<DEBERSHI MITRA>

SQL BUSINESS CASE → TARGET

Analyzing Customer table:

1. Total number of customers data we have → 99441

SQL query →
```
SELECT COUNT(distinct(c.customer_id))
    FROM `target-sql-project-391015.target_market.customers` c
```

2 Total number of unique customers → 96096

SQL query →
```
SELECT COUNT(distinct(c.customer_unique_id))
    FROM `target-sql-project-391015.target_market.customers` c
```

3 Number of unique zip code prefix → 14994

SQL query →
```
SELECT COUNT(distinct(c.customer_zip_code_prefix))
    FROM `target-sql-project-391015.target_market.customers` c
```

INSIGHT: This means that Target has Customers from 14994 different locations of Brazil.

---

4 Number of cities → 4119

SQL query →
```
SELECT COUNT(distinct(c.customer_city))
    FROM `target-sql-project-391015.target_market.customers` c
```

5 Number of states → 27

SQL query →
```
SELECT COUNT(distinct(c.customer_state))
```

```
FROM `target-sql-project-391015.target_market.customers` c
```

*INSIGHT: Customers are from 4119 different cities and 27 different states from **B**razil.*

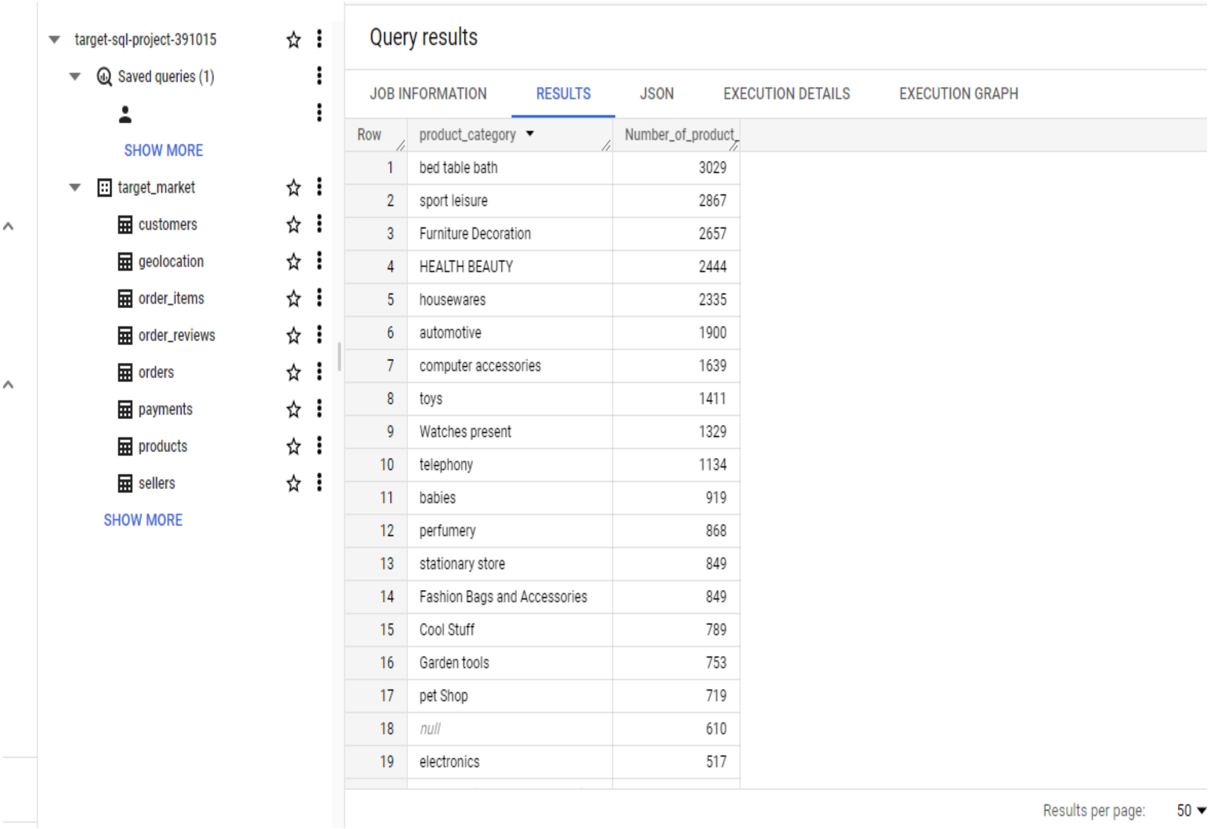
Analyzing Products:

1.Number of Unique Products available in Target ➔ 32951

SQL Query :
```
SELECT COUNT(product_id)
FROM `target-sql-project-391015.target_market.products`;
```

2. Number of products per category with the product category name:

SQL Query :
```
SELECT product_category
COUNT(DISTINCT(product_id)) AS Number_of_product_per_category
FROM `target-sql-project-391015.target_market.products’
GROUP BY product_category
ORDER BY COUNT(DISTINCT(product_id)) desc;
```



Query results

Row	product_category	Number_of_product
1	bed table bath	3029
2	sport leisure	2867
3	Furniture Decoration	2657
4	HEALTH BEAUTY	2444
5	housewares	2335
6	automotive	1900
7	computer accessories	1639
8	toys	1411
9	Watches present	1329
10	telephony	1134
11	babies	919
12	perfumery	868
13	stationary store	849
14	Fashion Bags and Accessories	849
15	Cool Stuff	789
16	Garden tools	753
17	pet Shop	719
18	null	610
19	electronics	517

Analyzing Sellers Data :

1. Number of Distinct seller Id's → 3095.

   SQL Query →
   ```
   SELECT
   COUNT(DISTINCT(seller_id))
   FROM `target-sql-project-391015.target_market.sellers` ;
   ```

2. Number of Seller's ZIP Code prefix → 2246.

   SQL Query →
   ```
   SELECT
   COUNT(DISTINCT(seller_zip_code_prefix))
   FROM `target-sql-project-391015.target_market.sellers` ;
   ```

3. Number of distinct seller city → 611

   SQL Query →
   ```
   SELECT
   COUNT(DISTINCT(seller_city))
   FROM `target-sql-project-391015.target_market.sellers` ;
   ```

4. Number of Seller State's → 23

   SQL Query →
   ```
   SELECT
   COUNT(DISTINCT(seller_state))
   FROM `target-sql-project-391015.target_market.sellers` ;
   ```

INSIGHTS:* *There are 3095 seller's data present here. These sellers are from 2246 locations, 611 cities and 23 different states in Brazil

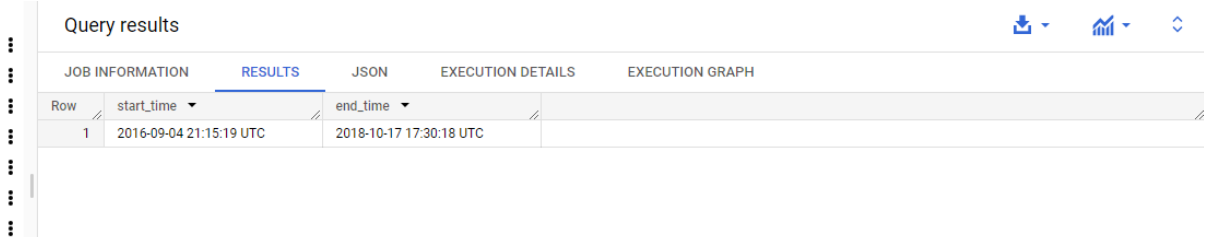
1.

B)

Time range between which the orders were placed ➔

SQL QUERY:

```
SELECT MIN(order_purchase_timestamp) AS start_time,
MAX(order_purchase_timestamp) AS end_time
FROM `target-sql-project-391015.target_market.orders`
```



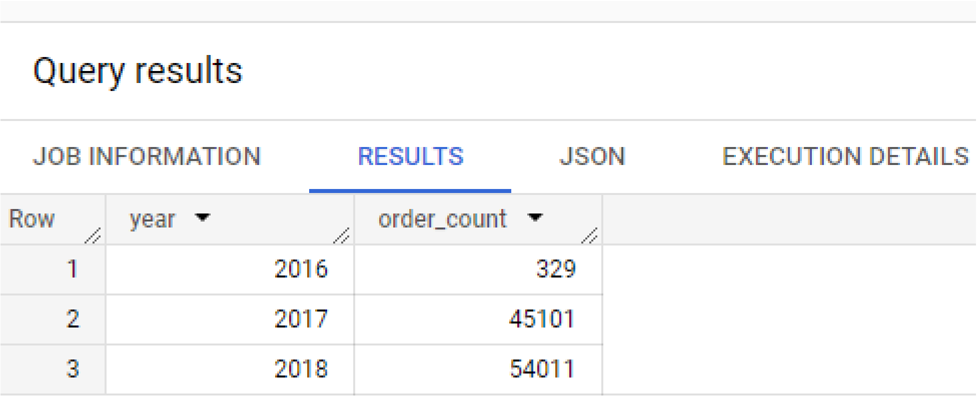
Row	start_time	end_time
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

INSIGHTS : The Time Range Between Which The orders were placed is from 2016-09-04 to 2018-10-17.

