TARGET CASE STUDY

Q.1 (a) Data types of the columns in a table.

A.1 (a) Going through all the tables schema in the given dataset under the schema column, these are the following datatypes present:

- String
- Integer
- Float
- Time-Stamp

Output: **Below is the schema of orderitems table which contains all the data types present in all other tables.**

Filter Enter property name or value

Field name	Type	Mode
order_id	STRING	NULLABLE
order_item_id	INTEGER	NULLABLE
product_id	STRING	NULLABLE
seller_id	STRING	NULLABLE
shipping_limit_date	TIMESTAMP	NULLABLE
price	FLOAT	NULLABLE
freight_value	FLOAT	NULLABLE

Q.1 (b) Time Period for which the data is given.

A.1 (b) Query :

```
SELECT  distinct extract (year from order_purchase_timestamp) as yearr
        ,min(order_purchase_timestamp) as first_purchase
        ,max(order_purchase_timestamp) as last_purchase
FROM `casestudy111.Target_database.orders`
group by 1
order by yearr;
```

Output:

Row	yearr	first_purchase	last_purchase
1	2016	2016-09-04 21:15:19 UTC	2016-12-23 23:16:47 UTC
2	2017	2017-01-05 11:56:06 UTC	2017-12-31 23:29:31 UTC
3	2018	2018-01-01 02:48:41 UTC	2018-10-17 17:30:18 UTC

Insights : The time period of the data spreads from 4th September 2016 to 17th October 2018.

Q.1 (c) Cities and States of the customers that ordered during the given period.

A.1 (c) Query:

```
select distinct customer_city,customer_state,extract (year from ord.order_purchase_timestamp) as yearr
from `Target_database.customers_table` cus
join `Target_database.orders` ord
on cus.customer_id=ord.customer_id
order by year
```

Output:

Row	customer_city	customer_state	yearr
1	itu	SP	2016
2	poa	SP	2016
3	bage	RS	2016
4	iacu	BA	2016
5	leme	SP	2016
6	lins	SP	2016
7	maua	SP	2016
8	cambe	PR	2016
9	cotia	SP	2016
10	macae	RJ	2016

Results per page: 50 1 – 50 of 6974

Q.2 (a) Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

A.2(a) Query:

```
SELECT  extract(year from order_purchase_timestamp) as year,extract (month from order_purchase_timestamp) as month,count(order_id) as number_of_orders

FROM `casestudy111.Target_database.orders`
group by 1,2
order by year,month;
```

