Business Case: Target

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 analyzing 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:

https://drive.google.com/drive/folders/1TGEc66YKbD443nsIRi1bWgVd238gJCnb?usp=sharing

Dataset:
https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb?usp=sharing

The data is available in 8 different csv files:

1. customers.csv
2. geolocation.csv
3. order_items.csv
4. payments.csv
5. reviews.csv
6. orders.csv
7. products.csv
8. sellers.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 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 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 shipping of the ordered product must be completed
price	Actual price of the products ordered
freight_value	Price rate at which a product is delivered from one point to another

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:

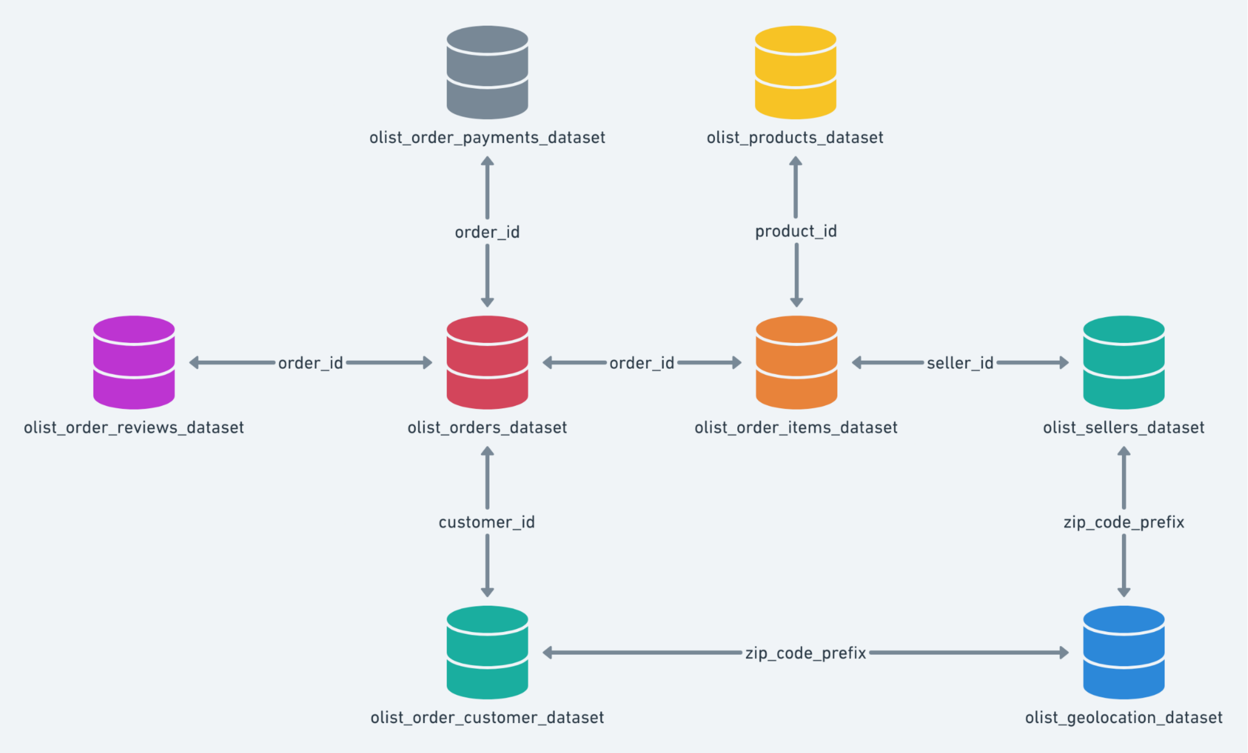
Features	Description
review_id	ID of the review given on the product ordered by the order id
order_id	A Unique ID of order made by the consumers
review_score	Review score given by the customer for each order on a scale of 1-5
review_comment_title	Title of the review
review_comment_message	Review comments 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:

