TARGET – BUISNESS CASE STUDY

➢ Company Description :

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analysing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

➢ Dataset:

Dataset: https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb

The data is available in 8 csv files:

1. customers.csv
2. sellers.csv
3. order_items.csv
4. geolocation.csv
5. payments.csv
6. reviews.csv
7. orders.csv
8. products.csv

The column description for these csv files is given below.

The **customers.csv** contain following features:

Features	Description
customer_id	ID of the consumer who made the purchase
customer_unique_id	Unique ID of the consumer

customer_zip_code_prefix	Zip Code of consumer’s location
customer_city	Name of the City from where order is made
customer_state	State Code from where order is made (Eg. são paulo - SP)

The **sellers.csv** contains following features:

Features	Description
seller_id	Unique ID of the seller registered
seller_zip_code_prefix	Zip Code of the seller’s location
seller_city	Name of the City of the seller
seller_state	State Code (Eg. são paulo - SP)

The **order_items.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
order_item_id	A Unique ID given to each item ordered in the order
product_id	A Unique ID given to each product available on the site
seller_id	Unique ID of the seller registered in Target
shipping_limit_date	The date before which the ordered product must be shipped
price	Actual price of the products ordered
freight_value	Price rate at which a product is delivered from one point to another

The **geolocations.csv** contain following features:

Features	Description
geolocation_zip_code_prefix	First 5 digits of Zip Code
geolocation_lat	Latitude
geolocation_lng	Longitude
geolocation_city	City
geolocation_state	State

The **payments.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
payment_sequential	Sequences of the payments made in case of EMI
payment_type	Mode of payment used (Eg. Credit Card)
payment_installments	Number of installments in case of EMI purchase
payment_value	Total amount paid for the purchase order

The **orders.csv** contain following features:

Features	Description
order_id	A Unique ID of order made by the consumers
customer_id	ID of the consumer who made the purchase
order_status	Status of the order made i.e. delivered, shipped, etc.

order_purchase_timestamp	Timestamp of the purchase
order_delivered_carrier_date	Delivery date at which carrier made the delivery
order_delivered_customer_date	Date at which customer got the product
order_estimated_delivery_date	Estimated delivery date of the products

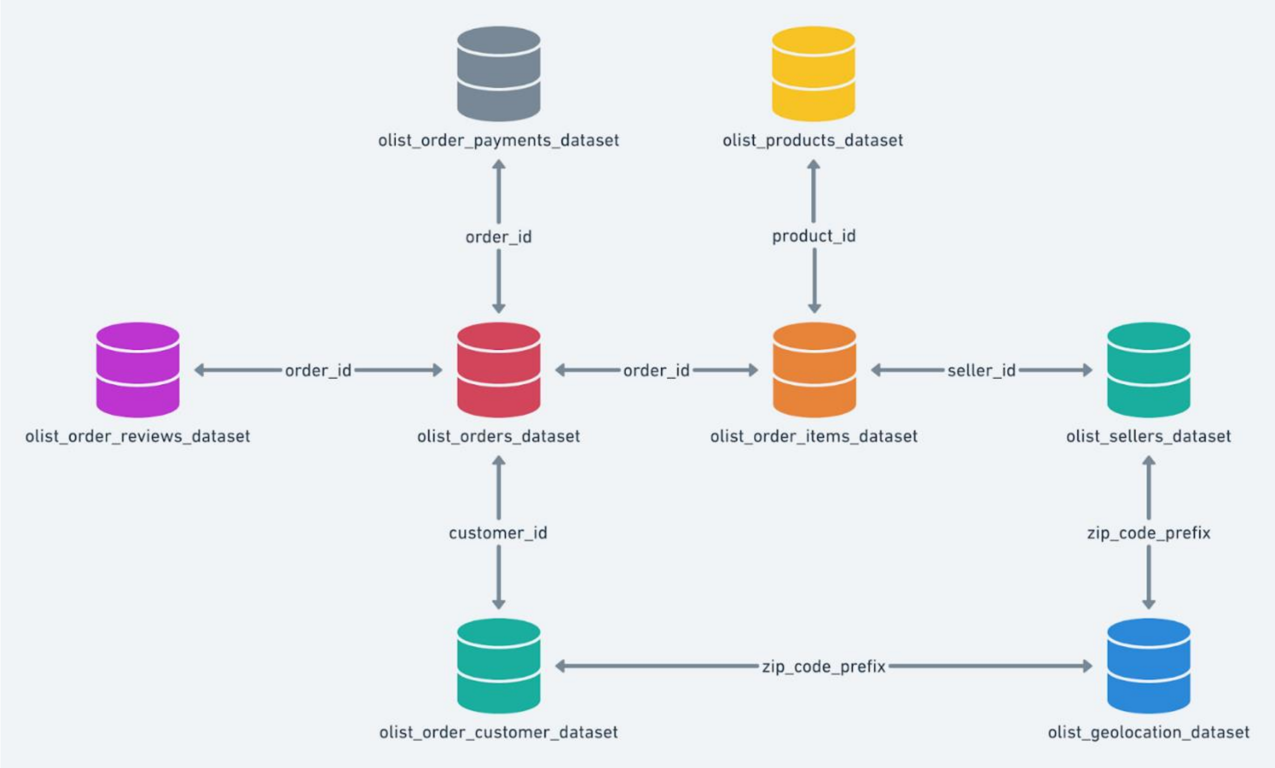
The **reviews.csv** contain following features:

Features	**Description**
review_id	ID of the review given on the product ordered by the order id
order_id	A Unique ID of order made by the consumers
review_score	Review score given by the customer for each order on a scale of 1-5
review_comment_title	Title of the review
review_comment_message	Review comments posted by the consumer for each order
review_creation_date	Timestamp of the review when it is created
review_answer_timestamp	Timestamp of the review answered

The **products.csv** contain following features:

Features	**Description**
product_id	A Unique identifier for the proposed project.
product_category_name	Name of the product category
product_name_lenght	Length of the string which specifies the name given to the products ordered
product_description_lenght	Length of the description written for each product ordered on the site
product_photos_qty	Number of photos of each product ordered available on the shopping portal
product_weight_g	Weight of the products ordered in grams
product_length_cm	Length of the products ordered in centimeters
product_height_cm	Height of the products ordered in centimeters
product_width_cm	Width of the product ordered in centimeters

Dataset schema:

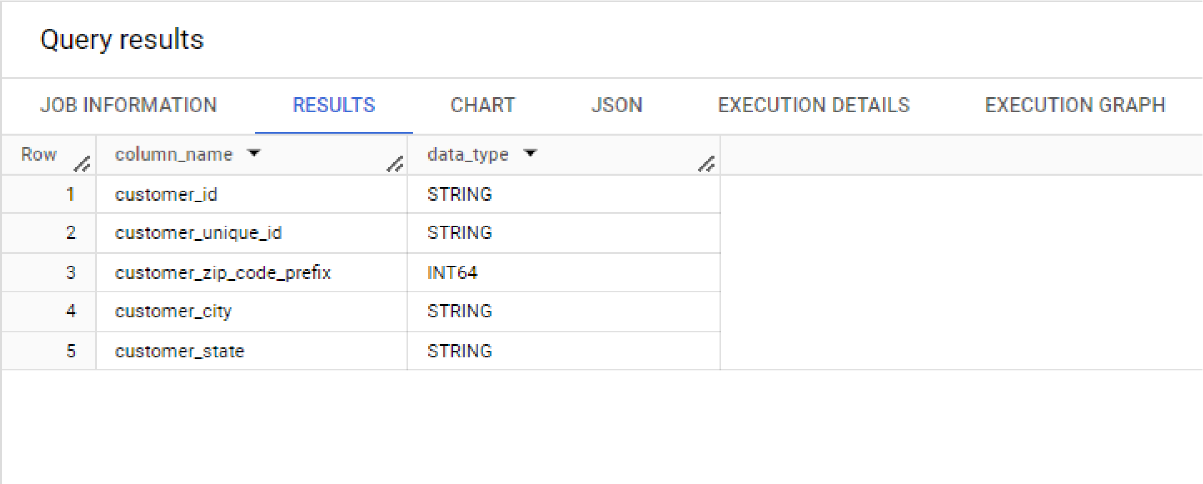


➢ Questions:

