes	
2	1	01-01-2021 19.10	20km	27 minutes	
3	1	03-01-2021 0.12	13.4km	20 mins	NaN
4	2	04-01-2021 13.53	23.4	40	NaN
5	3	08-01-2021 21.10	10	15	NaN
6	3	NULL	NULL	NULL	Cancellation
7	2	08-01-2021 21.30	25km	25mins	NULL
8	2	10-01-2021 0.15	23.4 km	15 minute	NULL
9	2	NULL	NULL	NULL	Customer Cancellation
10	1	11-01-2021 18.50	10km	10minutes	NULL



DRIVER_ORDER_NEW

order_id	driver_id	pickup_time	distance	duration	new_cancellation
1	1	01-01-2021 18.15	20km	32 minutes	1
2	1	01-01-2021 19.10	20km	27 minutes	1
3	1	03-01-2021 0.12	13.4km	20 mins	1
4	2	04-01-2021 13.53	23.4	40	1
5	3	08-01-2021 21.10	10	15	1
6	3	NULL	NULL	NULL	0
7	2	08-01-2021 21.30	25km	25mins	1
8	2	10-01-2021 0.15	23.4 km	15 minute	1
9	2	NULL	NULL	NULL	0
10	1	11-01-2021 18.50	10km	10minutes	1

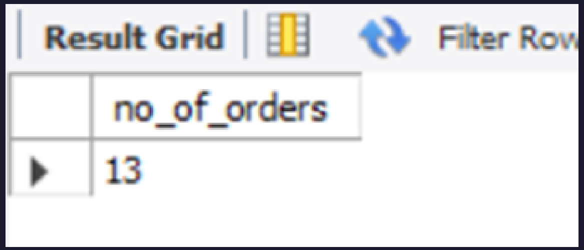
--> Creating a driver_order_new table

--> Making ‘cancellation’ and ‘customer cancellation’ to ‘0’ other to ‘1’

Question Statements

1)HOW MANY ROLLS WERE ORDERED?

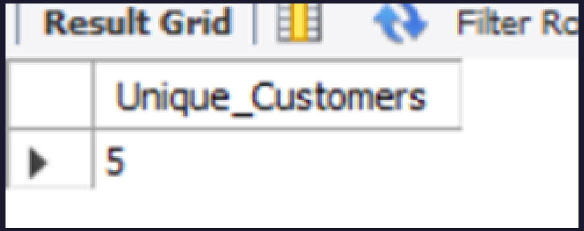
```
SELECT COUNT(roll_id) AS no_of_orders
FROM customer_order_new;
```



no_of_orders
13

2)HOW MANY UNIQUE CUSTOMERS WERE MADE ?

```
SELECT COUNT(*) AS Unique_Customers
FROM (SELECT customer_id FROM customer_order_new
  GROUP BY customer_id ) AS no_of_customer;
```



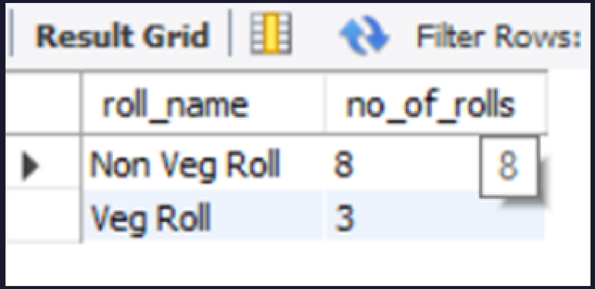
Unique_Customers
5

--> Total number of orders are 13

--> Among 5 unique customers

3)HOW MANY EACH TYPE OF ROLL WERE DELIVERED?

