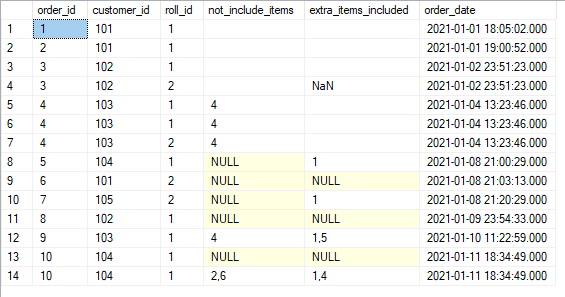
Data : C:\Users\Lenovo\AppData\Local\Temp\~vsB631.sql

Video link: https://www.youtube.com/watch?v=sCsXMDD0vls

Customers details:



Drivers orders details:

A screenshot of a number of vehicles

Description automatically generated

Ingredient details:

A screenshot of a menu

Description automatically generated

Drivers details: Rolls details: rolls ingredients:

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated A white rectangular object with black text

Description automatically generated

Solutions:

A. roll matrics:

1. how many rolls was ordered?

select count(roll\_id) as roll\_count from customer\_orders



2. how many unique customer orders were made?

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



3. success full orders delivered by drivers?

select \* from driver\_order

select driver\_id,count(distinct order\_id) as succesfuly\_delevered from driver\_order

where cancellation not in ('cancellation','Customer Cancellation')

group by driver\_id

A screenshot of a computer

Description automatically generated

4. how many rolls delivered( veg and nonveg)?

Step 1: get all the succesful delevery data

A screenshot of a data

Description automatically generated

Since here only 2 unsuccesful delivery so we need to remove it but if we use “where cancellation not in ('cancellation','Customer Cancellation')” it will remove null and blank also so we will use case.

select \* , case when cancellation in ('cancellation','Customer Cancellation')

then 'c' else 'nc' end as order\_cancel\_details from driver\_order

A screenshot of a screen

Description automatically generated

Details of succesful delivery:

select a.\* from (select \* , case when cancellation in ('cancellation','Customer Cancellation')

then 'c' else 'nc' end as order\_cancel\_details from driver\_order)a where order\_cancel\_details = 'nc'

A table with numbers and a few minutes

Description automatically generated with medium confidence

Step 2: get the roll information that is which customer order veg and nonveg rolls

succesfull customer orders:

select \* from customer\_orders where order\_id in

(select order\_id from (select \* , case when cancellation in ('cancellation','Customer Cancellation')

then 'c' else 'nc' end as order\_cancel\_details from driver\_order)a where order\_cancel\_details = 'nc')

A screenshot of a table

Description automatically generated

Final solution: So no of type of rolls sold

select roll\_id,count(roll\_id) Total\_count from customer\_orders where order\_id in

(select order\_id from (select \* , case when cancellation in ('cancellation','Customer Cancellation')

then 'c' else 'nc' end as order\_cancel\_details from driver\_order)a where order\_cancel\_details = 'nc')

group by roll\_id

A screenshot of a computer

Description automatically generated

5. how many veg and nonveg rolls were ordered by each cutomer?

Step 1: get the total rolls order by customers

select customer\_id,roll\_id,count(roll\_id) rollid\_counts from customer\_orders

group by customer\_id,roll\_id

A screenshot of a computer

Description automatically generated

Step 2: inner join rolls table

select a.\*, b.roll\_name from

(select customer\_id,roll\_id,count(roll\_id) rollid\_counts from customer\_orders

group by customer\_id,roll\_id)a inner join rolls b on a.roll\_id = b.roll\_id

A screenshot of a computer

Description automatically generated

6. what was the maximum no of rolls delivered in a single order?

Step: 1 remove all the cancelled order( code taken form question 4 step 1)

select order\_id from (select \* , case when cancellation in ('cancellation','Customer Cancellation')

then 'c' else 'nc' end as order\_cancel\_details from driver\_order)a where order\_cancel\_details = 'nc'

A screenshot of a cell phone

Description automatically generated

Step 2: details of all succesful delivery

select \* from customer\_orders where order\_id in (select order\_id from (select \* , case when cancellation in ('cancellation','Customer Cancellation')

then 'c' else 'nc' end as order\_cancel\_details from driver\_order)a where order\_cancel\_details = 'nc')

A screenshot of a computer

Description automatically generated

Step 3:

select order\_id,count(roll\_id) as rolls\_count from (select \* from customer\_orders where order\_id in

(select order\_id from (select \* , case when cancellation in ('cancellation','Customer Cancellation')

then 'c' else 'nc' end as order\_cancel\_details from driver\_order)a where order\_cancel\_details = 'nc'))b

group by order\_id

A screenshot of a computer

Description automatically generated

Step 4: find the max rolls count

select \* ,rank() over(order by rolls\_count desc) rnk from (select order\_id,count(roll\_id) as rolls\_count from (select \* from customer\_orders where order\_id in

