Business Case: Target SQL

Submitted By:
Rahul Kumar Verma
+91-8709530619
rahulvrm938@gmail.com

INDEX

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

1.1. Data type of columns in a table.

```
SELECT table_name,column_name,data_type
FROM Target.INFORMATION_SCHEMA.COLUMNS
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW

Row	table_name	column_name	data_type
1	order_items	order_id	STRING
2	order_items	order_item_id	INT64
3	order_items	product_id	STRING
4	order_items	seller_id	STRING
5	order_items	shipping_limit_date	TIMESTAMP
6	order_items	price	FLOAT64
7	order_items	freight_value	FLOAT64
8	sellers	seller_id	STRING

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

1.2. Time period for which the data is given.

```
select min(order_purchase_timestamp) from `Target.orders`;
select max(order_purchase_timestamp) from `Target.orders`;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	f0_
1	2016-09-04 21:15:19 UTC

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	f0_
1	2018-10-17 17:30:18 UTC

The data set is given for the time period : Sep'2016 to Oct'2018.

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

1.3. Cities and States covered in the dataset.

```
WITH base AS
(SELECT c.customer_city AS city, c.customer_state AS state
FROM `Target.orders` o LEFT JOIN `Target.customers` c
ON c.customer_id=o.customer_id)


SELECT state,city FROM base group by 1,2 order by state;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	city
1	AC	rio branco
2	AC	brasileia
3	AC	manoel urbano
4	AC	cruzeiro do sul
5	AC	xapuri

2. In-depth Exploration:

2.1. 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?

```
SELECT order_year, COUNT(*) AS ORDER_COUNT
FROM
        (SELECT *,EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
        EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year
        FROM `Target.orders`)base
GROUP BY order_year
ORDER BY order_year;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	order_year	ORDER_CO...
1	2016	329
2	2017	45101
3	2018	54011

```
SELECT order_year,order_month, COUNT(*) AS ORDER_COUNT
FROM
        (SELECT *,EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
        EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year
        FROM `Target.orders`)base
GROUP BY 1,2
ORDER BY 1,2;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	order_year	order_month	ORDER_CO...
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

```
SELECT order_month, COUNT(*) AS ORDER_COUNT
FROM
        (SELECT *,EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
        EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year
        FROM `Target.orders`)base
GROUP BY 1
ORDER BY 1;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	order_month	ORDER_CO...
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959
11	11	7544
12	12	5674

```
SELECT *
FROM
        (SELECT order_year,order_month, COUNT(*) AS ORDER_COUNT
        FROM
                (SELECT *,EXTRACT(MONTH FROM order_purchase_timestamp) AS order_month,
                EXTRACT(YEAR FROM order_purchase_timestamp) AS order_year
                FROM `Target.orders`)base
        GROUP BY 1,2
        )base_2
ORDER BY ORDER_COUNT DESC
LIMIT 1
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	order_year	order_month	ORDER_CO...
1	2017	11	7544

Yes, there is growing trend in e-commerce in Brazil as year-over-year we can see increase in count of orders.

We can see the trends of getting more orders during May, August and November month as Mother's Day is celebrated in Brazil on the second Sunday of May. Brazilians celebrate their dads on the second Sunday of August. In November, sales are high due to Chinese Double 11 and Black Friday.

2. In-depth Exploration:

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

```
SELECT TIME_DURATION,SUM(order_count) as Total_order
FROM
        (SELECT order_count, (CASE
        WHEN order_hour BETWEEN 0 AND 5 THEN 'DAWN'
        WHEN order_hour BETWEEN 6 AND 11 THEN 'MORNING'
        WHEN order_hour BETWEEN 12 AND 17 THEN 'AFTERNOON'
        WHEN order_hour BETWEEN 18 AND 23 THEN 'NIGHT'
                END) AS TIME_DURATION
                FROM
                        (SELECT order_hour, COUNT(*) AS order_count
                        FROM
                                (SELECT *,EXTRACT(TIME FROM order_purchase_timestamp) AS order_t
                                ime,
                                EXTRACT(HOUR FROM order_purchase_timestamp) AS order_hour
                                FROM `Target.orders`)base
                        GROUP BY 1
                        ORDER BY 1)base_2
        )base_3
GROUP BY 1
ORDER BY Total_order DESC;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	TIME_DURATION	Total_order
1	AFTERNOON	38361
2	NIGHT	34100
3	MORNING	22240
4	DAWN	4740

I have defined Time Interval as mentioned below:

12:00:01 am - 05:59:59 am : DAWN

06:00:01 am - 11:59:59 am : MORNING

12:00:01 pm - 05:59:59 pm : AFTERNOON

06:00:01 pm - 11:59:59 pm : NIGHT

Mostly, Brazilian customers tends to buy products during AFTERNOON and NIGHT.

