Business Case: Target SQL

This business case has data of 100k orders from the year 2016 to 2018 made at Target, Brazil. It is America's leading retailer business chain.

Data is available in 8 tables, which gives you information about the orders from different sections like order, details of the payment, time and location of the order, customers who made the purchases, items in the order, details of the product, information about the seller of the products, reviews of the order etc.

Analysis

1. Initial exploration of the dataset

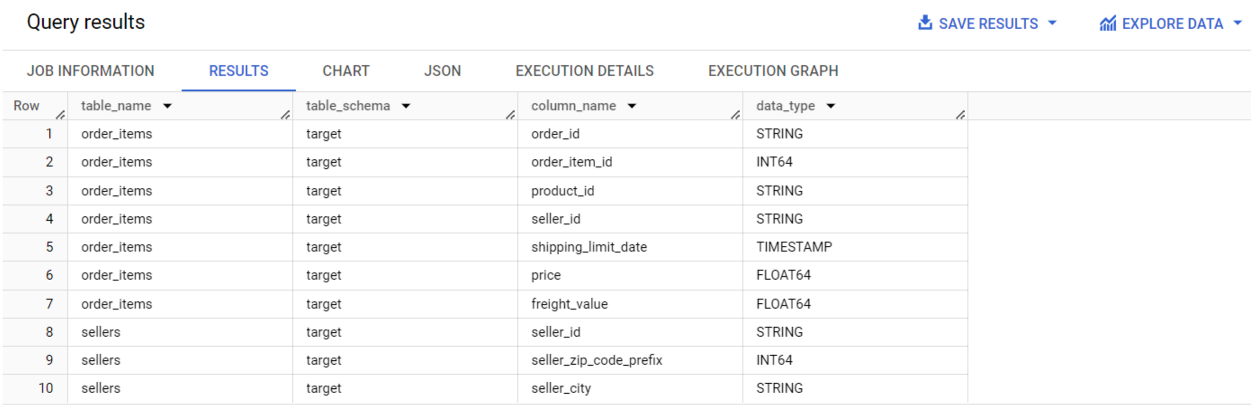
Here usual exploratory analysis and checking the structure and characteristics of the dataset is done.

1.1 Show all the table and all the columns present in each table along with its data type.

Query:

```
 SELECT
   table_schema,
   table_name,
   column_name,
    data_type
FROM
   `target`.INFORMATION_SCHEMA.COLUMNS;
```

RESULT:



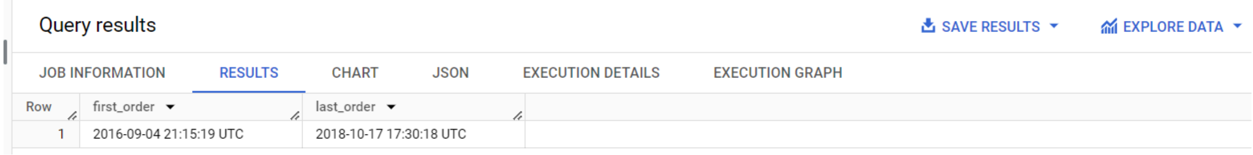
Row	table_name	table_schema	column_name	data_type
1	order_items	target	order_id	STRING
2	order_items	target	order_item_id	INT64
3	order_items	target	product_id	STRING
4	order_items	target	seller_id	STRING
5	order_items	target	shipping_limit_date	TIMESTAMP
6	order_items	target	price	FLOAT64
7	order_items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

1.2 Get the time range between which the orders were placed.

Query:

```
 SELECT
    MIN(order_purchase_timestamp) AS first_order,
    MAX(order_purchase_timestamp) AS last_order
FROM
    `target`.orders;
```

RESULT:



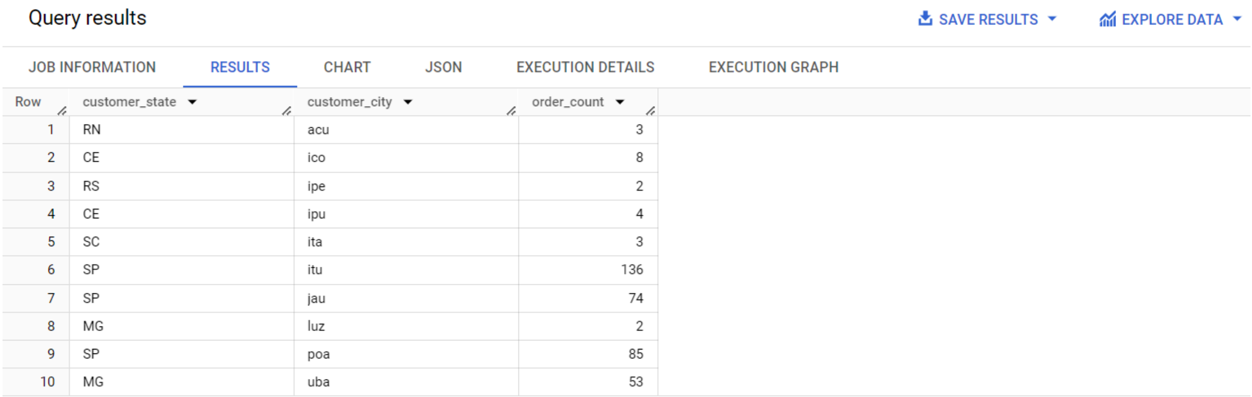
Row	first_order	last_order
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

1.3 Count the Cities & States of customers who ordered during the given period.

Query:

```
 SELECT
    customer_state,
    customer_city,
    COUNT * AS order_count
FROM
    `target`.customers
GROUP BY
    customer_state, customer_city;
```

RESULT:



Row	customer_state	customer_city	order_count
1	RN	acu	3
2	CE	ico	8
3	RS	ipe	2
4	CE	ipu	4
5	SC	ita	3
6	SP	itu	136
7	SP	jau	74
8	MG	luz	2
9	SP	poa	85
10	MG	uba	53

