SCALER

Target SQL - Business Case

Submitted by :

Harsha Srinivas, Tanna
harshasrinivas.tanna@gmail.com
Scaler DSML - Morning TTS Feb 2023
Submitted on April 5, 2023

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

Query :

```
SELECT column_name, data_type
FROM target-project-
bc-1.target_datasets.INFORMATION_SCHEMA.COLUMNS;
```

Query Result :

Query results

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

Row	column_name	data_type
1	order_id	STRING
2	order_item_id	INT64
3	product_id	STRING
4	seller_id	STRING
5	shipping_limit_date	TIMESTAMP
6	price	FLOAT64
7	freight_value	FLOAT64
8	seller_id	STRING
9	seller_zip_code_prefix	INT64
10	seller_city	STRING

1.2) Time period for which the data is given

Query :

```
SELECT MIN(order_purchase_timestamp) AS min_date,
MAX(order_purchase_timestamp) AS max_date
FROM `target-project-bc-1.target_datasets.orders`;
```

Query Result :

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETA

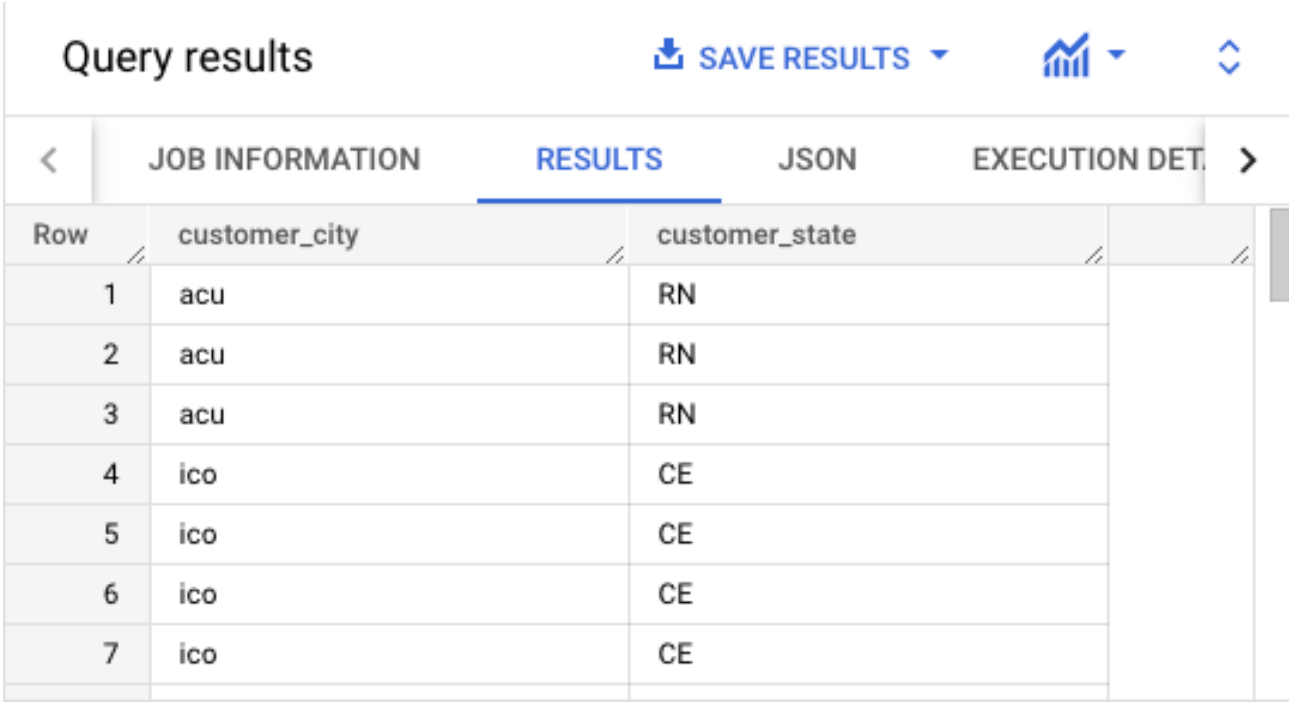
Row	min_date	max_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

1.3) Cities and States of customers ordered during the given period

Query :

```
SELECT customer_city, customer_state
FROM `target-project-bc-1.target_datasets.customers`;
```

Query Result :



Row	customer_city	customer_state
1	acu	RN
2	acu	RN
3	acu	RN
4	ico	CE
5	ico	CE
6	ico	CE
7	ico	CE

