

## FAASOS ROLLS SALES

1. How many rolls were ordered?

```
SELECT count(roll_id)
FROM customer_orders
```

Results		Messages	
	(No column name)		
1	14		

2. How many unique customer orders were made?

```
SELECT count(DISTINCT customer_id)
FROM customer_orders
```

Results		Messages	
	(No column name)		
1	5		

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

```
SELECT driver_id,
       count(order_id) AS No_of_orders_deleivered_by_Driver
FROM driver_order
WHERE duration is NOT null
GROUP BY driver_id
```

Results		Messages	
	driver_id	No_of_orders_deleivered_by_Driver	
1	1	4	
2	2	3	
3	3	1	

4. How many of each type of roll was delivered?

```
SELECT r.roll_name,
       c.roll_id,
       count(c.order_id)
FROM rolls r
JOIN customer_orders c
  ON r.roll_id = c.roll_id
JOIN driver_order d
  ON c.order_id = d.order_id
WHERE duration is NOT null
GROUP BY r.roll_name, c.roll_id
```

	roll_name	roll_id	(No column name)
1	Non Veg Roll	1	9
2	Veg Roll	2	3

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

```
SELECT customer_id,
       c.roll_id,
       count(c.roll_id) count,
       roll_name
FROM rolls r
JOIN customer_orders c
ON r.roll_id = c.roll_id
GROUP BY customer_id, c.roll_id, roll_name
ORDER BY roll_id
```

	customer_id	roll_id	count	roll_name
1	101	1	2	Non Veg Roll
2	102	1	2	Non Veg Roll
3	103	1	3	Non Veg Roll
4	104	1	3	Non Veg Roll
5	101	2	1	Veg Roll
6	102	2	1	Veg Roll
7	103	2	1	Veg Roll
8	105	2	1	Veg Roll

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

```
SELECT *
FROM (SELECT *,
       rank()
OVER (ORDER BY [rolls Ordered] desc) rnk
FROM (SELECT *
FROM (SELECT c.order_id,
       count(c.roll_id) AS [rolls Ordered]
FROM customer_orders c
JOIN driver_order d
ON c.order_id = d.order_id
WHERE duration is NOT null
GROUP BY c.order_id) a) b) c
WHERE rnk = 1
```

	order_id	rolls Ordered	mk
1	4	3	1

7. For each customer, how many delivered rolls had at least 1 change and how many had no change?

```
WITH temp_customer_orders (order_id,
    customer_id,
    roll_id,
    not_include_items,
    extra_items_included,
    order_date) as
(
    SELECT order_id,
        customer_id,
        roll_id,

        CASE
            WHEN not_include_items is null
                OR not_include_items = ' ' THEN
                '0'
            ELSE not_include_items
        END AS new_not_include_items,

        CASE
            WHEN extra_items_included is null
                OR extra_items_included = ' '
                OR extra_items_included = 'NaN' THEN
                '0'
            ELSE
                extra_items_included
        END AS
        new_extra_items_included,order_date
    FROM customer_orders
),

temp_driver_order (order_id,
    driver_id,
    pickup_time,
    distance,
    duration,
    new_cancellation) 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
)
SELECT customer_id,
    chg_no_chg,
    count(order_id) at_least_one_change from
```

```

(
SELECT *,

CASE
WHEN not_include_items = '0'
      AND extra_items_included = '0' THEN
  'no change'
ELSE 'change'
END chg_no_chg
FROM temp_customer_orders
WHERE order_id IN (
SELECT order_id
FROM temp_driver_order
WHERE new_cancellation!=0))a
GROUP BY customer_id, chg_no_chg;

```

Results Messages			
	customer_id	chg_no_chg	at_least_one_change
1	103	change	3
2	104	change	2
3	105	change	1
4	101	no change	2
5	102	no change	3
6	104	no change	1

8. How many rolls were delivered that had both exclusions and extras?

```
WITH temp_customer_orders (order_id,
    Customer_id,
    roll_id,
    not_include_items,
    extra_items_included,
    order_date)
as
(SELECT order_id,
    customer_id,
    roll_id,

    CASE
    WHEN not_include_items is null
        OR not_include_items = ' ' THEN
        '0'
    ELSE not_include_items
    END
AS new_not_included_items,

    CASE
    WHEN extra_items_included is null
        OR extra_items_included = 'NaN'
        OR extra_items_included = ' '
    then '0'
    ELSE extra_items_included end
AS new_extra_items_included,
    order_date
FROM customer_orders),

temp_driver_order (order_id,
    driver_id,
    pickup_time,
    distance,
    duration,
    new_cancellation) 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)

SELECT chg_no_chg,
    count(chg_no_chg) from
```

```

(
SELECT *,

CASE
WHEN not_include_items != '0'
      AND extra_items_included != '0' THEN
  'both inc exc'
ELSE 'either 1 inc
      OR exc'
END chg_no_chg
FROM temp_customer_orders
WHERE order_id IN (
SELECT order_id
FROM temp_driver_order
WHERE new_cancellation!=0))a
GROUP BY chg_no_chg;

```

Results		Messages
	chg_no_chg	(No column name)
1	both inc exc	1
2	either 1 inc      OR exc	11

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

```
SELECT hours_bucket,  
       count(hours_bucket) from  
(SELECT *,  
        concat(cast(datepart(hour,  
        order_date) AS varchar) ,  
        '-',  
        cast(datepart(hour,  
        order_date)+1 AS varchar))  
hours_bucket  
FROM customer_orders)a  
GROUP BY hours_bucket;
```

	hours_bucket	(No column name)
1	11-12	1
2	13-14	3
3	18-19	3
4	19-20	1
5	21-22	3
6	23-24	3