❖ Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

1. Data type of all columns in the "customers" table.

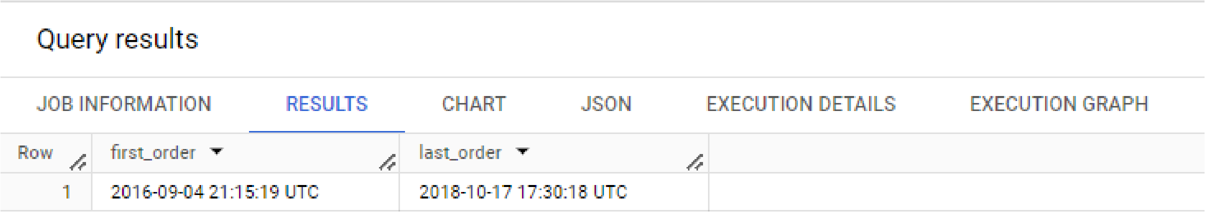
```
SELECT
  column_name,
  data_type
FROM
  `Data_Target.INFORMATION_SCHEMA.COLUMNS`
WHERE
  table_name = 'customers';
```



Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

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

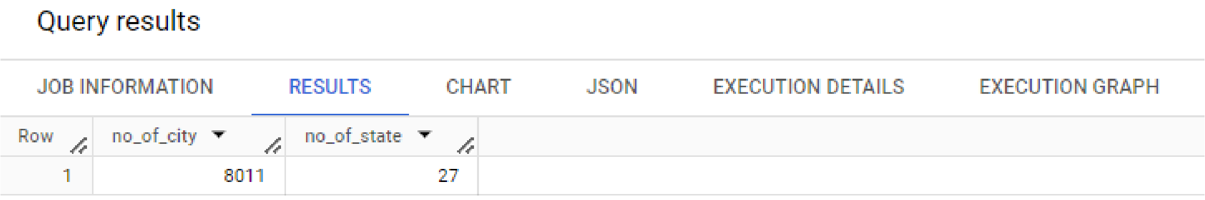
```
SELECT
  MIN(order_purchase_timestamp) AS first_order,
  MAX(order_purchase_timestamp) AS last_order
FROM
  `Data_Target.orders`;
```



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

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

```
SELECT
  COUNT(DISTINCT geolocation_city) AS no_of_city,
  COUNT(DISTINCT geolocation_state) AS no_of_state
FROM
  Data_Target.geolocation;
```

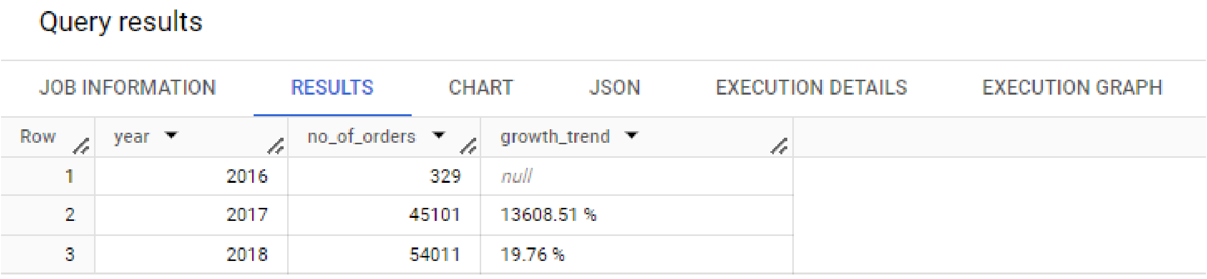


Row	no_of_city	no_of_state
1	8011	27

❖ In-depth Exploration:

1. Is there a growing trend in the no. of orders placed over the past years?

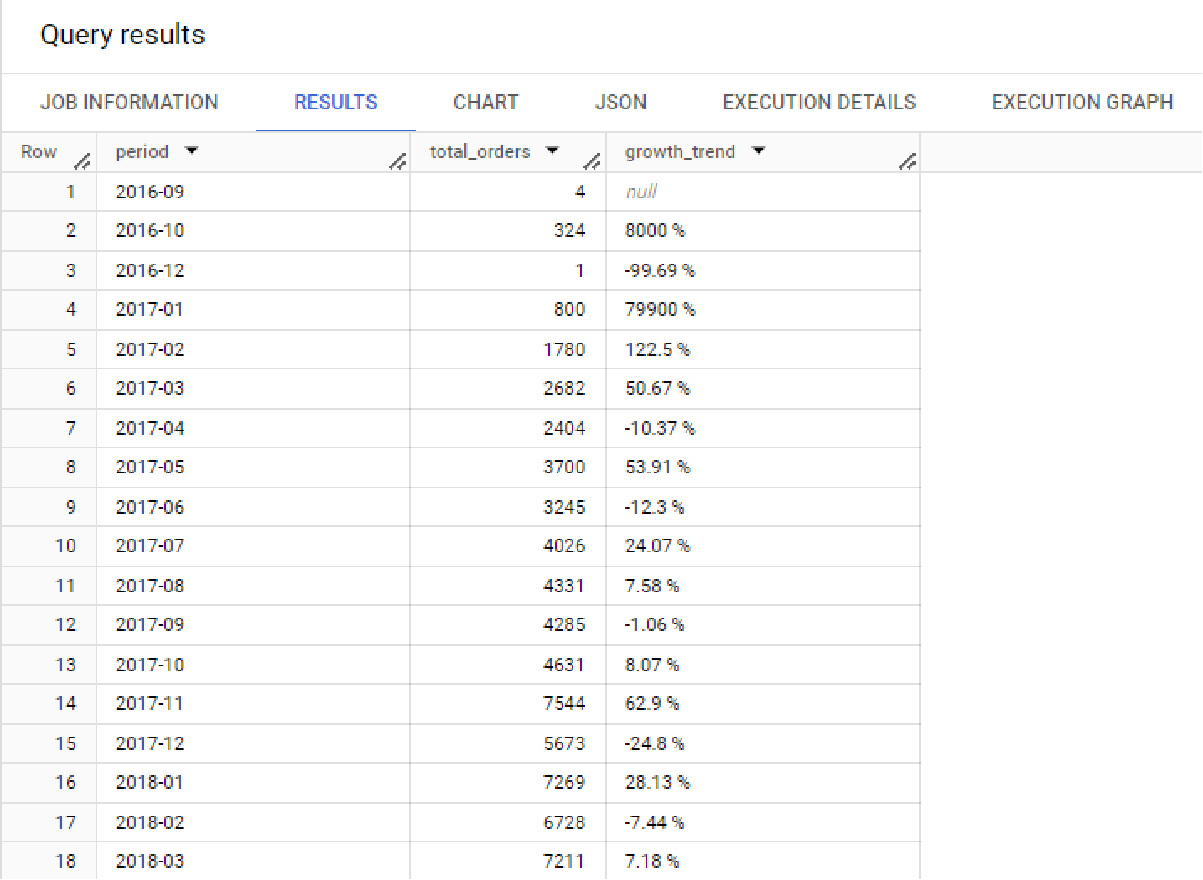
```
SELECT
  year,
  no_of_orders,
  CONCAT(
    ROUND(((no_of_orders - LAG(no_of_orders) OVER (ORDER BY year)) / LAG(no_of_orders)
OVER (ORDER BY year)) * 100, 2),
    ' %'
  ) AS growth_trend
FROM (
  SELECT
    EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
    COUNT(*) AS no_of_orders
  FROM
    `Data_Target.orders`
  GROUP BY
    year
  ORDER BY
    year
) AS yearly_orders
ORDER BY
  year;
```


Query results

Row	year	no_of_orders	growth_trend
1	2016	329	null
2	2017	45101	13608.51 %
3	2018	54011	19.76 %

2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

```
SELECT
  period,
  total_orders,
  CONCAT(
    ROUND(((total_orders - LAG(total_orders) OVER (ORDER BY period)) / LAG(total_orders)
OVER (ORDER BY period)) * 100, 2),
    ' %'
  ) AS growth_trend
FROM (
        SELECT
            FORMAT_TIMESTAMP('%Y-%m', order_purchase_timestamp) AS period,
            COUNT(*) AS total_orders
        FROM
            `Data_Target.orders`
        GROUP BY
            period
        ORDER BY
            period
    ) AS monthly_orders
    ORDER BY
        period;
```