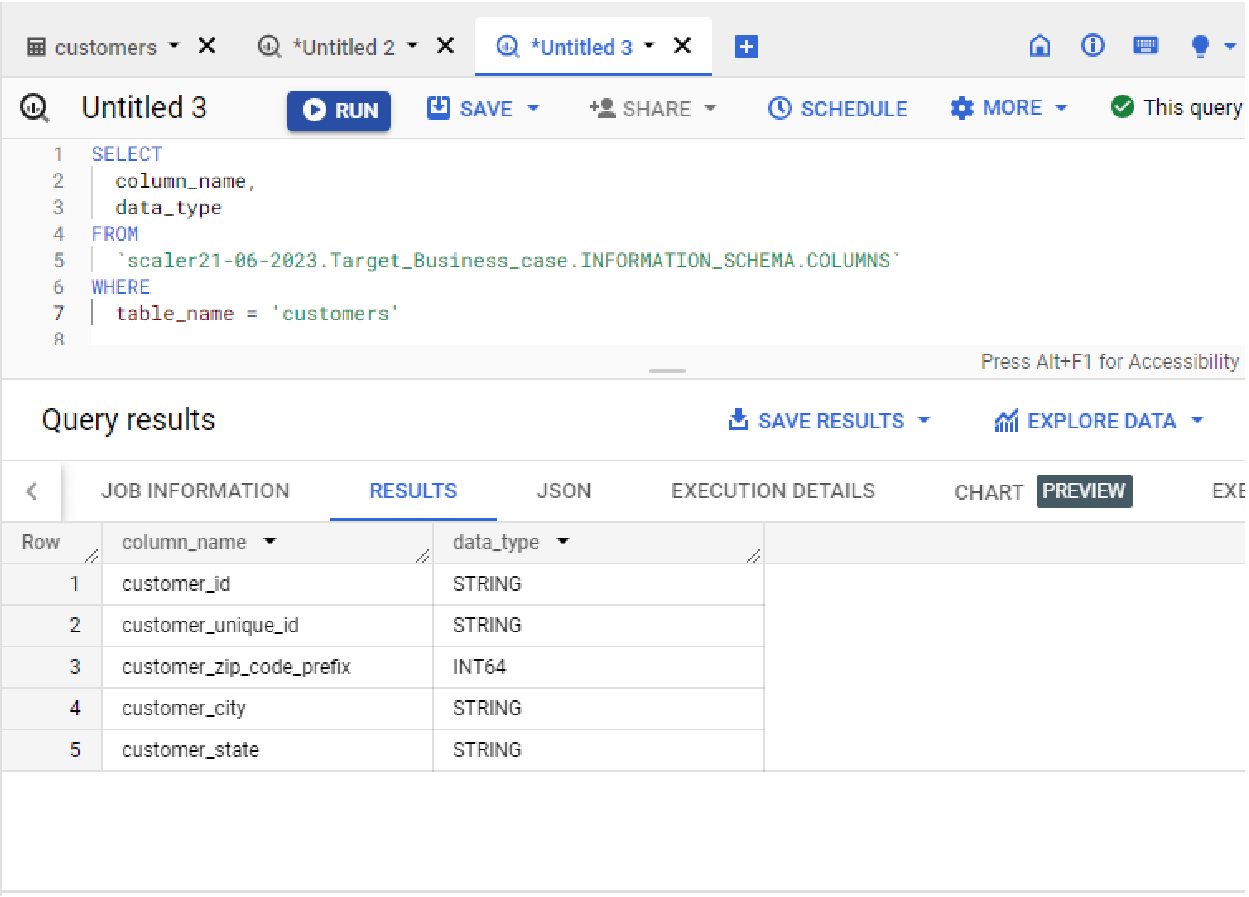


1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

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

```
SELECT
  column_name,
  data_type
FROM

`scaler21-06-2023.Target_Business_case.INFORMATION_SCHEMA.COLUMNS`
WHERE
  table_name = 'customers'
```



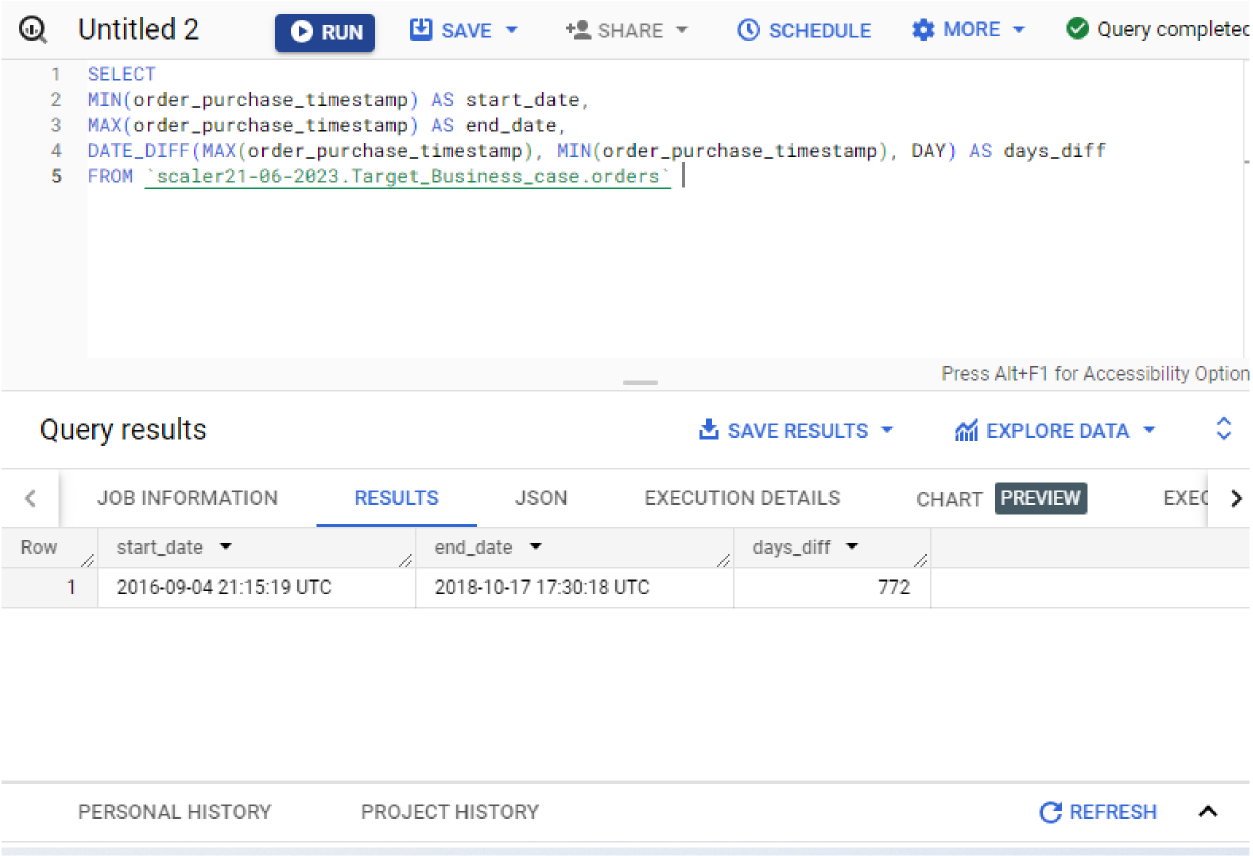
```
SELECT
  column_name,
  data_type
FROM
  `scaler21-06-2023.Target_Business_case.INFORMATION_SCHEMA.COLUMNS`
WHERE
  table_name = 'customers'
```

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

Here we can get the Data type of all columns in the “customers” table.