3. Evolution of E-commerce orders in the Brazil region:

3.1. Get month on month orders by region, states

```
WITH base AS
(SELECT c.customer_city AS city, c.customer_state AS state, o.order_id as order_id,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year
FROM `Target.orders` o LEFT JOIN `Target.customers` c
ON c.customer_id=o.customer_id),

base_2 AS (SELECT state,order_year,order_month,COUNT(order_id) AS total_order FROM base group by 1,2,3)

SELECT DISTINCT * FROM base_2 order by state,order_year,order_month;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	order_year	order_month	total_order
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5

```
WITH base AS
(SELECT c.customer_city AS city, c.customer_state AS state, o.order_id as order_id,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year
FROM `Target.orders` o LEFT JOIN `Target.customers` c
ON c.customer_id=o.customer_id),

base_2 AS (SELECT city,order_year,order_month,COUNT(order_id) AS total_order FROM base group by 1,2,3)

SELECT DISTINCT * FROM base_2 order by city,order_year,order_month;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	city	order_year	order_month	total_order
1	abadia dos dourados	2017	9	1
2	abadia dos dourados	2018	3	1
3	abadia dos dourados	2018	7	1
4	abadiania	2018	1	1
5	abaete	2017	2	1
6	abaete	2017	5	1
7	abaete	2017	7	2
8	abaete	2017	8	1

```
WITH base AS
(SELECT c.customer_city AS city, c.customer_state AS state, o.order_id as order_id,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year
FROM `Target.orders` o LEFT JOIN `Target.customers` c
ON c.customer_id=o.customer_id),

base_2 AS (SELECT state,city,order_year,order_month,COUNT(order_id) AS total_order FROM base grou
p by 1,2,3,4)

SELECT DISTINCT * FROM base_2 order by state,city,order_year,order_month;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	city	order_year	order_month	total_order
1	AC	brasileia	2017	2	1
2	AC	cruzeiro do sul	2017	12	2
3	AC	cruzeiro do sul	2018	5	1
4	AC	epitaciolandia	2017	10	1
5	AC	manoel urbano	2017	9	1
6	AC	porto acre	2017	4	1
7	AC	rio branco	2017	1	2
8	AC	rio branco	2017	2	2

3. Evolution of E-commerce orders in the Brazil region:

3.2. How are customers distributed in Brazil

```
WITH base AS
(SELECT c.customer_city AS city, c.customer_state AS state, o.order_id as order_id,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year
FROM `Target.orders` o LEFT JOIN `Target.customers` c
ON c.customer_id=o.customer_id),

base_2 AS (SELECT state,COUNT(order_id) OVER(PARTITION BY state) AS total_order FROM base
order by state,total_order desc)

select state,total_order from base_2 group by 1,2 order by total_order desc;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	total_order
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140

```
WITH base AS
(SELECT c.customer_city AS city, c.customer_state AS state, o.order_id as order_id,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year
FROM `Target.orders` o LEFT JOIN `Target.customers` c
ON c.customer_id=o.customer_id),

base_2 AS (SELECT city,COUNT(order_id) OVER(PARTITION BY city) AS total_order FROM base
order by city,total_order desc)

select city,total_order from base_2 group by 1,2 order by total_order desc;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	city	total_order
1	sao paulo	15540
2	rio de janeiro	6882
3	belo horizonte	2773
4	brasilia	2131
5	curitiba	1521
6	campinas	1444
7	porto alegre	1379
8	salvador	1245

```
WITH base AS
(SELECT c.customer_city AS city, c.customer_state AS state, o.order_id as order_id,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year
FROM `Target.orders` o LEFT JOIN `Target.customers` c
ON c.customer_id=o.customer_id),

base_2 AS (SELECT state,city,COUNT(order_id) OVER(PARTITION BY state,city) AS total_order
FROM base order by city,total_order desc),

base_3 AS (select DISTINCT state,city,total_order,ROW_NUMBER() OVER(PARTITION BY state,city
ORDER BY total_order DESC) AS RANKER from base_2 order by 1,3 desc)

SELECT state,city,total_order FROM base_3 where RANKER=1;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	city	total_order
1	AC	rio branco	70
2	AC	cruzeiro do sul	3
3	AC	senador guiomard	2
4	AC	xapuri	2
5	AC	manoel urbano	1
6	AC	brasileia	1
7	AC	epitaciolandia	1
8	AC	porto acre	1

During the period from 2016 to 2018, we have got maximum orders from sao paulo (Belongs to state of Sao Paulo). This is because it is having highest GDP & Per Capital Income and also densely populated city in Brazil.