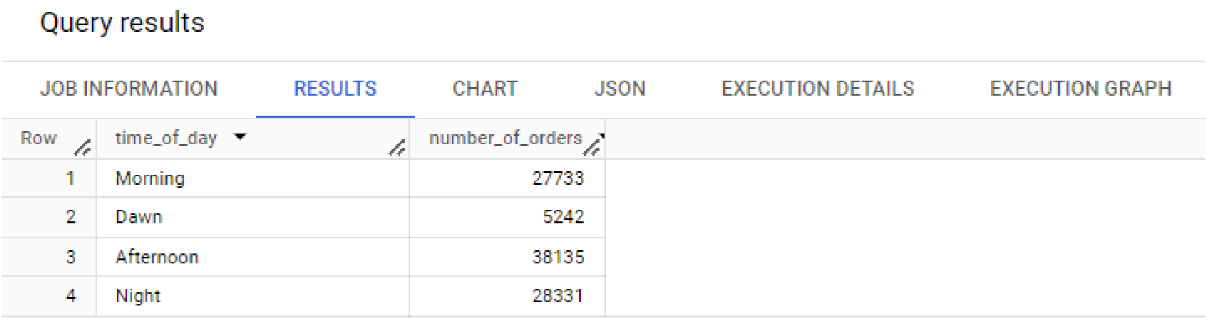


Row	period	total_orders	growth_trend
1	2016-09	4	null
2	2016-10	324	8000 %
3	2016-12	1	-99.69 %
4	2017-01	800	79900 %
5	2017-02	1780	122.5 %
6	2017-03	2682	50.67 %
7	2017-04	2404	-10.37 %
8	2017-05	3700	53.91 %
9	2017-06	3245	-12.3 %
10	2017-07	4026	24.07 %
11	2017-08	4331	7.58 %
12	2017-09	4285	-1.06 %
13	2017-10	4631	8.07 %
14	2017-11	7544	62.9 %
15	2017-12	5673	-24.8 %
16	2018-01	7269	28.13 %
17	2018-02	6728	-7.44 %
18	2018-03	7211	7.18 %

3. 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

```
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 number_of_orders
FROM
   `Data_Target.orders`
GROUP BY
   time_of_day;
```

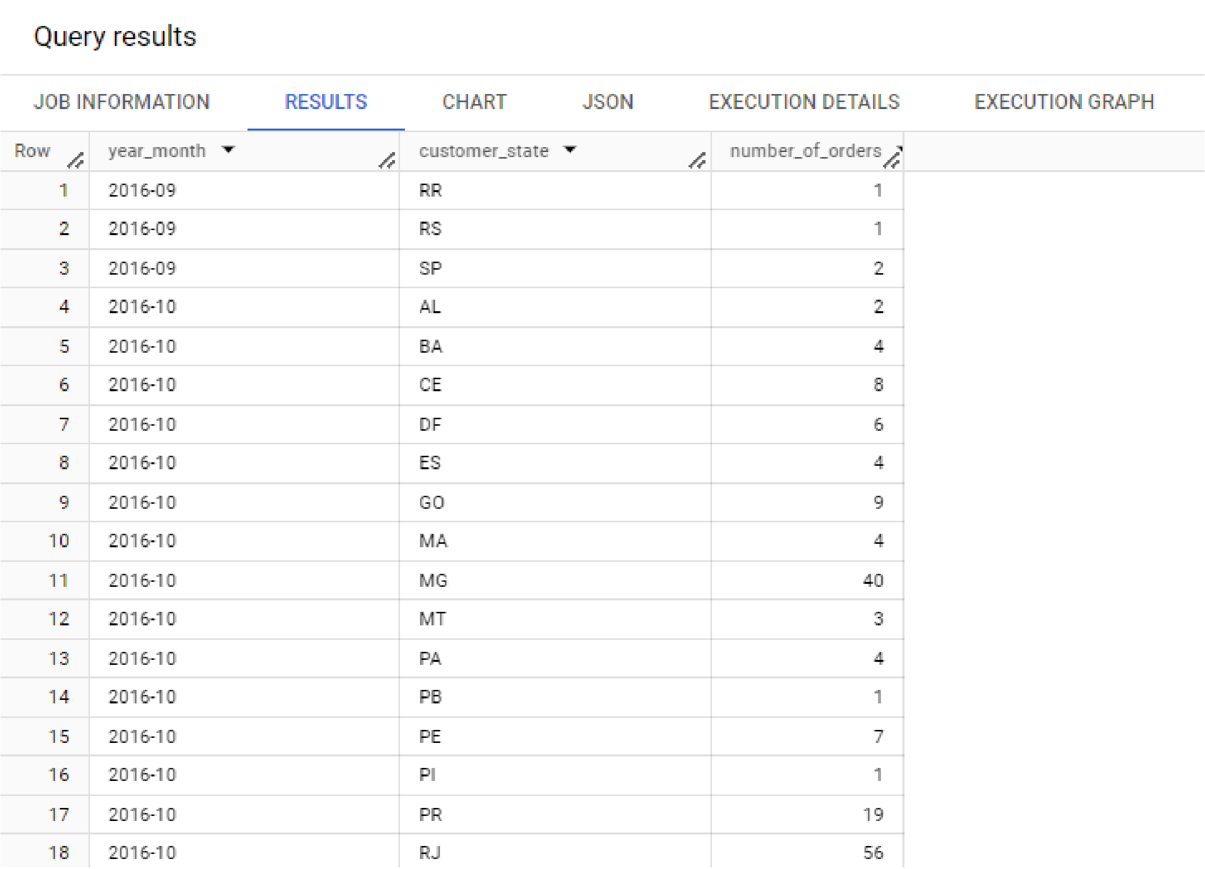


Row	time_of_day	number_of_orders
1	Morning	27733
2	Dawn	5242
3	Afternoon	38135
4	Night	28331

❖ Evolution of E-commerce orders in the Brazil region:

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

```
SELECT
  FORMAT_TIMESTAMP('%Y-%m', o.order_purchase_timestamp) AS year_month,
  c.customer_state,
  COUNT(*) AS number_of_orders
FROM
  `Data_Target.orders` o
JOIN
  `Data_Target.customers` c
ON
  o.customer_id = c.customer_id
GROUP BY
  year_month, c.customer_state
ORDER BY
  year_month, c.customer_state;
```