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

```
SELECT
MIN(order_purchase_timestamp) AS start_date,
MAX(order_purchase_timestamp) AS end_date,
DATE_DIFF(MAX(order_purchase_timestamp),
MIN(order_purchase_timestamp), DAY) AS days_diff
FROM `scaler21-06-2023.Target_Business_case.orders`
```



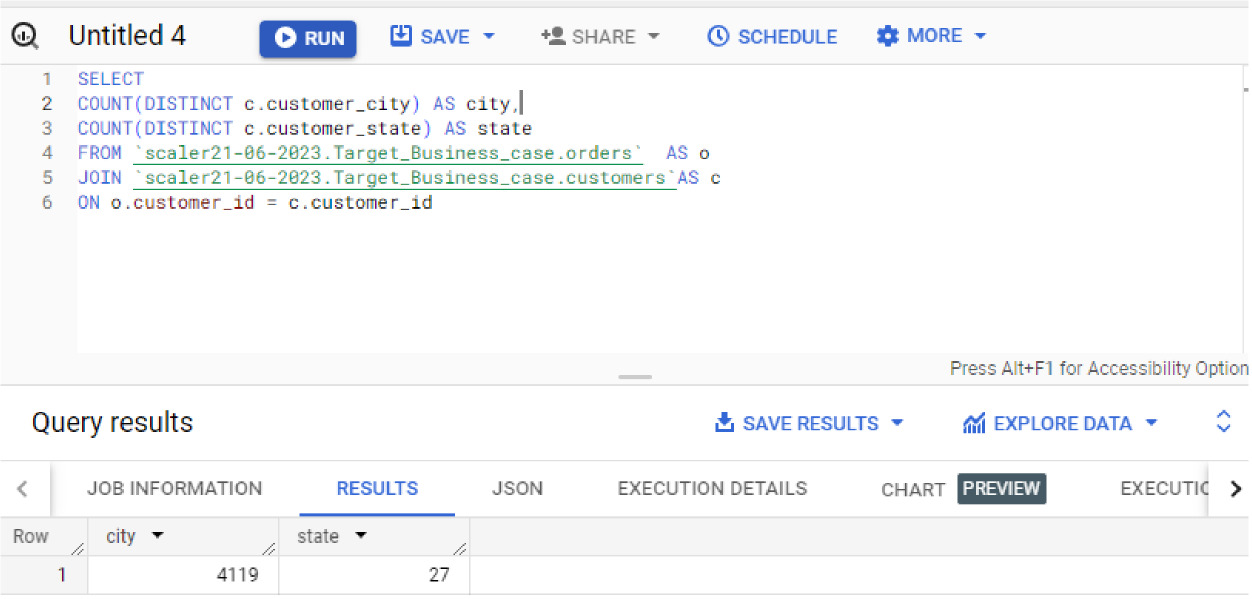
```
SELECT
MIN(order_purchase_timestamp) AS start_date,
MAX(order_purchase_timestamp) AS end_date,
DATE_DIFF(MAX(order_purchase_timestamp), MIN(order_purchase_timestamp), DAY) AS days_diff
FROM `scaler21-06-2023.Target_Business_case.orders`
```

Row	start_date	end_date	days_diff
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC	772

Here, we get the time range between which the orders are placed. The start date is **2016-09-04 21:15:19 UTC** and end date is **2018-10-17 17:30:18 UTC**. The difference between the start date and end date is **772** days

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

```
SELECT
COUNT(DISTINCT c.customer_city) AS city,
COUNT(DISTINCT c.customer_state) AS state
FROM `scaler21-06-2023.Target_Business_case.orders`  AS o
JOIN `scaler21-06-2023.Target_Business_case.customers`AS c
ON o.customer_id = c.customer_id
```



```
SELECT
COUNT(DISTINCT c.customer_city) AS city,
COUNT(DISTINCT c.customer_state) AS state
FROM `scaler21-06-2023.Target_Business_case.orders`  AS o
JOIN `scaler21-06-2023.Target_Business_case.customers`AS c
ON o.customer_id = c.customer_id
```