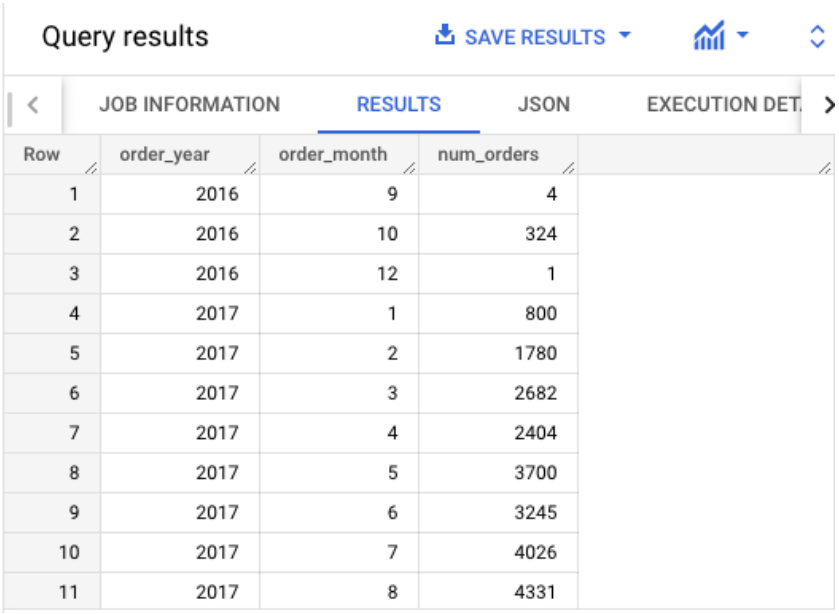
2)In-depth Exploration:

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?

Query :

```
SELECT
EXTRACT(YEAR from order_purchase_timestamp) AS order_year,
EXTRACT(MONTH from order_purchase_timestamp) AS order_month,
COUNT(*) AS num_orders
FROM `target_datasets.orders`
GROUP BY order_year,order_month
ORDER BY order_year,order_month;
```

Query Result :


Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DET.

Row	order_year	order_month	num_orders
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

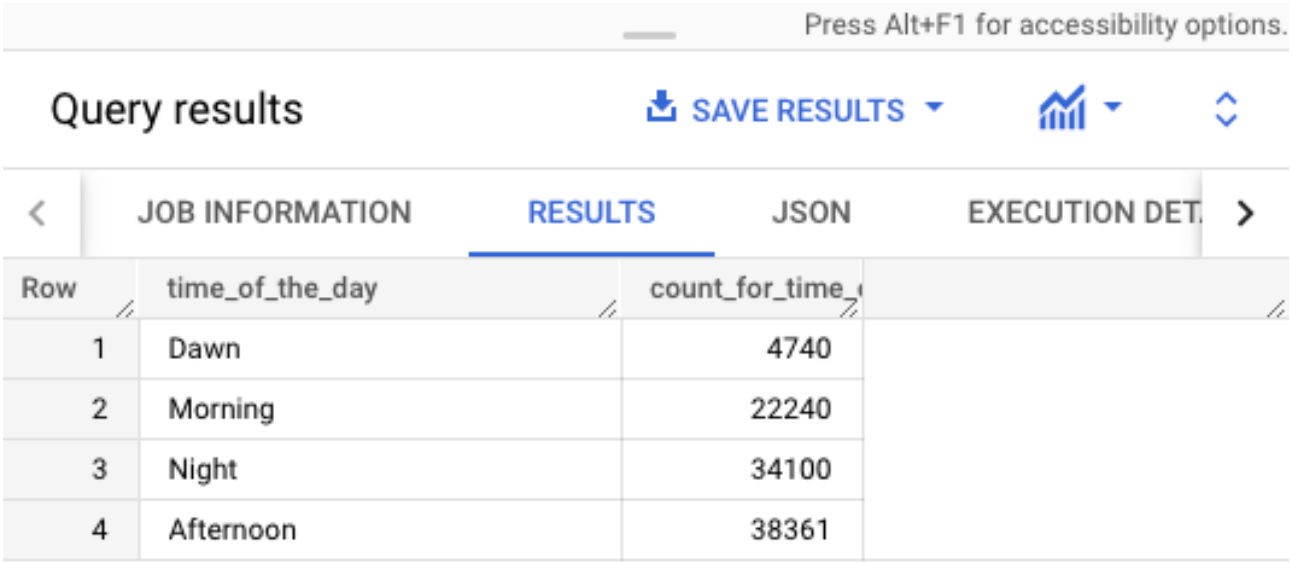
Conclusion : Yes, There is a growing trend on e-commerce in Brazil as it is evident from the Query Result the number of orders is increasing month after month.

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

Query :

```
SELECT time_of_the_day,
COUNT(*) AS count_for_time_of_the_day
FROM (
  SELECT
CASE WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 0
     AND EXTRACT(HOUR FROM order_purchase_timestamp) <6
     THEN 'Dawn'
     WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 6
     AND EXTRACT(HOUR FROM order_purchase_timestamp) <12
     THEN 'Morning'
     WHEN EXTRACT(HOUR FROM order_purchase_timestamp) >= 12
     AND EXTRACT(HOUR FROM order_purchase_timestamp) < 18
     THEN 'Afternoon'
     ELSE 'Night'
END AS time_of_the_day
FROM `target_datasets.orders`
) tbl_1
GROUP BY time_of_the_day
ORDER BY count_for_time_of_the_day;
```

Query Result :



Row	time_of_the_day	count_for_time_
1	Dawn	4740
2	Morning	22240
3	Night	34100
4	Afternoon	38361

Conclusion : For simplicity I considered Dawn to be from 12 mid night to 6 AM, Morning from 6AM to 12 PM, Afternoon from 12 PM to 6PM and Night from 6PM to 12 mid night.And we can observe from the query results that Brazilians tend to shop more during afternoon followed by night, morning and dawn.