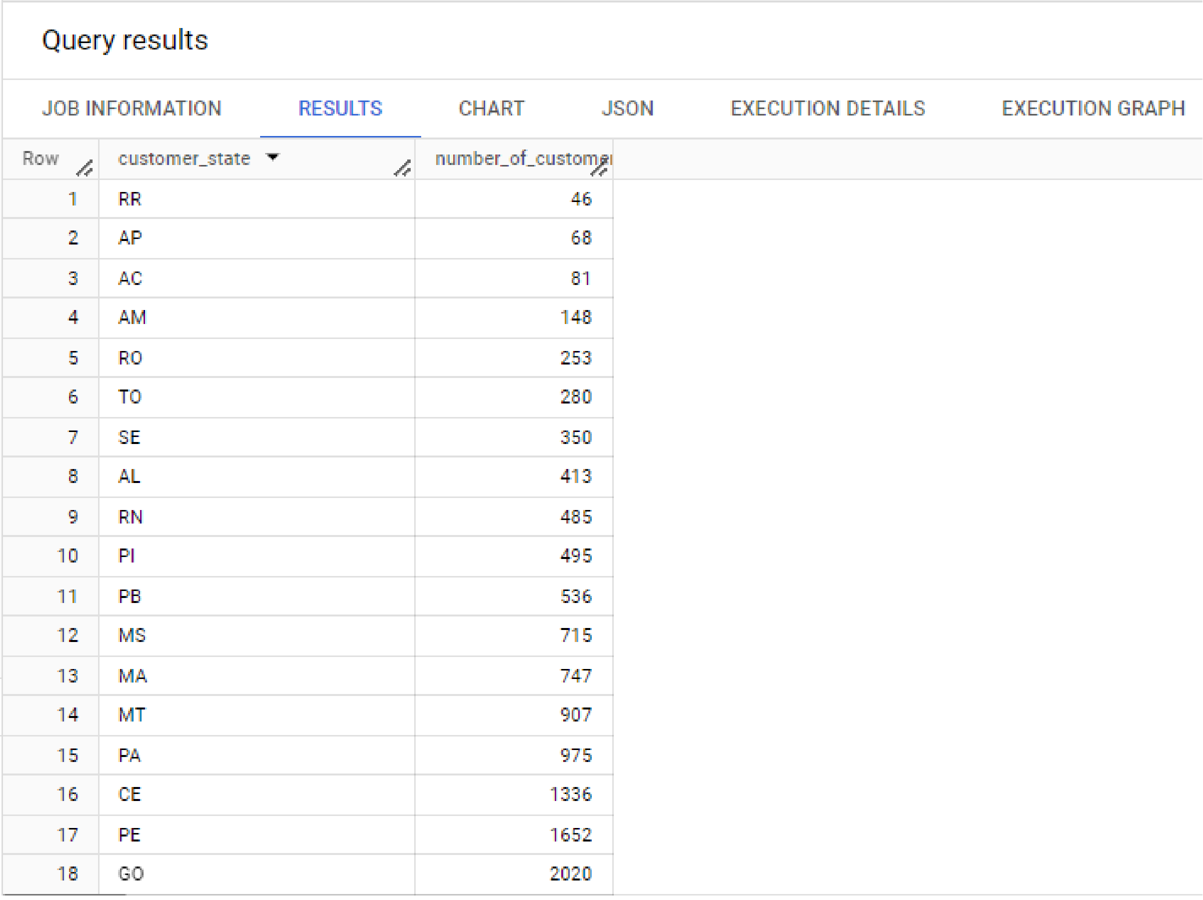


Row	year_month	customer_state	number_of_orders
1	2016-09	RR	1
2	2016-09	RS	1
3	2016-09	SP	2
4	2016-10	AL	2
5	2016-10	BA	4
6	2016-10	CE	8
7	2016-10	DF	6
8	2016-10	ES	4
9	2016-10	GO	9
10	2016-10	MA	4
11	2016-10	MG	40
12	2016-10	MT	3
13	2016-10	PA	4
14	2016-10	PB	1
15	2016-10	PE	7
16	2016-10	PI	1
17	2016-10	PR	19
18	2016-10	RJ	56

2. How are the customers distributed across all the states?

```
SELECT
  customer_state,
  COUNT(*) AS number_of_customers
FROM
  `Data_Target.customers`
```

```
GROUP BY
    customer_state
ORDER BY
    number_of_customers asc;
```

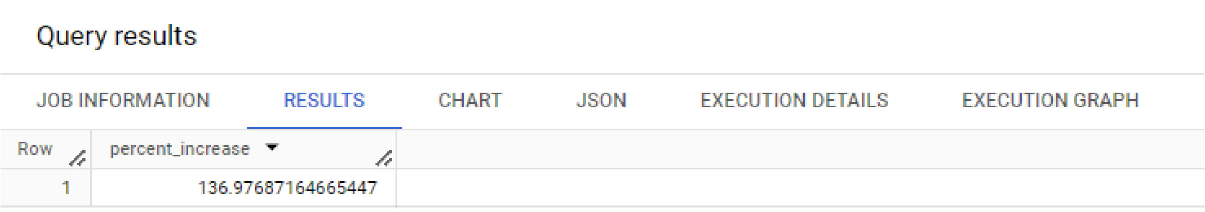


Row	customer_state	number_of_custome
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350
8	AL	413
9	RN	485
10	PI	495
11	PB	536
12	MS	715
13	MA	747
14	MT	907
15	PA	975
16	CE	1336
17	PE	1652
18	GO	2020

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

1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).You can use the "payment_value" column in the payments table to get the cost of orders.

```
SELECT
  (SUM(IF(EXTRACT(YEAR FROM order_purchase_timestamp) = 2018 AND EXTRACT(MONTH
FROM order_purchase_timestamp) BETWEEN 1 AND 8, payment_value, 0)) -
   SUM(IF(EXTRACT(YEAR FROM order_purchase_timestamp) = 2017 AND EXTRACT(MONTH
FROM order_purchase_timestamp) BETWEEN 1 AND 8, payment_value, 0))) /
   SUM(IF(EXTRACT(YEAR FROM order_purchase_timestamp) = 2017 AND EXTRACT(MONTH
FROM order_purchase_timestamp) BETWEEN 1 AND 8, payment_value, 0)) * 100.0 AS
percent_increase
FROM
  `Data_Target.payments` c
JOIN
  `Data_Target.orders` o
ON
  c.order_id = o.order_id;
```

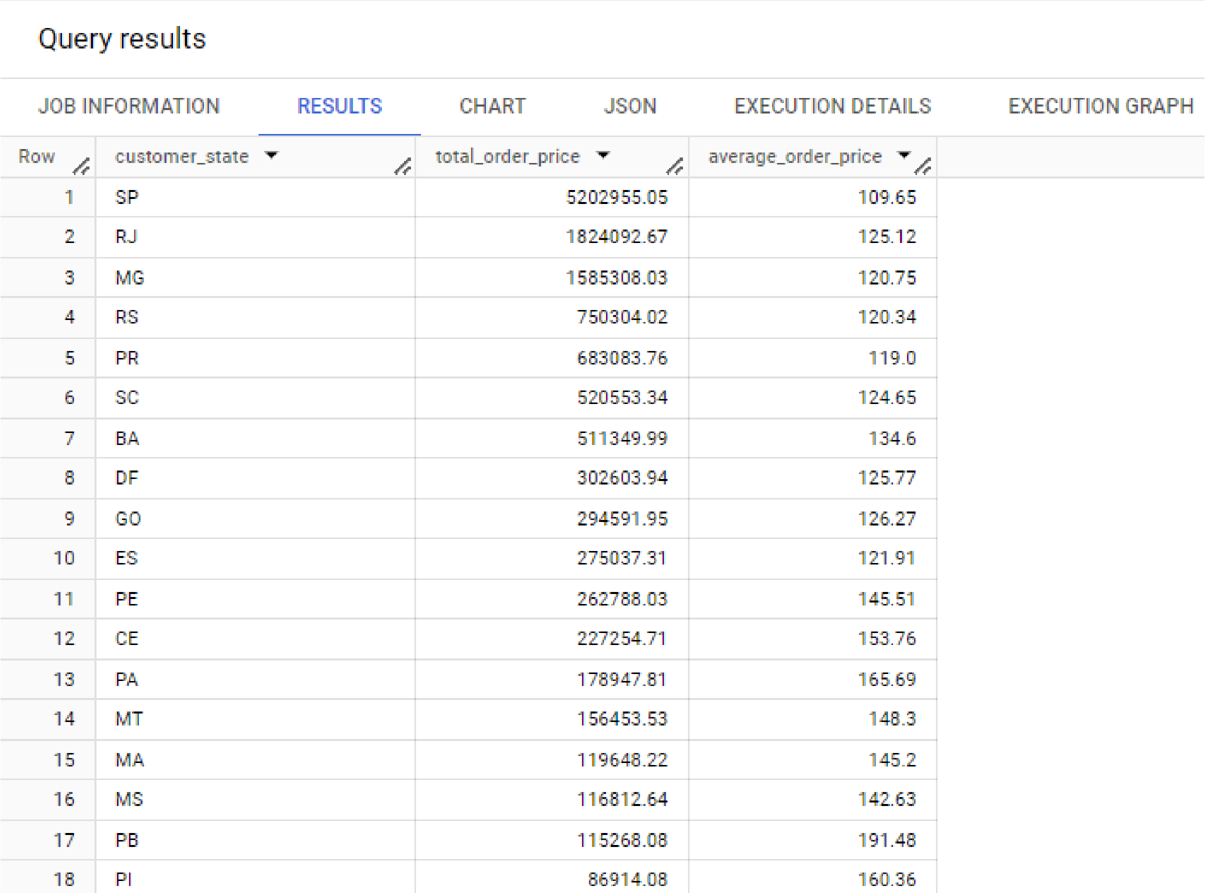

Query results

Row	percent_increase
1	136.97687164665447

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

```
SELECT
  c.customer_state,
  ROUND(SUM(ot.price), 2) AS total_order_price,
  ROUND(AVG(ot.price), 2) AS average_order_price
FROM
  `Data_Target.orders` o
JOIN
  `Data_Target.customers` c
ON
  o.customer_id = c.customer_id
JOIN
  `Data_Target.order_items` ot
ON
  o.order_id = ot.order_id
GROUP BY
  c.customer_state
```