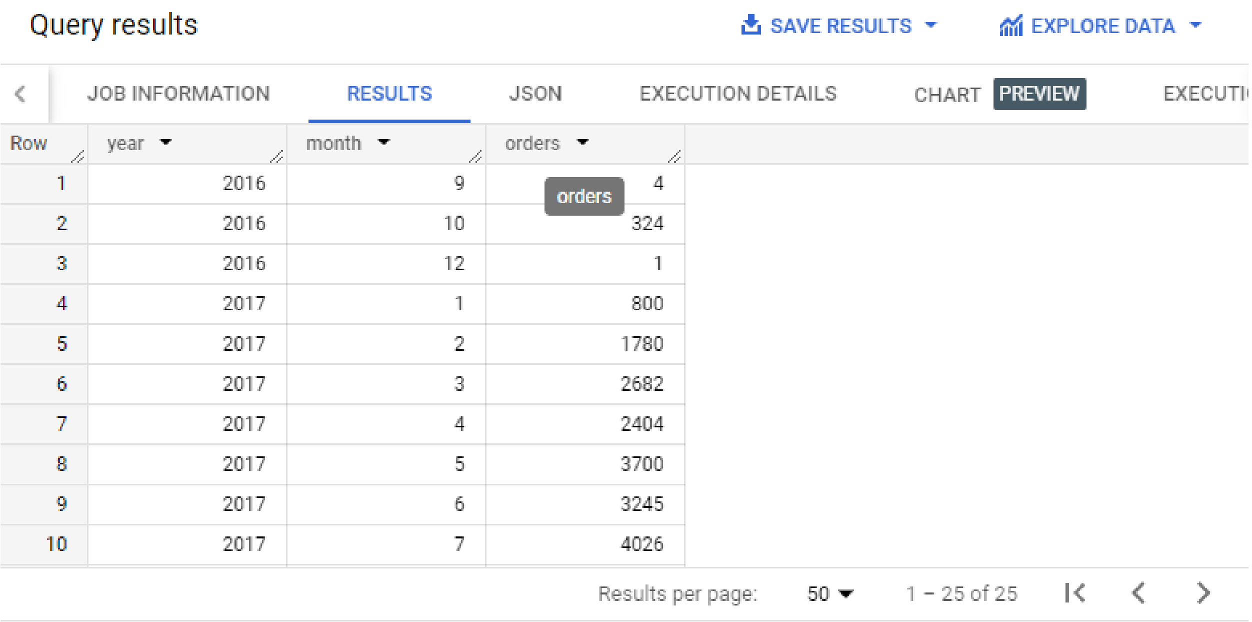
Row	city	state
1	4119	27

Here we count the cities & States of customers who ordered during the given period. The count of **cities** is **4119** and count of **state** is **27**

2. In-depth Exploration:

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

```
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
COUNT(order_id) AS orders
FROM `scaler21-06-2023.Target_Business_case.orders`
GROUP BY year, month
ORDER BY year, month
```

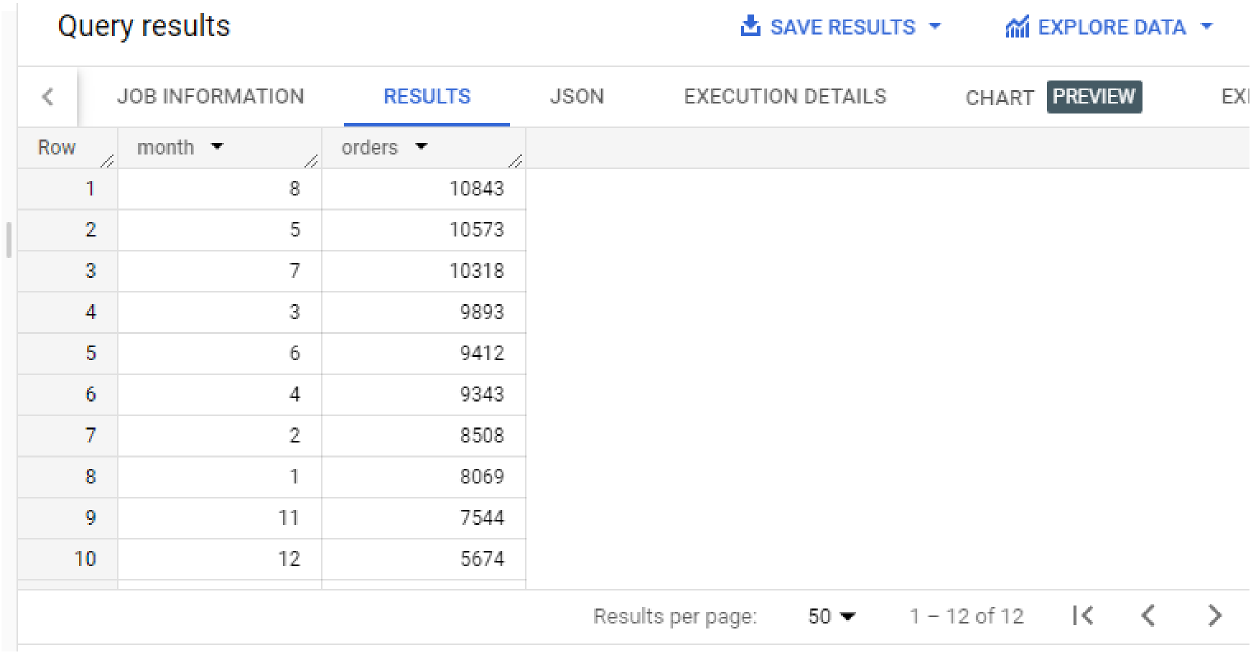


Row	year	month	orders
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026

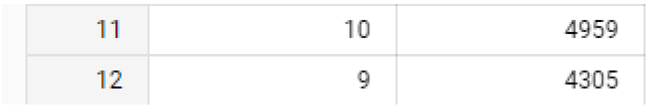
Here we get the insights about the growing trend in the number of orders placed over the past years

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

```
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
COUNT(order_id) AS orders
FROM `scaler21-06-2023.Target_Business_case.orders`
GROUP BY  month
ORDER BY orders DESC
```



Row	month	orders
1	8	10843
2	5	10573
3	7	10318
4	3	9893
5	6	9412
6	4	9343
7	2	8508
8	1	8069
9	11	7544
10	12	5674