4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

4.1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only)

```
WITH base AS (SELECT oi.price AS cost,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year
FROM `Target.order_items` oi JOIN `Target.orders` o ON o.order_id=oi.order_id),

base_2 AS (SELECT order_year, ROUND(sum(cost),2) AS total_cost FROM base
WHERE order_month BETWEEN 1 AND 8 group by order_year order by order_year),

base_3 AS (SELECT order_year,total_cost,
LEAD(total_cost,1,0) OVER(order by order_year ASC) AS next_year_total_cost FROM base_2
order by order_year asc)

SELECT order_year,total_cost,next_year_total_cost,(CASE
WHEN percent_increase_cost>=0 THEN percent_increase_cost
WHEN percent_increase_cost<0 THEN 0
END) AS yoy_increase_cost
FROM
        (SELECT *,
        ROUND(((next_year_total_cost-total_cost)/(total_cost))*100,2) AS percent_increase_cost FROM base_3);
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	order_year	total_cost	next_year_total_cost	yoy_increase_cost
1	2017	3113000.32	7385905.8	137.26
2	2018	7385905.8	0.0	0.0

4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

4.2. Mean & Sum of price and freight value by customer state.

```
WITH base AS (SELECT oi.price AS price,
oi.freight_value AS freight_value,oi.price+oi.freight_value as total_price ,
c.customer_state AS state
FROM `Target.order_items` oi JOIN `Target.orders` o ON o.order_id=oi.order_id
JOIN `Target.customers` c on c.customer_id=o.customer_id)

SELECT state,ROUND(AVG(price),2) AS mean_price,
ROUND(AVG(freight_value),2) AS mean_freight_value,
ROUND(AVG(total_price),2) AS mean_total_price,
ROUND(SUM(price),2) AS sum_price,
ROUND(SUM(freight_value),2) AS sum_freight_value,
ROUND(SUM(total_price),2) AS sum_total_price
FROM base
GROUP BY 1 ORDER BY 1;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	mean_price	mean_freight_value	mean_total_price	sum_price	sum_freight_value	sum_total_price
1	AC	173.73	40.07	213.8	15982.95	3686.75	19669.7
2	AL	180.89	35.84	216.73	80314.81	15914.59	96229.4
3	AM	135.5	33.21	168.7	22356.84	5478.89	27835.73
4	AP	164.32	34.01	198.33	13474.3	2788.5	16262.8
5	BA	134.6	26.36	160.97	511349.99	100156.68	611506.67
6	CE	153.76	32.71	186.47	227254.71	48351.59	275606.3
7	DF	125.77	21.04	146.81	302603.94	50625.5	353229.44
8	ES	121.91	22.06	143.97	275037.31	49764.6	324801.91

5. Analysis on sales, freight and delivery time

5.1 Calculate days between purchasing, delivering and estimated delivery

5.2 Create columns:

time_to_delivery = order_purchase_timestamp-order_delivered_customer_date

diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

5.3 Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

5.4 Sort the data to get the following:

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

Top 5 states with highest/lowest average time to delivery

Top 5 states where delivery is really fast/ not so fast compared to estimated date

5.1 & 5.2 >

```
SELECT order_id,
DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp, DAY) AS time_to_delivery,
DATE_DIFF(order_delivered_customer_date,order_estimated_delivery_date, DAY) AS diff_estimated_delivery
FROM `Target.orders`;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	order_id	time_to_delivery	diff_estimated...
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

5.3>

```
WITH base AS (SELECT c.customer_state AS state,o.order_id AS order_id,oi.freight_value AS freight_value,
DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, DAY) AS time_to_delivery,
DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY) AS diff_estimated_delivery
FROM `Target.customers` c JOIN `Target.orders` o ON o.customer_id=c.customer_id
JOIN `Target.order_items` oi ON oi.order_id=o.order_id)
```

```
SELECT state,ROUND(AVG(freight_value),2) AS mean_freight_value, ROUND(AVG(time_to_delivery),2) A
S mean_time_to_delivery, ROUND(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM base GROUP BY 1 ORDER BY 1;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	AC	40.07	20.33	-20.01
2	AL	35.84	23.99	-7.98
3	AM	33.21	25.96	-18.98
4	AP	34.01	27.75	-17.44
5	BA	26.36	18.77	-10.12
6	CE	32.71	20.54	-10.26
7	DF	21.04	12.5	-11.27
8	ES	22.06	15.19	-9.77

