FASSO'S SALES ANALYSIS

SQL PROJECT

BY:RAMAN PANDEY

CONTENTS

- About the Project
- Purpose
- About Data
- Analysis List
- Approach Used
- Report
- Conclusion

ABOUT

This project aims to explore the Fasso's sales data to understand top performing food items, ordering trends of customers and sales pattern of various items. This aims to study how sales strategy can be improved and optimized.

PURPOSE

The major aim of this project is to gain insights into the sales data of Fasso's to analyze the following:-

- Roll Metrics

- Driver and Customer Experience

- Ingredient Optimization

- Pricing and Ratings

With the help of insights the management will take data driven decisions which will help in growing the business.

▶ ABOUT DATA

Creation of Database and insertion of Data

```
drop table if exists driver;
CREATE TABLE driver(driver_id integer,reg_date date);

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 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 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');
```

Creation of Database and insertion of Data

```sql
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');
```

```sql
drop table if exists driver_order;
CREATE TABLE driver_order(order_id integer,driver_id integer,pickup_time datetime,distance VARCHAR(7),duration VARCHAR(10),cancellation VARCHAR(23));
INSERT INTO driver_order(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,null,'Cancellation'),
(7,2,'01-08-2020 21:30:45','25km','25mins',null),
(8,2,'01-10-2020 00:15:02','23.4 km','15 minute',null),
(9,2,null,null,null,'Customer Cancellation'),
(10,1,'01-11-2020 18:50:20','10km','10minutes',null);
```

Creation of Database and insertion of Data

```sql
drop table if exists customer_orders;
CREATE TABLE customer_orders(order_id integer,customer_id integer,roll_id integer,
not_included_items VARCHAR(4),extra_items_included VARCHAR(4),order_date datetime);
INSERT INTO customer_orders(order_id,customer_id,roll_id,not_included_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');
```

ANALYSIS LIST

► ITEM ANALYSIS – Conducted analysis on the data to understand the different item lines, which item lines is performing best and the item lines that need to be improved.

► ORDER ANALYSIS - This analysis aims to answer the question of the ordering trends of items. The result of this can help use measure the effectiveness of each sales strategy the business applies and what modifications are needed to gain more sales.

- CUSTOMER ANALYSIS – This analysis aims to uncover the different customers segments, purchase trends and the profitability of each customer segment.

APPROACH USED

1. Data Wrangling and Feature Engineering:-

Through these the inappropriate data values were handled by either modifying them directly or adding new columns so as to query the database in a manner which does not create failure of our queries to extract meaningful insights.
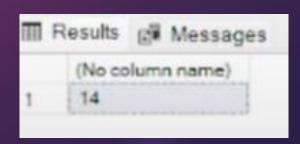
2. EXPLORATORY DATA ANALYSIS:- EDA is done to find the answer of the listed questions and aim of the project

1.HOW MANY ROLLS WERE ORDERED?

REPORT

SQL QUERY

```
select count(roll_id) from customer_orders;
```
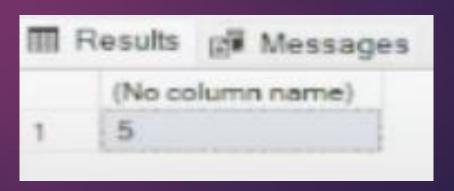
RESULT

	Results	Messages
	(No column name)	
1	14	

2.HOW MANY UNIQUE CUSTOMERS MADE ORDERS?

REPORT

SQL QUERY

```
select count(distinct customer_id) from customer_orders;
```

RESULT

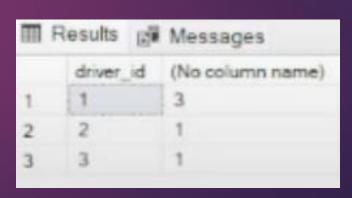|	Results	Messages
	(No column name)	
1	5	

3.HOW MANY SUCCESSFUL ORDERS WERE DELIVERED BY EACH DRIVER?