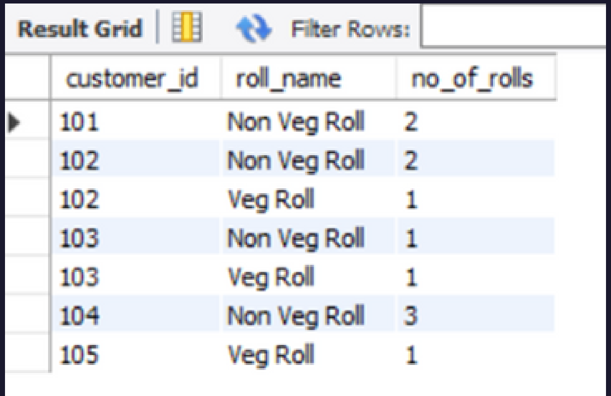
```
SELECT roll_name,COUNT(a.roll_id) AS no_of_rolls
FROM (SELECT order_id,rolls.roll_id,roll_name FROM customer_order_new con
   JOIN rolls
  ON rolls.roll_id = con.roll_id
   WHERE order_id IN ( SELECT order_id FROM driver_order_new
     WHERE new_cancellation <> 0 ) ) a
GROUP BY roll_name;
```



roll_name	no_of_rolls
Non Veg Roll	8
Veg Roll	3

4)HOW MANY VEG AND NON-VEG ROLLS WERE ORDERED BY EACH CUSTOMER?

```
SELECT customer_id,roll_name,COUNT(rolls.roll_id) AS no_of_rolls
FROM (SELECT * FROM customer_order_new
   WHERE order_id IN ( SELECT order_id FROM driver_order_new
        WHERE new_cancellation <> 0 )) AS customer_roll , rolls
WHERE customer_roll.roll_id = rolls.roll_id
GROUP BY customer_id,roll_name;
```



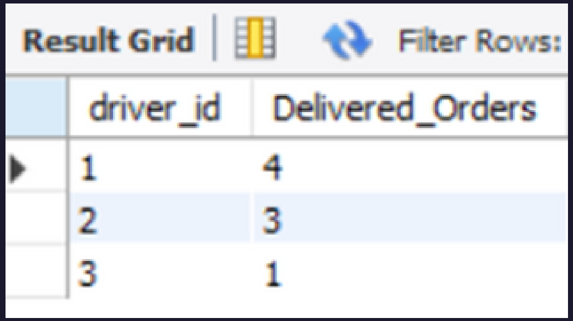
customer_id	roll_name	no_of_rolls
101	Non Veg Roll	2
102	Non Veg Roll	2
102	Veg Roll	1
103	Non Veg Roll	1
103	Veg Roll	1
104	Non Veg Roll	3
105	Veg Roll	1

--> If a total of 13 rolls were ordered but only 11(8+3) were delivered, then two orders were cancelled

--> Most of the rolls delivered are Non-Veg(8)

5)HOW MANY SUCCESSFUL ORDERS WERE DELIVERD BY EACH DRIVER?

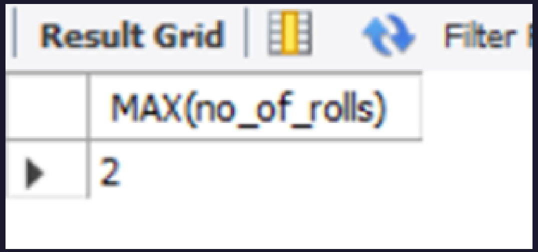
```
SELECT driver_id,COUNT(order_id) AS Delivered_Orders
FROM (SELECT * FROM driver_order_new
  WHERE new_cancellation != 0  ) a
GROUP BY driver_id;
```



driver_id	Delivered_Orders
1	4
2	3
3	1

6)WHAT WAS THE MAXIMAM NUMBER OF ROLLS DELIVERED IN SINGLE ORDER?

```
SELECT MAX(no_of_rolls) FROM (
 SELECT order_id,COUNT(roll_id) AS no_of_rolls FROM (
  SELECT * FROM customer_order_new
    WHERE order_id IN ( SELECT order_id FROM driver_order_new
        WHERE new_cancellation <> 0 ))a
 GROUP BY order_id) b;
```