5.4.1>

```
WITH base AS (SELECT c.customer_state AS state,o.order_id AS order_id,oi.freight_value AS freight_value,
DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, DAY) AS time_to_delivery,
DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY) AS diff_estimated_delivery,
FROM `Target.customers` c JOIN `Target.orders` o ON o.customer_id=c.customer_id
JOIN `Target.order_items` oi ON oi.order_id=o.order_id)

SELECT state,ROUND(AVG(freight_value),2) AS mean_freight_value, ROUND(AVG(time_to_delivery),2) A
S mean_time_to_delivery, ROUND(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM base GROUP BY 1 ORDER BY 2 ASC LIMIT 5;
```

Top 5 states with lowest average freight value

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	mean_freigh...	mean_time_...	mean_diff_e...
1	SP	15.15	8.26	-10.27
2	PR	20.53	11.48	-12.53
3	MG	20.63	11.52	-12.4
4	RJ	20.96	14.69	-11.14
5	DF	21.04	12.5	-11.27

```
WITH base AS (SELECT c.customer_state AS state,o.order_id AS order_id,oi.freight_value AS freight_value,
DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, DAY) AS time_to_delivery,
DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY) AS diff_estimated_delivery,
FROM `Target.customers` c JOIN `Target.orders` o ON o.customer_id=c.customer_id
JOIN `Target.order_items` oi ON oi.order_id=o.order_id)
```

```
SELECT state,ROUND(AVG(freight_value),2) AS mean_freight_value, ROUND(AVG(time_to_delivery),2) A
S mean_time_to_delivery, ROUND(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery F
ROM base GROUP BY 1 ORDER BY 2 DESC LIMIT 5;
```

Top 5 states with highest average freight value

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	mean_freigh...	mean_time_...	mean_diff_e...
1	RR	42.98	27.83	-17.43
2	PB	42.72	20.12	-12.15
3	RO	41.07	19.28	-19.08
4	AC	40.07	20.33	-20.01
5	PI	39.15	18.93	-10.68

5.4.2>

```
WITH base AS (SELECT c.customer_state AS state,o.order_id AS order_id,oi.freight_value AS freight_value,
DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, DAY) AS time_to_delivery,
DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY) AS diff_estimated_delivery,
FROM `Target.customers` c JOIN `Target.orders` o ON o.customer_id=c.customer_id
JOIN `Target.order_items` oi ON oi.order_id=o.order_id)

SELECT state,ROUND(AVG(freight_value),2) AS mean_freight_value, ROUND(AVG(time_to_delivery),2) A
S mean_time_to_delivery, ROUND(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM base GROUP BY 1 ORDER BY 3 ASC LIMIT 5;
```

Top 5 states with lowest average time to delivery

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	mean_freigh...	mean_time_...	mean_diff_e...
1	SP	15.15	8.26	-10.27
2	PR	20.53	11.48	-12.53
3	MG	20.63	11.52	-12.4
4	DF	21.04	12.5	-11.27
5	SC	21.47	14.52	-10.67

```
WITH base AS (SELECT c.customer_state AS state,o.order_id AS order_id,oi.freight_value AS freight_value,
DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, DAY) AS time_to_delivery,
DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY) AS diff_estimated_delivery,
FROM `Target.customers` c JOIN `Target.orders` o ON o.customer_id=c.customer_id
JOIN `Target.order_items` oi ON oi.order_id=o.order_id)
```

```
SELECT state,ROUND(AVG(freight_value),2) AS mean_freight_value, ROUND(AVG(time_to_delivery),2) A
S mean_time_to_delivery, ROUND(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM base GROUP BY 1 ORDER BY 3 DESC LIMIT 5;
```

Top 5 states with highest average time to delivery

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	mean_freigh...	mean_time_...	mean_diff_e...
1	RR	42.98	27.83	-17.43
2	AP	34.01	27.75	-17.44
3	AM	33.21	25.96	-18.98
4	AL	35.84	23.99	-7.98
5	PA	35.83	23.3	-13.37

5.4.3>

```
WITH base AS (SELECT c.customer_state AS state,o.order_id AS order_id,oi.freight_value AS freight_value,
DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, DAY) AS time_to_delivery,
DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY) AS diff_estimated_delivery,
FROM `Target.customers` c JOIN `Target.orders` o ON o.customer_id=c.customer_id
JOIN `Target.order_items` oi ON oi.order_id=o.order_id)

SELECT state,ROUND(AVG(freight_value),2) AS mean_freight_value, ROUND(AVG(time_to_delivery),2) A
S mean_time_to_delivery, ROUND(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM base GROUP BY 1 ORDER BY 4 ASC LIMIT 5;
```