To determine if there is a growing trend in the number of orders placed over the past years, Assuming that the **order_purchase_timestamp** column represents the timestamp of the purchase,

SQL Query :

```
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
COUNT(*) AS order_count
FROM `target-sql-project-391015.target_market.orders’
GROUP BY year
ORDER BY year;
```

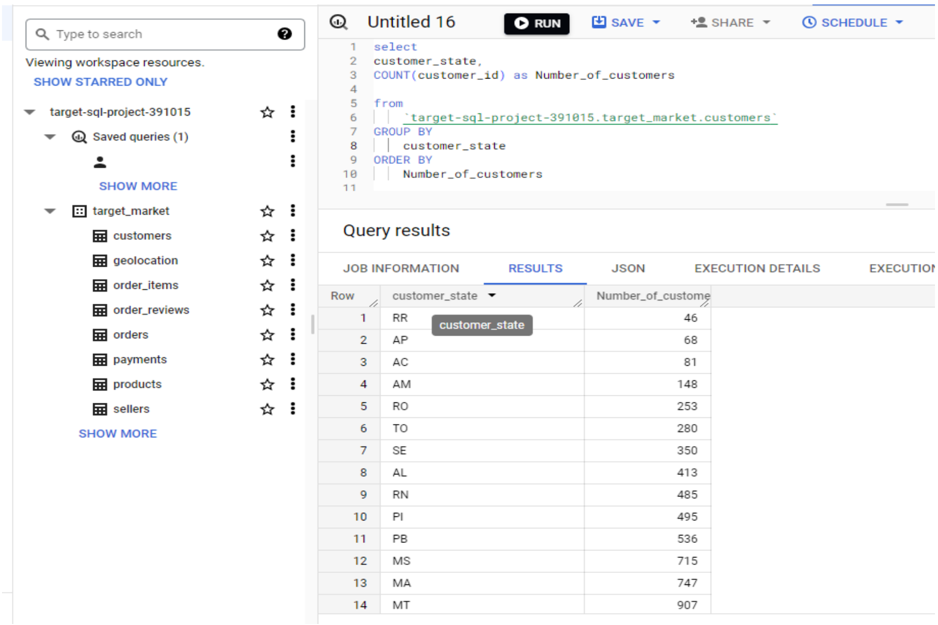


Row	year	order_count
1	2016	329
2	2017	45101
3	2018	54011

C.Number of Customers from Each State :

SQL query →

```
SELECT customer_state,

COUNT(customer_id) as Number_of_customers

FROM `target-sql-project-391015.target_market.customers`
GROUP BY customer_state
ORDER BY Number_of_customers
```



```
select
customer_state,
COUNT(customer_id) as Number_of_customers

from
  `target-sql-project-391015.target_market.customers`
GROUP BY
  customer_state
ORDER BY
  Number_of_customers
```

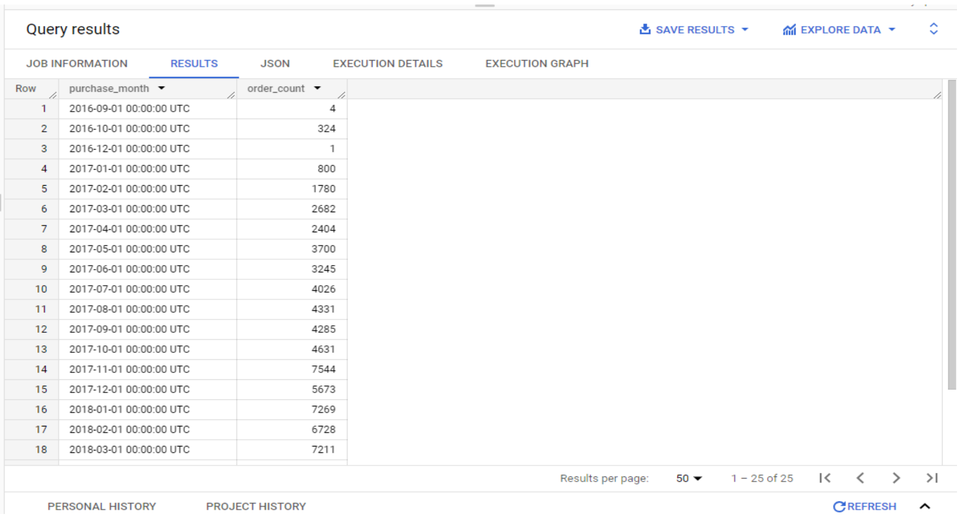
Row	customer_state	Number_of_custome
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350
8	AL	413
9	RN	485
10	PI	495
11	PB	536
12	MS	715
13	MA	747
14	MT	907

INSIGHT: The Top 3 States with highest Number of Customers are : SP(41746),RJ(12852),MG(16351)

B. Monthly Seasonality in terms of the Number of orders being placed?

SQL Query :

```
            SELECT
                    DATE_TRUNC(order_purchase_timestamp, MONTH) AS purchase_month,
                    COUNT(*) AS order_count
                    FROM `target-sql-project-391015.target_market.orders`
                    GROUP BY purchase_month
                    ORDER BY purchase_month;
```



Row	purchase_month	order_count
1	2016-09-01 00:00:00 UTC	4
2	2016-10-01 00:00:00 UTC	324
3	2016-12-01 00:00:00 UTC	1
4	2017-01-01 00:00:00 UTC	800
5	2017-02-01 00:00:00 UTC	1780
6	2017-03-01 00:00:00 UTC	2682
7	2017-04-01 00:00:00 UTC	2404
8	2017-05-01 00:00:00 UTC	3700
9	2017-06-01 00:00:00 UTC	3245
10	2017-07-01 00:00:00 UTC	4026
11	2017-08-01 00:00:00 UTC	4331
12	2017-09-01 00:00:00 UTC	4285
13	2017-10-01 00:00:00 UTC	4631
14	2017-11-01 00:00:00 UTC	7544
15	2017-12-01 00:00:00 UTC	5673
16	2018-01-01 00:00:00 UTC	7269
17	2018-02-01 00:00:00 UTC	6728
18	2018-03-01 00:00:00 UTC	7211

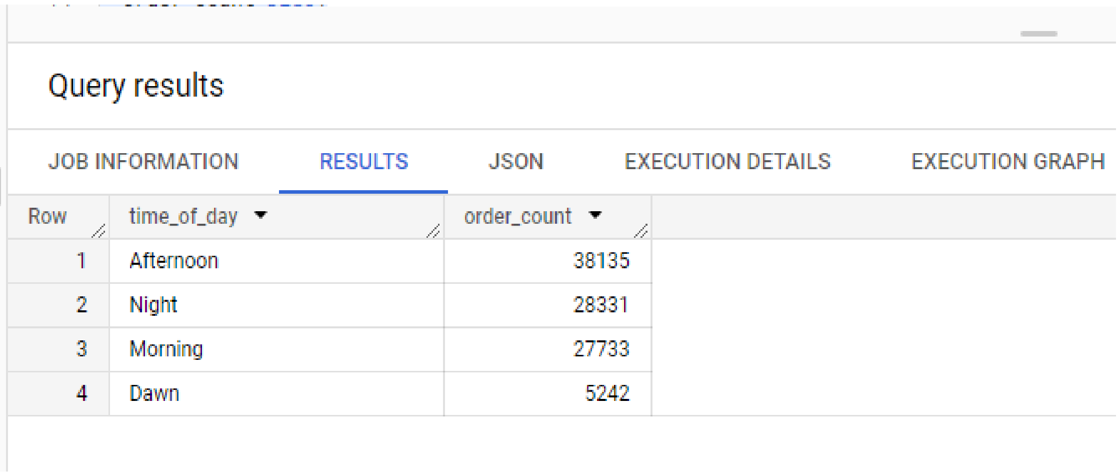
C. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

> 0-6 hrs : Dawn
> 7-12 hrs : Mornings
> 13-18 hrs : Afternoon
> 19-23 hrs : Night