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

```
SELECT dow,  
       count(DISTINCT order_id) from  
(SELECT *,  
        DATENAME(dw,  
        order_date) dow  
FROM customer_orders) a  
GROUP BY dow
```

	dow	(No column name)
1	Friday	5
2	Monday	2
3	Saturday	2
4	Sunday	1

11. What was the average time in minutes it took for each driver to arrive at the fasoos HQ to pickup the order?

```
SELECT driver_id,  
       sum(diff)/count(order_id) avg_mins from  
(SELECT * from  
(SELECT *,  
         row_number() over(partition by order_id  
ORDER BY diff) rnk from  
(SELECT a.order_id,  
         a.customer_id,  
         a.roll_id,  
         a.not_include_items,  
         a.extra_items_included,  
         a.order_date,  
  
         b.driver_id,  
         b.pickup_time,  
         b.distance,  
         b.duration,  
         b.cancellation,  
         DATEDIFF(minute,  
         a.order_date,  
         b.pickup_time) diff  
FROM customer_orders a  
JOIN driver_order b  
  ON a.order_id = b.order_id  
WHERE b.pickup_time is NOT null)a)b  
WHERE rnk =1)c  
GROUP BY driver_id
```

Results		Messages
	driver_id	avg_mins
1	1	14
2	2	20
3	3	10



12. Is there any relationship between the number of rolls and how long the order takes to prepare?

```
SELECT order_id,  
       count(roll_id) cnt,  
       sum(diff)/count(roll_id) tmt from  
(SELECT a.order_id,  
        a.customer_id,  
        a.roll_id,  
        a.not_include_items,  
        a.extra_items_included,  
        a.order_date,  
  
        b.driver_id,  
        b.pickup_time,  
        b.distance,  
        b.duration,  
        b.cancellation,  
        DATEDIFF(minute,  
        a.order_date,  
        b.pickup_time) diff  
FROM customer_orders a  
JOIN driver_order b  
  ON a.order_id = b.order_id  
WHERE b.pickup_time is NOT null)a  
GROUP BY order_id
```

	order_id	cnt	tmt
1	1	1	10
2	2	1	10
3	3	2	21
4	4	3	30
5	5	1	10
6	7	1	10
7	8	1	21
8	10	2	16

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

```
SELECT customer_id,  
       sum(distance)/count(order_id) avg_distance from  
(SELECT * from  
(SELECT *,  
         row_number() over(partition by order_id  
ORDER BY diff) rnk from  
(SELECT a.order_id,  
         a.customer_id,  
         a.roll_id,  
         a.not_include_items,  
         a.extra_items_included,  
         a.order_date,  
  
         b.driver_id,  
         b.pickup_time,  
  
         cast(trim(replace(lower(b.distance),  
                           'km',  
                           '')) AS decimal(4,  
                           2)) distance  
,  
         b.duration,  
         b.cancellation,  
         DATEDIFF(minute,  
         a.order_date,  
         b.pickup_time) diff  
FROM customer_orders a  
JOIN driver_order b  
  ON a.order_id = b.order_id  
WHERE b.pickup_time is NOT null)a)b  
  WHERE rnk =1)c  
GROUP BY customer_id
```

Results		Messages
	customer_id	avg_distance
1	101	20.000000
2	102	18.400000
3	103	23.400000
4	104	10.000000
5	105	25.000000

14. What is the difference between the longest and shortest delivery times for all orders?

```
SELECT max(duration) - min(duration) diff from
(SELECT cast(
CASE
WHEN duration LIKE '%min%' THEN
left(duration, charindex('m',duration)-1)else
duration
END AS integer) AS duration from
driver_order
WHERE duration is NOT null)a
```

Results		Messages	
diff			
1	30		

15. What was the average speed for each driver for each delivery and do you notice any trends for these values?

```
SELECT a.order_id,
a.driver_id,
a.distance/a.duration speed,
b.cnt from
(SELECT order_id,
driver_id,
CAST(trim(replace(lower(distance),
'km',
'')) AS decimal(4,
2)) distance
,
cast(
CASE
WHEN duration LIKE '%min%' THEN
left(duration, charindex('m', duration)-1)else
duration
END AS integer) AS duration
FROM driver_order
WHERE distance is NOT null)a inner join
(SELECT order_id,
count(roll_id)cnt
FROM customer_orders
GROUP BY order_id)b
ON a.order_id = b.order_id
```

	order_id	driver_id	speed	cnt
1	1	1	0.62500000000000	1
2	2	1	0.7407407407407	1
3	3	1	0.67000000000000	2
4	4	2	0.58500000000000	3
5	5	3	0.66666666666666	1
6	7	2	1.00000000000000	1
7	8	2	1.56000000000000	1
8	10	1	1.00000000000000	2

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

```

SELECT driver_id,
       (s*1.0/t)*100 cancelled_per from
(SELECT driver_id,
       sum(can_per) s,
       count(driver_id) t from
(SELECT driver_id,

       CASE
       WHEN lower(cancellation) LIKE '%cancel%' THEN
       0
       ELSE 1
       END AS can_per
FROM driver_order)
a
GROUP BY driver_id)b

```

	driver_id	cancelled_per
1	1	100.000000000000
2	2	75.000000000000
3	3	50.000000000000