Top 5 states where delivery is fast compared to estimated date

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	mean_freigh...	mean_time_...	mean_diff_e...
1	AC	40.07	20.33	-20.01
2	RO	41.07	19.28	-19.08
3	AM	33.21	25.96	-18.98
4	AP	34.01	27.75	-17.44
5	RR	42.98	27.83	-17.43

```
WITH base AS (SELECT c.customer_state AS state,o.order_id AS order_id,oi.freight_value AS freight_value,
DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp, DAY) AS time_to_delivery,
DATE_DIFF(o.order_delivered_customer_date,o.order_estimated_delivery_date, DAY) AS diff_estimated_delivery,
FROM `Target.customers` c JOIN `Target.orders` o ON o.customer_id=c.customer_id
JOIN `Target.order_items` oi ON oi.order_id=o.order_id)

SELECT state,ROUND(AVG(freight_value),2) AS mean_freight_value, ROUND(AVG(time_to_delivery),2) A
S mean_time_to_delivery, ROUND(AVG(diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM base GROUP BY 1 ORDER BY 4 DESC LIMIT 5;
```

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

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	state	mean_freigh...	mean_time_...	mean_diff_e...
1	AL	35.84	23.99	-7.98
2	MA	38.26	21.2	-9.11
3	SE	36.65	20.98	-9.17
4	ES	22.06	15.19	-9.77
5	BA	26.36	18.77	-10.12

6. Payment type analysis:

6.1. Month over Month count of orders for different payment types.

```
WITH base AS (SELECT p.payment_type AS payment_type,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month, o.order_id AS order_id
FROM `Target.orders` o JOIN `Target.payments` p ON p.order_id=o.order_id)

SELECT payment_type,order_year,order_month,COUNT(order_id)
FROM base GROUP BY 1,2,3 ORDER BY 1,2,3;
```

We have observed an increase in payment year over year for UPI, Credit Card and Debit Card. This is because of the improvement in Fintech/Banking and ease of access to the Internet.

6. Payment type analysis:

6.2. Distribution of payment installments and count of orders.

```
WITH base AS (SELECT p.payment_type AS payment_type,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month, o.order_id AS order_id,
p.payment_installments AS payment_installments
FROM `Target.orders` o JOIN `Target.payments` p ON p.order_id=o.order_id)

SELECT payment_installments,COUNT(order_id) FROM base GROUP BY 1 ORDER BY 1;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	payment_in...	f0_
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626

Results per page: 50 1 – 24 of 24

```
WITH base AS (SELECT p.payment_type AS payment_type,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month, o.order_id AS order_id,
p.payment_installments AS payment_installments
FROM `Target.orders` o JOIN `Target.payments` p ON p.order_id=o.order_id)