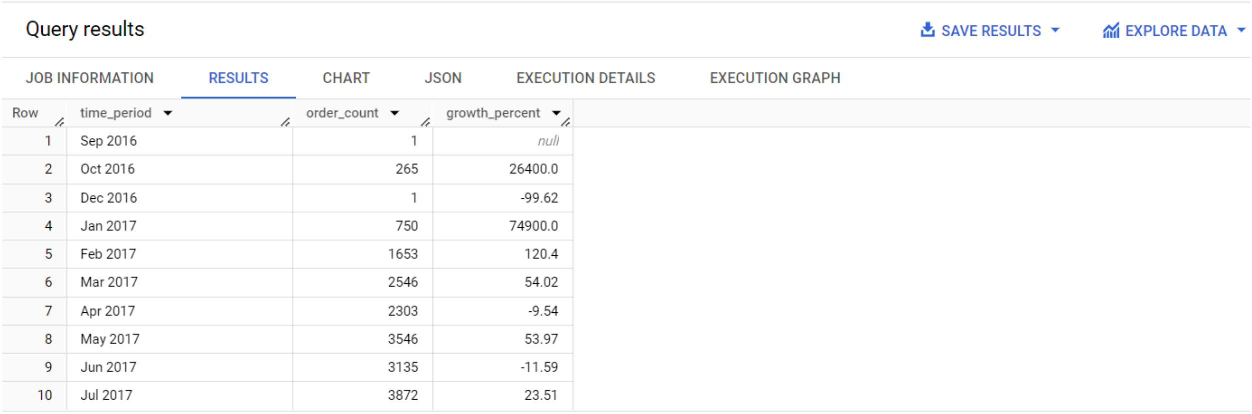
2 .In- depth exploration of dataset

2.1 Is there a going trend in the number of orders placed over the past years ?

Query:

```
SELECT
  time_period,
  order_count,
  ROUND((((order_count - LAG(order_count) OVER(ORDER BY t1.YEAR, t1.month)) /
LAG(order_count) OVER(ORDER BY t1.YEAR, t1.month))* 100), 2) AS growth_percent
FROM (
  SELECT
    EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
    EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
    FORMAT_DATE('%b %Y', DATE(ORDER_PURCHASE_TIMESTAMP)) AS time_period,
    COUNT(order_id) AS order_count
  FROM
    `target.orders`
  WHERE
    order_status = "delivered"
  GROUP BY
    month, year, time_period) t1
ORDER BY
  year, month;
```

RESULT:



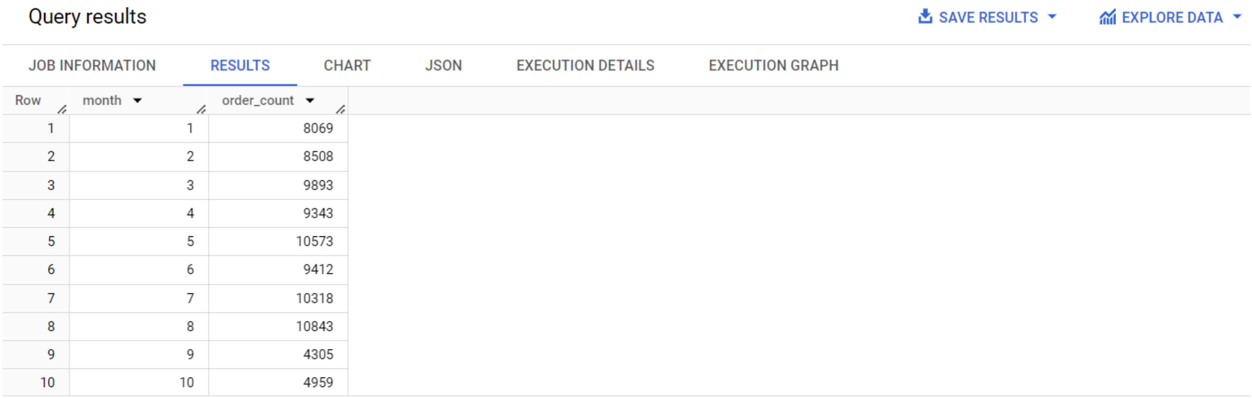
Row	time_period	order_count	growth_percent
1	Sep 2016	1	null
2	Oct 2016	265	26400.0
3	Dec 2016	1	-99.62
4	Jan 2017	750	74900.0
5	Feb 2017	1653	120.4
6	Mar 2017	2546	54.02
7	Apr 2017	2303	-9.54
8	May 2017	3546	53.97
9	Jun 2017	3135	-11.59
10	Jul 2017	3872	23.51

2.2 Can we some some kind of monthly seasonality in terms of the number of orders being placed ?

Query:

```
SELECT
    EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
    COUNT(*) AS order_count
FROM
    `target`.orders
GROUP BY
    month
ORDER BY
    Month;
```

RESULT :



Row	month	order_count
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

Can we see some seasonality with peaks at specific months?
No, as for some months data is showing orders of only 1 or 2 records and also there is huge spike in orders is seen in different months so we cannot comment on seasonality.
Overall business is in uptrend and sharp spike in orders in seen MoM basis.

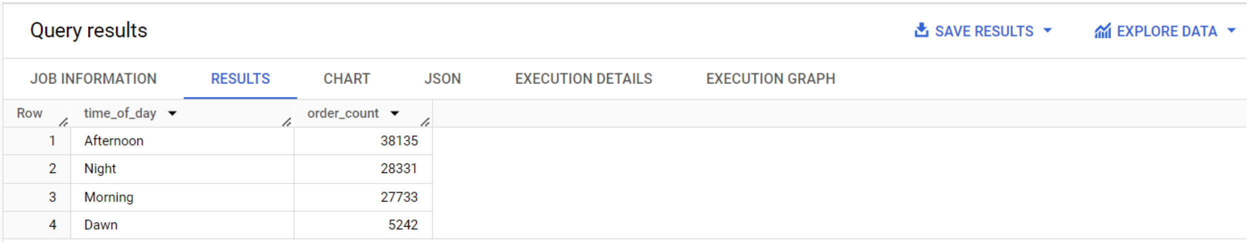
2.3 During what time of the day, do the Brazilian customers mostly place their orders ? (Dawn, Morning,Afternoon or Night)

Query:

```
SELECT
    CASE
        WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN 'Dawn'
        WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 7 AND 12 THEN
'Morning'
```

```
        WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN
'Afternoon'
        ELSE 'Night'
    END AS time_of_day,
    COUNT(*) AS order_count
FROM
    `target`.orders
GROUP BY
    time_of_day
ORDER BY
    order_count DESC;
```

RESULT :



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

Brazilian customers usually tend to buy in afternoon and night.

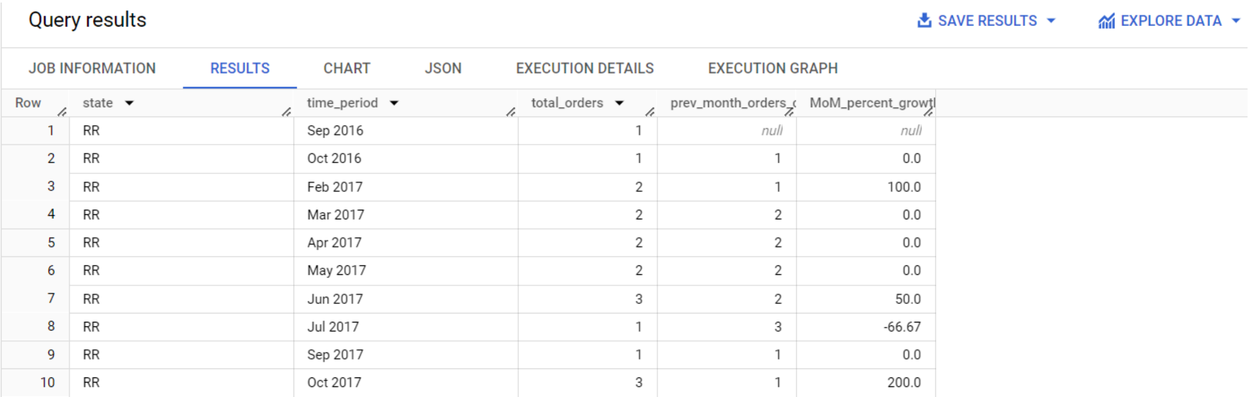
3. Evolution of E-commerce orders in the brazil region.