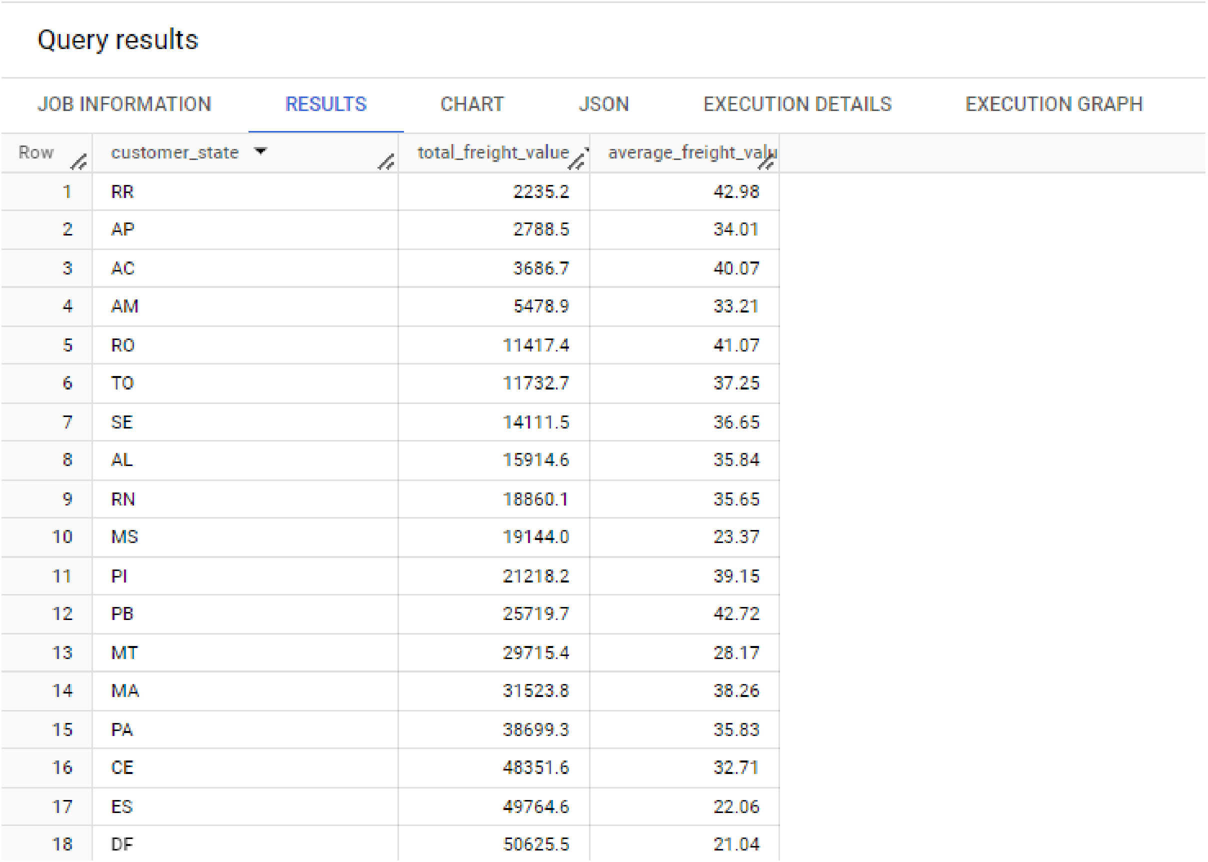
```
ORDER BY
    total_order_price DESC;
```



Row	customer_state	total_order_price	average_order_price
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	MG	1585308.03	120.75
4	RS	750304.02	120.34
5	PR	683083.76	119.0
6	SC	520553.34	124.65
7	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27
10	ES	275037.31	121.91
11	PE	262788.03	145.51
12	CE	227254.71	153.76
13	PA	178947.81	165.69
14	MT	156453.53	148.3
15	MA	119648.22	145.2
16	MS	116812.64	142.63
17	PB	115268.08	191.48
18	PI	86914.08	160.36

3. Calculate the Total & Average value of order freight for each state.

```
SELECT
    c.customer_state,
    Round(SUM(ot.freight_value),1) AS total_freight_value,
    Round(AVG(ot.freight_value),2) AS average_freight_value
FROM
    `Data_Target.orders` o
JOIN
    `Data_Target.customers` c
ON
    o.customer_id = c.customer_id
JOIN
    `Data_Target.order_items` ot
ON
    o.order_id = ot.order_id
GROUP BY
    c.customer_state
ORDER BY
    total_freight_value asc;
```



Row	customer_state	total_freight_value	average_freight_valu
1	RR	2235.2	42.98
2	AP	2788.5	34.01
3	AC	3686.7	40.07
4	AM	5478.9	33.21
5	RO	11417.4	41.07
6	TO	11732.7	37.25
7	SE	14111.5	36.65
8	AL	15914.6	35.84
9	RN	18860.1	35.65
10	MS	19144.0	23.37
11	PI	21218.2	39.15
12	PB	25719.7	42.72
13	MT	29715.4	28.17
14	MA	31523.8	38.26
15	PA	38699.3	35.83
16	CE	48351.6	32.71
17	ES	49764.6	22.06
18	DF	50625.5	21.04

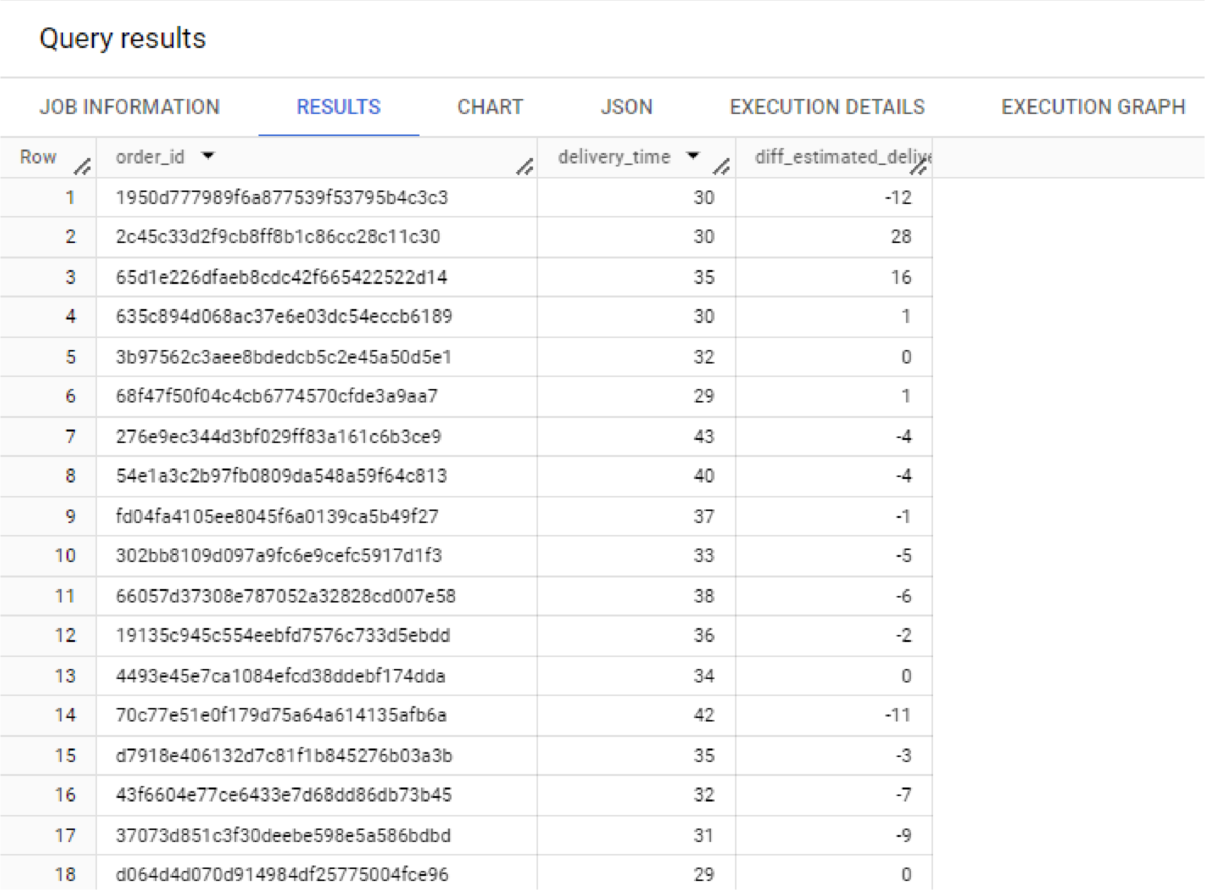
❖ Analysis based on sales, freight and delivery time.