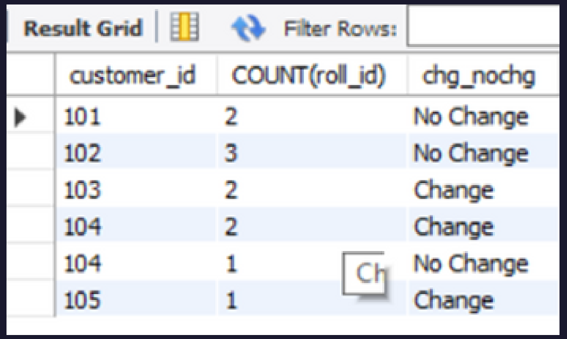
MAX(no_of_rolls)
2

--> Driver_id ‘1’ has delivered maximum no_of_orders:4

--> Maximum rolls delivered in a single order: 2

7)FOR EACH CUSTOMER, HOW MANY DELIVERED ROLLS (HAD AT LEAST 1 CHANGE) AND (HOW MANY DON'T HAVE)?

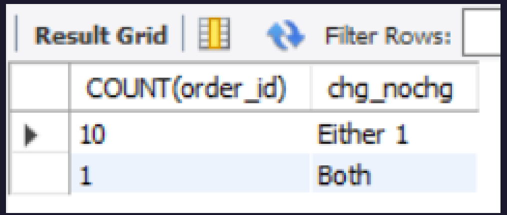
```
SELECT customer_id,COUNT(roll_id),chg_nochg
FROM(SELECT *,CASE WHEN new_not_include_items = '0' AND new_extra_items_incuded =
'0'
 THEN 'No Change' ELSE 'Change' END chg_nochg FROM customer_order_new
WHERE order_id IN ( SELECT order_id FROM driver_order_new
     WHERE new_cancellation <> 0 ))a
GROUP BY customer_id,chg_nochg
ORDER BY customer_id;
```



customer_id	COUNT(roll_id)	chg_nochg
101	2	No Change
102	3	No Change
103	2	Change
104	2	Change
104	1	No Change
105	1	Change

8)HOW MANY ROLLS WERE DELIVERED THAT HAD BOTH EXCLUSINS AND EXTRAS?

```
SELECT COUNT(order_id),chg_nochg
FROM (SELECT *,CASE WHEN new_not_include_items != '0' AND new_extra_items_incuded !='0'
          THEN 'Both' ELSE 'Either 1' END chg_nochg FROM customer_order_new
WHERE order_id IN ( SELECT order_id FROM driver_order_new
                              WHERE new_cancellation <> 0 ) ) AS a
GROUP BY chg_nochg;
```



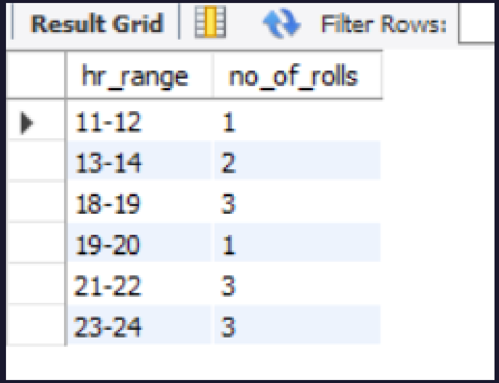
COUNT(order_id)	chg_nochg
10	Either 1
1	Both

--> More than 50% of customers has ordered to made changes in their rolls

--> Only 1 roll has both exclusion and extra items

9)WHAT WAS THE TOTAL NUMBER OF ROLLS ORDERED FOR EACH HOUR OF THE DAY?