SQL Query ➔

```
SELECT

  CASE
    WHEN EXTRACT(HOUR FROM CAST(order_purchase_timestamp AS TIMESTAMP)) BETWEEN 0 AND 6
THEN 'Dawn'
    WHEN EXTRACT(HOUR FROM CAST(order_purchase_timestamp AS TIMESTAMP)) BETWEEN 7 AND 12
THEN 'Morning'
    WHEN EXTRACT(HOUR FROM CAST(order_purchase_timestamp AS TIMESTAMP)) BETWEEN 13 AND 18
THEN 'Afternoon'
```

```
    WHEN EXTRACT(HOUR FROM CAST(order_purchase_timestamp AS TIMESTAMP)) BETWEEN 19 AND 23
THEN 'Night'
  END AS time_of_day,
  COUNT(*) AS order_count
FROM
  `target-sql-project-391015.target_market.orders`
GROUP BY
  time_of_day
ORDER BY
    order_count DESC;
```



Query results

Row	time_of_day	order_count
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

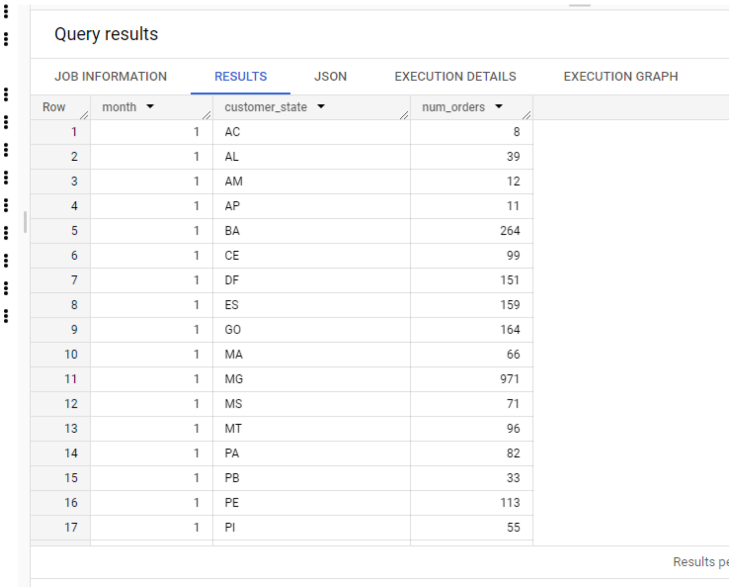
INSIGHTS → According to the Data Brazilian Customers mostly prefer Afternoon to place their orders.

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

1.Get the month on month no. of orders placed in each state.

SQL Query:

```
SELECT
month,
customer_state,
COUNT(DISTINCTorder_id)ASnum_orders
FROM (
SELECT
EXTRACT(MONTH FROM TIMESTAMP_TRUNC(order_purchase_timestamp, MONTH)) AS month,
customer_state,
order_id
FROM
`target-sql-project-391015.target_market.orders` o
JOIN
`target-sql-project-391015.target_market.customers` c ON o.customer_id = c.customer_id
)
GROUP BY
month,
customer_state
ORDER BY month, customer_state;
```

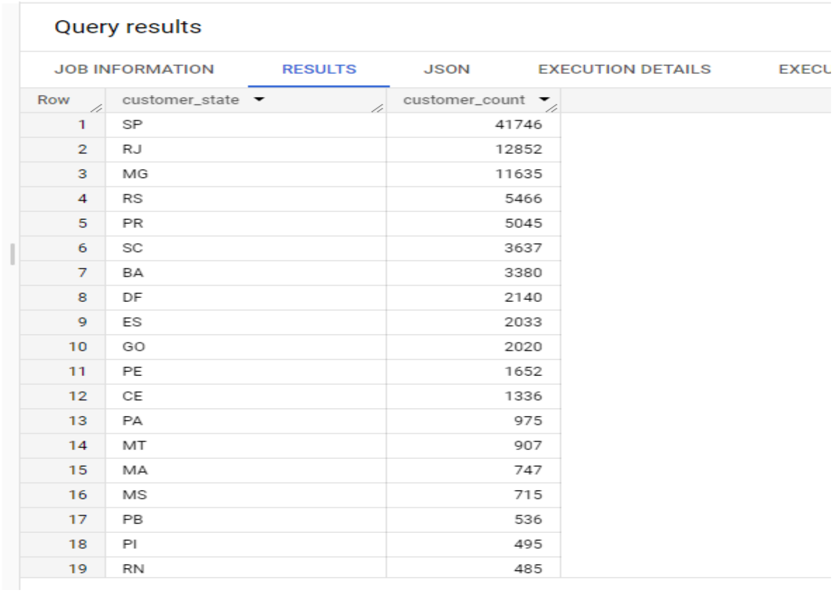


Row	month	customer_state	num_orders
1	1	AC	8
2	1	AL	39
3	1	AM	12
4	1	AP	11
5	1	BA	264
6	1	CE	99
7	1	DF	151
8	1	ES	159
9	1	GO	164
10	1	MA	66
11	1	MG	971
12	1	MS	71
13	1	MT	96
14	1	PA	82
15	1	PB	33
16	1	PE	113
17	1	PI	55

2 To determine how customers are distributed across all the states, you can use the following SQL query:

SQL Query →

```
SELECT customer_state, COUNT(*) AS customer_count
FROM `target-sql-project-391015.target_market.customers`
GROUP BY customer_state
ORDER BY customer_count DESC;
```



Query results

Row	customer_state	customer_count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020
11	PE	1652
12	CE	1336
13	PA	975
14	MT	907
15	MA	747
16	MS	715
17	PB	536
18	PI	495
19	RN	485

***INSIGHT** – Number of Unique customers in each state is*

SP – 41740

RJ- 12852

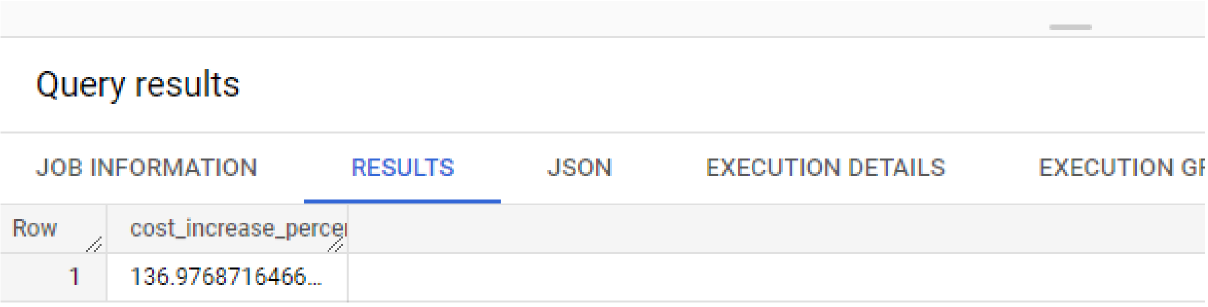
MG- 16351

Impact on Economy:

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

   SQL Query →

```
SELECT
  (SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018 THEN p.payment_value
ELSE 0 END) -
   SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 THEN p.payment_value
ELSE 0 END)) /
  SUM(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 THEN p.payment_value
ELSE 0 END) * 100 AS cost_increase_percentage
FROM
  `target-sql-project-391015.target_market.orders` o
JOIN
  `target-sql-project-391015.target_market.payments` p
ON
  o.order_id = p.order_id
WHERE
  EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018)
  AND EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
```



Query results

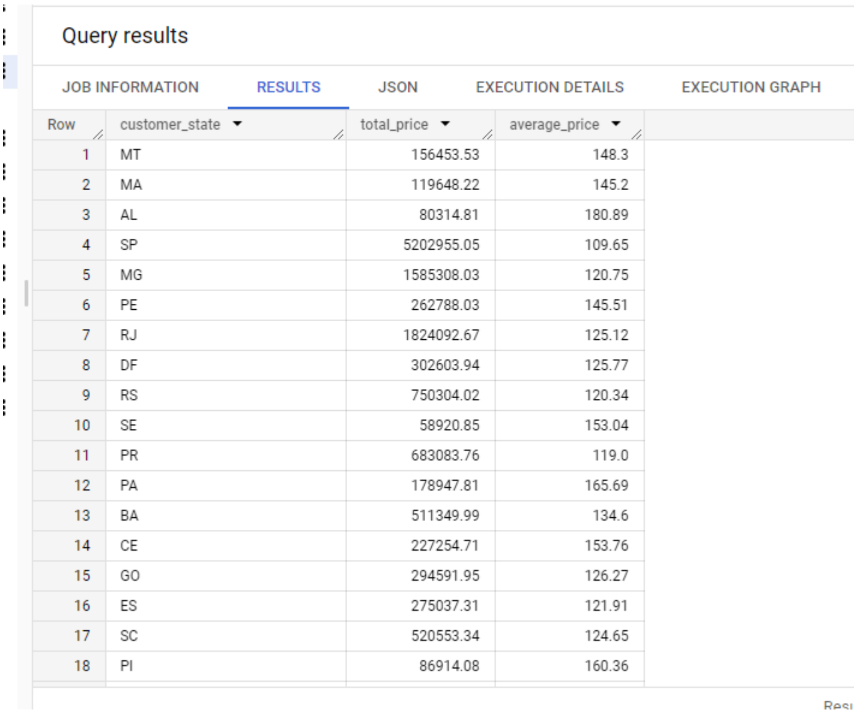
Row	cost_increase_perce
1	136.9768716466...