1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
   Also, calculate the difference (in days) between the estimated & actual delivery date of an order. Do this in a single query.

   You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

   - **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
   - **diff_estimated_delivery** = order_delivered_customer_date - order_estimated_delivery_date

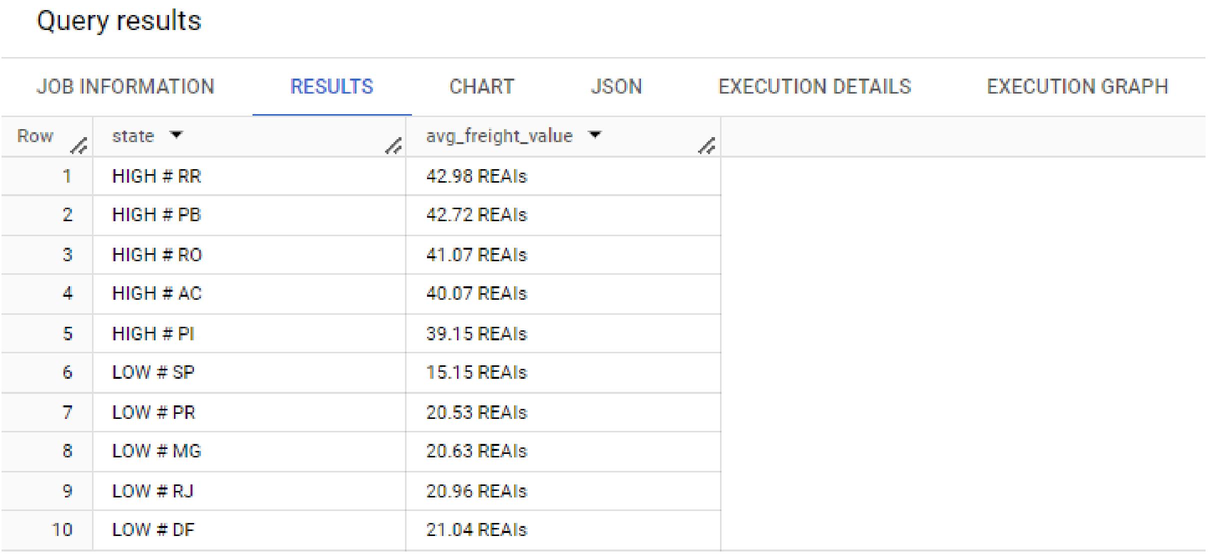
```
SELECT
  order_id,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
delivery_time,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS
diff_estimated_delivery
FROM
  `Data_Target.orders`;
```



Row	order_id	delivery_time	diff_estimated_deliv
1	1950d777989f6a877539f53795b4c3c3	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28c11c30	30	28
3	65d1e226dfaeb8cdc42f665422522d14	35	16
4	635c894d068ac37e6e03dc54eccb6189	30	1
5	3b97562c3aee8bdedcb5c2e45a50d5e1	32	0
6	68f47f50f04c4cb6774570cfde3a9aa7	29	1
7	276e9ec344d3bf029ff83a161c6b3ce9	43	-4
8	54e1a3c2b97fb0809da548a59f64c813	40	-4
9	fd04fa4105ee8045f6a0139ca5b49f27	37	-1
10	302bb8109d097a9fc6e9cefc5917d1f3	33	-5
11	66057d37308e787052a32828cd007e58	38	-6
12	19135c945c554eebfd7576c733d5ebdd	36	-2
13	4493e45e7ca1084efcd38ddebf174dda	34	0
14	70c77e51e0f179d75a64a614135afb6a	42	-11
15	d7918e406132d7c81f1b845276b03a3b	35	-3
16	43f6604e77ce6433e7d68dd86db73b45	32	-7
17	37073d851c3f30deebe598e5a586bdbd	31	-9
18	d064d4d070d914984df25775004fce96	29	0

2. Find out the top 5 states with the highest & lowest average freight value.

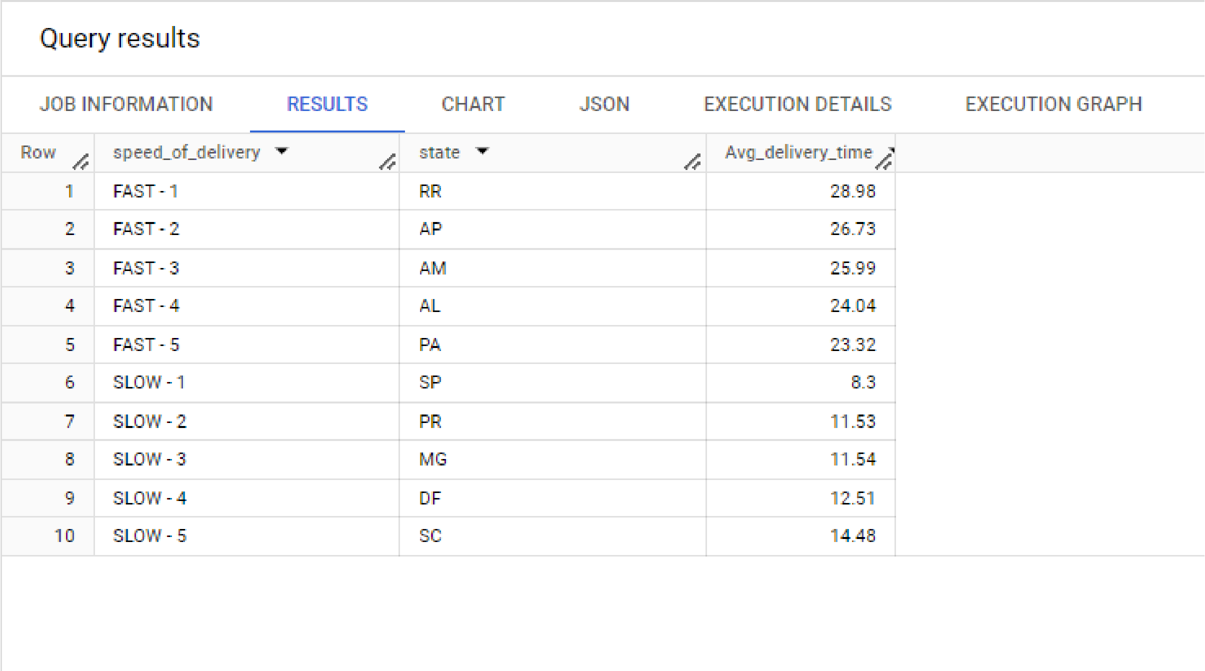
```
SELECT state, avg_freight_value
FROM (
  (
    SELECT
      CONCAT('HIGH # ', c.customer_state) AS state,
      MAX(ot.freight_value) AS High_freight_value,
      CONCAT(ROUND(AVG(ot.freight_value), 2), ' REAIs') AS avg_freight_value
    FROM
      `Data_Target.customers` c
    JOIN
      `Data_Target.orders` o
    ON
      c.customer_id = o.customer_id
    JOIN
      `Data_Target.order_items` ot
    ON
      o.order_id = ot.order_id
    GROUP BY
      c.customer_state
    ORDER BY
      avg_freight_value DESC
    LIMIT 5
  )
  UNION ALL
  (
    SELECT
      CONCAT('LOW # ', c.customer_state) AS state,
      MIN(ot.freight_value) AS low_freight_value,
      CONCAT(ROUND(AVG(ot.freight_value), 2), ' REAIs') AS avg_freight_value
    FROM
      `Data_Target.customers` c
    JOIN
      `Data_Target.orders` o
    ON
      c.customer_id = o.customer_id
    JOIN
      `Data_Target.order_items` ot
    ON
      o.order_id = ot.order_id
    GROUP BY
      c.customer_state
    ORDER BY
      avg_freight_value ASC
    LIMIT 5
  )
) AS t;
```