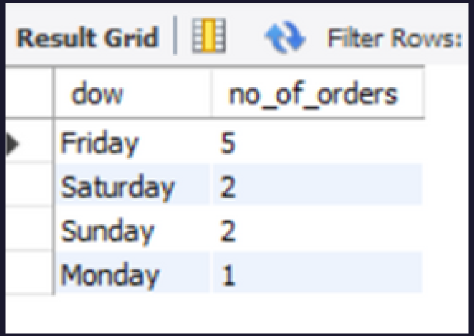
```
SELECT hr_range,COUNT(roll_id) AS no_of_rolls FROM(
SELECT *,CONCAT(EXTRACT(hour FROM order_date) ,'-',EXTRACT(hour FROM order_date)+1)
AS hr_range FROM customer_order_new) AS a
GROUP BY hr_range
ORDER BY hr_range;
```



hr_range	no_of_rolls
11-12	1
13-14	2
18-19	3
19-20	1
21-22	3
23-24	3

10)WHAT WAS THE NUMBER OF ORDERS FOR EACH DAY OF WEEK?

```
SELECT dow,COUNT(DISTINCT order_id) AS no_of_orders
FROM(SELECT *,dayname(order_date) AS dow FROM customer_order_new) a
GROUP BY dow
ORDER BY no_of_orders DESC;
```



dow	no_of_orders
Friday	5
Saturday	2
Sunday	2
Monday	1

--> Rush hours during which demand is significantly higher, between 21:00 and 24:00, especially on Fridays

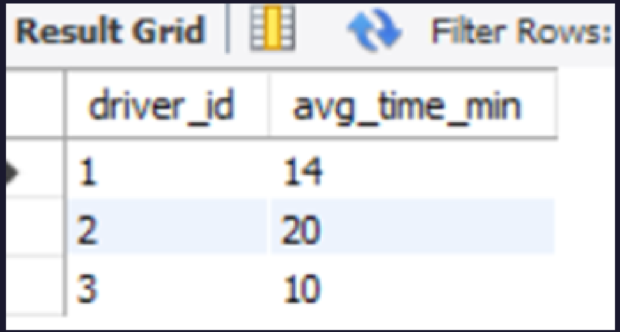
-->Then comes Saturday and sunday

11)WHAT WAS THE AVERAGE TIME IN MINUTES IT TOOK FOR EACH DRIVER TO ARRIVE AT THE FAASOS HQ TO PICKUP THE ORDER?

```
SELECT driver_id, ROUND(AVG(DIFF)) AS avg_time_min FROM (
SELECT a.order_id ,d.driver_id,order_date,d.pickup_time ,time_to_sec((TIMEDIFF(pickup_time,order_date)))/60 AS DIFF
FROM (SELECT order_id,order_date FROM customer_order_new
   GROUP BY order_id,order_date) As a
   JOIN driver_order_new AS d
  ON a.order_id = d.order_id
   WHERE pickup_time IS NOT NULL ) b
GROUP BY driver_id;
```

--> Driver arrival time ranges between 10-20 Min

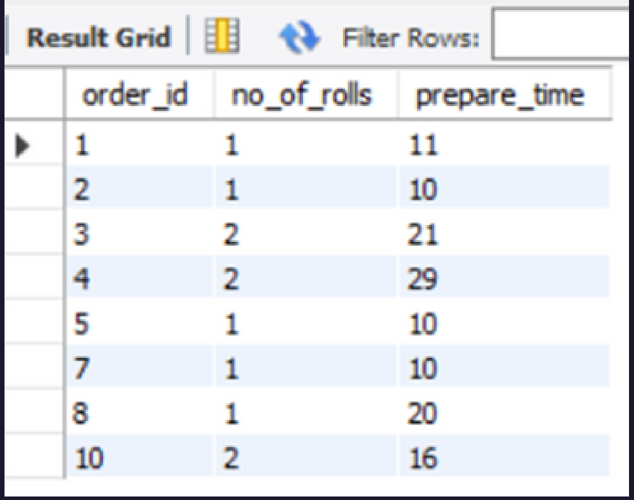
--> As most of the orders are delivered by driver_id ‘1’, It’s average time came out to be 14 Min