3.1 Get the month on month number of orders placed in each state.

Query:

```
 SELECT
  state , time_period , total_orders,
  LAG(total_orders) OVER(PARTITION BY state ORDER BY year, month ) AS
prev_month_orders_count,
  ROUND(((total_orders - LAG(total_orders) OVER(PARTITION BY state ORDER BY year,
month )) / LAG(total_orders) OVER(PARTITION BY state ORDER BY year, month))* 100,2) AS
MoM_percent_growth
FROM (
  SELECT
    state,
    time_period,
    year,
    month,
    COUNT(*) AS total_orders
  FROM (
    SELECT
      o.order_id, o.order_purchase_timestamp,
      EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
      EXTRACT(Month FROM order_purchase_timestamp) AS month,
      FORMAT_DATE('%b %Y', DATE(ORDER_PURCHASE_TIMESTAMP)) AS time_period,
      c.customer_state AS state
    FROM
      `target.orders` o
    JOIN
      `target.customers` c
    USING
      (customer_id)
    ORDER BY
       year, month) t1
  GROUP BY
    state, time_period, year, month) t2;
```

RESULT :



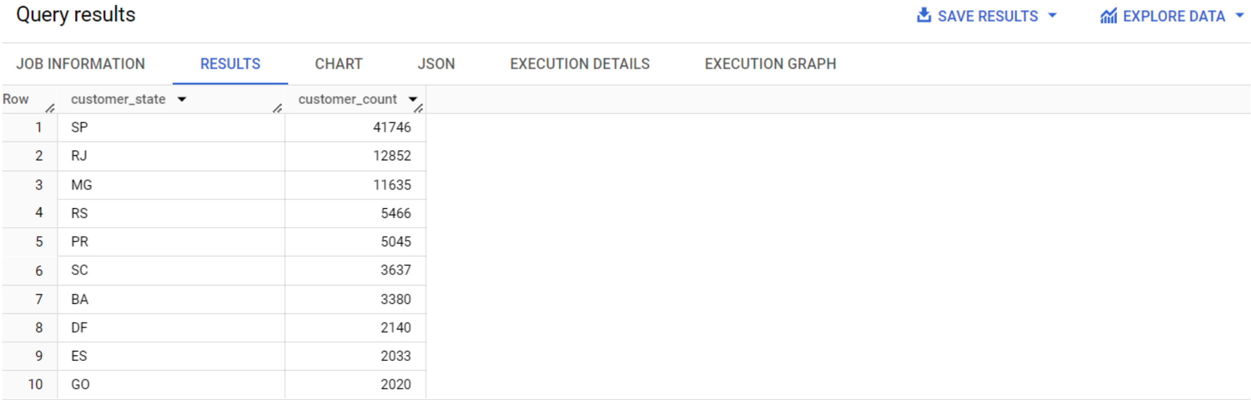
Row	state	time_period	total_orders	prev_month_orders_	MoM_percent_growt
1	RR	Sep 2016	1	null	null
2	RR	Oct 2016	1	1	0.0
3	RR	Feb 2017	2	1	100.0
4	RR	Mar 2017	2	2	0.0
5	RR	Apr 2017	2	2	0.0
6	RR	May 2017	2	2	0.0
7	RR	Jun 2017	3	2	50.0
8	RR	Jul 2017	1	3	-66.67
9	RR	Sep 2017	1	1	0.0
10	RR	Oct 2017	3	1	200.0

3.2 How are the customers distributed across all the state ?

Query:

```
SELECT
    customer_state,
    COUNT(*) AS customer_count
FROM
    `target`.customers
GROUP BY
    customer_state
ORDER BY
    customer_count DESC;
```

RESULT :



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

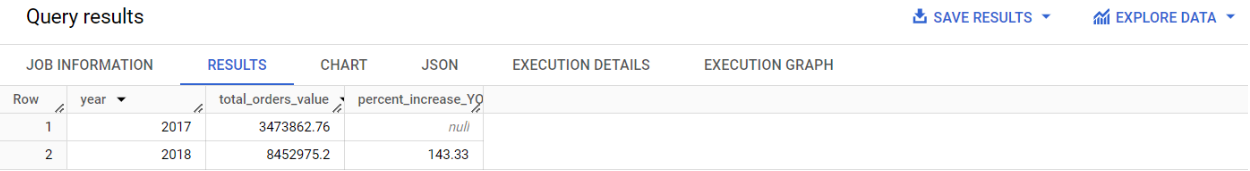
4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

4.1 Get the percentage increase in the cost of orders from year 2017 to 2018 (include months between January to August only). You can use the “payments_value” column in the payments table to get the cost of orders.

Query:

```
SELECT
  year,
  ROUND(SUM(payment_value), 2) AS total_orders_value,
  ROUND(
    (SUM(payment_value) - LAG(SUM(payment_value)) OVER(ORDER BY year)) /
NULLIF(LAG(SUM(payment_value)) OVER(ORDER BY year), 0) * 100, 2) AS
percent_increase_YOY
FROM (
  SELECT
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
    p.payment_value
  FROM
    `target.orders` o
  JOIN
    `target.payments` p ON o.order_id = p.order_id
  WHERE
    o.order_status = "delivered" AND
    EXTRACT(MONTH FROM o.order_purchase_timestamp) BETWEEN 1 AND 8
) t1
GROUP BY
  year
ORDER BY
  year;
```

RESULT :



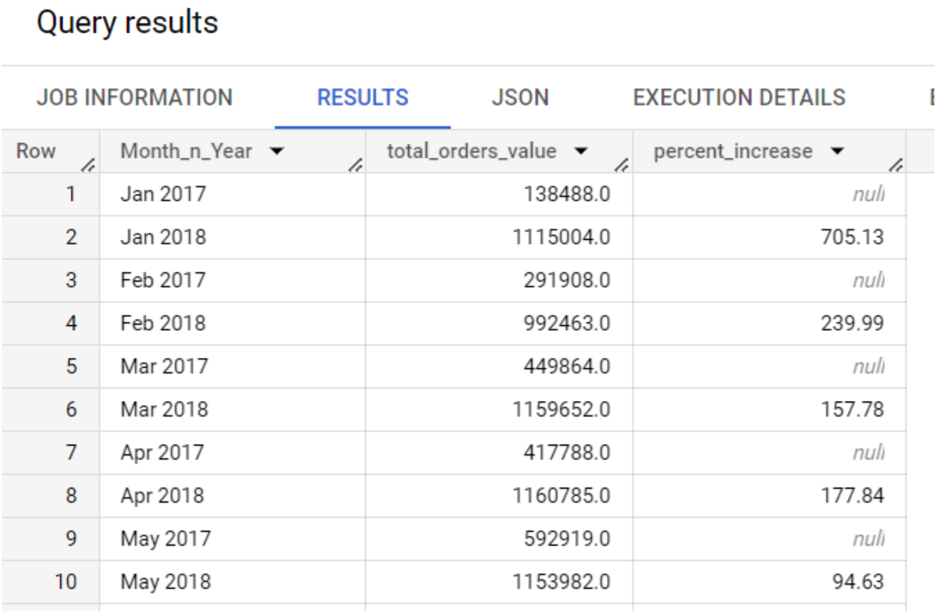
Row	year	total_orders_value	percent_increase_YO
1	2017	3473862.76	*null*
2	2018	8452975.2	143.33