11	10	4959
12	9	4305

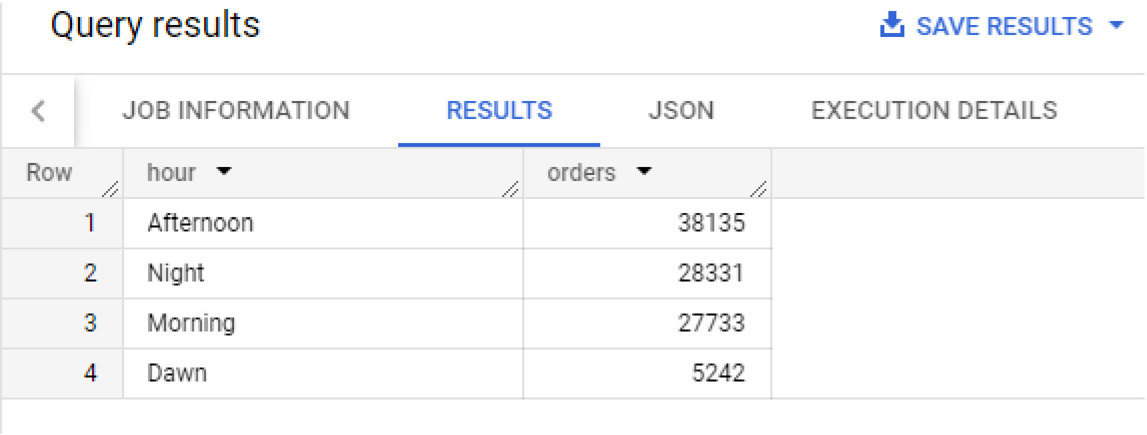
Here we get the insights about kind of seasonality in terms of the number of orders being placed.

August, May, July are the months with the **most number of orders** being placed.

September, October are the months with **least number of orders** being placed

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
    a. 0-6 hrs : Dawn
    b. 7-12 hrs : Mornings
    c. 13-18 hrs : Afternoon
    d. 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'
      WHEN EXTRACT(HOUR FROM order_purchase_timestamp) BETWEEN 19 AND 23 THEN
'Night'
  END AS hour,
COUNT(order_id) AS orders
FROM `scaler21-06-2023.Target_Business_case.orders`
GROUP BY hour
ORDER BY orders DESC
```



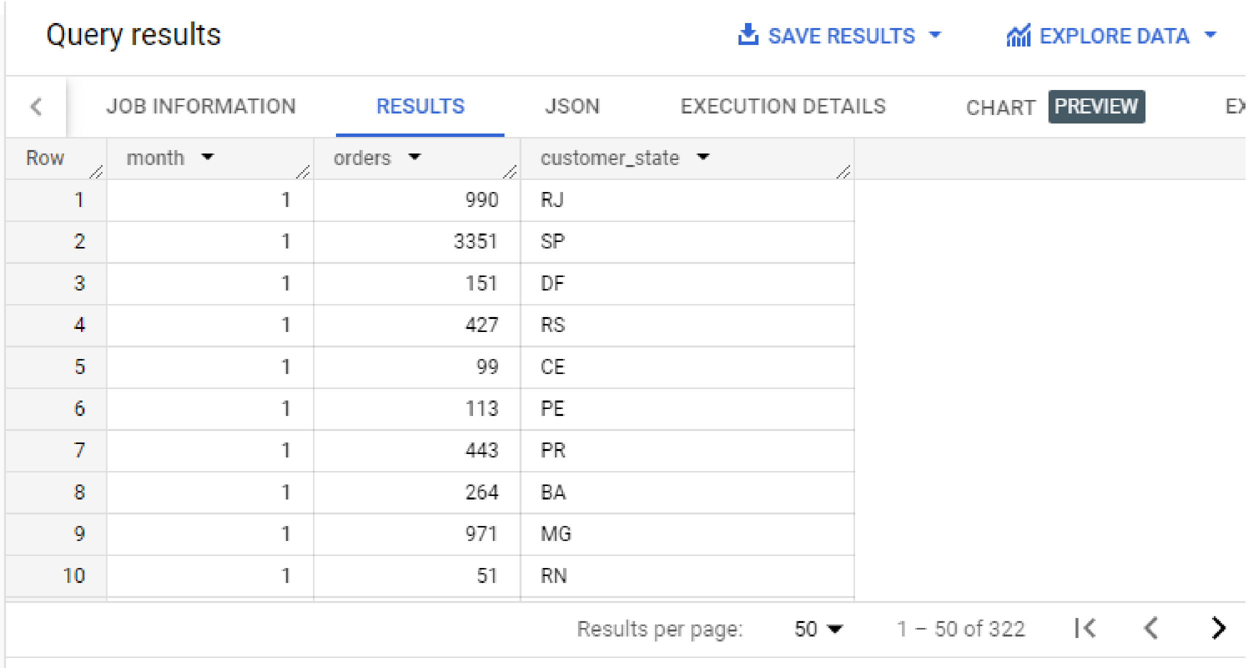
Row	hour	orders
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

Brazilian customers tend to place the **highest** number of orders during the “**Afternoon”**, while the **lowest** volume of orders is typically observed during the “**Dawn”** hours.

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

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

```
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
COUNT(order_id) AS orders,
c.customer_state
FROM `scaler21-06-2023.Target_Business_case.orders` AS o
JOIN `scaler21-06-2023.Target_Business_case.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY month, c.customer_state
ORDER BY month
```

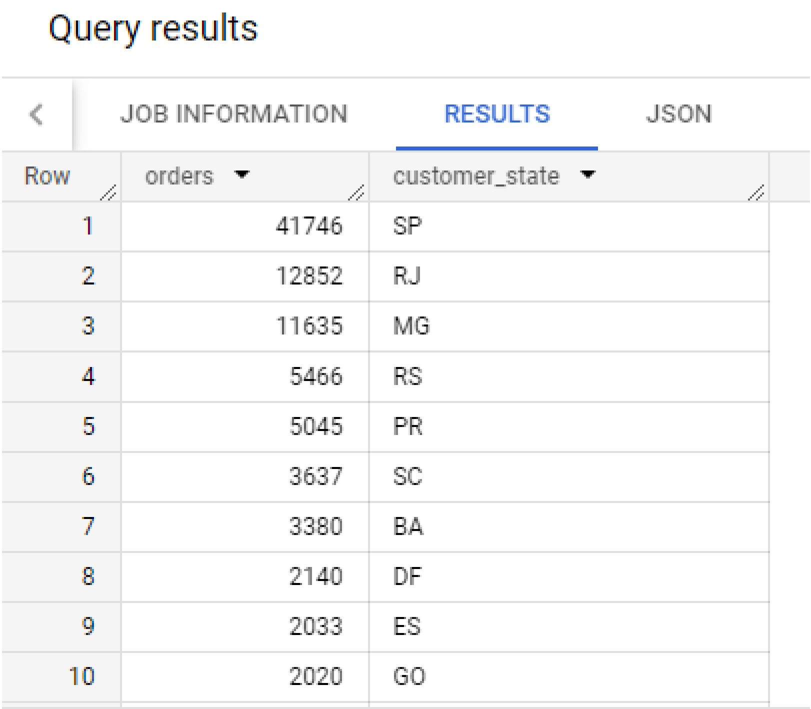


Row	month	orders	customer_state
1	1	990	RJ
2	1	3351	SP
3	1	151	DF
4	1	427	RS
5	1	99	CE
6	1	113	PE
7	1	443	PR
8	1	264	BA
9	1	971	MG
10	1	51	RN

Here we get the insights about month on month number of orders placed in each state

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