Row	state	avg_freight_value
1	HIGH # RR	42.98 REAls
2	HIGH # PB	42.72 REAls
3	HIGH # RO	41.07 REAls
4	HIGH # AC	40.07 REAls
5	HIGH # PI	39.15 REAls
6	LOW # SP	15.15 REAls
7	LOW # PR	20.53 REAls
8	LOW # MG	20.63 REAls
9	LOW # RJ	20.96 REAls
10	LOW # DF	21.04 REAls

3. Find out the top 5 states with the highest & lowest average delivery time.

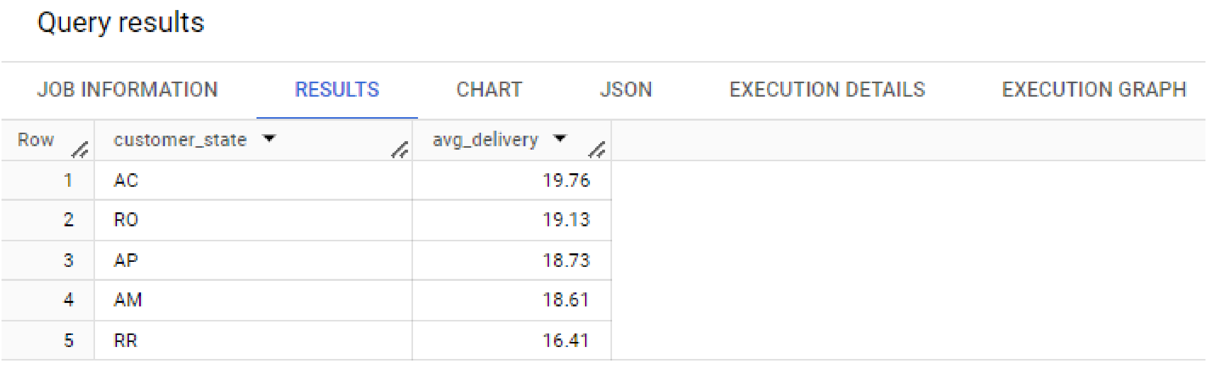
```
SELECT
  CONCAT(val, ' - ', rnk) AS speed_of_delivery,
  state,
  ROUND(avg_delivery_time, 2) AS Avg_delivery_time
FROM (
  SELECT
    state,
    'FAST' AS val,
    AVG(delivery_time) AS avg_delivery_time,
    DENSE_RANK() OVER (ORDER BY AVG(delivery_time) DESC) AS rnk
  FROM (
    SELECT
      customer_state AS state,
      DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
delivery_time
    FROM
      `Data_Target.customers` AS c
    JOIN
      `Data_Target.orders` AS o
    ON
      c.customer_id = o.customer_id
    WHERE
      order_delivered_customer_date IS NOT NULL
      AND order_purchase_timestamp IS NOT NULL
    GROUP BY
      state, order_delivered_customer_date, order_purchase_timestamp
  ) nt1
  GROUP BY
    state
    UNION ALL

    SELECT
        state,
        'SLOW' AS val,
        AVG(delivery_time) AS avg_delivery_time,
        DENSE_RANK() OVER (ORDER BY AVG(delivery_time) ASC) AS rnk
    FROM (
        SELECT
            customer_state AS state,
            DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
delivery_time
        FROM
            `Data_Target.customers` AS c
        JOIN
            `Data_Target.orders` AS o
        ON
            c.customer_id = o.customer_id
        WHERE
            order_delivered_customer_date IS NOT NULL
            AND order_purchase_timestamp IS NOT NULL
        GROUP BY
            state, order_delivered_customer_date, order_purchase_timestamp
    ) nt2
    GROUP BY
        state
) cte
WHERE rnk <= 5
ORDER BY speed_of_delivery;
```



Row	speed_of_delivery	state	Avg_delivery_time
1	FAST - 1	RR	28.98
2	FAST - 2	AP	26.73
3	FAST - 3	AM	25.99
4	FAST - 4	AL	24.04
5	FAST - 5	PA	23.32
6	SLOW - 1	SP	8.3
7	SLOW - 2	PR	11.53
8	SLOW - 3	MG	11.54
9	SLOW - 4	DF	12.51
10	SLOW - 5	SC	14.48

4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery. You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

```
SELECT customer_state,
Round(AVG(DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)),2)
AS avg_delivery
FROM
   `Data_Target.orders`o
JOIN
   `Data_Target.customers` c
ON
   o.customer_id = c.customer_id
GROUP BY
   customer_state
ORDER BY
   avg_delivery DESC
LIMIT 5;
```

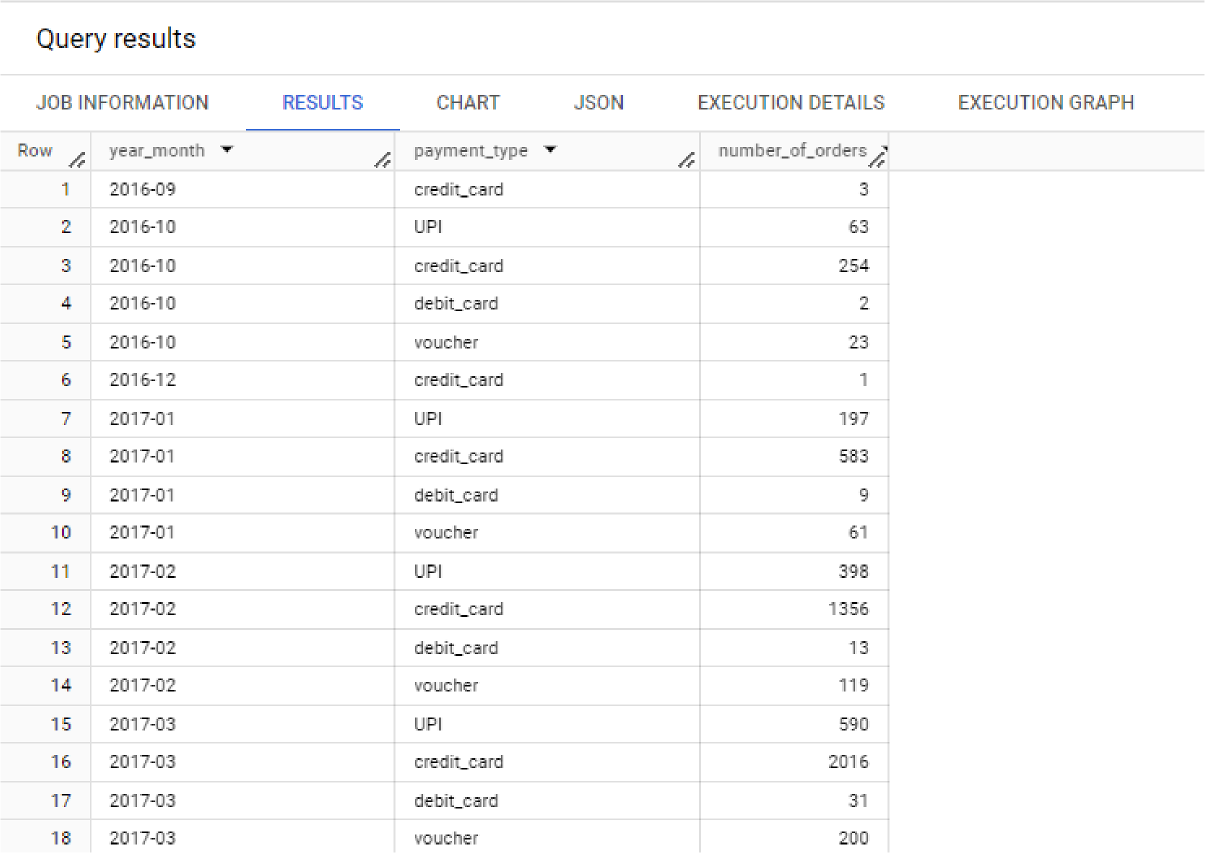


Row	customer_state	avg_delivery
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