4.2 Calculate the Total and Average value of order price for each state.

Query:

```
SELECT
    customer_state,
    ROUND(SUM(order_items.price), 2) AS total_price,
    ROUND(AVG(order_items.price), 2) AS avg_price
FROM
    `target`.order_items
JOIN
    orders ON order_items.order_id = orders.order_id
JOIN
    customers ON orders.customer_id = customers.customer_id
GROUP BY
    Customer_state;
```

RESULT :



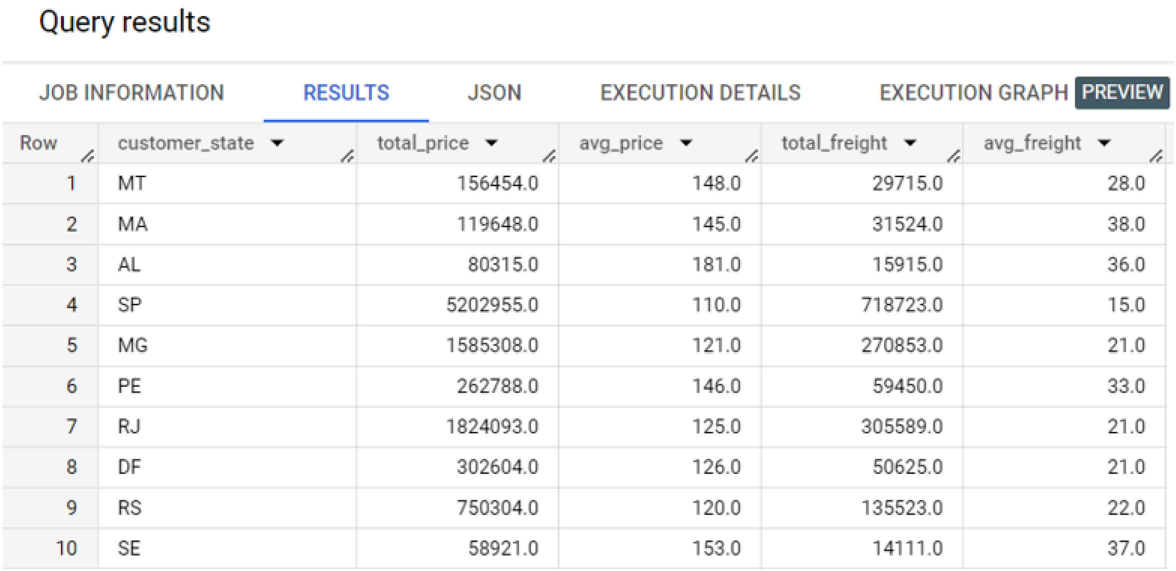
Row	Month_n_Year	total_orders_value	percent_increase
1	Jan 2017	138488.0	null
2	Jan 2018	1115004.0	705.13
3	Feb 2017	291908.0	null
4	Feb 2018	992463.0	239.99
5	Mar 2017	449864.0	null
6	Mar 2018	1159652.0	157.78
7	Apr 2017	417788.0	null
8	Apr 2018	1160785.0	177.84
9	May 2017	592919.0	null
10	May 2018	1153982.0	94.63

4.3 Calculate the Total and Average value of order freight for each state ?

Query:

```
SELECT
   customer_state,
   ROUND(SUM(order_items.freight_value), 2) AS total_freight,
   ROUND(AVG(order_items.freight_value), 2) AS avg_freight
FROM
   `target`.order_items
JOIN
   orders ON order_items.order_id = orders.order_id
JOIN
   customers ON orders.customer_id = customers.customer_id
GROUP BY
   Customer_state;
```

RESULT :



Row	customer_state	total_price	avg_price	total_freight	avg_freight
1	MT	156454.0	148.0	29715.0	28.0
2	MA	119648.0	145.0	31524.0	38.0
3	AL	80315.0	181.0	15915.0	36.0
4	SP	5202955.0	110.0	718723.0	15.0
5	MG	1585308.0	121.0	270853.0	21.0
6	PE	262788.0	146.0	59450.0	33.0
7	RJ	1824093.0	125.0	305589.0	21.0
8	DF	302604.0	126.0	50625.0	21.0
9	RS	750304.0	120.0	135523.0	22.0
10	SE	58921.0	153.0	14111.0	37.0

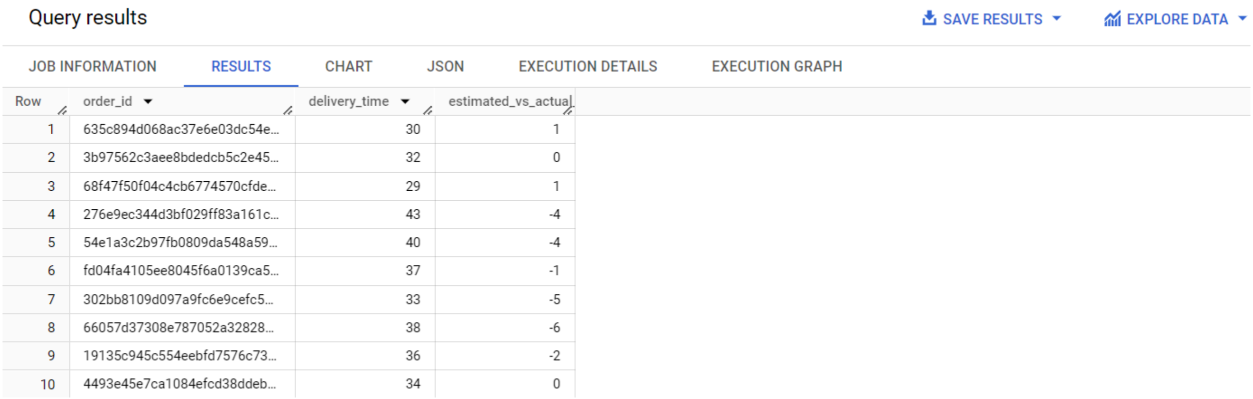
5. Analysis based on sales, freight and delivery time.