```
SELECT
COUNT(order_id) AS orders,
c.customer_state
FROM `scaler21-06-2023.Target_Business_case.orders` AS o
JOIN `scaler21-06-2023.Target_Business_case.customers` AS c
ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY orders DESC
```



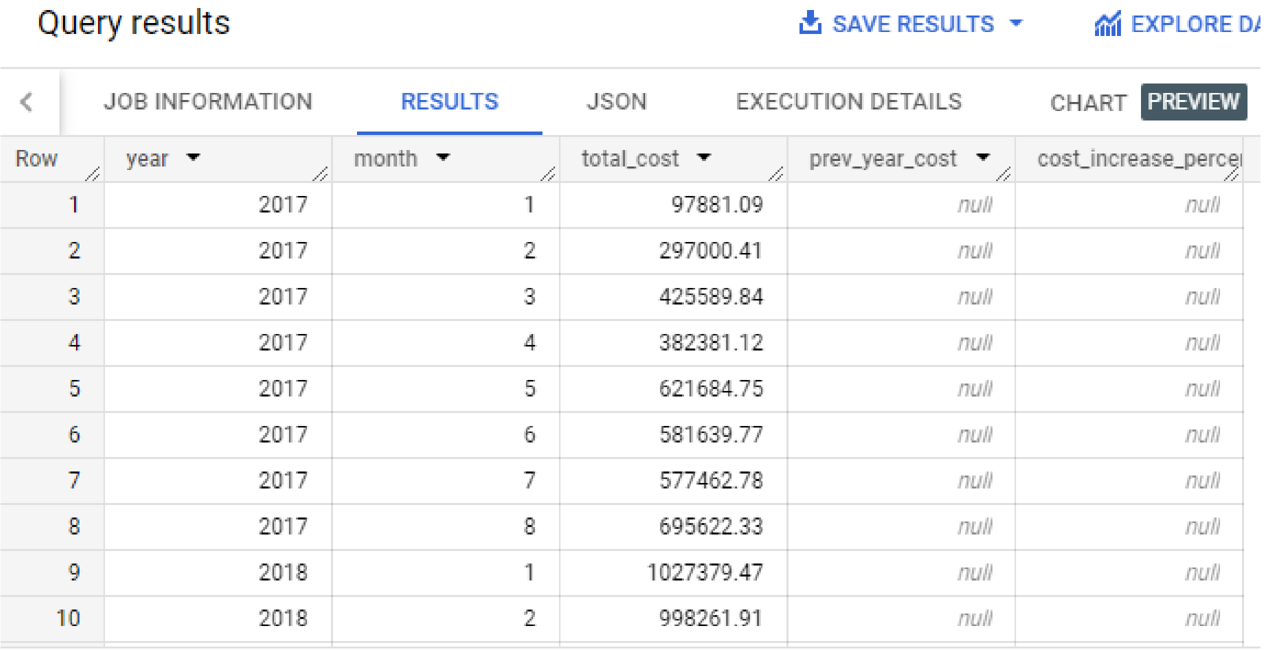
Row	orders	customer_state
1	41746	SP
2	12852	RJ
3	11635	MG
4	5466	RS
5	5045	PR
6	3637	SC
7	3380	BA
8	2140	DF
9	2033	ES
10	2020	GO

Here we get the insights about the customers distributed across all the states in Brazil

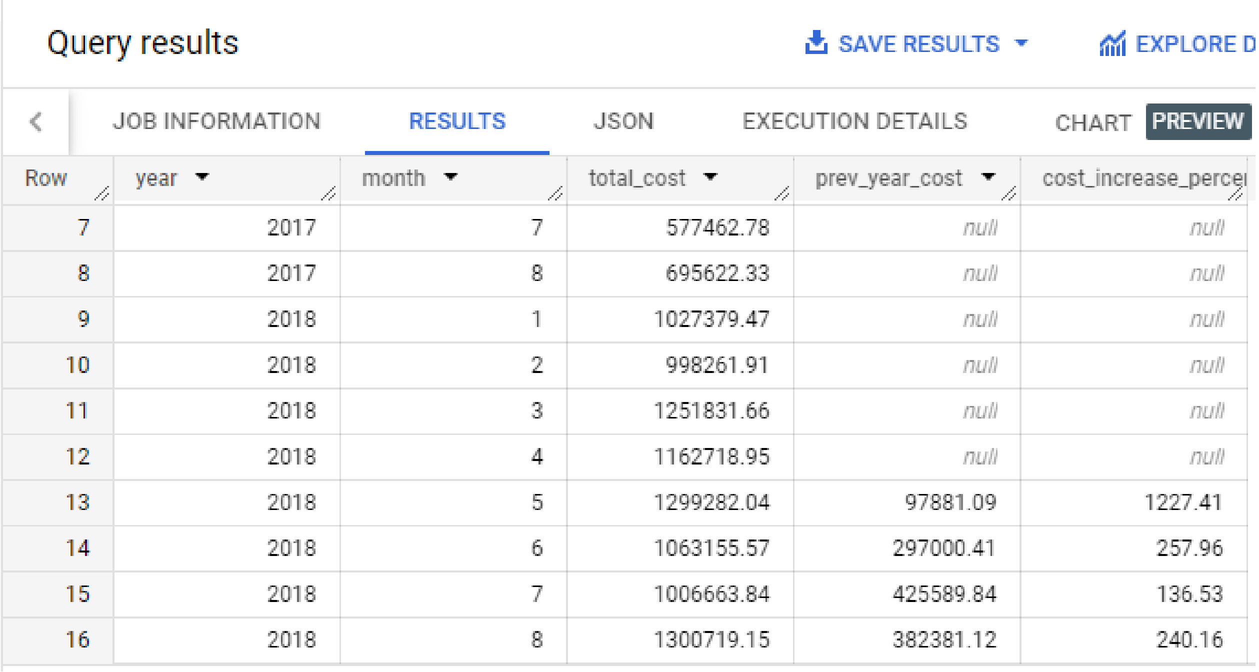
4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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