REPORT

SQL QUERY

```
select driver_id ,count(distinct order_id) from driver_order where cancellation not in ('Cancellation','Customer Cancellation')
group by driver_id;
```
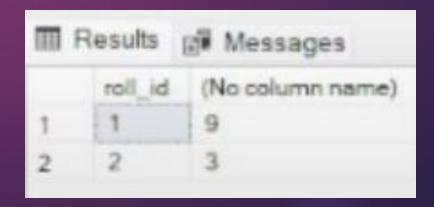
RESULT

	driver_id	(No column name)
1	1	3
2	2	1
3	3	1

4.HOW MANY OF EACH TYPE OF ROLL WAS DELIVERED?

REPORT

SQL
QUERY

```sql
select roll_id,count(roll_id) 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;
```

RESULT

	roll_id	(No column name)
1	1	9
2	2	3

5.HOW MANY VEG AND NON VEG ROLLS WERE ORDERED BY EACH CUSTOMER?

REPORT

SQL QUERY

```
select a.*,b.roll_name from
(
select customer_id,roll_id,count(roll_id) cnt
from customer_orders
group by customer_id,roll_id)a inner join rolls b on a.roll_id=b.roll_id;
```
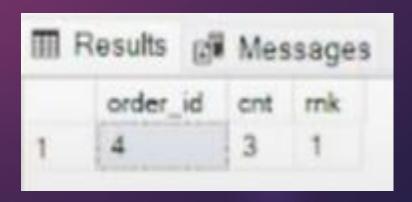
RESULT

	customer_id	roll_id	cnt	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?

REPORT

SQL
QUERY

```
select * from
(
select *,rank() over(order by cnt desc) rnk from
(
select order_id,count(roll_id) cnt
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;
```

RESULT

	order_id	cnt	rnk
1	4	3	1

7. FOR EACH CUSTOMER HOW MANY ROLLS HAVE AT LEAST 1 CHANGE AND HOW MANY HAD NONE?

REPORT

SQL QUERY

```sql
with temp_customer_orders(order_id, customer_id, roll_id, not_included_items,extra_items_included,
order_date)
as
select order_id, customer_id, roll_id,
case when not_included_items is null or not_included
_items=' '
then '0'
else not_included_items end as new_not_include_items,
case when extra_items_included is null or extra_items_included='0'
or extra_items_included='NaN' or extra_items_included='NULL' then '0' else 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 else 1
end as new_cancellation from driver order)

select customer_id, chg_no_chg, count (order_id) at_least_1_change from
(
select *, case when not_included_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;
```
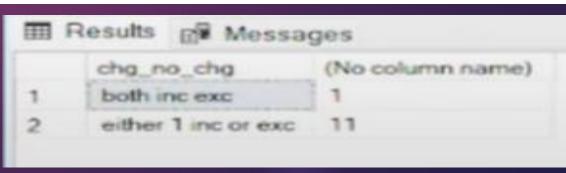
RESULT

	customer_id	chg_no_chg	at_least_1_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 THAT WERE DELIVERED HAD BOTH EXCLUSIONS AND EXTRAS?

REPORT

SQL QUERY

```sql
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 'e' 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'
or extra_items_included='NULL' then '0' else extra_items 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 else 1 end as new_cancellation
from driver order
)

select chg_no_chg, count (chg_no_chg) from
(select case when not_include_items!-'e' 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;
```

RESULT

	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 IN EACH HOUR OF THE DAY?

REPORT

SQL QUERY

```
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;
```
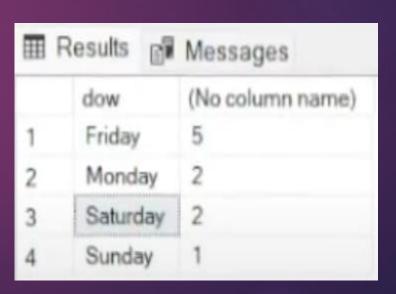
RESULT

	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 ON EACH DAY OF THE WEEK?

REPORT

SQL QUERY

```
select dow,count(distinct order_id) from
(select *,datename(dw,order_date) dow from customer_orders)a
group by dow;
```

RESULT

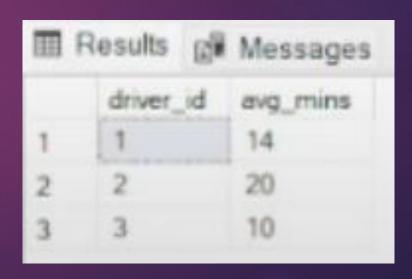|	dow	(No column name)
1	Friday	5
2	Monday	2
3	Saturday	2
4	Sunday	1

11.WHAT WAS THE AVERAGE TIME IT TOOK FOR EACH OF THE DRIVERS TO REACH THE FASSO'S HQ?