5.1 Calculate days between purchasing, delivering and estimated delivery.

Query:

```
SELECT
    order_id,
    TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
delivery_time,
    TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)
AS estimated_vs_actual_delivery
FROM
    `target`.orders
WHERE
    order_status = 'delivered';
```

RESULT :



Row	order_id	delivery_time	estimated_vs_actual
1	635c894d068ac37e6e03dc54e...	30	1
2	3b97562c3aee8bdedcb5c2e45...	32	0
3	68f47f50f04c4cb6774570cfde...	29	1
4	276e9ec344d3bf029ff83a161c...	43	-4
5	54e1a3c2b97fb0809da548a59...	40	-4
6	fd04fa4105ee8045f6a0139ca5...	37	-1
7	302bb8109d097a9fc6e9cefc5...	33	-5
8	66057d37308e787052a32828...	38	-6
9	19135c945c554eebfd7576c73...	36	-2
10	4493e45e7ca1084efcd38ddeb...	34	0

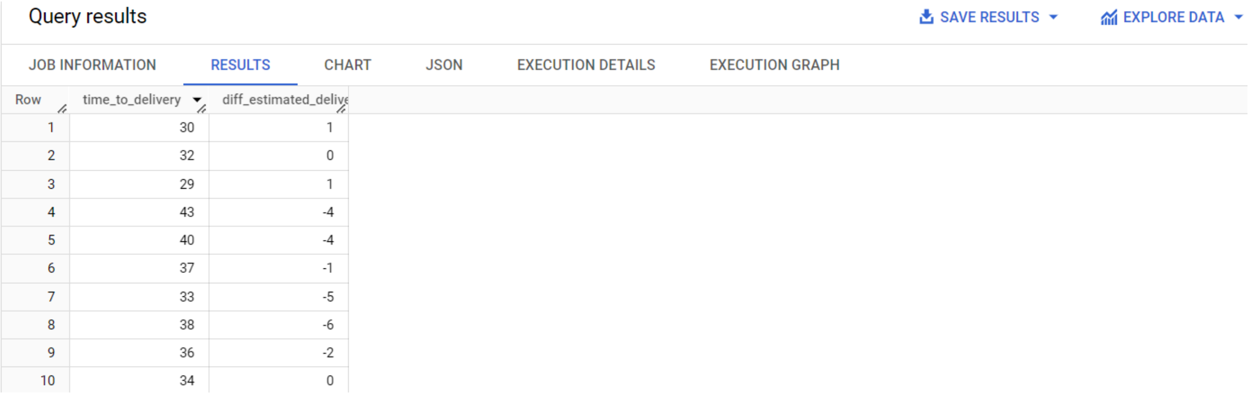
5.2 Calculate days between the estimated & actual delivery date of an order.

Query:

```
  SELECT

    TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_delivery,
    TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)
AS diff_estimated_delivery
FROM
    `target`.orders`
WHERE
    order_status = 'delivered';
```

RESULT :



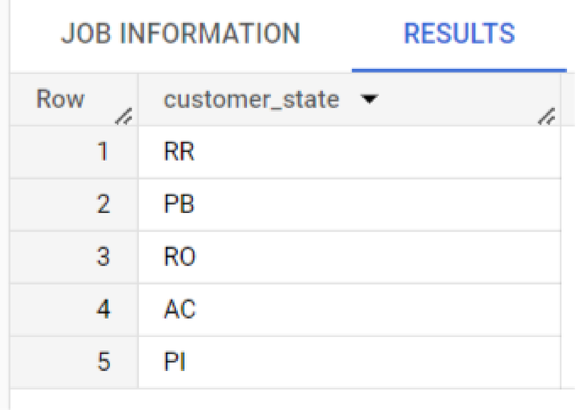
Row	time_to_delivery	diff_estimated_delive
1	30	1
2	32	0
3	29	1
4	43	-4
5	40	-4
6	37	-1
7	33	-5
8	38	-6
9	36	-2
10	34	0

5.3 Top 5 states with Highest & Lowest average freight value.

Query: (Highest Average Freight value)

```
SELECT
    customer_state,
    ROUND(AVG(order_items.freight_value), 2) AS avg_freight
FROM
    `target`.order_items
JOIN
    orders ON order_items.order_id = orders.order_id
JOIN
    customers ON orders.customer_id = customers.customer_id
GROUP BY
    customer_state
ORDER BY
    avg_freight DESC
LIMIT 5;
```

RESULT :



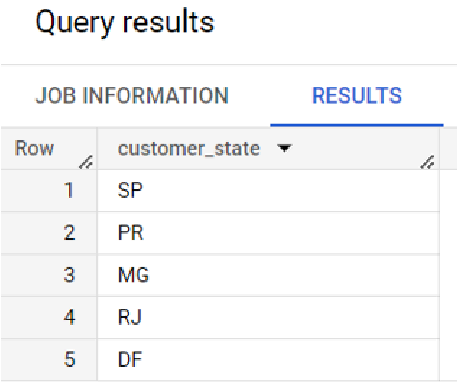
Row	customer_state
1	RR
2	PB
3	RO
4	AC
5	PI

5.3 Top 5 states with Highest & Lowest average freight value.

Query: (Lowest Average Freight value)

```
SELECT
    customer_state,
    ROUND(AVG(order_items.freight_value), 2) AS avg_freight
FROM
    order_items
JOIN
    orders ON order_items.order_id = orders.order_id
JOIN
    customers ON orders.customer_id = customers.customer_id
GROUP BY
    customer_state
ORDER BY
    avg_freight ASC
LIMIT 5;
```

RESULT :



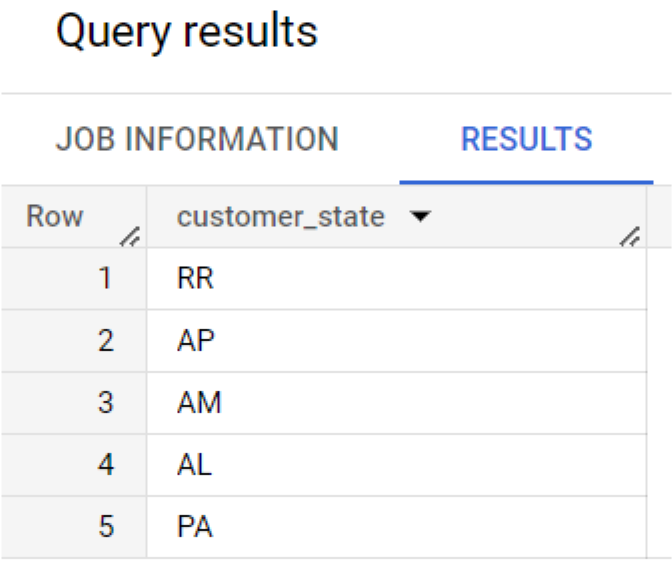
Row	customer_state
1	SP
2	PR
3	MG
4	RJ
5	DF