(select order\_id from (select \* , case when cancellation in ('cancellation','Customer Cancellation')

then 'c' else 'nc' end as order\_cancel\_details from driver\_order)a where order\_cancel\_details = 'nc'))b

group by order\_id)c

A screenshot of a table

Description automatically generated

Final solution:

select \* from

(select \* ,rank() over(order by rolls\_count desc) rnk from (select order\_id,count(roll\_id) as rolls\_count from (select \* from customer\_orders where order\_id in

(select order\_id from (select \* , case when cancellation in ('cancellation','Customer Cancellation')

then 'c' else 'nc' end as order\_cancel\_details from driver\_order)a where order\_cancel\_details = 'nc'))b

group by order\_id)c)d where rnk =1



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

Step 1: clean the data ( remove null and vacant spaces)

Creating temporary table:

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='NaN' or extra\_items\_included=' ' then '0' else extra\_items\_included

end as new\_extra\_items\_included, order\_date from customer\_orders

)

select \* from temp\_customer\_orders

A screenshot of a number list

Description automatically generated

Now removing the cancellation and noncancellation from drivers orders:

with 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 \* from temp\_driver\_order

A screenshot of a data table

Description automatically generated

Now remove cancelled:

select \* from temp\_driver\_order where new\_cancellation !=0

A screenshot of a data

Description automatically generated

Final solution: join tempory customer and deliver

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='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 \* from temp\_customer\_orders where order\_id in (

select order\_id from temp\_driver\_order where new\_cancellation !=0)

A screenshot of a number list

Description automatically generated

Now checking for changes in order i.e any item included or removed:

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='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,

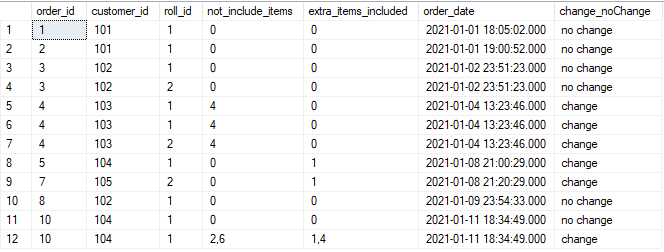
case when cancellation in ('cancellation','Customer Cancellation') then 0 else 1 end as new\_cancellation

from driver\_order

)

select \*, case when not\_include\_items = '0' and extra\_items\_included = '0' then 'no change' else 'change' end as change\_noChange

from temp\_customer\_orders where order\_id in (select order\_id from temp\_driver\_order where new\_cancellation !=0)



Final solution:

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='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 customer\_id,change\_nochange,count(order\_id) as at\_least\_1\_change from

(

select \*, case when not\_include\_items = '0' and extra\_items\_included = '0' then 'no change' else 'change' end as change\_noChange

from temp\_customer\_orders where order\_id in (select order\_id from temp\_driver\_order where new\_cancellation !=0))a

group by customer\_id,change\_noChange

A screenshot of a computer

Description automatically generated

8. how many rolls were deliverd 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\_include\_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 \*, case when not\_include\_items != '0' and extra\_items\_included != '0' then 'both included extras' else 'either 1 included or extra'

end as change\_noChange

from temp\_customer\_orders where order\_id in (select order\_id from temp\_driver\_order where new\_cancellation !=0)

A white paper with numbers and a number on it

Description automatically generated

Final solution:

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='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 change\_noChange,count(change\_noChange) as cnt from

(select \*, case when not\_include\_items != '0' and extra\_items\_included != '0' then 'both included extras' else 'either 1 included or extra'

end as change\_noChange

from temp\_customer\_orders where order\_id in (select order\_id from temp\_driver\_order where new\_cancellation !=0))a

group by change\_noChange

A close-up of a message

Description automatically generated

9. what is the total number of rolls ordered for each hour of the day?

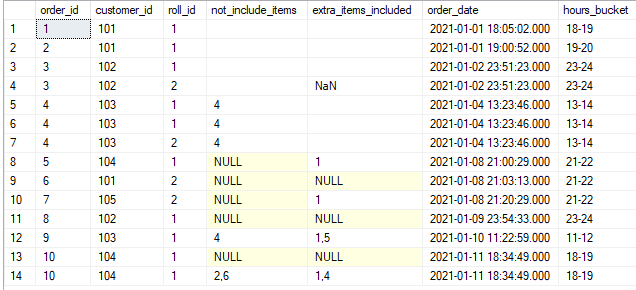
Step:1 select \*,datepart(hour,order\_date) as hr from customer\_orders

A screenshot of a table

Description automatically generated

Step 2:

select \*,concat(cast(datepart(hour,order\_date) as varchar),'-',cast(datepart(hour,order\_date)+1 as varchar)) hours\_bucket from customer\_orders



Final solution:

select hours\_bucket,count(hours\_bucket) as cnt 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