INSIGHTS: *The % increase in the cost of orders from year 2017 to 2018 is 136.97 %*

2. Calculate the total & Average value of order price for each state.

SQL QUERY ➔

```
SELECT
c.customer_state,
round(SUM(oi.price),2) AS total_price,
round(AVG(oi.price),2) AS average_price
FROM
`target-sql-project-391015.target_market.orders` AS o
JOIN
`target-sql-project-391015.target_market.order_items` AS oi ON o.order_id = oi.order_id
JOIN
`target-sql-project-391015.target_market.customers` c on o.customer_id = c.customer_id
GROUP BY
c.customer_state
```

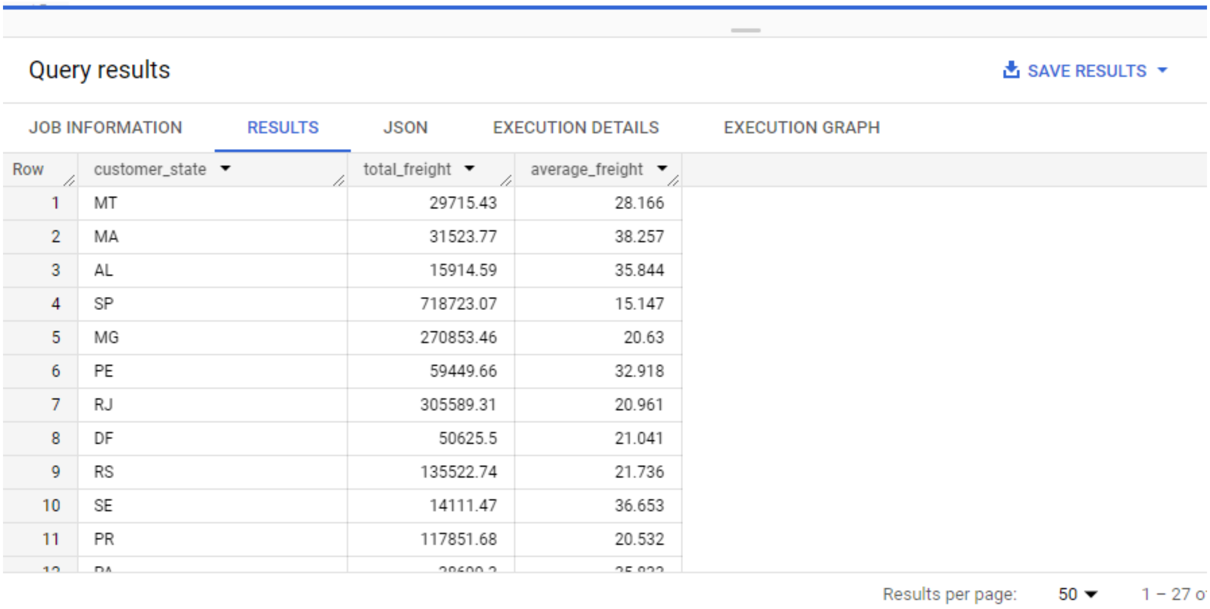


Row	customer_state	total_price	average_price
1	MT	156453.53	148.3
2	MA	119648.22	145.2
3	AL	80314.81	180.89
4	SP	5202955.05	109.65
5	MG	1585308.03	120.75
6	PE	262788.03	145.51
7	RJ	1824092.67	125.12
8	DF	302603.94	125.77
9	RS	750304.02	120.34
10	SE	58920.85	153.04
11	PR	683083.76	119.0
12	PA	178947.81	165.69
13	BA	511349.99	134.6
14	CE	227254.71	153.76
15	GO	294591.95	126.27
16	ES	275037.31	121.91
17	SC	520553.34	124.65
18	PI	86914.08	160.36

3 . To calculate the total and average value of order freight for each state:

SQL Query:

```
SELECT

  customer_state,

  round(SUM(freight_value),3) AS total_freight,

  round(AVG(freight_value),3) AS average_freight

FROM

  `target-sql-project-391015.target_market.orders` AS o

JOIN

  `target-sql-project-391015.target_market.order_items` AS oi ON o.order_id = oi.order_id

JOIN

  `target-sql-project-391015.target_market.customers` AS c ON o.customer_id = c.customer_id

GROUP BY

  customer_state
```



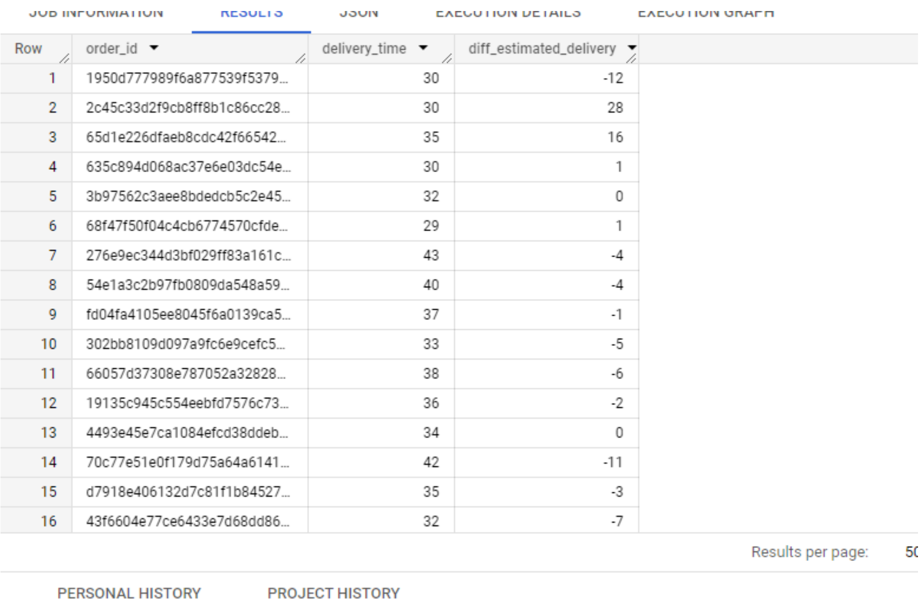
Row	customer_state	total_freight	average_freight
1	MT	29715.43	28.166
2	MA	31523.77	38.257
3	AL	15914.59	35.844
4	SP	718723.07	15.147
5	MG	270853.46	20.63
6	PE	59449.66	32.918
7	RJ	305589.31	20.961
8	DF	50625.5	21.041
9	RS	135522.74	21.736
10	SE	14111.47	36.653
11	PR	117851.68	20.532
12	PA	38699.3	35.832

Analysis Based On sales, freight and delivery time.