5.4 Top 5 states with highest and lowest average delivery time:

Query: (Highest average delivery time)

```
SELECT
    customer_state,
    ROUND(AVG(TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY)), 2) AS avg_delivery_time
FROM
    orders
JOIN
    customers ON orders.customer_id = customers.customer_id
WHERE
    order_status = 'delivered'
GROUP BY
    customer_state
ORDER BY
    avg_delivery_time DESC
LIMIT 5;
```

RESULT :

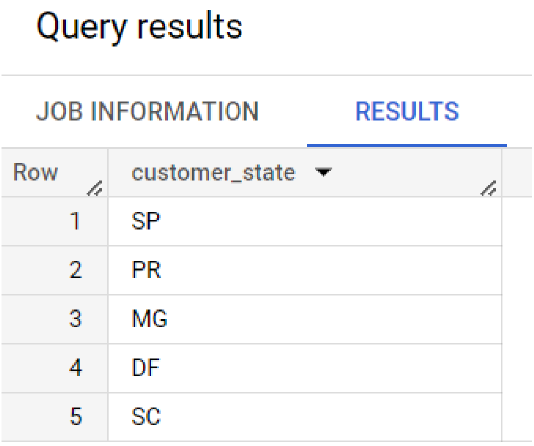


Row	customer_state
1	RR
2	AP
3	AM
4	AL
5	PA

5.4 Top 5 states with highest and lowest average delivery time:
 Query: (Lowest average delivery time)

```
 SELECT
    customer_state,
    ROUND(AVG(TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY)), 2) AS avg_delivery_time
FROM
    orders
JOIN
    customers ON orders.customer_id = customers.customer_id
WHERE
    order_status = 'delivered'
GROUP BY
    customer_state
ORDER BY
    avg_delivery_time ASC
LIMIT 5;
```

RESULT:



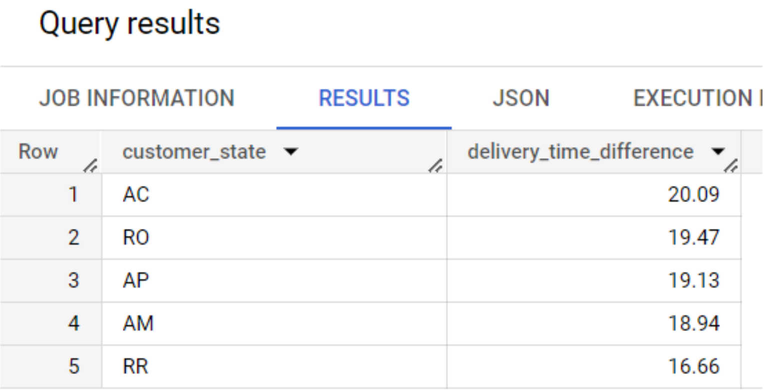
Row	customer_state
1	SP
2	PR
3	MG
4	DF
5	SC

5.5 Top 5 states with fastest and lowest delivery compared to the estimated date:

Query: (states with fastest delivery compared to the estimated date)

```
 SELECT
    customer_state,
    ROUND(AVG(TIMESTAMP_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY)), 2) AS avg_difference
FROM
    orders
JOIN
    customers ON orders.customer_id = customers.customer_id
WHERE
    order_status = 'delivered'
GROUP BY
    customer_state
ORDER BY
    avg_difference DESC
LIMIT 5;
```

RESULT:

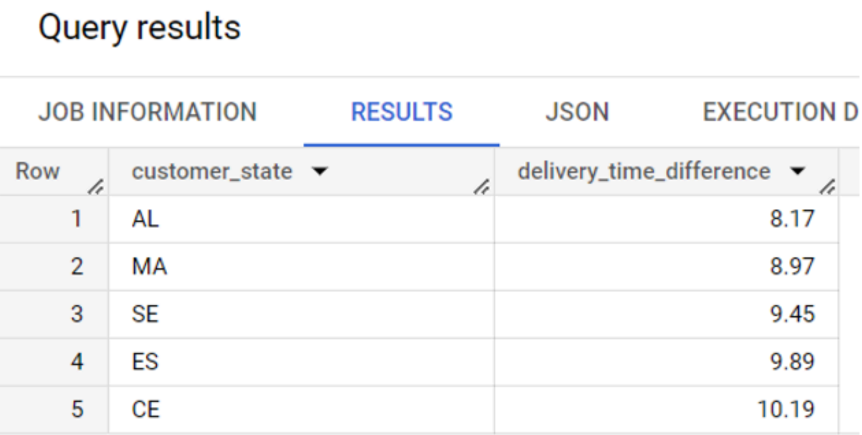


Row	customer_state	delivery_time_difference
1	AC	20.09
2	RO	19.47
3	AP	19.13
4	AM	18.94
5	RR	16.66

5.5 Top 5 states with fastest and lowest delivery compared to the estimated date:
Query: (states with lowest delivery compared to the estimated date)

```
SELECT
    customer_state,
    ROUND(AVG(TIMESTAMP_DIFF(order_estimated_delivery_date,
order_delivered_customer_date, DAY)), 2) AS avg_difference
FROM
    orders
JOIN
    customers ON orders.customer_id = customers.customer_id
WHERE
    order_status = 'delivered'
GROUP BY
    customer_state
ORDER BY
    avg_difference ASC
LIMIT 5;
```

RESULT:



Row	customer_state	delivery_time_difference
1	AL	8.17
2	MA	8.97
3	SE	9.45
4	ES	9.89
5	CE	10.19

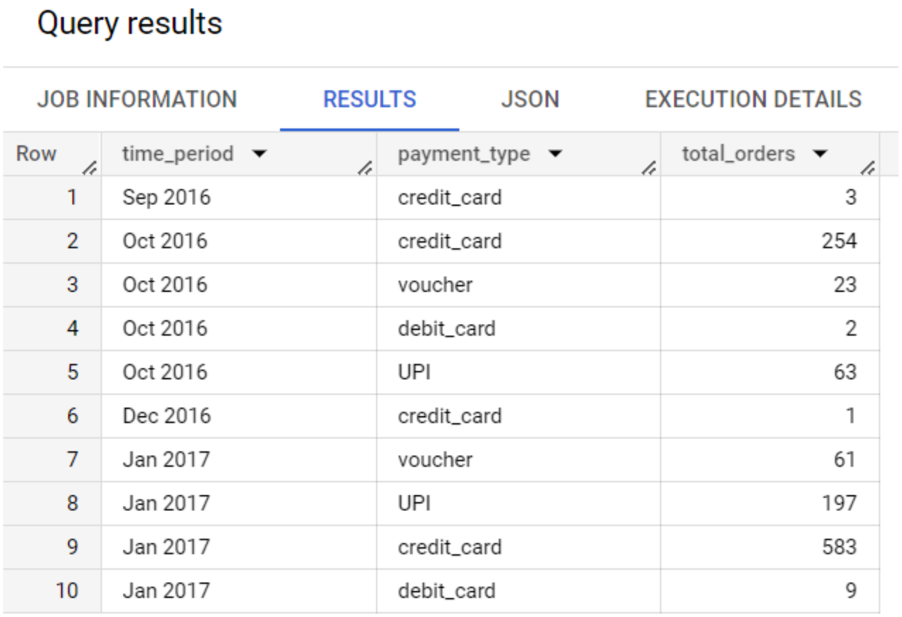
6. Payment type Analysis.