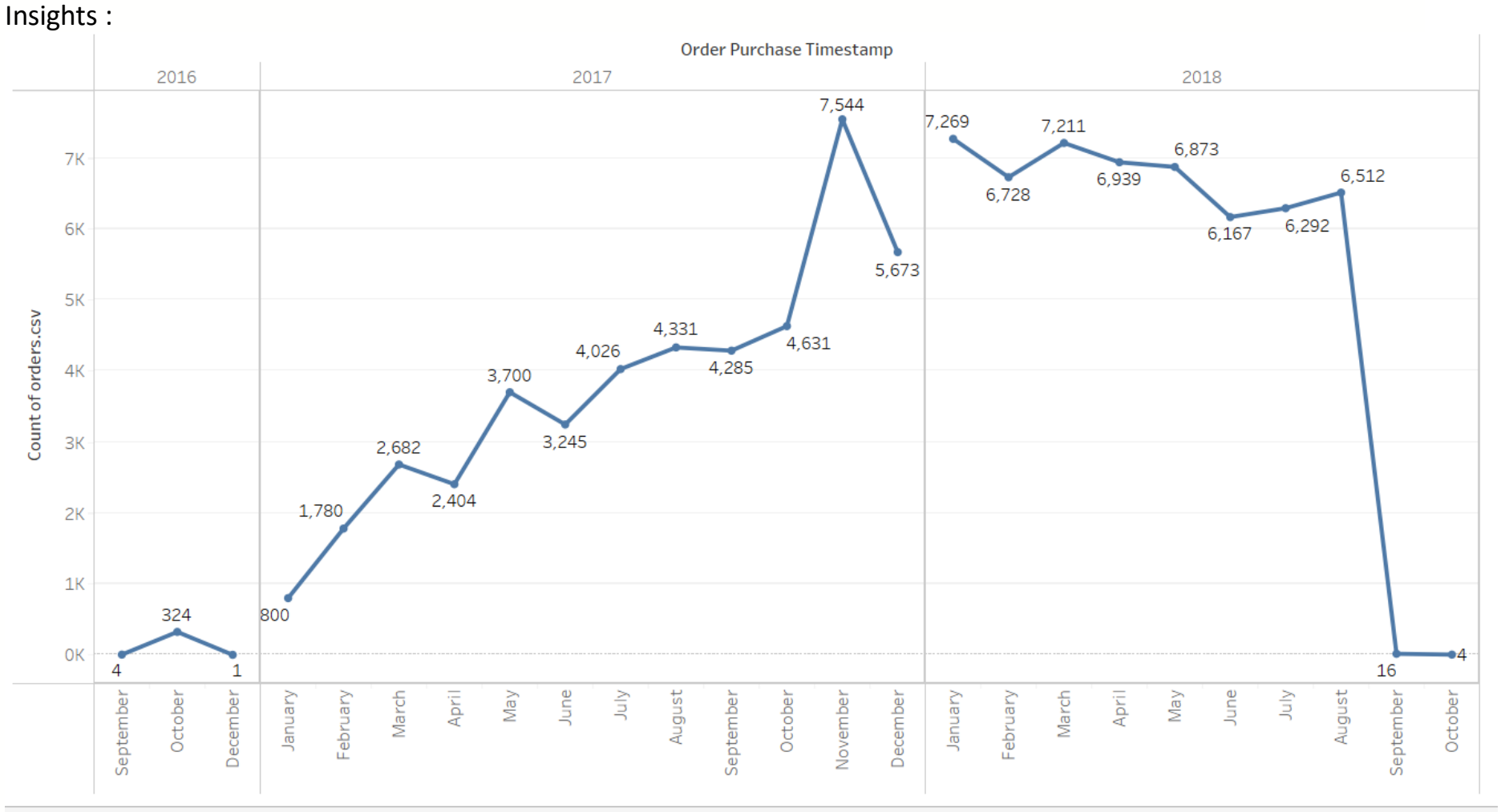
Output:

JOB INFORMATION RESULTS JSON EXECUTION

Row	year	month	number_of_orde
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331

Results per page: 50 1 – 25 of 25

Insights :



- **From the above output as well as the graph a couple of things can be understood, firstly that we see a massive change in the number of orders from the year 2016 to 2017 in Brazil mainly because in 2016 Brazil suffered one of its worst recession hits and the GDP growth was -2.1%, hence we can see the number of orders in the last quarter of 2016 being almost negligible.**
- **Secondly in the year 2017 there was labour market reform which strengthened incentives for job creation and hence due to more salaried individuals the retail shopping growth increased in Brazil which can be seen by the number of orders in 2017.**

Q.2(b) What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

A.2(b) Query:

```
with order_time as (

select extract(HOUR from order_purchase_timestamp) as hour,
        count(*) as orde

from `Target_database.orders`
group by 1
)
select case
        when hour>=0 and hour<=6 then 'Dawn'
        when hour>6 and hour<=12 then 'Afternoon'
        when hour>12 and hour<=18 then 'Evening'
        when hour>18 and hour<=23 then 'Night'
        end as time_of_day,
        sum(orde) as total_orders

from order_time
group by 1
order by total_orders desc
```

Output:

JOB INFORMATION | RESULTS | JSON

Row	time_of_day	total_orders
1	Evening	38135
2	Night	28331
3	Afternoon	27733
4	Dawn	5242

Q.3 (a) Get month on month orders by states.

A.3 (a) Query:

```
select extract(YEAR from order_purchase_timestamp) as year
,extract(MONTH from order_purchase_timestamp) as month
,count(*) as num_orders
,cus.customer_state

from `Target_database.orders` ord
 join `Target_database.customers_table` cus
on ord.customer_id=cus.customer_id
group by 1,2,4
order by 1,2;
```

Output:

Row	year	month	num_orders	customer_state
1	2016	9	1	RR
2	2016	9	1	RS
3	2016	9	2	SP
4	2016	10	113	SP
5	2016	10	24	RS
6	2016	10	56	RJ
7	2016	10	3	MT
8	2016	10	9	GO
9	2016	10	40	MG
10	2016	10	8	CE

Q.3 (b) Distribution of customers across the states in Brazil.