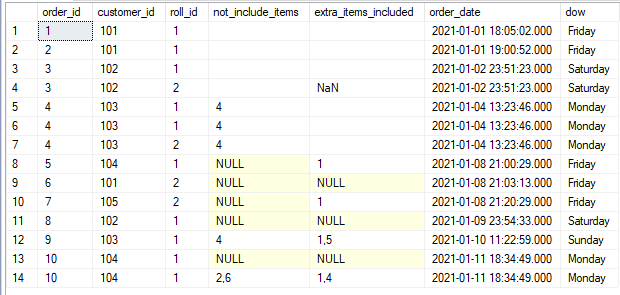
A screenshot of a computer

Description automatically generated

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

Step 1:

select \* ,datename(dw, order\_date) dow from customer\_orders

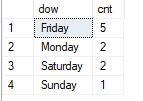


Step 2:

select dow,count(distinct order\_id) cnt from

(select \* ,datename(dw, order\_date) dow from customer\_orders)a

group by dow



B. driver and customer experience:

1. what is the average time in minutes it took for each driver to arrive at the fasoos hq to pickup the order?

select driver\_id,sum(Time\_taken\_in\_minutes)/count(order\_id) from

(select \* from

(select \*,row\_number()over(partition by order\_id order by Time\_taken\_in\_minutes)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.duration,b.cancellation,datediff(minute,a.order\_date,b.pickup\_time) Time\_taken\_in\_minutes

from customer\_orders a inner 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

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

select order\_id,count(roll\_id) as counts,sum(Time\_taken\_in\_minutes)/count(roll\_id) tyme 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.duration,b.cancellation,datediff(minute,a.order\_date,b.pickup\_time) Time\_taken\_in\_minutes

from customer\_orders a inner join driver\_order b on a.order\_id=b.order\_id

where b.pickup\_time is not null)a

group by order\_id

A screenshot of a computer

Description automatically generated

3. what was the average distance travel for each customer?

A screenshot of a computer

Description automatically generated

Note: since in distance column, it consist of km so we need to replace km to do sum other wise it Is a varchar so we need to convert to flaot by using cast. And in row 8 there is a space between 23.4 and km so we use trim to remove this space

select customer\_id,sum(distance)/count(order\_id) as average\_distance from

(select \* from

(select \*,row\_number()over(partition by order\_id order by Time\_taken\_in\_minutes)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) Time\_taken\_in\_minutes

from customer\_orders a inner 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

A screenshot of a data

Description automatically generated

4. what is the difference between the longest and shortes delivery time for all orders?

Step 1:

select duration from driver\_order where duration is not null

A screenshot of a computer

Description automatically generated

Here we are seeing different kind of minutes so we need to replace it using charindex

select duration,charindex('m',duration) from driver\_order where duration is not null

A screenshot of a computer

Description automatically generated

So with the help of charindex we identify the position from where minute or min or minutes is starting the we will use left

left(duration,charindex('m',duration,-1)) else duration

this code will select all the values before ‘m’ i.e 32,27 etc and if ‘m; is not available it will return the value present in the cell.

select duration, case when duration like '%min%' then left(duration,charindex('m',duration)-1) else

duration end as duration

from driver\_order where duration is not null

A screenshot of a computer

Description automatically generated

This is still varchar we need to convert to decimal or integer

(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)

now duration column is converted to integer datatype:

final solution:

select max(duration) as max\_durations,min(duration) as min\_durations,(max(duration)-min(duration)) as 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)f

A white rectangular object with black text

Description automatically generated

5. what was the average speed for each driver for each delivery and do you notice any trend for these values?

Step:1

select order\_id,count(roll\_id) from customer\_orders group by order\_id

A screenshot of a computer

Description automatically generated

Step 2:

select a.order\_id, a.driver\_id,cast(a.distance/a.duration as decimal(10,3)) as 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

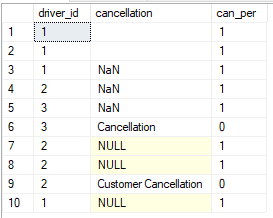
(select order\_id,count(roll\_id) as cnt from customer\_orders group by order\_id)b on a.order\_id = b.order\_id

A screenshot of a data

Description automatically generated

6. what is the succesful delivery percentage for each driver?

select driver\_id ,cancellation,case when lower(cancellation) like '%cancel%' then 0 else 1 end can\_per from driver\_order



select driver\_id,s,t,s/t as cancelled\_percentage from

(select driver\_id,sum(can\_per) s, count(driver\_id) t from

(select driver\_id ,cancellation,case when lower(cancellation) like '%cancel%' then 0 else 1 end can\_per from driver\_order)a

group by driver\_id)b

A screenshot of a calendar

Description automatically generated

So it is an incorrect because there are some cancelled delivery for driver 2 but cancelled percentage is 0, it is because we are not converting it to decimal. So it is giving whole number.

select driver\_id,s,t,s\*1.0/t as cancelled\_percentage from

(select driver\_id,sum(can\_per) s, count(driver\_id) t from

(select driver\_id ,cancellation,case when lower(cancellation) like '%cancel%' then 0 else 1 end can\_per from driver\_order)a

group by driver\_id)b

A screenshot of a computer

Description automatically generated

Now converting to percentage:

Final result:

select driver\_id,s,t,(s\*1.0/t)\*100 as cancelled\_percentage from

(select driver\_id,sum(can\_per) s, count(driver\_id) t from

(select driver\_id ,cancellation,case when lower(cancellation) like '%cancel%' then 0 else 1 end can\_per from driver\_order)a

group by driver\_id)b

A screenshot of a computer

Description automatically generated