6.1 Month-on-Month count of orders by payment type:

Query:

```
 SELECT
    EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
    EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
    payment_type,
    COUNT(*) AS order_count
FROM
    `target`.orders
JOIN
    payments ON orders.order_id = payments.order_id
GROUP BY
    year, month, payment_type
ORDER BY
    year, month;
```

RESULT:



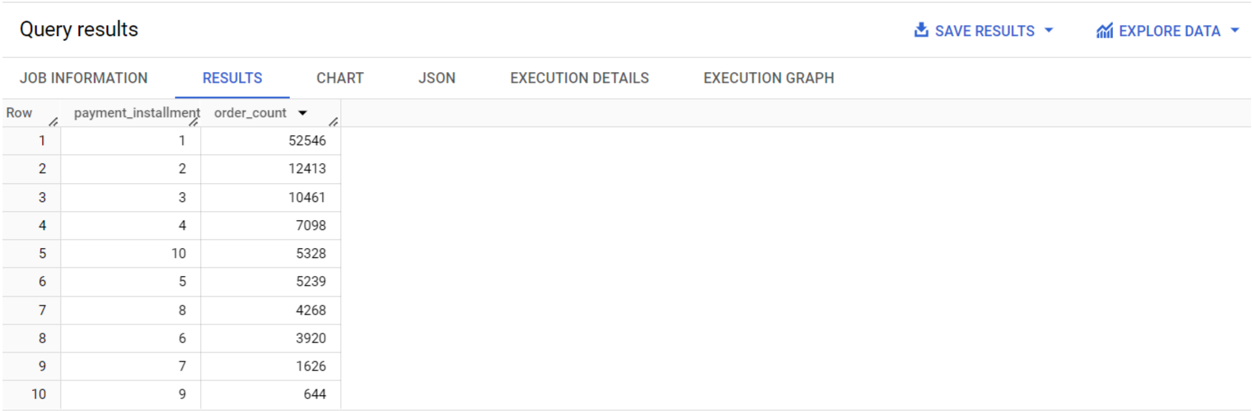
Row	time_period	payment_type	total_orders
1	Sep 2016	credit_card	3
2	Oct 2016	credit_card	254
3	Oct 2016	voucher	23
4	Oct 2016	debit_card	2
5	Oct 2016	UPI	63
6	Dec 2016	credit_card	1
7	Jan 2017	voucher	61
8	Jan 2017	UPI	197
9	Jan 2017	credit_card	583
10	Jan 2017	debit_card	9

6.2 Count of orders based on payment installments:

Query:

```
SELECT
    payment_installments,
    COUNT(*) AS order_count
FROM
    `target`.payments
GROUP BY
    payment_installments
ORDER BY
    order_count DESC;
```

RESULT:

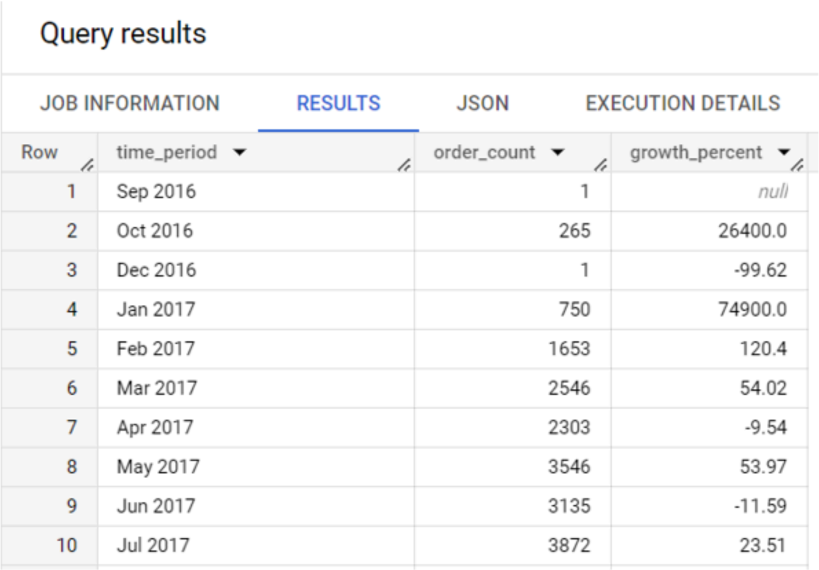


Row	payment_installment	order_count
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644

7. Actionable Insights.

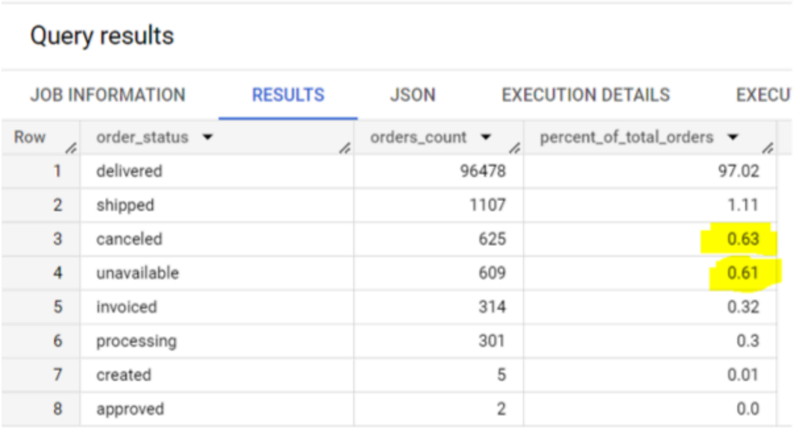
- Total of 609 orders were unavailable and 625 orders were canceled, comprising around 1.2% of total orders.
- Abrupt increase in order volume indicates rapid growth, suggesting that additional workforce, potentially on a contractual basis, may be required.
- States with high average delivery times may need improved logistics for better customer satisfaction

- We can see how the orders trajectory is showing very abrupt increase in orders volume with in very short time.
- Looking at overall trend, it is seen that business is picking up very fast in brazil so company has to be ready with extra workforce to avoid high risk , it can be consider hiring contractural employees.