driver_id	avg_time_min
1	14
2	20
3	10

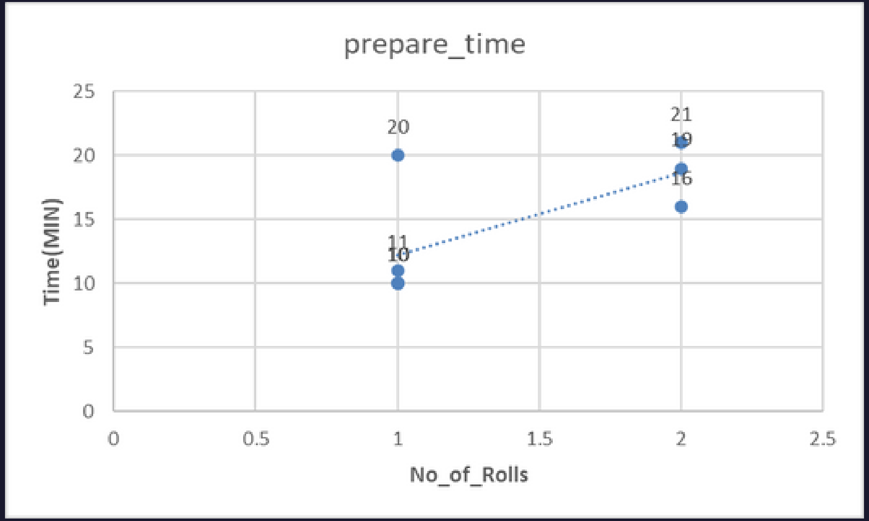
12)IS THERE ANY RELATION BETWEEN THE NUMBER OF ROLLS AND HOW LONG THE ORDER TO PREPARE?

```
SELECT order_id,no_of_rolls, ROUND( time_to_sec((TIMEDIFF(pickup_time,order_date)))/60) AS prepare_time
FROM (SELECT order_id,customer_id,order_date,pickup_time,count(roll_id) AS no_of_rolls
FROM (SELECT con.*,d.pickup_time FROM customer_order_new con
   JOIN driver_order_new d
  ON d.order_id = con.order_id
   WHERE d.pickup_time IS NOT NULL ) a
GROUP BY order_id,customer_id,order_date,pickup_time) b;
```



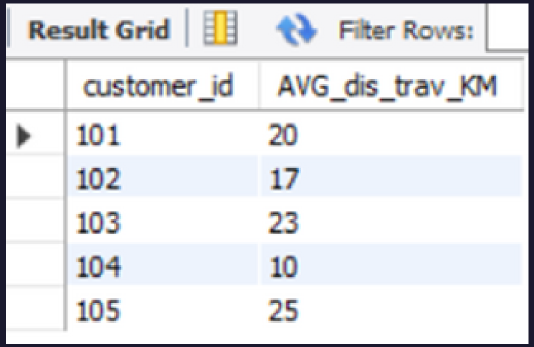
order_id	no_of_rolls	prepare_time
1	1	11
2	1	10
3	2	21
4	2	29
5	1	10
7	1	10
8	1	20
10	2	16

--> It was observed that the time required to prepare orders is directly related to the number of rolls in the order. Specifically, for each roll added to an order, the preparation time increases by 10 minutes



13)WHAT WAS THE AVERAGE DISTANCE TRAVELLED FOR EACH CUSTOMER

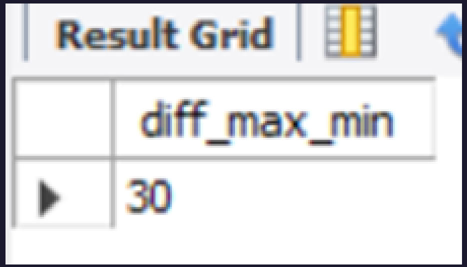
```
SELECT con.customer_id, ROUND(AVG(don.distance)) AS AVG_dis_trav_KM
FROM driver_order_new don, customer_order_new con
WHERE don.order_id=con.order_id AND pickup_time <> 0
GROUP BY customer_id;
```