A.3 (b) Query:

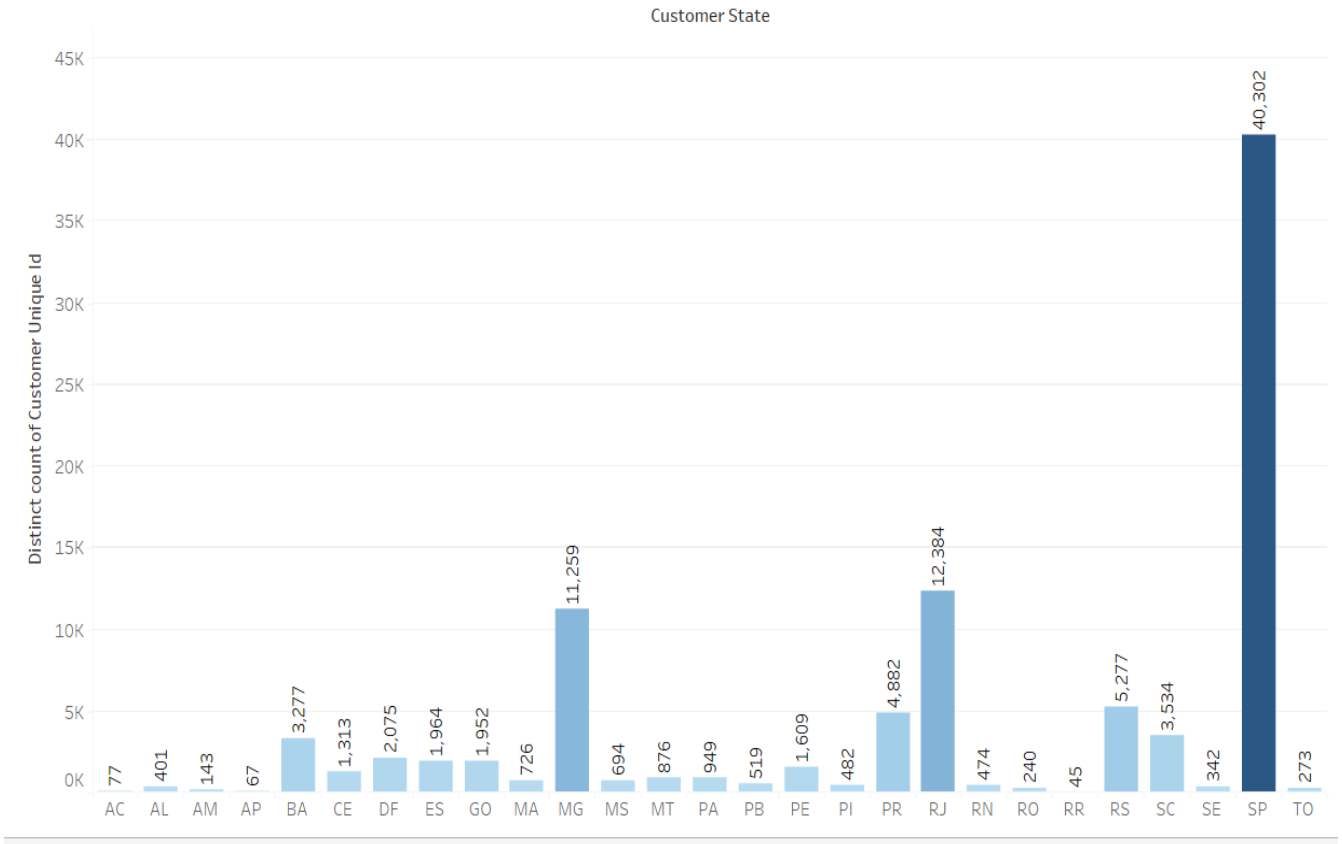
```
select customer_state
,count(distinct customer_unique_id) as num_cust
 from `Target_database.customers_table`
group by 1
order by 2 desc
```