REPORT

SQL QUERY

```
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 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;
```

RESULT

	driver_id	avg_mins
1	1	14
2	2	20
3	3	10

12.IS THERE ANY RELATIONSHIP BETWEEN THE NUMBER OF ROLLS AND TIME TOOK TO PREPARE THEM?

REPORT

SQL QUERY

```
select order_id,count(roll_id) cnt,sum(diff)/count(roll_id) tym 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 inner join driver_order b on a.order_id=b.order_id
where b.pickup_time is not null)a
group by order_id;
```
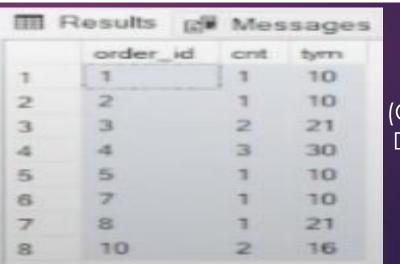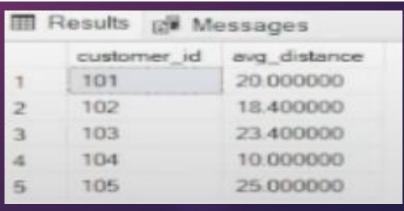
RESULT

	order_id	cnt	tym
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

(CLEARLY THERE IS A DIRECT RELATIONSHIP)

13.WHAT WAS THE AVERAGE DISTANCE TRAVELLED FOR EACH CUSTOMER?

REPORT

SQL QUERY

```sql
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 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;
```
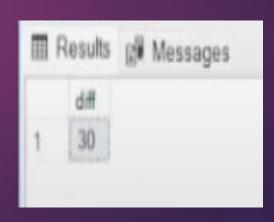
RESULT

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

Results   Messages

14.WHAT IS THE DIFFERENCE BETWEEN HIGHEST AND LOWEST DELIVERY TIME?

REPORT

SQL QUERY

```sql
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;
```

RESULT

	diff
1	30

15.WHAT IS THE AVERAGE SPEED OF DRIVERS FOR EACH DELIVERY AND IS THERE ANY TREND PRESENT?

REPORT

SQL QUERY

```
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;
```

RESULT

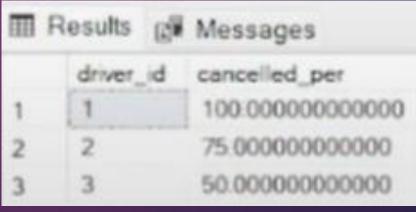|	order_id	driver_id	cnt	speed
1	1	1	1	0.6250000000000
2	2	1	1	0.74074074074074
3	3	1	2	0.6700000000000
4	4	2	3	0.5850000000000
5	5	3	1	0.66666666666666
6	7	2	1	1.0000000000000
7	8	2	1	1.5600000000000
8	10	1	2	1.0000000000000

(Avg speed decreases as No. of rolls Increases)

16.WHAT IS THE SUCCESSFUL DELIVERY PERCENTAGE FOR EACH DRIVER?

REPORT

SQL QUERY

```
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 else 1 end as 0 can_per from driver_order) a group by driver_id)b;
```

RESULT

	driver_id	cancelled_per
1	1	100.000000000000
2	2	75.000000000000
3	3	50.000000000000

CONCLUSIONS-

- TOTAL ROLLS ORDERED=14

- NON VEG ROLL IS PREFERABLE

- LATE AFTERNOON TO EARLY EVENING AND LATE NIGHTS ARE THE PREFERRED TIMINGS

- ON FRIDAY HIGHEST NUMBER OF ORDERS WERE MADE

- AVERAGE TIME REQUIRED TO REACH FASOO'S HQ IS APPROXIMATELY 14 MINUTES

- MAXIMUM ROLLS WERE ORDERED WITH EITHER INGREDIENT INCLUSIONS OR EXCLUSIONSBUT NOT BOTH