```
WITH MonthlyCosts AS (
  SELECT
    EXTRACT(YEAR FROM o.shipping_limit_date) AS year,
    EXTRACT(MONTH FROM o.shipping_limit_date) AS month,
    ROUND(SUM(o.price + o.freight_value),2) AS total_cost
  FROM
    `scaler21-06-2023.Target_Business_case.order_items` AS o
  JOIN
    `scaler21-06-2023.Target_Business_case.payments` AS p
  ON
    o.order_id = p.order_id
  WHERE
    EXTRACT(YEAR FROM o.shipping_limit_date) IN (2017, 2018)
    AND EXTRACT(MONTH FROM o.shipping_limit_date) BETWEEN 1 AND 8
  GROUP BY year,month
)
SELECT
  year,month, total_cost,
  LAG(total_cost, 12) OVER (ORDER BY year, month) AS prev_year_cost,
  ROUND(((total_cost - LAG(total_cost, 12) OVER (ORDER BY year, month)) /
LAG(total_cost, 12) OVER (ORDER BY year, month)) * 100, 2) AS cost_increase_percentage
FROM
  MonthlyCosts
ORDER BY year ASC, month ASC
```



Row	year	month	total_cost	prev_year_cost	cost_increase_perce
1	2017	1	97881.09	null	null
2	2017	2	297000.41	null	null
3	2017	3	425589.84	null	null
4	2017	4	382381.12	null	null
5	2017	5	621684.75	null	null
6	2017	6	581639.77	null	null
7	2017	7	577462.78	null	null
8	2017	8	695622.33	null	null
9	2018	1	1027379.47	null	null
10	2018	2	998261.91	null	null



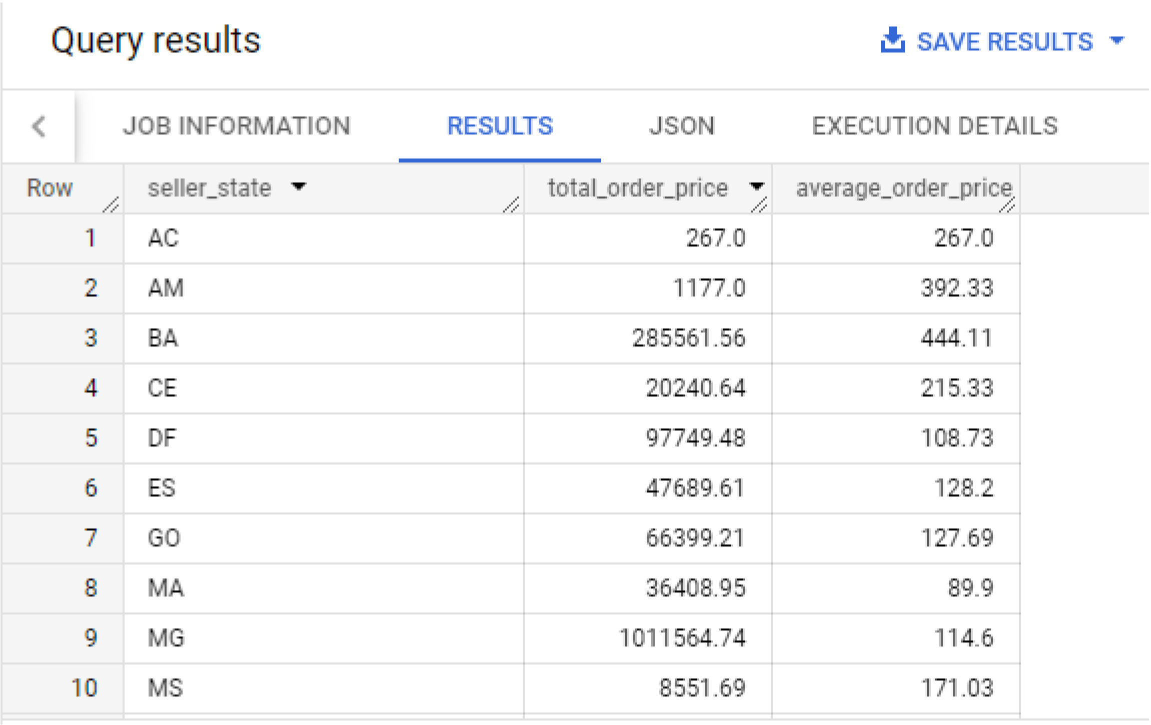
Row	year	month	total_cost	prev_year_cost	cost_increase_perce
7	2017	7	577462.78	null	null
8	2017	8	695622.33	null	null
9	2018	1	1027379.47	null	null
10	2018	2	998261.91	null	null
11	2018	3	1251831.66	null	null
12	2018	4	1162718.95	null	null
13	2018	5	1299282.04	97881.09	1227.41
14	2018	6	1063155.57	297000.41	257.96
15	2018	7	1006663.84	425589.84	136.53
16	2018	8	1300719.15	382381.12	240.16

Here we get the insights about the percentage increase in the cost of order from year **2017** to **2018**, including months between **January** to **August**.

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