A. *The Number of days taken to deliver each order from the order's purchase date as delivery time between the estimated and actual delivery date of an order.*

SQL Query ➔

```
SELECT
  order_id,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS delivery_time,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS
diff_estimated_delivery
FROM
  `target-sql-project-391015.target_market.orders`
```



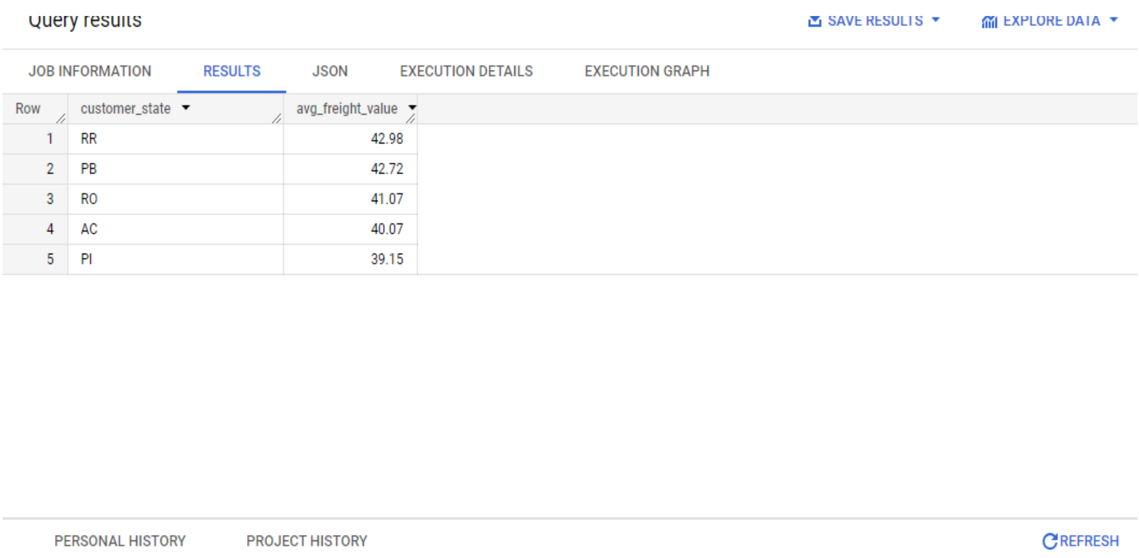
Row	order_id	delivery_time	diff_estimated_delivery
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	30	28
3	65d1e226dfaeb8cdc42f66542...	35	16
4	635c894d068ac37e6e03dc54e...	30	1
5	3b97562c3aee8bdedcb5c2e45...	32	0
6	68f47f50f04c4cb6774570cfde...	29	1
7	276e9ec344d3bf029ff83a161c...	43	-4
8	54e1a3c2b97fb0809da548a59...	40	-4
9	fd04fa4105ee8045f6a0139ca5...	37	-1
10	302bb8109d097a9fc6e9cefc5...	33	-5
11	66057d37308e787052a32828...	38	-6
12	19135c945c554eebfd7576c73...	36	-2
13	4493e45e7ca1084efcd38ddeb...	34	0
14	70c77e51e0f179d75a64a6141...	42	-11
15	d7918e406132d7c81f1b84527...	35	-3
16	43f6604e77ce6433e7d68dd86...	32	-7

B.

1. Top 5 States with Highest Average Freight Value:

SQL Query:

```
SELECT
  customer_state,
  round(AVG(freight_value), 2) AS avg_freight_value
  FROM `target-sql-project-391015.target_market.order_items` AS oi
  JOIN `target-sql-project-391015.target_market.orders` AS o ON oi.order_id =
o.order_id

  JOIN `target-sql-project-391015.target_market.customers` AS c ON
o.customer_id = c.customer_id
  GROUP BY customer_state
  ORDER BY avg_freight_value DESC
  LIMIT 5
```

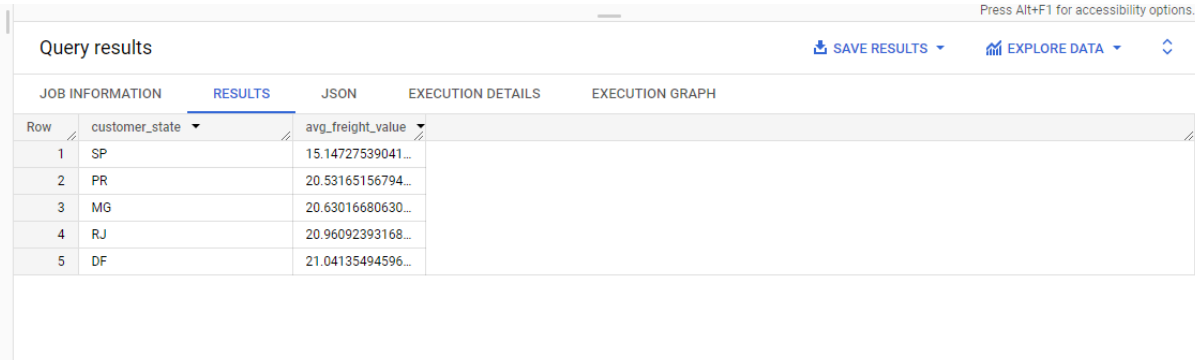


Row	customer_state	avg_freight_value
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

2) Top 5 states with lowest average freight value:

SQL Query:

```
SELECT
 customer_state,
  AVG(freight_value) AS avg_freight_value
FROM
  `target-sql-project-391015.target_market.order_items` AS oi
JOIN
  `target-sql-project-391015.target_market.orders` AS o ON oi.order_id = o.order_id
JOIN
  `target-sql-project-391015.target_market.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
  customer_state
ORDER BY
  avg_freight_value ASC
LIMIT
  5
```



Row	customer_state	avg_freight_value
1	SP	15.14727539041…
2	PR	20.53165156794…
3	MG	20.63016680630…
4	RJ	20.96092393168…
5	DF	21.04135494596…

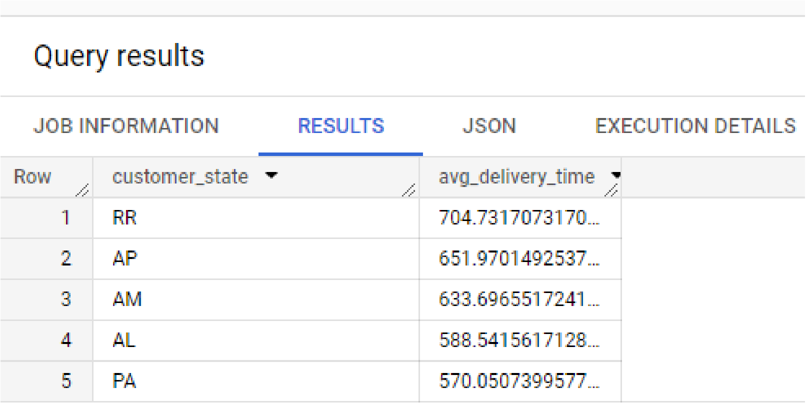
C.

1.

A) Top 5 States with The highest Delivery time:

SQL Query →

```
WITH order_delivery_duration AS (
  SELECT
    o.order_id,
    o.order_status,
    o.order_delivered_customer_date,
    o.order_purchase_timestamp,
    c.customer_state,
    TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, HOUR) AS
delivery_duration
  FROM
    `target-sql-project-391015.target_market.orders` AS o
  INNER JOIN
    `target-sql-project-391015.target_market.customers` AS c ON o.customer_id =
c.customer_id
  WHERE
    o.order_status = 'delivered'
)

SELECT
  customer_state,
  AVG(delivery_duration) AS avg_delivery_time
FROM
  order_delivery_duration
GROUP BY
  customer_state
ORDER BY
avg_delivery_time DESC
LIMIT 5
```



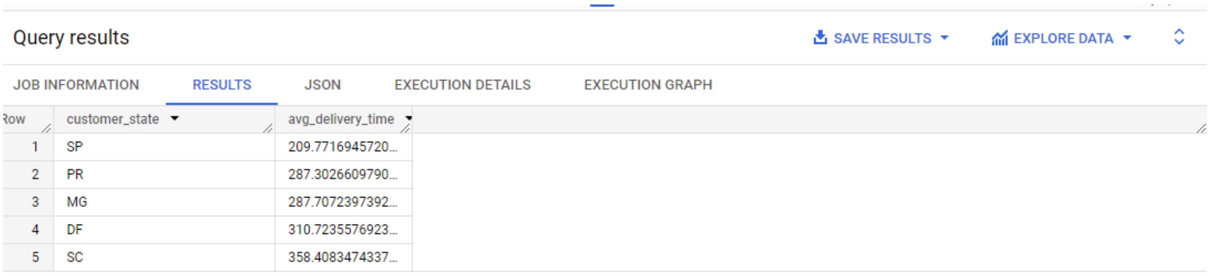
Row	customer_state	avg_delivery_time
1	RR	704.7317073170...
2	AP	651.9701492537...
3	AM	633.6965517241...
4	AL	588.5415617128...
5	PA	570.0507399577...

2.