customer_id	AVG_dis_trav_KM
101	20
102	17
103	23
104	10
105	25

14)WHAT IS THE DIFFERENCE BETWEEN THE LONGEST AND SHORTEST DELIVERY TIMES FOR ALL ORDERS?

```
SELECT MAX(duration_new)-MIN(duration_new) AS diff_max_min FROM (
SELECT duration,CASE WHEN duration LIKE '%min%' THEN LEFT(duration,position('m' IN
duration)-1)
ELSE duration END AS duration_new FROM driver_order_new
WHERE duration <> 0)a;
```



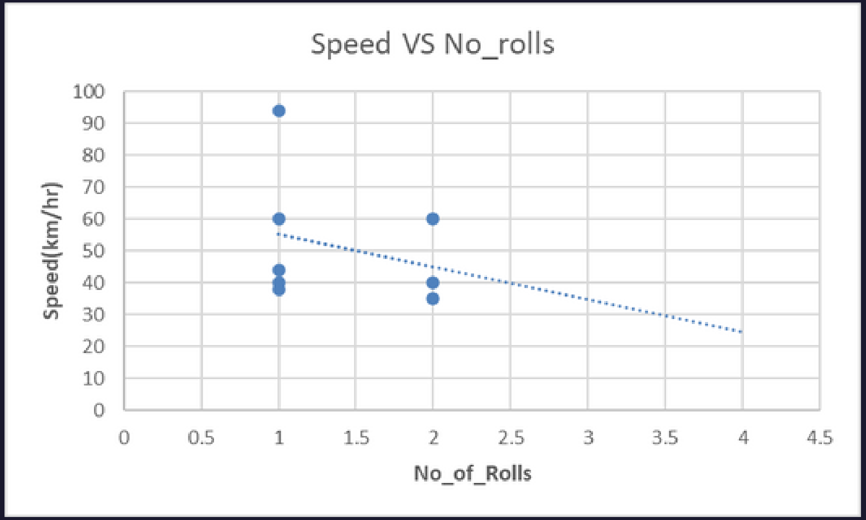
diff_max_min
30

--> Average distance traveled for each customer ranges between 10-25 km

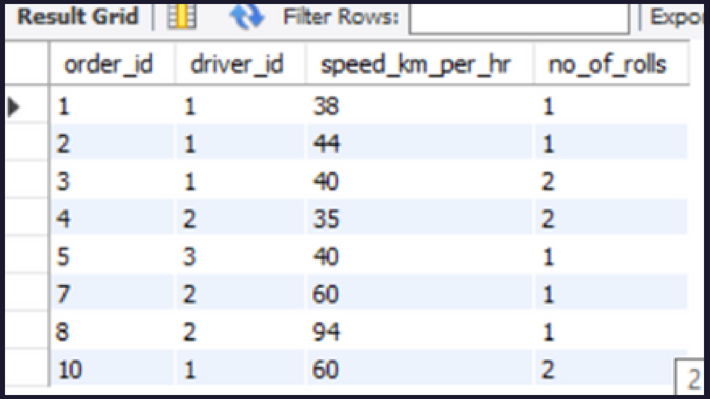
--> Difference between the longest and shortest delivery time is 30 minutes

15)WHAT WAS THE AVERAGE SPEED FOR EACH DRIVER FOR EACH DELIVERY AND DO YOU NOTICE ANY TREND FOR THESE VALUES?

```
SELECT a.order_id,driver_id,ROUND(distance_km/(duration_min/60)) AS speed_km_per_hr,COUNT(roll_id) AS
no_of_rolls FROM(
SELECT order_id,driver_id,
   CASE WHEN distance LIKE '%km%' THEN LEFT(distance,position('k'IN distance)-1) ELSE distance END AS
distance_km,
      CASE WHEN duration LIKE '%min%' THEN LEFT(duration,position('m' IN duration)-1) ELSE duration END AS
duration_min
FROM driver_order_new
WHERE pickup_time <> 0) a,customer_order_new AS con
WHERE a.order_id = con.order_id
GROUP BY a.order_id,driver_id,speed_km_per_hr;
```



