**Faasos Case Study**

1. **Roll Metrics**
2. **How many rolls were ordered**

select count(order\_id) as total\_rolls\_ordered

from customer\_order;



1. **How many unique orders were made**

select count(distinct order\_id) as unique\_orders

from customer\_order;



1. **How many unique customers**

select count(distinct customer\_id) as unique\_customers

from customer\_order;



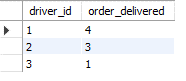
1. **How many successful orders delivered by each driver**

select driver\_id, count(order\_id) as order\_delivered

from driver\_order

where duration is not null

group by driver\_id;



1. **How many of each type of roll was delivered**

select roll\_id, count(roll\_id) as num\_of\_each\_roll\_delivered

from driver\_order d

inner join customer\_order c

on d.order\_id = c.order\_id

where d.duration is not null

group by roll\_id;



1. **How many veg and non-veg rolls were ordered by each customer**

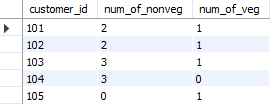
select customer\_id,

sum(case when roll\_id = 1 then 1 else 0 end) as num\_of\_nonveg,

sum(case when roll\_id <> 1 then 1 else 0 end) as num\_of\_veg

from customer\_order

group by customer\_id;



1. **Maximum number of rolls delivered in a single order**

select c.order\_id, count(c.order\_id) as max\_rolls\_per\_order

from customer\_order c

inner join driver\_order d

on c.order\_id = d.order\_id

where d.duration is not null

group by c.order\_id

order by 2 desc

limit 1;



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

with cte as

(select

case when not\_include\_items is null or not\_include\_items = '' then 0 else not\_include\_items end as exclude\_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 add\_items, c.\*

from customer\_order c

inner join driver\_order d

on c.order\_id = d.order\_id

where d.duration is not null)

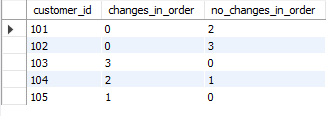
select customer\_id,

sum(case when exclude\_items > 0 or add\_items > 0 then 1 else 0 end) as changes\_in\_order,

sum(case when exclude\_items = 0 and add\_items = 0 then 1 else 0 end) as no\_changes\_in\_order

from cte

group by customer\_id;



1. **How many rolls were delivered that had both exclusions and extras**

with cte as

(select case when not\_include\_items is null or not\_include\_items = '' then 0 else not\_include\_items end as exclude\_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 add\_items, c.\*

from customer\_order c

inner join driver\_order d

on c.order\_id = d.order\_id

where d.duration is not null)

select order\_id, count(order\_id) order\_with\_both\_changes

from cte

where exclude\_items > 0 and add\_items > 0

group by order\_id;



1. **What was the total number of rolls ordered at each hour of the day**

with cte1 as

(select \*, hour(order\_date) as hour\_of\_day,

hour(addtime(order\_date, "01:00:00")) as hour\_plus\_one

from customer\_order),

cte2 as

(select \*, concat(hour\_of\_day, "-", hour\_one) as hour\_interval

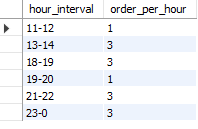
from cte1)

select hour\_interval, count(roll\_id) as order\_per\_hour

from cte2

group by hour\_interval

order by 1;



1. **what was the number of orders for each day of the week**

with cte as

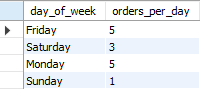
(select \*, dayname(order\_date) as day\_of\_week

from customer\_order)

select day\_of\_week, count(order\_id) orders\_per\_day

from cte

group by day\_of\_week;



1. **Driver and Customer Experience Metrics**
2. **What was the average time (in minutes) it took each driver to arrive at the Faasos outlet to pick up the order**

with cte1 as

(select c.\*, d.driver\_id, d.pickup\_time, timestampdiff(minute, order\_date, pickup\_time) as time\_to\_pickup

from customer\_order c

inner join driver\_order d

on c.order\_id = d.order\_id

where d.pickup\_time is not null),

cte2 as

(select \*,

row\_number() over(partition by order\_id order by time\_to\_pickup) as rnum

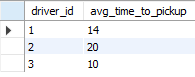
from cte1)

select driver\_id, round(avg(time\_to\_pickup),0) as avg\_time\_to\_pickup

from cte2

where rnum = 1

group by driver\_id;



1. **Is there any relation between the number of rolls and how long the order takes to prepare**

with cte as

(select c.\*, d.driver\_id, d.pickup\_time, timestampdiff(minute, order\_date, pickup\_time) as time\_to\_prepare\_order

from customer\_order c

inner join driver\_order d

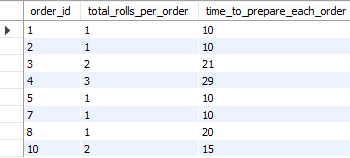
on c.order\_id = d.order\_id

where d.pickup\_time is not null)

select order\_id, count(order\_id) total\_rolls\_per\_order, time\_to\_prepare\_each\_order

from cte

group by order\_id**;**

****

1. **What was the average distance travelled for each customer**

with cte as

(select c.\*, trim(replace(d.distance, 'km', '')) distance\_travelled

from customer\_order c

inner join driver\_order d

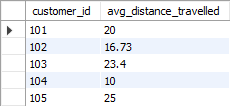
on c.order\_id = d.order\_id

where d.pickup\_time is not null)

select customer\_id, round(avg(distance\_travelled), 2) avg\_distance\_travelled

from cte

group by customer\_id;



1. **What was the difference between the longest and the shortest times for all orders**

with cte as

(select \*, left(duration,2) time\_taken\_to\_deliver

from driver\_order

where duration is not null)

select max(time\_taken\_to\_deliver) - min(time\_taken\_to\_deliver) range\_of\_delivery\_time

from cte;



1. **What was the average speed for each driver for each delivery is there any trend for these values**

with cte1 as

(select \*, trim(replace(distance, 'km', '')) distance\_travelled,

left(duration,2) time\_taken\_to\_deliver

from driver\_order

where duration is not null),

cte2 as

(select \*, distance\_travelled / time\_taken\_to\_deliver as speed\_of\_delivery

from cte1)

select c2.order\_id, driver\_id, round(avg(speed\_of\_delivery),2) as avg\_speed\_per\_delivery,

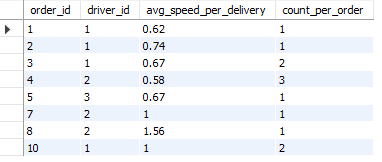
count(c.order\_id) count\_per\_order

from cte2 as c2

inner join customer\_order c

on c2.order\_id = c.order\_id

group by c2.order\_id, driver\_id;



1. **What is the successful delivery percentage for each driver**

select driver\_id,

round(sum(case when cancellation is null or cancellation = 'NaN' or cancellation = '' then 1 else 0 end)\*100.0 / count(\*),0) as completion\_percent,

round(sum(case when cancellation in ('Cancellation', 'Customer Cancellation') then 1 else 0 end)\*100.0 / count(\*),0) as cancellation\_percent

from driver\_order

group by driver\_id;