Top 5 States with Lowest Average delivery time:

SQL Query:

```
WITH order_delivery_duration AS (
  SELECT
    o.order_id,
    o.order_status,
    o.order_delivered_customer_date,
    o.order_purchase_timestamp,
    c.customer_state,
    TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, HOUR) AS
delivery_duration
  FROM
    `target-sql-project-391015.target_market.orders` AS o
  INNER JOIN
    `target-sql-project-391015.target_market.customers` AS c ON o.customer_id =
c.customer_id
  WHERE
    o.order_status = 'delivered'
)

SELECT
  customer_state,
  AVG(delivery_duration) AS avg_delivery_time
FROM
  order_delivery_duration
GROUP BY
  customer_state
ORDER BY
  avg_delivery_time
LIMIT 5
```

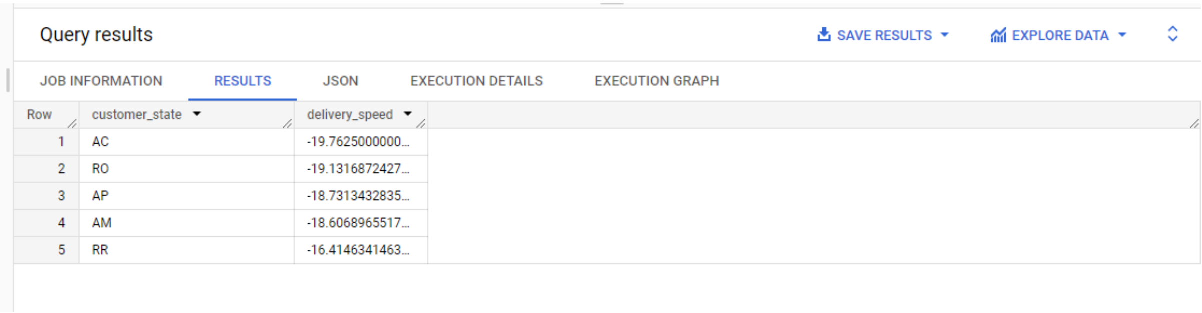


Row	customer_state	avg_delivery_time
1	SP	209.7716945720...
2	PR	287.3026609790...
3	MG	287.7072397392...
4	DF	310.7235576923...
5	SC	358.4083474337...

D. Top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

SQL Query →

```
SELECT
customer_state,
AVG(DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, DAY)) AS
delivery_speed

FROM
  `target-sql-project-391015.target_market.orders` o
JOIN `target-sql-project-391015.target_market.customers` c on o.customer_id = c.customer_id
WHERE
  order_status = 'delivered'
GROUP BY
  customer_state
ORDER BY
  delivery_speed ASC
LIMIT
  5;
```



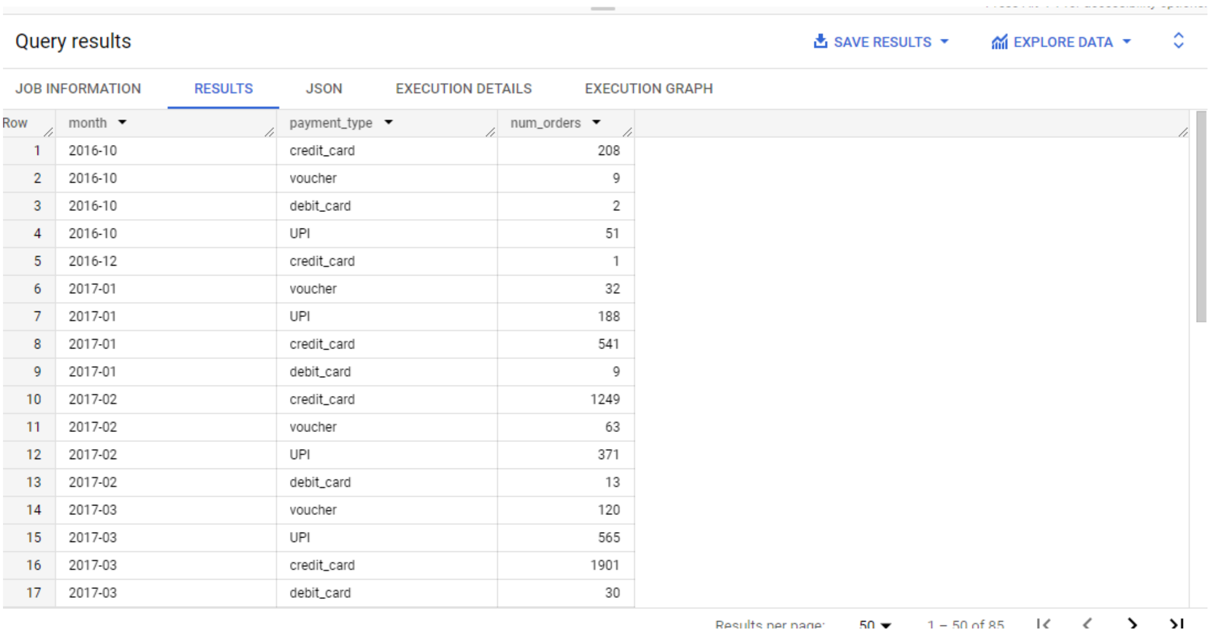
Row	customer_state	delivery_speed
1	AC	-19.7625000000...
2	RO	-19.1316872427...
3	AP	-18.7313432835...
4	AM	-18.6068965517...
5	RR	-16.4146341463...

Analysis Based on the payments:

1. *Month on Month Number of orders places using different payment types:*

SQL Query →

```
WITH monthly_orders AS
(
  SELECT
    FORMAT_TIMESTAMP('%Y-%m', order_purchase_timestamp) AS month,
    COUNT(DISTINCT o.order_id) AS num_orders,
    payment_type
  FROM
    `target-sql-project-391015.target_market.orders` o
  INNER JOIN
    `target-sql-project-391015.target_market.payments` p ON o.order_id = p.order_id
  WHERE
    order_status = 'delivered'
  GROUP BY
    month,
    payment_type
)

SELECT
  month,
  payment_type,
  num_orders
FROM
  monthly_orders
ORDER BY
  month
```



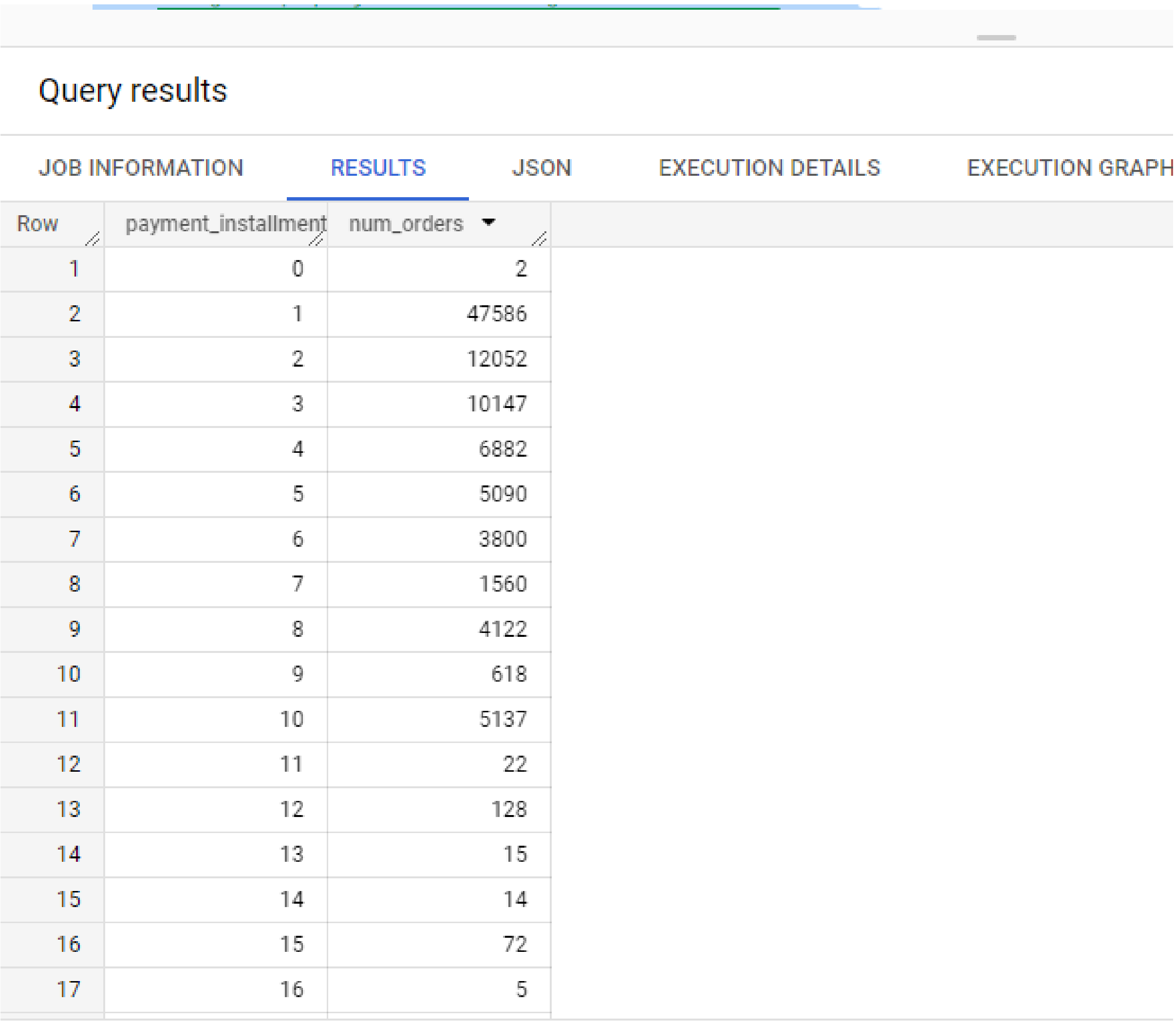
Row	month ▾	payment_type ▾	num_orders ▾
1	2016-10	credit_card	208
2	2016-10	voucher	9
3	2016-10	debit_card	2
4	2016-10	UPI	51
5	2016-12	credit_card	1
6	2017-01	voucher	32
7	2017-01	UPI	188
8	2017-01	credit_card	541
9	2017-01	debit_card	9
10	2017-02	credit_card	1249
11	2017-02	voucher	63
12	2017-02	UPI	371
13	2017-02	debit_card	13
14	2017-03	voucher	120
15	2017-03	UPI	565
16	2017-03	credit_card	1901
17	2017-03	debit_card	30