3)Evolution of E-commerce orders in the Brazil region:
3.1) Get month on month orders by states

Query :

```
SELECT
c.customer_state,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS order_month,
COUNT(*) AS count_of_orders
FROM `target_datasets.orders` o
JOIN
`target_datasets.customers` c
ON
o.customer_id = c.customer_id
GROUP BY c.customer_state,order_year,order_month
ORDER BY c.customer_state,order_year,order_month;
```

Query Result :

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECU

Row	customer_state	order_year	order_month	count_of_orders
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
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6
11	AC	2017	11	5

3.2) Distribution of customers across the states in Brazil

Query :

```
SELECT
DISTINCT customer_state,
COUNT(*) OVER(PARTITION BY customer_state) AS statewise_count
FROM `target_datasets.customers`
ORDER BY statewise_count;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DET

Row	customer_state	statewise_count
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350

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) - You can use "payment_value" column in payments table

Query :

```
WITH cte_table AS (SELECT year,month,purchase_value_by_month FROM (
SELECT
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
SUM(p.payment_value)
OVER(PARTITION BY EXTRACT(YEAR FROM
o.order_purchase_timestamp),EXTRACT(MONTH FROM o.order_purchase_timestamp))
AS purchase_value_by_month
FROM
`target-project-bc-1.target_datasets.payments` p
JOIN
`target-project-bc-1.target_datasets.orders` o
ON p.order_id = o.order_id
ORDER BY EXTRACT(MONTH FROM o.order_purchase_timestamp) ASC,
o.order_purchase_timestamp ASC) tbl_1
WHERE year IN (2017,2018) AND month IN (1,2,3,4,5,6,7,8)
GROUP BY year,month,purchase_value_by_month
ORDER BY year,month)

SELECT *,
100 * (r.purchase_value_by_month - l.purchase_value_by_month)/
l.purchase_value_by_month AS percent_diff_month_comparison
 FROM cte_table l JOIN cte_table r ON
l.year <> r.year AND l.month = r.month
ORDER BY l.year,l.month
LIMIT 8;
```

Query results

SAVE RESULTS ▾ EXPLORE DATA ▾

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

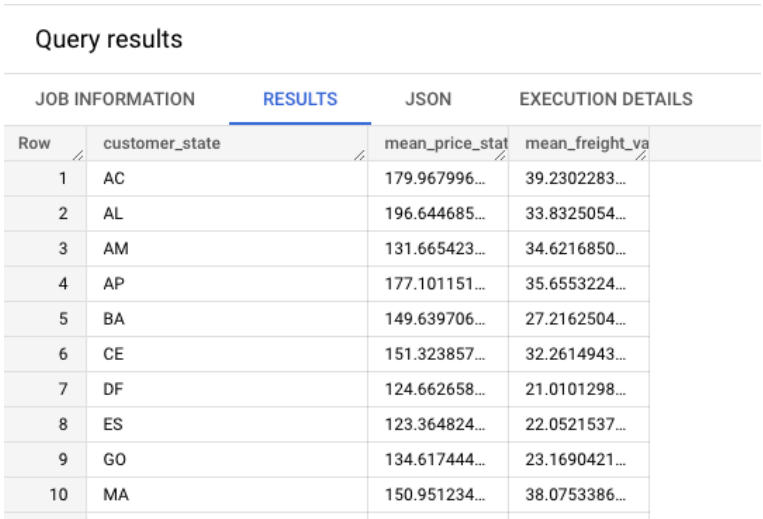
Row	year	month	purchase_value_	year_1	month_1	purchase_value_	percent_diff_mo
1	2017	1	138488.04	2018	1	1115004.18	705.126695...
2	2017	2	291908.01	2018	2	992463.34	239.991814...
3	2017	3	449863.6	2018	3	1159652.12	157.778606...
4	2017	4	417788.03	2018	4	1160785.48	177.840770...
5	2017	5	592918.82	2018	5	1153982.15	94.6273437...
6	2017	6	511276.38	2018	6	1023880.5	100.259691...
7	2017	7	592382.92	2018	7	1066540.75	80.0424546...
8	2017	8	674396.32	2018	8	1022425.32	51.6060052...

Conclusion:As we can see from result the % increase is highest for the January and least for Aug when comparing 2017 to 2018 data.

4.2) Mean & Sum of price and freight value by customer state
Query :