SELECT payment_installments,COUNT(order_id) AS count_order
FROM base GROUP BY 1 ORDER BY 2 desc;
```

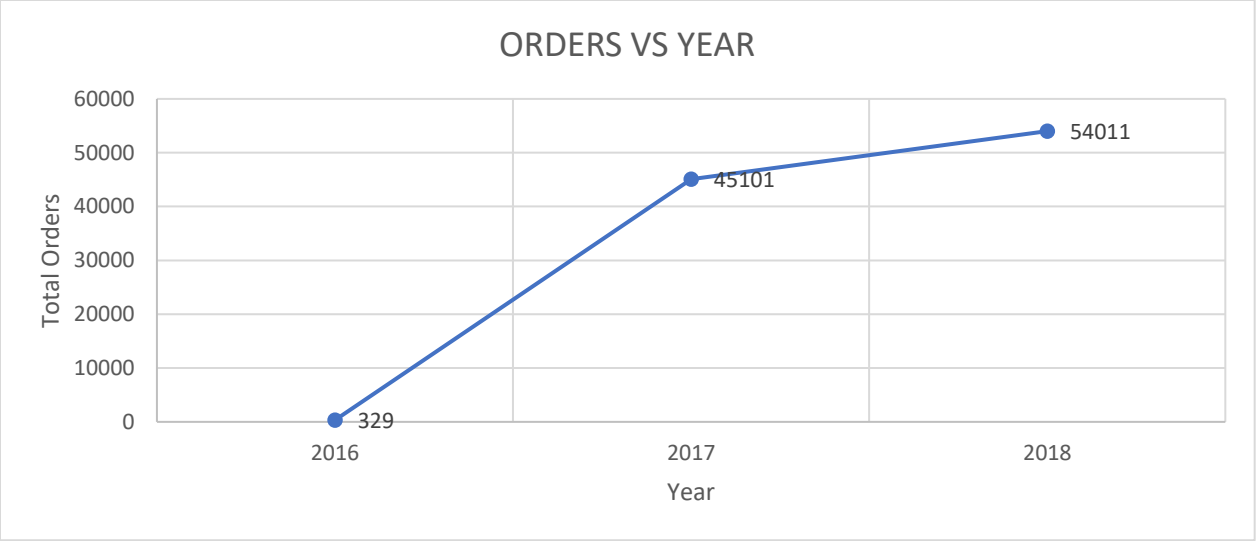
Query results

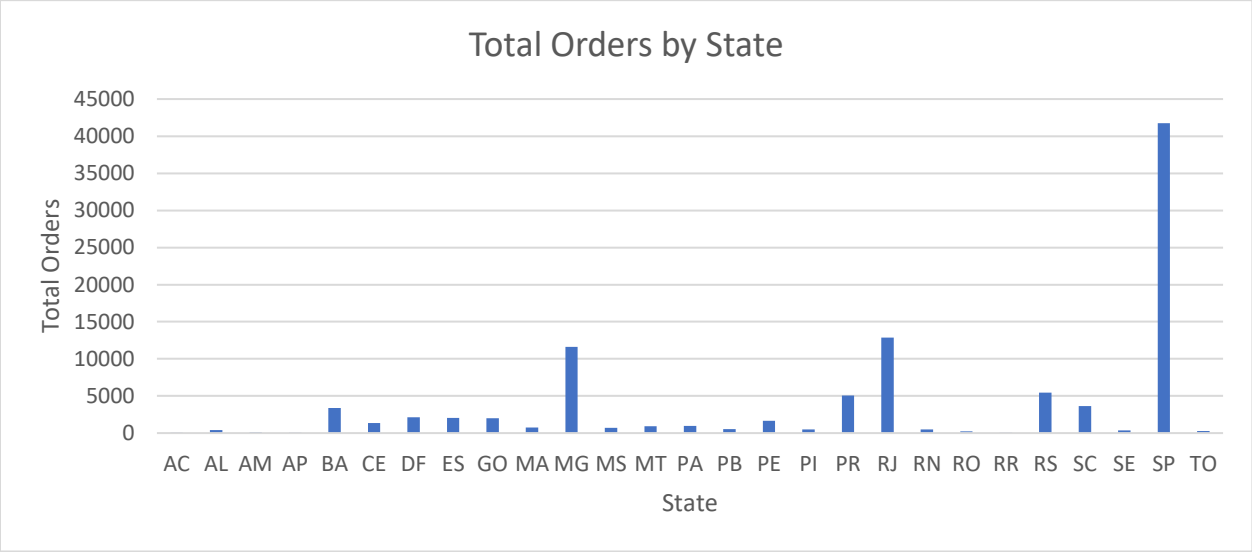
SAVE RESULTS

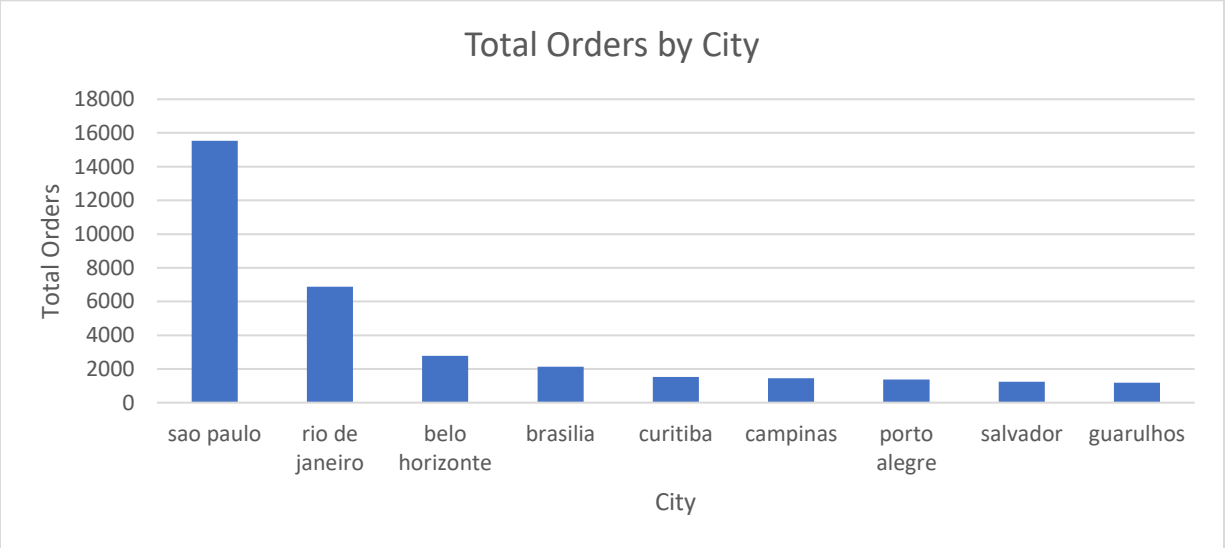
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

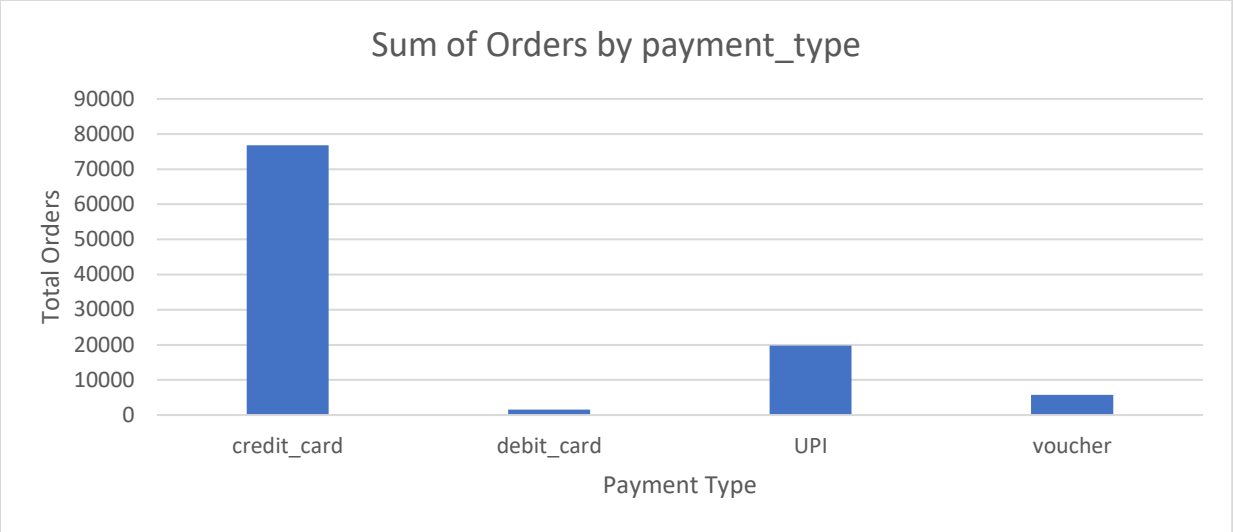
Row	payment_installments	count_order
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920

From the above table, we can conclude that customers generally opt for less no. of installment.

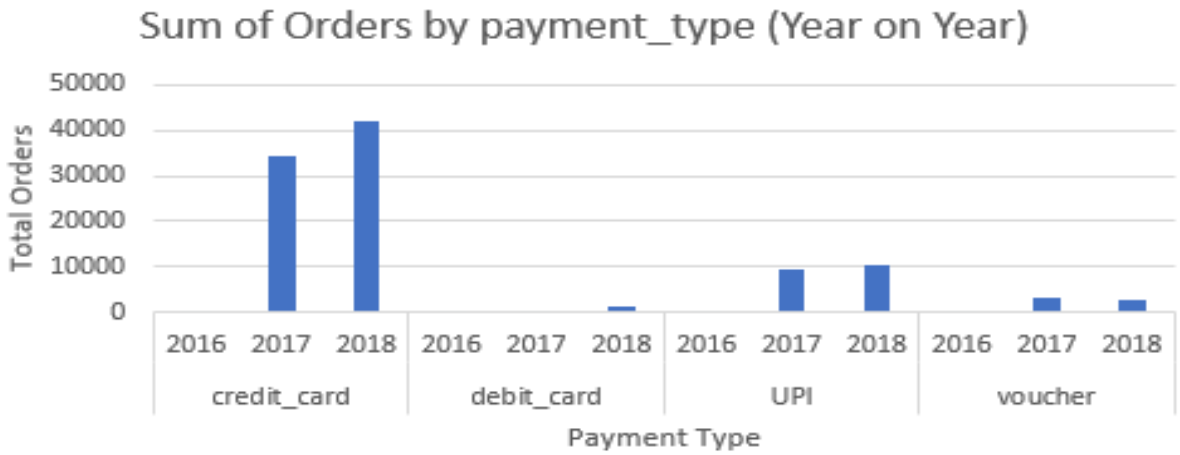

ORDERS VS YEAR
60000
50000
40000
30000
20000
10000
0
Total Orders
329
45101
54011
2016
2017
2018
Year


Total Orders by State
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
Total Orders
AC AL AM AP BA CE DF ES GO MA MG MS MT PA PB PE PI PR RJ RN RO RR RS SC SE SP TO
State

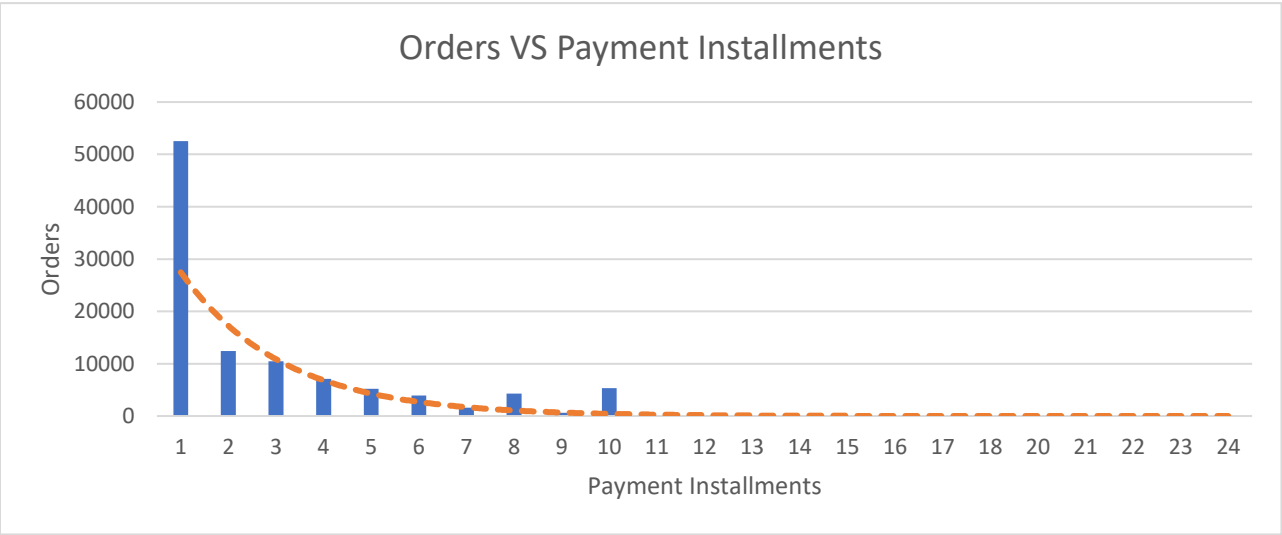

Total Orders by City