INSIGHTS: The query provides insights into the month-on-month number of orders placed using different payment types, helping identify trends and preferences in consumer behaviour and payment methods over time.

2.) Number of orders places on the basis of the payment installments that have been paid:

SQL Query ➔

```
SELECT p.payment_installments, COUNT(DISTINCT p.order_id) AS num_orders
FROM `target-sql-project-391015.target_market.orders` AS o
JOIN `target-sql-project-391015.target_market.payments` AS p
ON o.order_id = p.order_id
WHERE order_status = 'delivered'
GROUP BY payment_installments
ORDER BY payment_installments;
```



Row	payment_installment	num_orders
1	0	2
2	1	47586
3	2	12052
4	3	10147
5	4	6882
6	5	5090
7	6	3800
8	7	1560
9	8	4122
10	9	618
11	10	5137
12	11	22
13	12	128
14	13	15
15	14	14
16	15	72
17	16	5

INSIGHTS : Overall, analyzing the relationship between payment installments and the number of orders can offer valuable insights for optimizing pricing, marketing, and customer engagement strategies to drive sales and enhance customer satisfaction.

Actionable insights and recommendations based on payments *:*
Insights:

1. Identify the payment types preferred by customers month-on-month.
2. Track changes in the number of orders placed over time for each payment type.
3. Determine if certain payment types are gaining or losing popularity over the months.

Action Items:

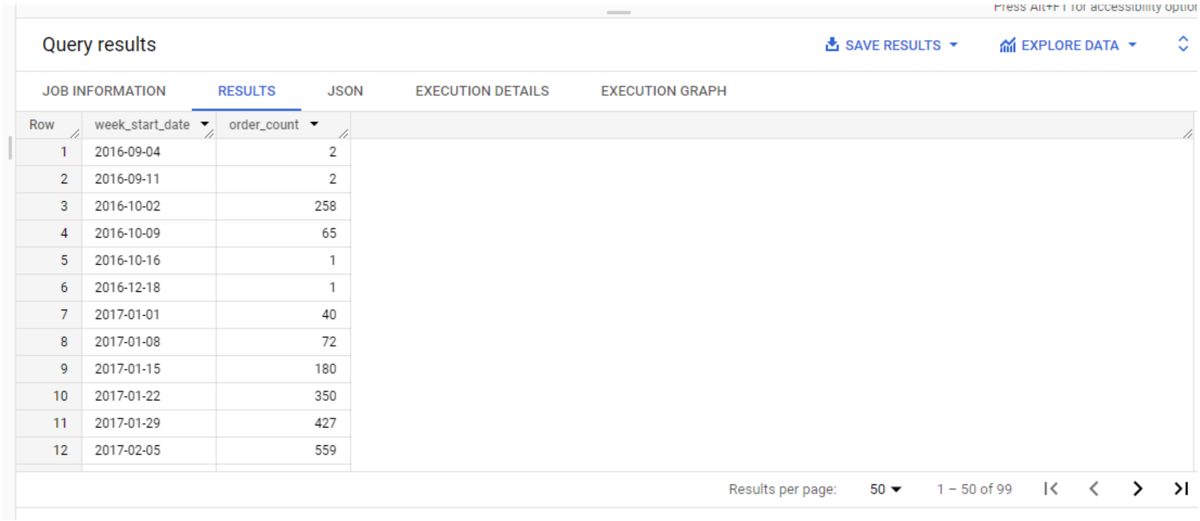
1. Promote and incentivize the use of preferred payment types to increase customer satisfaction and streamline payment processes.
2. Analyze the reasons behind fluctuations in the number of orders for different payment types and take appropriate measures to address any issues or improve customer experience.
3. Offer additional payment options based on customer preferences and market trends to expand the range of available choices.

INSIGHTS:

Number of orders per week :

SQL Querry ➔

```
SELECT
DATE_TRUNC(DATE(order_purchase_timestamp), WEEK) AS week_start_date,
COUNT(DISTINCT order_id) AS order_count
FROM `target-sql-project-391015.target_market.orders`
GROUP BY
week_start_date
ORDER BY
week_start_date
```



Row	week_start_date	order_count
1	2016-09-04	2
2	2016-09-11	2
3	2016-10-02	258
4	2016-10-09	65
5	2016-10-16	1
6	2016-12-18	1
7	2017-01-01	40
8	2017-01-08	72
9	2017-01-15	180
10	2017-01-22	350
11	2017-01-29	427
12	2017-02-05	559

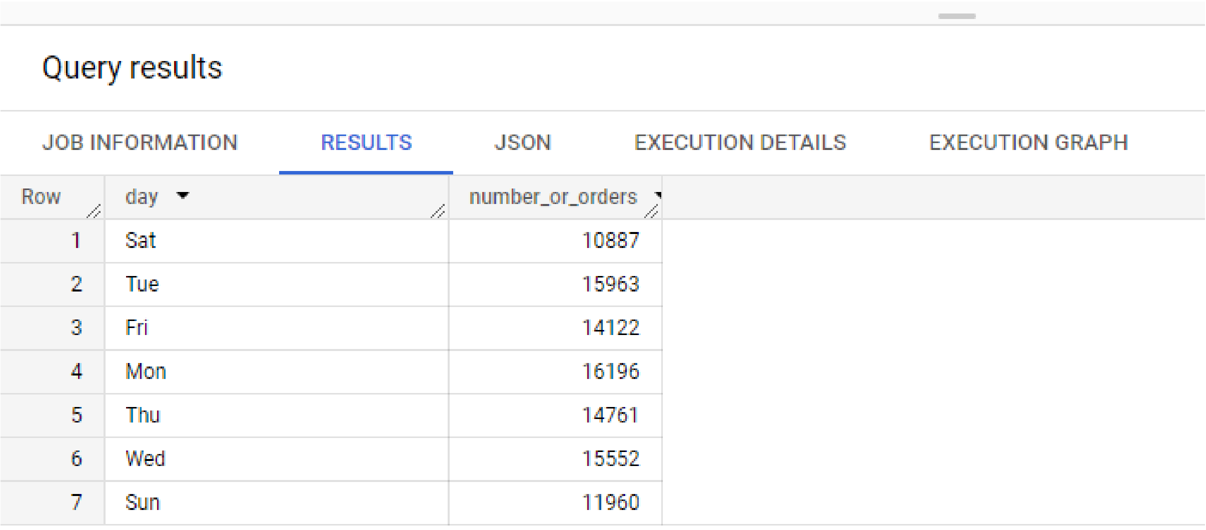
Results per page: 50 1 – 50 of 99

INSIGHTS:

Number of orders per Day:

SQL QUERRY→

```
SELECT
FORMAT_TIMESTAMP("%a", order_purchase_timestamp) as day,
count(order_id) as number_or_orders
from `target-sql-project-391015.target_market.orders`
group by day
```