```
SELECT
DISTINCT c.customer_state,
AVG(oi.price) AS mean_price_statewise,
AVG(oi.freight_value) AS mean_freight_value_statewise
FROM `target-project-bc-1.target_datasets.order_items` oi JOIN `target-project-bc-1.target_datasets.orders` o
ON oi.order_id = o.order_id
JOIN `target-project-bc-1.target_datasets.customers` c
ON o.customer_id = c.customer_id
JOIN `target-project-bc-1.target_datasets.geolocation` g
ON c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
GROUP BY c.customer_state
```

Query Result :



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	mean_price_stat	mean_freight_va
1	AC	179.967996...	39.2302283...
2	AL	196.644685...	33.8325054...
3	AM	131.665423...	34.6216850...
4	AP	177.101151...	35.6553224...
5	BA	149.639706...	27.2162504...
6	CE	151.323857...	32.2614943...
7	DF	124.662658...	21.0101298...
8	ES	123.364824...	22.0521537...
9	GO	134.617444...	23.1690421...
10	MA	150.951234...	38.0753386...

5) Analysis on sales, freight and delivery time
5.1) Calculate days between purchasing, delivering and estimated delivery

Query :

```
SELECT order_id,days_to_delivery,estimated_days_to_delivery FROM
(SELECT *,
DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) AS days_to_delivery,
```

```
DATE_DIFF(order_estimated_delivery_date,order_purchase_timestamp, DAY) AS
estimated_days_to_delivery
FROM `target-project-bc-1.target_datasets.orders` ) tbl_1
WHERE days_to_delivery IS NOT NULL
AND estimated_days_to_delivery IS NOT NULL
ORDER BY order_id;
```

Query Result :

Query results

SAVE RESULTS

B INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_id	days_to_delivery	estimated_days
1	00010242fe8c5a6d1ba2dd792...	7	15
2	00018f77f2f0320c557190d7a1...	16	18
3	000229ec398224ef6ca0657da...	7	21
4	00024acbcdf0a6daa1e931b03...	6	11
5	00042b26cf59d7ce69dfabb4e...	25	40
6	00048cc3ae777c65dbb7d2a06...	6	21
7	00054e8431b9d7675808bcb8...	8	24
8	000576fe39319847cbb9d288c...	5	20
9	0005a1a1728c9d785b8e2b08...	9	9
10	0005f50442cb953dcd1d21e1f...	2	20
11	00061f2a7bc09da83e415a52d...	4	15

5.2) Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

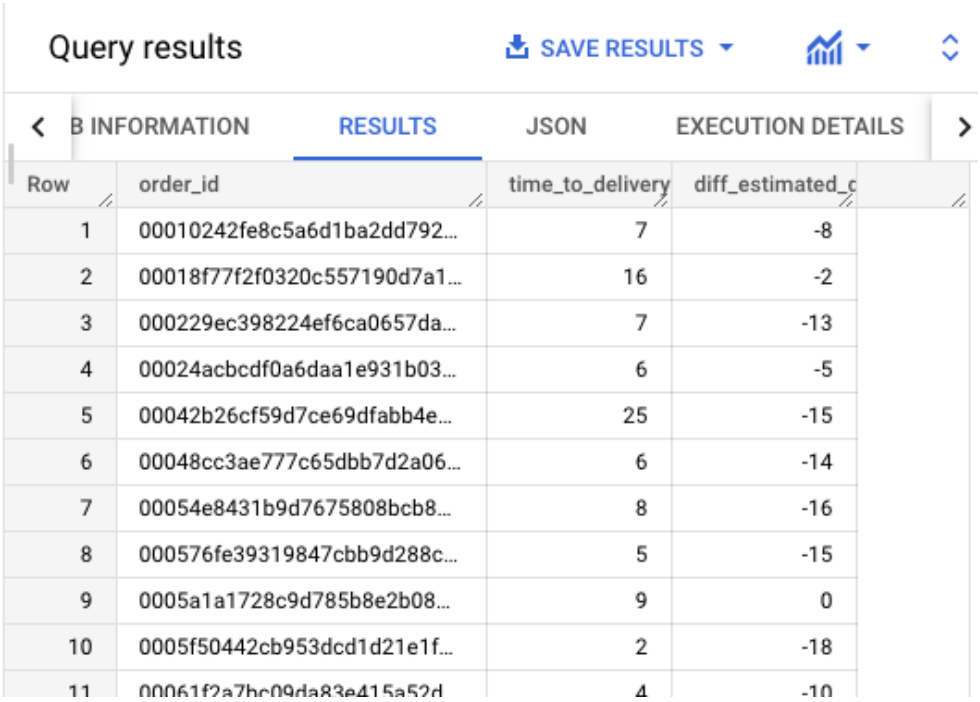
- time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

Query :

```
SELECT order_id,time_to_delivery,diff_estimated_delivery FROM
(SELECT *,
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-project-bc-1.target_datasets.orders` ) tbl_1
WHERE time_to_delivery IS NOT NULL
```

```
AND diff_estimated_delivery IS NOT NULL
ORDER BY order_id;
```

Query Result :


Query results

SAVE RESULTS

B INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_id	time_to_delivery	diff_estimated_c
1	00010242fe8c5a6d1ba2dd792...	7	-8
2	00018f77f2f0320c557190d7a1...	16	-2
3	000229ec398224ef6ca0657da...	7	-13
4	00024acbcdf0a6daa1e931b03...	6	-5
5	00042b26cf59d7ce69dfabb4e...	25	-15
6	00048cc3ae777c65dbb7d2a06...	6	-14
7	00054e8431b9d7675808bcb8...	8	-16
8	000576fe39319847cbb9d288c...	5	-15
9	0005a1a1728c9d785b8e2b08...	9	0
10	0005f50442cb953dcd1d21e1f...	2	-18
11	00061f2a7bc09da83e415a52d	4	-10

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

Query :

