```
create database fasoo;
use fasoo;
drop table driver
CREATE TABLE if not exists driver(driver_id integer,reg_date varchar(30));
INSERT INTO driver(driver_id,reg_date)
VALUES (1,'01-01-2021'),
(2,'01-03-2021'),
(3,'01-08-2021'),
(4,'01-15-2021');
drop table if exists ingredients;
CREATE TABLE ingredients(ingredients_id integer,ingredients_name varchar(60));
INSERT INTO ingredients(ingredients_id ,ingredients_name)
VALUES (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');
drop table if exists rolls;
CREATE TABLE rolls(roll_id integer,roll_name varchar(30));
INSERT INTO rolls(roll_id ,roll_name)
VALUES (1
               ,'Non Veg Roll'),
(2
       ,'Veg Roll');
drop table if exists rolls_recipes;
CREATE TABLE rolls_recipes(roll_id integer,ingredients varchar(24));
INSERT INTO rolls_recipes(roll_id ,ingredients)
VALUES (1,"1,2,3,4,5,6,8,10"),
(2,"4,6,7,9,11,12");
drop table if exists driver_order1;
CREATE TABLE driver_order1(order_id integer,driver_id integer,pickup_time varchar(30),distance
VARCHAR(7), duration VARCHAR(10), cancellation VARCHAR(23));
INSERT INTO driver_order1(order_id,driver_id,pickup_time,distance,duration,cancellation)
VALUES(1,1,'01-01-2021 18:15:34','20km','32 minutes',"),
(2,1,'01-01-2021 19:10:54','20km','27 minutes',''),
(3,1,'01-03-2021 00:12:37','13.4km','20 mins','NaN'),
(4,2,'01-04-2021 13:53:03','23.4','40','NaN'),
(5,3,'01-08-2021 21:10:57','10','15','NaN'),
(6,3,null,null,'Cancellation'),
(7,2,'01-08-2021 21:30:45','25km','25mins',null),
(8,2,'01-10-2021 00:15:02','23.4 km','15 minute',null),
```

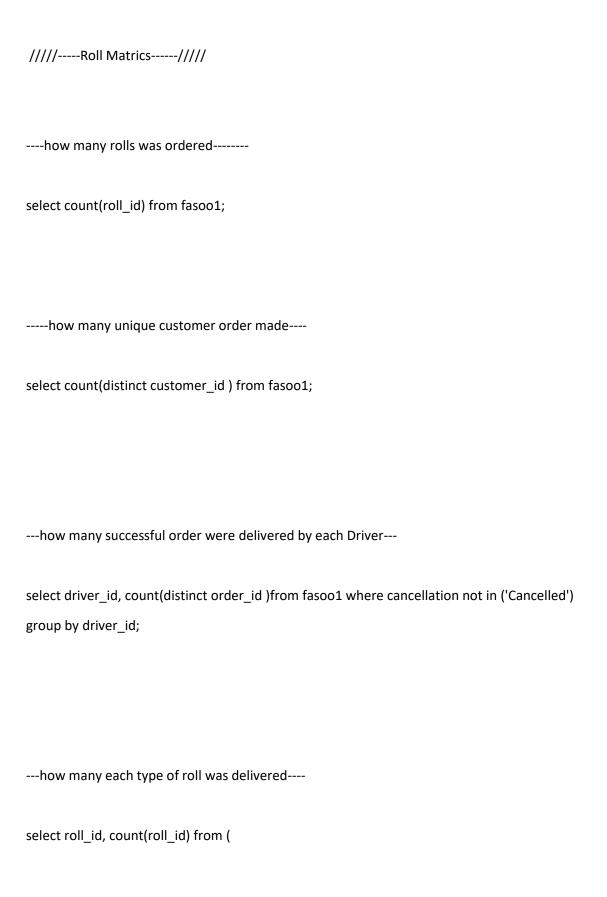
```
(9,2,null,null,'Customer Cancellation'),
(10,1,'01-11-2021 18:50:20','10km','10minutes',null);
drop table if exists customer_orders;
CREATE TABLE customer_orders(order_id integer,customer_id integer,roll_id
integer,not_include_items VARCHAR(4),extra_items_included VARCHAR(4), order_date varchar(30));
INSERT INTO
customer_orders(order_id,customer_id,roll_id,not_include_items,extra_items_included,order_date)
values (1,101,1,",",'01-01-2021 18:05:02'),
(2,101,1,",",'01-01-2021 19:00:52'),
(3,102,1,",",'01-02-2021 23:51:23'),
(3,102,2,",'NaN','01-02-2021 23:51:23'),
(4,103,1,'4',",'01-04-2021 13:23:46'),
(4,103,1,'4',",'01-04-2021 13:23:46'),
(4,103,2,'4',",'01-04-2021 13:23:46'),
(5,104,1,null,'1','01-08-2021 21:00:29'),
(6,101,2,null,null,'01-08-2021 21:03:13'),
(7,105,2,null,'1','01-08-2021 21:20:29'),
(8,102,1,null,null,'01-09-2021 23:54:33'),
(9,103,1,'4','1,5','01-10-2021 11:22:59'),
(10,104,1,null,null,'01-11-2021 18:34:49'),
(10,104,1,'2,6','1,4','01-11-2021 18:34:49');
select * from customer_orders;
select * from driver_order;
select * from ingredients;
select * from driver;
select * from rolls;
```

```
select * from rolls_recipes;
desc
      customer_orders;
----//converted varch to date format----//
update
 driver
set
 reg_date = STR_TO_DATE(reg_date, '%m-%d-%Y');
 ////-----/////****////
 ----creating a Master table ----not able to join all-----
 create table if not exists Fasoo as
 select c.order_id, c.customer_id,c.roll_id,c.not_include_items,c.extra_items_included,c.order_date,
 d.driver\_id, d.pickup\_time, d.distance, d.duration, d.cancellation,\\
 e.reg_date,
 r.ingredients,
 s.roll_name,
 i.ingredients_name
from customer_orders c
```