❖ Analysis based on the payments:

1. Find the month-on-month no. of orders placed using different payment types.

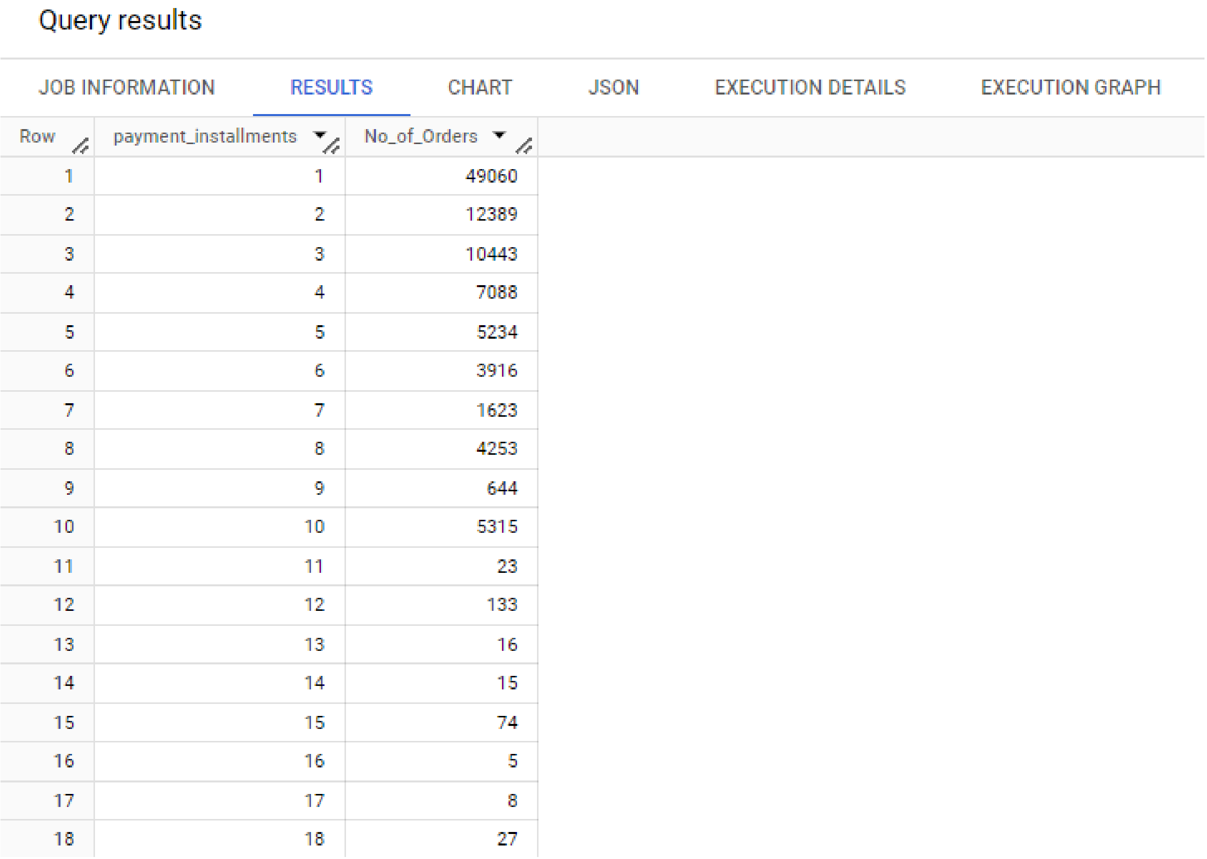
```
SELECT
  FORMAT_TIMESTAMP('%Y-%m', order_purchase_timestamp) AS year_month,
  payment_type,
  COUNT(*) AS number_of_orders
FROM
  `Data_Target.payments`p
JOIN
  `Data_Target.orders` o
ON
  p.order_id = o.order_id
GROUP BY
  year_month,
  payment_type
ORDER BY
  year_month,
  payment_type;
```



Row	year_month	payment_type	number_of_orders
1	2016-09	credit_card	3
2	2016-10	UPI	63
3	2016-10	credit_card	254
4	2016-10	debit_card	2
5	2016-10	voucher	23
6	2016-12	credit_card	1
7	2017-01	UPI	197
8	2017-01	credit_card	583
9	2017-01	debit_card	9
10	2017-01	voucher	61
11	2017-02	UPI	398
12	2017-02	credit_card	1356
13	2017-02	debit_card	13
14	2017-02	voucher	119
15	2017-03	UPI	590
16	2017-03	credit_card	2016
17	2017-03	debit_card	31
18	2017-03	voucher	200

2. Find the no. of orders placed on the basis of the payment installments that have been paid.

```
SELECT
  payment_installments,
  COUNT(DISTINCT order_id) AS No_of_Orders
FROM
  `Data_Target.payments`
WHERE
  payment_installments != 0
GROUP BY
  payment_installments
ORDER BY
  payment_installments;
```



Query results

Row	payment_installments	No_of_Orders
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	5	5234
6	6	3916
7	7	1623
8	8	4253
9	9	644
10	10	5315
11	11	23
12	12	133
13	13	16
14	14	15
15	15	74
16	16	5
17	17	8
18	18	27

➢ Insights:

1. **Data Types and Schema Understanding:**
   - The structure and data types of the "customers" table have been reviewed. This ensures we understand the type of data we are working with, such as IDs, zip codes, city names, and state codes.
2. **Order Time Range:**
   - Orders were placed between [earliest date] and [latest date]. This indicates the active period for the dataset, allowing us to focus on this specific timeframe for detailed analysis.
3. **Customer Distribution by City and State:**
   - Certain cities and states have higher order counts. For example, São Paulo (SP) might have significantly more orders compared to other regions. This suggests regional popularity and higher market activity in specific areas.

➢ Recommendations:

1. **Target Marketing Efforts:**
   - **High-Order Regions:** Focus marketing campaigns in regions with high order counts, such as São Paulo (SP). Tailored promotions and localized advertising can further increase sales in these areas.
   - **Low-Order Regions:** Investigate regions with low order counts to identify potential barriers. Consider localized promotions, partnerships with local influencers, or improving delivery logistics to boost sales.
2. **Enhance Customer Experience:**
   - **Popular Cities:** For cities with a high volume of orders, ensure that the logistics and supply chain are optimized to handle the demand efficiently. This includes maintaining adequate stock levels and ensuring timely deliveries.
   - **Feedback and Reviews:** Encourage customers in high-order regions to leave reviews and feedback. Use this information to improve products and services, enhancing customer satisfaction and loyalty.
3. **Logistics and Supply Chain Optimization:**
   - **Delivery Efficiency:** Analyze delivery times and delays to identify patterns or bottlenecks. Focus on improving the supply chain efficiency, particularly in regions where delivery delays are common.
   - **Freight Cost Management:** Monitor and manage freight costs, especially in regions where these costs are high. Negotiate better rates with shipping partners or explore alternative delivery methods.
4. **Seasonal and Temporal Promotions:**
   - **Order Trends:** Use insights from monthly and yearly order trends to plan seasonal promotions. If there is a noticeable peak in orders during certain months, align marketing efforts to capitalize on these trends.
   - **Time of Day:** If certain times of the day show higher order activity, consider running time-limited promotions or advertising campaigns during these peak hours to maximize engagement and sales.
5. **Payment Methods:**
   - **Popular Payment Types:** Identify the most commonly used payment methods and ensure they are prominently featured and seamlessly integrated into the checkout process. Consider offering incentives for using preferred payment methods to streamline transactions.

➢ Further Analysis:

To deepen the analysis and refine recommendations, consider the following steps:

1. **Order Trends Analysis:** Examine monthly and yearly order trends to identify growth patterns or seasonal peaks.
2. **Delivery Time and Efficiency:** Analyze delivery times and deviations from estimated delivery dates to improve logistics.
3. **Customer Feedback Analysis:** Investigate customer reviews and feedback to understand satisfaction levels and areas for improvement.
4. **Sales and Revenue Analysis:** Evaluate total and average order values, including freight costs, to identify revenue patterns and optimize pricing strategies.

By leveraging these insights and recommendations, Target can improve its operations, enhance customer satisfaction, and drive growth in the Brazilian market.