Row	day	number_or_orders
1	Sat	10887
2	Tue	15963
3	Fri	14122
4	Mon	16196
5	Thu	14761
6	Wed	15552
7	Sun	11960

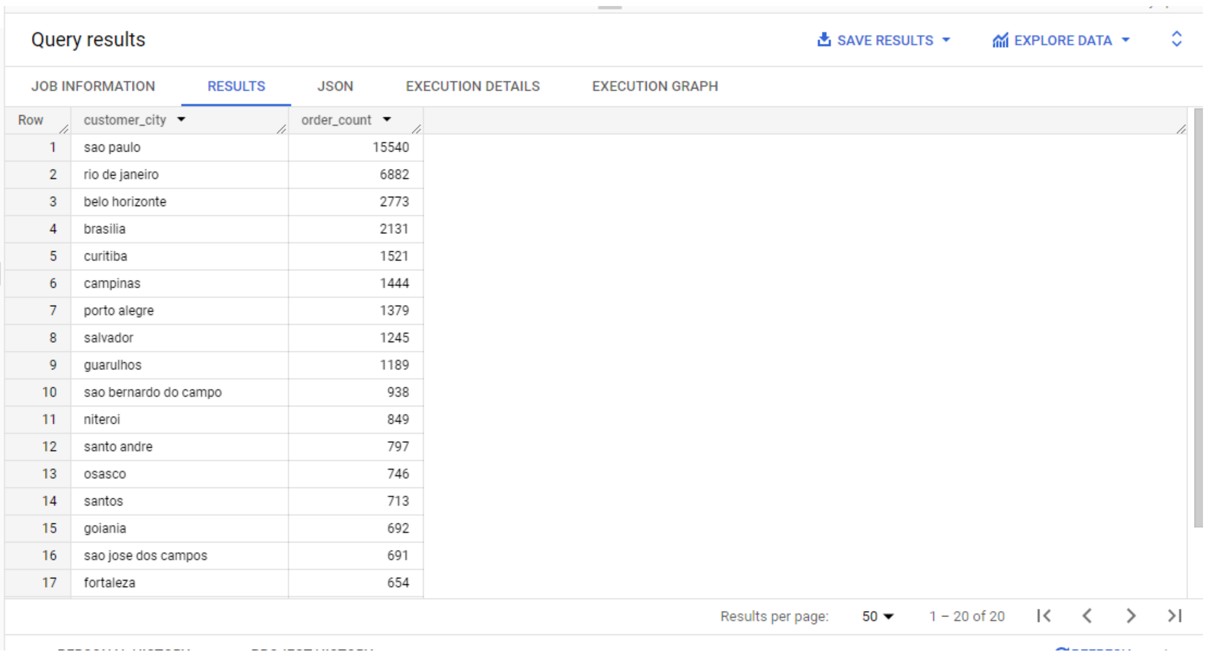
INSIGHTS:

TOP 20 CITIES where highest Number of Orders coming From:

To determine the overall revenue generated by Number of Customers From Each City.

SQL QUERRY →

```
SELECT customer_city, COUNT(order_id) AS order_count
FROM `target-sql-project-391015.target_market.orders` o
JOIN `target-sql-project-391015.target_market.customers` c ON o.customer_id = c.customer_id
GROUP BY customer_city
ORDER BY order_count DESC
LIMIT 20;
```



Row	customer_city	order_count
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
9	guarulhos	1189
10	sao bernardo do campo	938
11	niteroi	849
12	santo andre	797
13	osasco	746
14	santos	713
15	goiania	692
16	sao jose dos campos	691
17	fortaleza	654

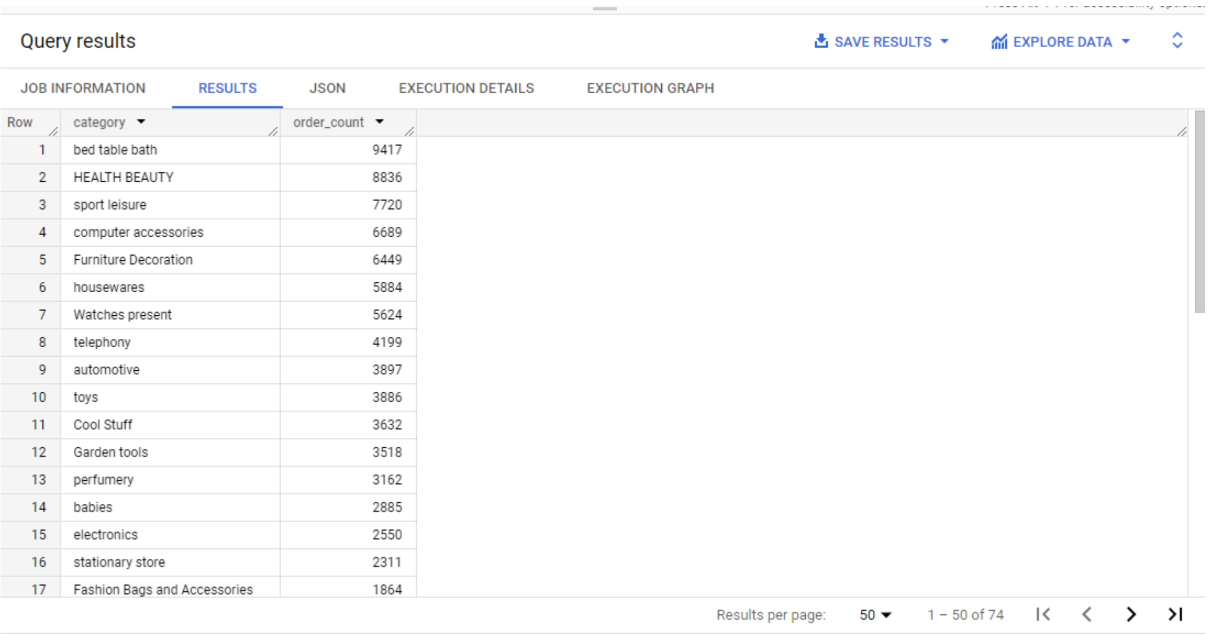
INSIGHTS:

Health and Beauty, Watches present, bed table bath, sport leisure, computer accessories, Furniture Decoration, housewares, Automotive are some of the top selling product categories. health and beauty products are top selling having highest orders. PCs and Musical Instruments category have relatively less number of products , but contributes in a high revenue.

Top selling product categories and the number of orders placed per category:

SQL Query :

```
SELECT
  p.product_category AS category,
  COUNT(DISTINCT o.order_id) AS order_count
FROM
  `target-sql-project-391015.target_market.orders` AS o
JOIN
  `target-sql-project-391015.target_market.order_items` AS oi ON o.order_id = oi.order_id
JOIN
  `target-sql-project-391015.target_market.products` AS p ON oi.product_id = p.product_id
GROUP BY
  category
ORDER BY
  order_count DESC;
```



Row	category	order_count
1	bed table bath	9417
2	HEALTH BEAUTY	8836
3	sport leisure	7720
4	computer accessories	6689
5	Furniture Decoration	6449
6	housewares	5884
7	Watches present	5624
8	telephony	4199
9	automotive	3897
10	toys	3886
11	Cool Stuff	3632
12	Garden tools	3518
13	perfumery	3162
14	babies	2885
15	electronics	2550
16	stationary store	2311
17	Fashion Bags and Accessories	1864

Insights and Recommendations :

- We have 99,441 customers of data available.
- We have 96096 number of Unique Customers ids.
- 14994 different locations of customers
- Customers are from different 4119 cities and 27 states from Brazil.
- total 99441 customers are there in given data.
- from total 99441 orders , 1107 are shipped ,625 were canceled, 96478 are delivered.
- Total 3095 sellers who are from 611 different cities and 23 states in Brazil and from 2246 different areas as per zip-code data.
- São Paulo state has the highest numbers of sellers in country.

Analysis of sales and revenue as per time :

- ➢ Time period for which the data is given is 25 months.
- ➢ Tuesday,Monday and Wednesdays have a relatively higher number of orders.

Customer_purchasing Behavior:

- customers are purchasing during moring 8am to late evening 11pm.
- afternoon and evening orders are very high , compare to morning , and night time.

Recommendations :

- Top selling items are between 10-100 dollars,introducing new different more products from top selling categories can increase revenue more.
- It was observed an increasing trend in revenue and orders over time , yet during october and january sales are decreasing probably after Festival Sales. Introducing possible discount on not so running product can help sell more products during those low going months.