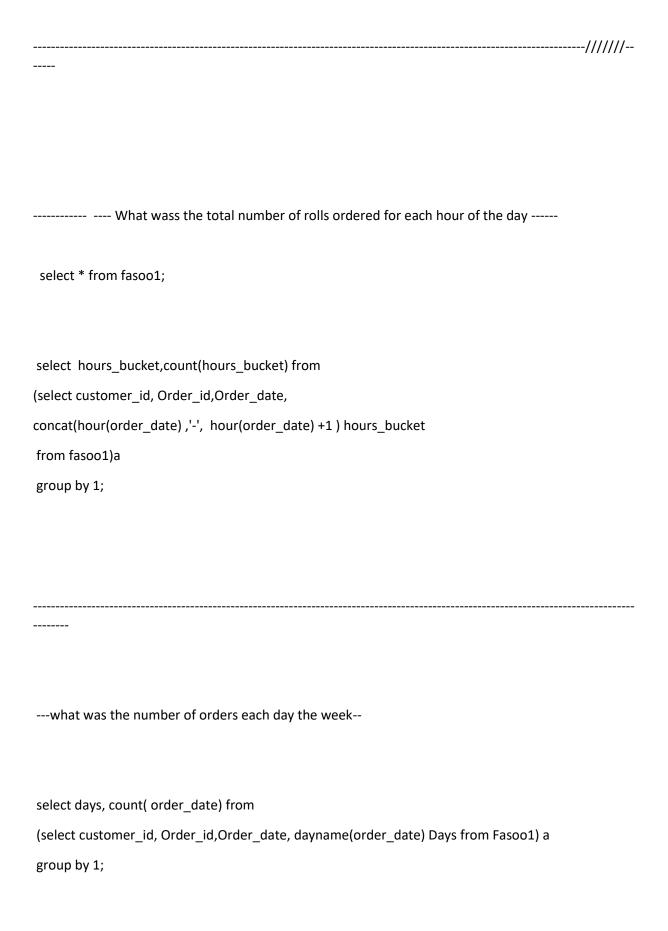
```
left outer join driver_order d on c.order_id = d.order_id
left outer join driver e on d.driver_id = e.driver_id
left outer join rolls_recipes r on c.roll_id = r.roll_id
left outer join rolls s on c.roll_id = s.roll_id
left outer join ingredients i on r.ingredients = i.ingredients_id
------
(creating extra one for testing some codes)
 create table if not exists Fasoo1 as
 select c.customer_id, c.order_id,c.order_date,d.pickup_time, c.roll_id, s.roll_name, r.ingredients,
c.not_include_items,c.extra_items_included,
d.distance,d.duration,d.cancellation,
 d.driver_id, e.reg_date
from customer_orders c
left outer join driver_order d on c.order_id = d.order_id
left outer join driver e on d.driver_id = e.driver_id
left outer join rolls_recipes r on c.roll_id = r.roll_id
left outer join rolls s on c.roll_id = s.roll_id;
```