```
SELECT
c.customer_state,
AVG(oi.freight_value) AS mean_freight_value,
AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY))
AS time_to_delivery,
AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date,DAY)) AS diff_estimated_delivery
FROM `target_datasets.order_items` oi JOIN `target_datasets.orders`o
ON oi.order_id = o.order_id
JOIN `target_datasets.customers`c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY c.customer_state ASC;
```

Query Result :

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GF

Row	customer_state	mean_freight_va	time_to_delivery	diff_estimated_c
1	AC	40.0733695...	20.3296703...	20.0109890...
2	AL	35.8436711...	23.9929742...	7.97658079...
3	AM	33.2053939...	25.9631901...	18.9754601...
4	AP	34.0060975...	27.7530864...	17.4444444...
5	BA	26.3639589...	18.7746402...	10.1194678...
6	CE	32.7142016...	20.5371669...	10.2566619...
7	DF	21.0413549...	12.5014861...	11.2747346...
8	ES	22.0587765...	15.1928089...	9.76853932...
9	GO	22.7668152...	14.9481774...	11.3728590...
10	MA	38.2570024...	21.2037500...	9.10999999...

5.4) Sort the data to get the following:

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

Query : Top 5 states with lowest average freight value

```
SELECT
c.customer_state,
AVG(oi.freight_value) AS mean_freight_value,
AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY))
AS time_to_delivery,
AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date,DAY)) AS diff_estimated_delivery
FROM `target_datasets.order_items` oi JOIN `target_datasets.orders`o
ON oi.order_id = o.order_id
JOIN `target_datasets.customers`c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY mean_freight_value ASC
LIMIT 5;
```

Query results

SAVE RESULTS EXPLORE

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	mean_freight_va	time_to_delivery	diff_estimated_c
1	SP	15.1472753...	8.25960855...	10.2655943...
2	PR	20.5316515...	11.4807930...	12.5338998...
3	MG	20.6301668...	11.5155221...	12.3971510...
4	RJ	20.9609239...	14.6893821...	11.1444931...
5	DF	21.0413549...	12.5014861...	11.2747346...

Query : Top 5 states with highest average freight value

```
SELECT
c.customer_state,
AVG(oi.freight_value) AS mean_freight_value,
AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY))
AS time_to_delivery,
AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date,DAY)) AS diff_estimated_delivery
FROM `target_datasets.order_items` oi JOIN `target_datasets.orders`o
ON oi.order_id = o.order_id
JOIN `target_datasets.customers`c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY mean_freight_value DESC
LIMIT 5;
```

Query results

SAVE RESULTS EXPLOR

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	mean_freight_va	time_to_delivery	diff_estimated_c
1	RR	42.9844230...	27.8260869...	17.4347826...
2	PB	42.7238039...	20.1194539...	12.1501706...
3	RO	41.0697122...	19.2820512...	19.0805860...
4	AC	40.0733695...	20.3296703...	20.0109890...
5	PI	39.1479704...	18.9311663...	10.6826003...

5.6 - Top 5 states with highest/lowest average time to delivery

Query : Top 5 states with lowest average time to delivery