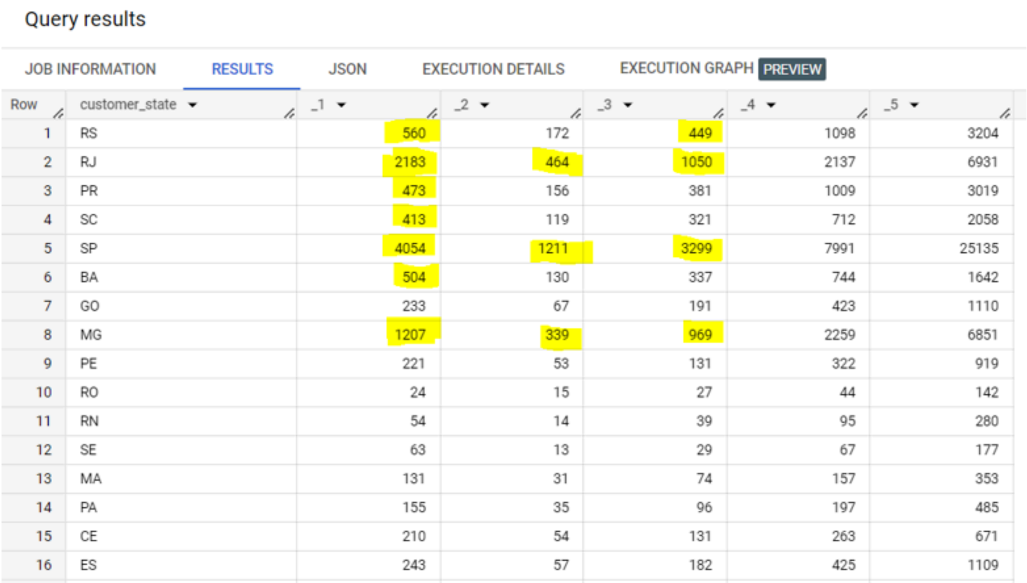
Row	time_period	order_count	growth_percent
1	Sep 2016	1	null
2	Oct 2016	265	26400.0
3	Dec 2016	1	-99.62
4	Jan 2017	750	74900.0
5	Feb 2017	1653	120.4
6	Mar 2017	2546	54.02
7	Apr 2017	2303	-9.54
8	May 2017	3546	53.97
9	Jun 2017	3135	-11.59
10	Jul 2017	3872	23.51

- Company received low rating for maximum orders in highlighted states.
- We need to study further about the reasons for customer dissatisfaction to such great extent in these states.



Row	order_status	orders_count	percent_of_total_orders
1	delivered	96478	97.02
2	shipped	1107	1.11
3	canceled	625	0.63
4	unavailable	609	0.61
5	invoiced	314	0.32
6	processing	301	0.3
7	created	5	0.01
8	approved	2	0.0

- This is the query for counting the number of rating for each state



Query results

Row	customer_state	_1	_2	_3	_4	_5
1	RS	560	172	449	1098	3204
2	RJ	2183	464	1050	2137	6931
3	PR	473	156	381	1009	3019
4	SC	413	119	321	712	2058
5	SP	4054	1211	3299	7991	25135
6	BA	504	130	337	744	1642
7	GO	233	67	191	423	1110
8	MG	1207	339	969	2259	6851
9	PE	221	53	131	322	919
10	RO	24	15	27	44	142
11	RN	54	14	39	95	280
12	SE	63	13	29	67	177
13	MA	131	31	74	157	353
14	PA	155	35	96	197	485
15	CE	210	54	131	263	671
16	ES	243	57	182	425	1109

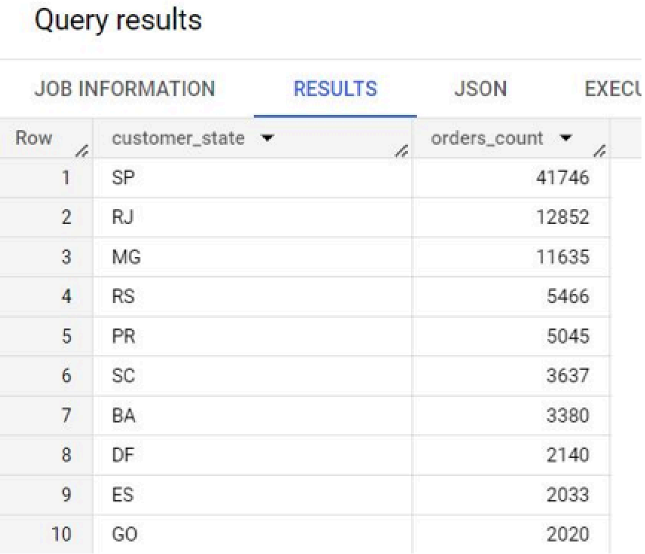
```
SELECT *

FROM (
SELECT
c.customer_state,
orv_review_score
FROM
`target`.order_reviews orv
JOIN
 `target`.orders o
 USING
  (order_id)
JOIN
`target`.customers c
USING
(customer_id) PIVOT(COUNT(*) FOR review_score IN (1,2,3,4,5)):
```

8.Recommendations.

1.As Brazilian customers usually tend to buy in afternoon and night, we can increase staff in during this time frame in order to manage the customer's requests and services better during this time by reducing workforce of morning and dawn.

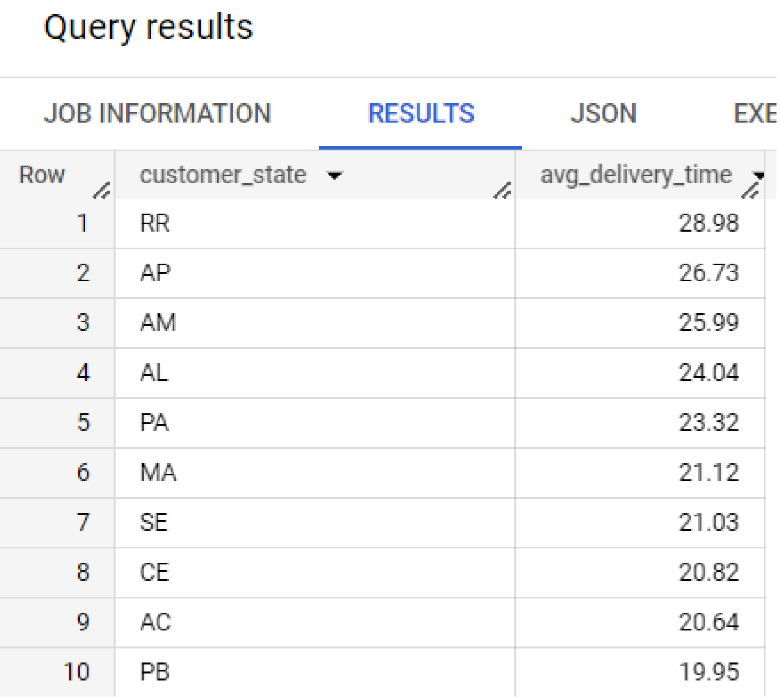
2.We can see, only 3 states contribute for maximum volume, and rest of the state need to be focussed on improving the business.



Row	customer_state	orders_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

3.Averge delivery time is high for most of those states from where company is receiving quite less volume of orders, detailed study is needed further for checking the other reasons behind such low volume of orders from majority of states,huge delivery time can be one of the reasons and need to work on it.

Stats with highest average delivery time -



Query results

Row	customer_state	avg_delivery_time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32
6	MA	21.12
7	SE	21.03
8	CE	20.82
9	AC	20.64
10	PB	19.95