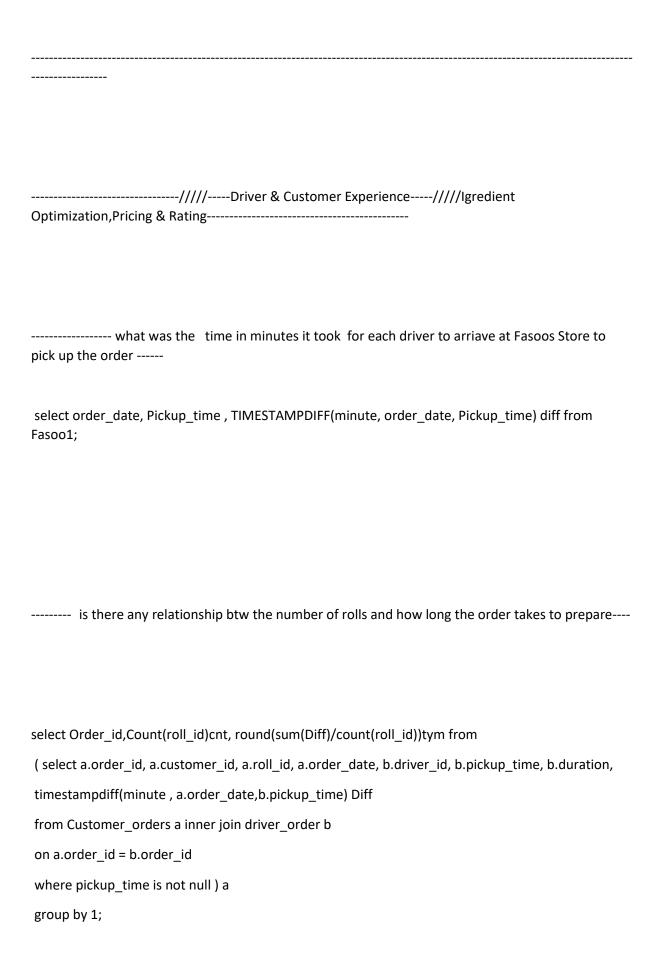
---- handling the null values in cancellation column---

```
UPDATE fasoo1
SET Cancellation =
 CASE
   WHEN Cancellation IN ('Cancellation', 'Customer Cancellation') then 'Cancelled'
   ELSE 'Delivered'
 END;
select * from fasoo1
-----
-working on blank data---
UPDATE fasoo1
SET not_include_items =
 CASE
  WHEN not_include_items IS NULL OR not_include_items = " THEN '0'
  ELSE not_include_items
 END,
 extra_items_included =
 CASE
  WHEN extra_items_included IS NULL OR extra_items_included = "
  OR extra_items_included = 'NaN' OR extra_items_included = 'NULL' THEN '0'
  ELSE extra_items_included
 END;
///////-----
```



```
select * from fasoo1 where cancellation = 'Delivered') a
group by 1;
---how many veg and non veg rolls were order by each customer---
       select *from fasoo1;
select customer_id, roll_id, roll_name, count(roll_id) cnt from fasoo1
group by 1,2,3;
---for each customer, how many delivered rolls had at least 1 change how many had no change-----
select customer_id, chg_no_chg, count(chg_no_chg) from
select *, case when not_include_items = '0' and extra_items_included = '0'then 'No Changes' else
'Change' end chg_no_chg from
(
select * from fasoo1 where cancellation = 'Delivered'
) a )b
group by 1,2;
```





/// 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 ) rnk from
( sELECT Customer_id, order_id, Driver_id, trim(replace((distance), 'km', '')) distance
FROM Fasoo1
where distance is not null )a )b
where rnk = 1)c
group by 1
/// what was the diference btw the longest & shortest delivery times for all orders////
select duration, locate("m", duration) from fasoo1;

select Max(duration) - min(duration) diff from
( Select
case when duration like '%min%' then
left(duration,locate('m', duration) -1) else
duration end as duration
from driver_order
where duration is not null ) a;
Not getting entire data ,so case when is used
Select duration,
left(duration,locate('m', duration) -1) as duration
from driver_order
where duration is not null;
//////
// What was the Avg speed of each driver for each delivery and do u noticed any trend for these values
Speed = d/t

```
Select distinct order_id, Driver_id, Day_Name, Timings,round(distance/duration, 2.1) Avg_speed
from
( Select order_id,Driver_id,dayname(order_date)Day_Name, hour(order_date)Timings,
trim(replace((distance), 'km', ")) distance,
case when duration like '%min%' then
             left(duration,locate('m', duration) -1) else
   duration end as duration
from Fasoo1
where distance is not null )a
Sucessful deliivery % for each driver--
SDP = total Sucessful Delivery/total orders taken
Select driver_id, round((c/t*1-1)*100, 2.2) Cancel_Per from
(select Driver_id, sum(can_Per) c, count(driver_id) t from
(Select driver_id,
case when lower(cancellation) like '%Cancelled%' then 0 else 1 end as can_per
from Fasoo1)a
group by 1)b;
```