Output:

Row	customer_state	num_cust
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952

Results per page: 50 1 – 27 of 27

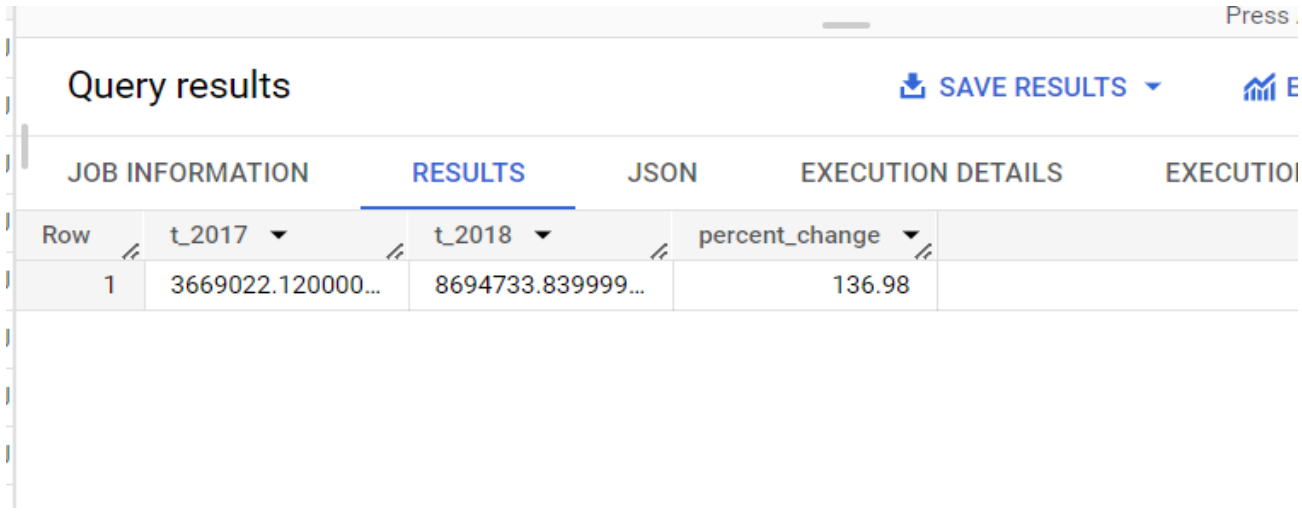
Insights : The State Sao Paulo being the most populated state in Brazil drives in the most number of customers for Target throughout the country.



Q.4 (a) Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only)
A.4 (a) Query:

```
        with t1 as(


select extract(YEAR from order_purchase_timestamp) as year
     ,sum(p.payment_value) as t_2017
from `Target_database.orders` o
join `Target_database.payments` p
on o.order_id=p.order_id
            where extract(YEAR from order_purchase_timestamp) in (2017) and extract(MONTH from order_purchase_timestamp) not in (9,10,11,12)
group by 1
order by 1
        ),

        t2 as(

select extract(YEAR from order_purchase_timestamp) as year
     ,sum(p.payment_value) as t_2018
from `Target_database.orders` o
join `Target_database.payments` p
on o.order_id=p.order_id
        where extract(YEAR from order_purchase_timestamp) in (2018) and extract(MONTH from order_purchase_timestamp) not in (9,10,11,12)
group by 1
order by 1
        )

select t1.t_2017,t2.t_2018,
((t2.t_2018-t1.t_2017)/(t1.t_2017))*100 as per_change
from t1,t2;
```

Output:



Q.4 (b) Mean & Sum of price and freight value by customer state.
A.4 (b) Query:

```
select c.customer_state

        ,round(sum(freight_value),2) as freight_sum
        ,round(sum(price),2) as price_sum
        ,round(avg(freight_value),2) as freight_avg
        ,round(avg(price),2) as price_avg

from `Target_database.orderitems`o
join `Target_database.orders` orde
on o.order_id=orde.order_id
join `Target_database.customers_table` c
on orde.customer_id=c.customer_id
group by 1
order by 1
```

Output:

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAI

Row	customer_state	freight_sum	price_sum	freight_avg	price_avg
1	AC	3686.75	15982.95	40.07	173.73
2	AL	15914.59	80314.81	35.84	180.89
3	AM	5478.89	22356.84	33.21	135.5
4	AP	2788.5	13474.3	34.01	164.32
5	BA	100156.68	511349.99	26.36	134.6
6	CE	48351.59	227254.71	32.71	153.76
7	DF	50625.5	302603.94	21.04	125.77
8	ES	49764.6	275037.31	22.06	121.91
9	GO	53114.98	294591.95	22.77	126.27
10	MA	31523.77	119648.22	38.26	145.2
11	MG	270853.46	1585308.03	20.63	120.75

Results per page: 50 1 – 27 of 27

Q.5(a) Calculate days between purchasing, delivering and estimated delivery.
A.5(a) Query:

```
    select order_id

,date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY) AS actual_delivery
,date_diff(order_estimated_delivery_date,order_purchase_timestamp,DAY) AS estimated_delivery


    from `Target_database.orders`
where order_status='delivered';
```

Output:

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECU

Row	order_id	actual_delivery	estimated_delivery
1	635c894d068ac37e6e03dc54e...	30	32
2	3b97562c3aee8bdedcb5c2e45...	32	33
3	68f47f50f04c4cb6774570cfde...	29	31
4	276e9ec344d3bf029ff83a161c...	43	39
5	54e1a3c2b97fb0809da548a59...	40	36
6	fd04fa4105ee8045f6a0139ca5...	37	35
7	302bb8109d097a9fc6e9cefc5...	33	28
8	66057d37308e787052a32828...	38	32
9	19135c945c554eebfd7576c73...	36	33
10	4493e45e7ca1084efcd38ddeb...	34	33
11	70c77e51e0f179d75a64a6141...	42	31

Results per page: 50 1 – 50 of 96478

REFRESH

Q.5 (b) Find time_to_delivery & diff_estimated_delivery.

A.5 (b) Query:

```
select order_id
,date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY) AS time_to_delivery
,date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY) AS diff_estimated_delivery

from `Target_database.orders` ;
```

Output:

Query results

SAVE RESULTS | EX

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAI

Row	order_id	time_to_delivery	diff_estimated_c
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5

Results per page: 50 1 – 50 of 99441

Q.5(c) Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

- Top 5 states with highest average freight value - sort in desc limit 5

Query:

```
select c.customer_state
        ,round(avg(freight_value),2) as average_freight
        ,round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY)),2) AS time_to_delivery
        ,round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY)),2) AS      diff_estimated_delivery


from `Target_database.orders` o
join `Target_database.orderitems` orde
on o.order_id=orde.order_id
join `Target_database.customers_table` c
on o.customer_id=c.customer_id
group by 1
order by 2 desc
limit 5
```

Output:

Query results

JOB INFORMATION | RESULTS | JSON | EX

Row	customer_state	average_freight
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

- Top 5 states with highest average time to delivery.

Query:

```
select c.customer_state
        ,round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY)),2) AS time_to_delivery
        ,round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY)),2) AS diff_estimated_delivery
         ,round(avg(freight_value),2) as average_freight

from `Target_database.orders` o
join `Target_database.orderitems` orde
on o.order_id=orde.order_id
join `Target_database.customers_table` c
on o.customer_id=c.customer_id
group by 1
order by 2 desc
```

Output:

JOB INFORMATION | RESULTS | JSON | EXE

Row	customer_state	time_to_delivery
1	RR	27.83
2	AP	27.75
3	AM	25.96
4	AL	23.99
5	PA	23.3
6	MA	21.2
7	SE	20.98
8	CE	20.54
9	AC	20.33
10	PB	20.12
11	RO	19.28

- Top 5 states where delivery is not so fast compared to estimated date.