```
SELECT
c.customer_state,
AVG(oi.freight_value) AS mean_freight_value,
AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY))
AS time_to_delivery,
AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date,DAY)) AS diff_estimated_delivery
FROM `target_datasets.order_items` oi JOIN `target_datasets.orders`o
ON oi.order_id = o.order_id
JOIN `target_datasets.customers`c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY time_to_delivery ASC
LIMIT 5;
```

Query Result :

Query results

SAVE RESULTS EXPLORE DAT

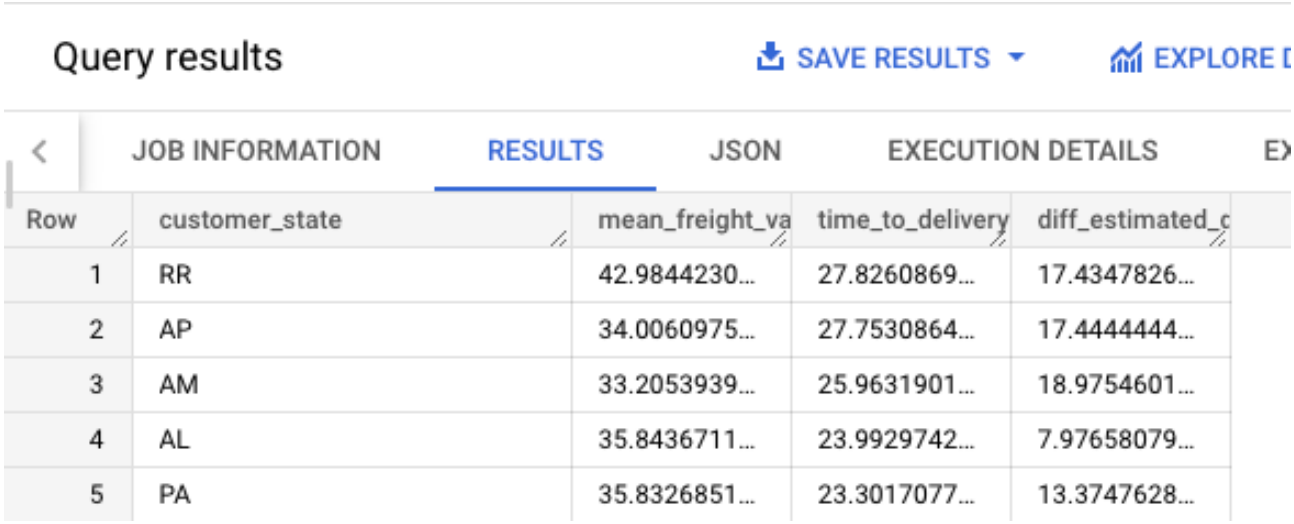
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXEC

Row	customer_state	mean_freight_va	time_to_delivery	diff_estimated_c
1	SP	15.1472753...	8.25960855...	10.2655943...
2	PR	20.5316515...	11.4807930...	12.5338998...
3	MG	20.6301668...	11.5155221...	12.3971510...
4	DF	21.0413549...	12.5014861...	11.2747346...
5	SC	21.4703687...	14.5209858...	10.6688628...

Query : Top 5 states with highest average time to delivery

```
SELECT
c.customer_state,
AVG(oi.freight_value) AS mean_freight_value,
AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY))
AS time_to_delivery,
AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date,DAY)) AS diff_estimated_delivery
FROM `target_datasets.order_items` oi JOIN `target_datasets.orders`o
ON oi.order_id = o.order_id
JOIN `target_datasets.customers`c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY time_to_delivery DESC
```

Query Result :



Row	customer_state	mean_freight_va	time_to_delivery	diff_estimated_c
1	RR	42.9844230...	27.8260869...	17.4347826...
2	AP	34.0060975...	27.7530864...	17.4444444...
3	AM	33.2053939...	25.9631901...	18.9754601...
4	AL	35.8436711...	23.9929742...	7.97658079...
5	PA	35.8326851...	23.3017077...	13.3747628...

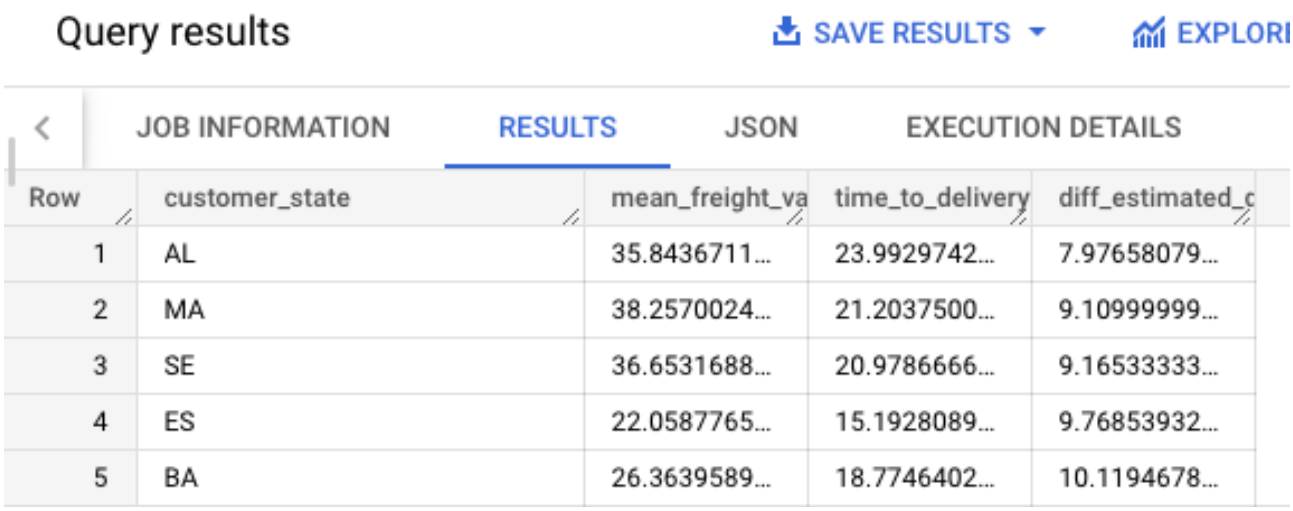
5.7)Top 5 states where delivery is really fast/ not so fast compared to estimated date

Query : Top 5 states where delivery is really fast

```
SELECT
c.customer_state,
AVG(oi.freight_value) AS mean_freight_value,
AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY))
AS time_to_delivery,
```