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Total Orders
sao paulo
rio de janeiro
belo horizonte
brasilia
curitiba
campinas
porto alegre
salvador
guarulhos
City



Sum of Orders



payment_type order_year



Insights:

- there is growing trend in e-commerce in Brazil as count of orders increased year-over-year.
- We can see the trends of getting more orders during May, August and November month as Mother's Day is celebrated in Brazil on the second Sunday of May. Brazilians celebrate their dads on the second Sunday of August. In November, sales are high due to Chinese Double 11 and Black Friday.
- During the period from 2016 to 2018, we have got maximum orders from sao paulo (Belongs to state of Sao Paulo). This is because it is having highest GDP & Per Capital Income and also densely populated city amongst Brazilian cities.
- We have observed an increase in payment year over year for UPI, Credit Card and Debit Card. This is because of the improvement in Fintech/Banking and ease of access to the Internet.
- States where delivery is not so fast as compared to estimated date are AL, MA, SE, ES & BA.

Recommendations:

- As we are having growing e-commerce market in Brazil, we need to increase product variety and no. of suppliers to meet the demand.
- As the demand is high in the months of May, August and November. Suppliers should be equipped with more stocks, and we can give discounts to attract more customers.
- As we are getting more orders from Sao Paulo, we can increase more sales by offering delivery within 2 days. For that we can increase warehouse/hub in that state.
- The cashless payment mode has had more influence. So, we roll out even our credit-card.
- The states where delivery is not so fast generally belongs to the Eastern Region. So, we need to set up hubs and need to find more suppliers for the Eastern Region.