Query:

```
select c.customer_state
,round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY)),2) AS diff_estimated_delivery
,round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY)),2) AS time_to_delivery
,round(avg(freight_value),2) as average_freight

from `Target_database.orders` o
join `Target_database.orderitems` orde
on o.order_id=orde.order_id
join `Target_database.customers_table` c
on o.customer_id=c.customer_id
group by 1
order by 2 ASC
```

Output:

Row	customer_state	diff_estimated_delive
1	AL	7.98
2	MA	9.11
3	SE	9.17
4	ES	9.77
5	BA	10.12
6	CE	10.26
7	SP	10.27
8	MS	10.34
9	SC	10.67
10	PI	10.68

Q.6 (a) Month over Month count of orders for different payment types.

A.6 (a) Query:

```
select extract(YEAR from order_purchase_timestamp) as year
,extract(MONTH from order_purchase_timestamp) as month
,count(*) as num_orders
,p.payment_type

from `Target_database.orders` o
join `Target_database.payments` p
on o.order_id=p.order_id
group by 1,2,4
order by 1,2;
```

Output:

Row	year	month	num_orders	payment_type
1	2016	9	3	credit_card
2	2016	10	254	credit_card
3	2016	10	63	UPI
4	2016	10	23	voucher
5	2016	10	2	debit_card
6	2016	12	1	credit_card
7	2017	1	583	credit_card
8	2017	1	197	UPI
9	2017	1	61	voucher
10	2017	1	9	debit_card
11	2017	2	1356	credit_card

Results per page: 50 1 – 50 of 90

Insights: From the output and the graph we can see the payment type for the various order made from September 2016 to October 2018 we can see how credit card is the most used payment method and due to growth in technology UPI being the second most used payment type.

Q.6 (b) Count of orders based on the no. of payment installments.
A.6 (b) Query:

```
    select extract(YEAR from order_purchase_timestamp) as year
,extract(MONTH from order_purchase_timestamp) as month
,count(*) as num_orders
,p.payment_installments

    from `Target_database.orders` o
    join `Target_database.payments` p
    on o.order_id=p.order_id
    group by 1,2,4
    order by 1,2,4;
```

Output :

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	year	month	num_orders	payment_installments
1	2016	9	1	1
2	2016	9	1	2
3	2016	9	1	3
4	2016	10	144	1
5	2016	10	30	2
6	2016	10	43	3
7	2016	10	26	4
8	2016	10	20	5
9	2016	10	18	6
10	2016	10	13	7
11	2016	10	3	8

Results per page: 50 1 – 50 of 324

INSIGHTS

After performing in depth Exploratory Data Analysis on the Target_Brazil dataset these are the multiple insights that I have gathered:

- The data involves customers and their multiple orders made by them at the Target retail store over the years 2016-2018.

- 2016 being the year where Brazil was hit with severe recession, the GDP growth being -2.1%, the number of orders made throughout the last quarter was negligible.

- Moreover 2017 being the year where labour market reforms took place due to which incentives for job creations were strengthened and hence there is an immense increase in the retail market thus an immense growth can be seen from coming out of the recession period in the Target sales.

- Sao Paolo (SP) being the most populated state drives the maximum number of customers (40,302) where as SP was also the state where maximum of the orders was sold by the sellers (70,188)

- Post analysis the Product category with the maximum orders were: Bed Table Bath, Health Beauty & Sport Leisure.

- Product Category with minimum orders were: Insurance & Services, Fashion-Children's Clothing & PC gamers.

- Maximum sales revenue was generated by the Health & Beauty category.

- Apart from Credit Card there is quite an increase in the use of UPI as a payment type and almost every transaction made was done via installments.

RECOMMENDATIONS

- In order to increase revenue as well as increase number of orders the first step of action would be to boost the categories with least orders in the form of fancy marketing and ads. For e.g., Fashion Children's clothing is something that could be boosted very easily by making sure we are showcasing different clothes on the website as per seasonal change, every parent keeps their child first and would definitely go for that.

- Secondly Brazil being a massive footballing nation is always going to worship Football. The Sport Leisure category could definitely become the top revenue category of Target if we focus on good branding and good marketing/deals based on the football season.

- Lastly with growing technology we can see how UPI is becoming the go-to for every customer with a sudden increase in the payment type, how we can benefit from this is as via the data we can see the number of customers using the installments option making schemes and certain campaigns promoting No-cost E.M.I option and certain cashback option on the purchase of certain products via CC or UPI would definitely attract more customers as well.