```
AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date,DAY)) AS diff_estimated_delivery
FROM `target_datasets.order_items` oi JOIN `target_datasets.orders`o
ON oi.order_id = o.order_id
JOIN `target_datasets.customers`c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY diff_estimated_delivery ASC
LIMIT 5;
```

Query Result :

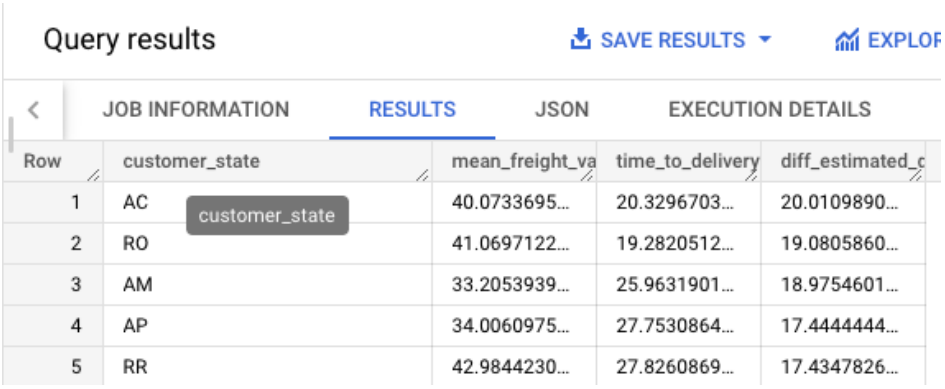


Row	customer_state	mean_freight_va	time_to_delivery	diff_estimated_c
1	AL	35.8436711...	23.9929742...	7.97658079...
2	MA	38.2570024...	21.2037500...	9.10999999...
3	SE	36.6531688...	20.9786666...	9.16533333...
4	ES	22.0587765...	15.1928089...	9.76853932...
5	BA	26.3639589...	18.7746402...	10.1194678...

Query : Top 5 states where delivery is really slow

```
SELECT
c.customer_state,
AVG(oi.freight_value) AS mean_freight_value,
AVG(DATE_DIFF(o.order_delivered_customer_date,o.order_purchase_timestamp,DAY))
AS time_to_delivery,
AVG(DATE_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date,DAY)) AS diff_estimated_delivery
FROM `target_datasets.order_items` oi JOIN `target_datasets.orders`o
ON oi.order_id = o.order_id
JOIN `target_datasets.customers`c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY diff_estimated_delivery DESC
LIMIT 5;
```

Query Result :



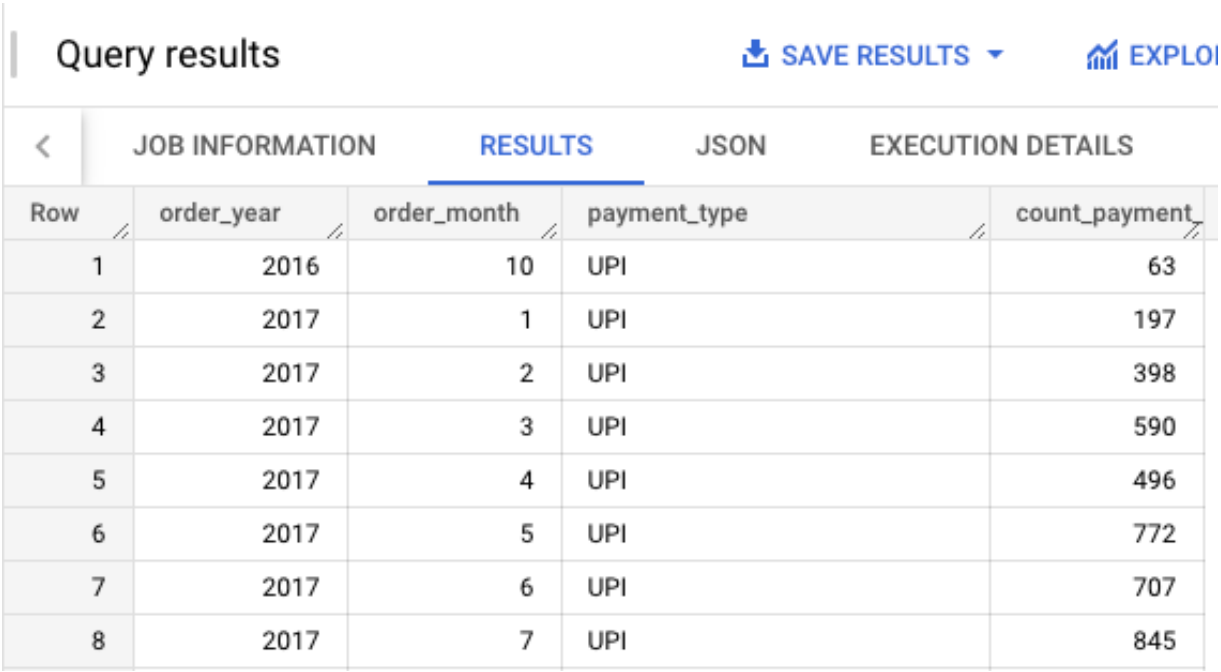
Row	customer_state	mean_freight_va	time_to_delivery	diff_estimated_c
1	AC	40.0733695...	20.3296703...	20.0109890...
2	RO	41.0697122...	19.2820512...	19.0805860...
3	AM	33.2053939...	25.9631901...	18.9754601...
4	AP	34.0060975...	27.7530864...	17.4444444...
5	RR	42.9844230...	27.8260869...	17.4347826...

6) Payment type analysis:

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

Query :