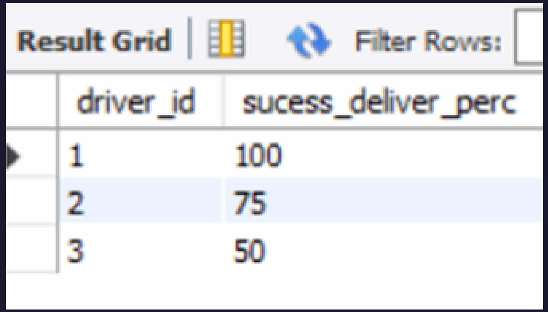
--> This indicates that as the number of rolls in an order increases, drivers tend to operate at a slightly lower speed, possibly due to the additional handling required for larger orders.



order_id	driver_id	speed_km_per_hr	no_of_rolls
1	1	38	1
2	1	44	1
3	1	40	2
4	2	35	2
5	3	40	1
7	2	60	1
8	2	94	1
10	1	60	2

16)WHAT IS THE SUCCESFUL DELIVERY PERCENTAGE FOR EACH DRIVER?

```
SELECT driver_id,(delivered/orders)*100 AS sucess_deliver_perc FROM (
SELECT driver_id,SUM(new_cancellation) AS delivered,COUNT(new_cancellation) AS orders
FROM driver_order_new
GROUP BY driver_id) a
```



driver_id	sucess_deliver_perc
1	100
2	75
3	50

CONCLUSION

Driver performance stands out, with Driver_id ‘1’ achieving a remarkable 100% success rate. Recognizing and motivating such high-performing drivers can significantly boost overall operational efficiency and enhance customer satisfaction. Customers' delivery distances vary, ranging from 10 to 25 kilometers, underlining the diverse delivery needs within the customer base. The delivery times exhibit a 30-minute difference between the longest and shortest, underscoring the need for consistent and efficient delivery processes.

Roll preparation time increases by 10 minutes for each roll added to an order, highlighting the importance of communicating this correlation to customers for more accurate order placement expectations. Driver arrival times typically fall within a prompt 10-20 minute range, averaging 14 minutes, indicating swift pickups from the FAASOS HQ. Recognizing peak demand periods, notably between 21:00 and 24:00, especially on Fridays, is crucial for effective resource allocation and service optimization during these high-demand windows.

Moreover, customer order customization is common, with over 50% of customers requesting changes to their rolls, emphasizing the value of customization options in meeting individual preferences. Interestingly, orders with both exclusions and extra items are relatively rare, suggesting a less common order customization scenario.

RECOMMENDATION

- **Peak Hour Preparedness:** During rush hours, prepare additional resources and drivers to meet increased demand, ensuring timely deliveries.
- **Driver Recognition:** Recognize and incentivize high-performing drivers, such as Driver ‘1’, to maintain exceptional success rates and service quality.
- **Efficient Packaging:** Optimize packaging for larger orders to minimize handling time while maintaining order quality.
- **Customization Options:** Expand customization options and promote them to cater to customer preferences. Develop recommendations for common customization scenarios.
- **Driver Training:** Continue to train drivers on efficient routes and order handling for larger orders. Emphasize the importance of prompt pickups from the FAASOS HQ.
- **Order Consistency:** Ensure consistent delivery processes to reduce the 30-minute variance in delivery times. Monitor and address any process bottlenecks.
- **Feedback Mechanism:** Establish a feedback mechanism to gather insights from both drivers and customers. Use this data to continually refine and improve operations.