```
SELECT
EXTRACT(YEAR from o.order_purchase_timestamp) AS order_year,
EXTRACT(MONTH from o.order_purchase_timestamp) AS order_month,
p.payment_type,
COUNT(p.order_id) AS count_payment_type_wise
FROM `target_datasets.payments`  p
JOIN
`target_datasets.orders`o
ON p.order_id = o.order_id
GROUP BY p.payment_type,order_year,order_month
ORDER BY p.payment_type,order_year,order_month;
```

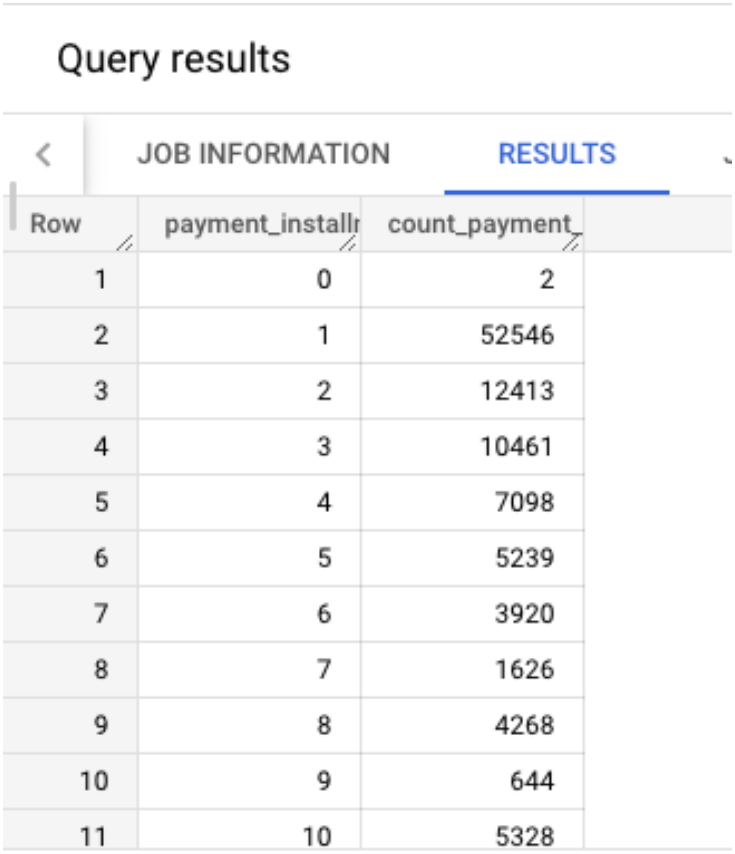


Row	order_year	order_month	payment_type	count_payment_
1	2016	10	UPI	63
2	2017	1	UPI	197
3	2017	2	UPI	398
4	2017	3	UPI	590
5	2017	4	UPI	496
6	2017	5	UPI	772
7	2017	6	UPI	707
8	2017	7	UPI	845

6.2) Count of orders based on the no. of payment installments

Query :

```
SELECT
payment_installments,
COUNT(order_id) AS count_payment_installment_wise
FROM `target_datasets.payments`
GROUP BY payment_installments
ORDER BY payment_installments;
```



Query results

JOB INFORMATION | RESULTS

Row	payment_installr	count_payment_
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644
11	10	5328

7.Actionable Insights

- We can reduce the difference between estimated delivery time and delivery time by working on the logistics

- Offer targeted promotions during the afternoon and evening hours (since these are the peak hours when Brazilians tend to shop more) to capitalize on peak purchasing times. This could include discounts or special offers on popular items or products that are frequently purchased during those times.

- Consider adjusting pricing for products based on the time of day to encourage more sales during off-peak hours. For example, offering lower prices on items during the morning hours could help increase sales during that time.

- Improve the customer experience during peak purchasing times by ensuring that website performance and customer service are optimized. This could include offering live chat support or increasing staffing levels during busy periods.

- Explore the possibility of expanding product offerings during peak purchasing times to meet customer demand. This could include adding new products or increasing inventory levels for popular items.

8. Recommendations :

- Consider expanding inventory levels for popular products: If certain products are consistently popular during peak purchasing times, it may be worthwhile to increase inventory levels to ensure that these products are always available for customers. This can help to prevent customers from leaving the website to search for the same products elsewhere, potentially losing the sale.

- Your analysis revealed that customers have different purchasing patterns throughout the day. By using data to analyze which products are most popular during different times of day, the e-commerce website can optimize its product offerings to better match customer demand. For example, if customers are more likely to purchase electronics during the afternoon hours, the website could highlight these products during that time.