```
WITH StateOrderData AS (
  SELECT
    p.seller_state,
    ROUND(SUM(o.price),2) AS total_order_price,
    ROUND(AVG(o.price),2) AS average_order_price
  FROM
    `scaler21-06-2023.Target_Business_case.order_items` AS o
  JOIN
    `scaler21-06-2023.Target_Business_case.sellers` AS p
  ON
    o.seller_id = p.seller_id

  GROUP BY
    p.seller_state
)
SELECT
  seller_state,
  total_order_price,
  average_order_price
FROM
  StateOrderData
ORDER BY
  seller_state ASC
```



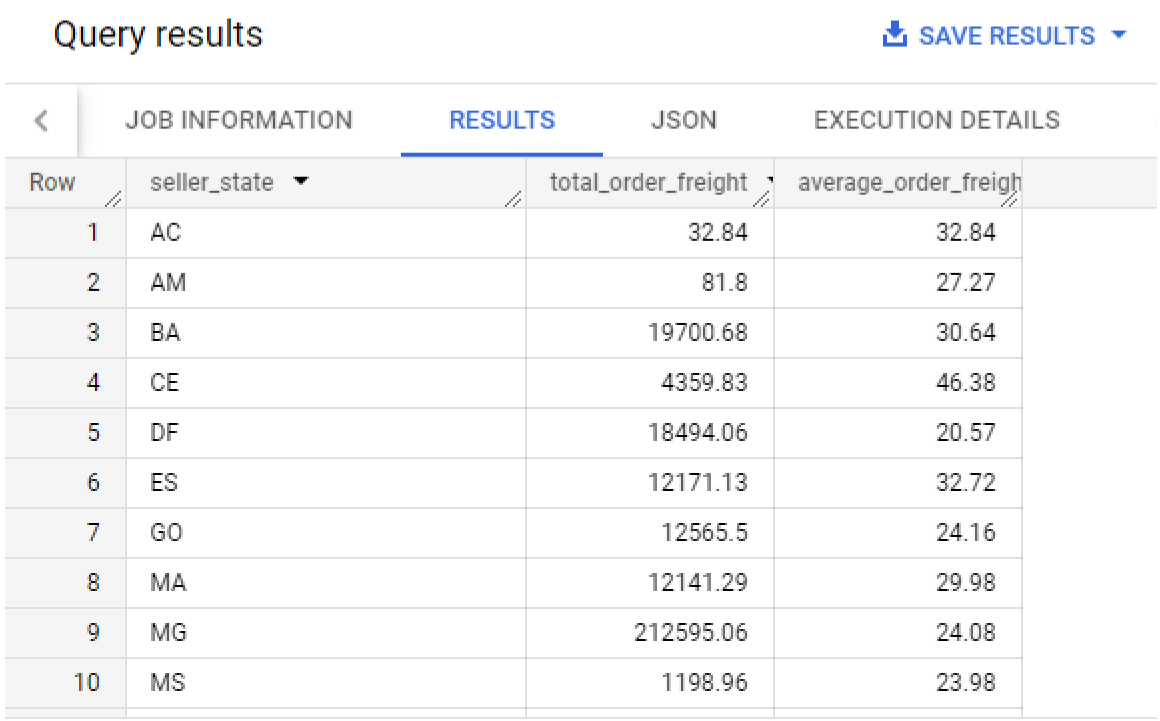
Row	seller_state	total_order_price	average_order_price
1	AC	267.0	267.0
2	AM	1177.0	392.33
3	BA	285561.56	444.11
4	CE	20240.64	215.33
5	DF	97749.48	108.73
6	ES	47689.61	128.2
7	GO	66399.21	127.69
8	MA	36408.95	89.9
9	MG	1011564.74	114.6
10	MS	8551.69	171.03

Here we get the insights about the **Total & Average value** of order price for each **state**.

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

```sql
WITH StateOrderData AS (
  SELECT
    p.seller_state,
    ROUND(SUM(o.freight_value),2) AS total_order_freight,
    ROUND(AVG(o.freight_value),2) AS average_order_freight
  FROM
    `scaler21-06-2023.Target_Business_case.order_items` AS o
  JOIN
    `scaler21-06-2023.Target_Business_case.sellers` AS p
  ON
    o.seller_id = p.seller_id

  GROUP BY
    p.seller_state
)
SELECT
  seller_state,
  total_order_freight,
  average_order_freight
FROM
  StateOrderData
ORDER BY
  seller_state ASC
```



Row	seller_state	total_order_freight	average_order_freigh
1	AC	32.84	32.84
2	AM	81.8	27.27
3	BA	19700.68	30.64
4	CE	4359.83	46.38
5	DF	18494.06	20.57
6	ES	12171.13	32.72
7	GO	12565.5	24.16
8	MA	12141.29	29.98
9	MG	212595.06	24.08
10	MS	1198.96	23.98

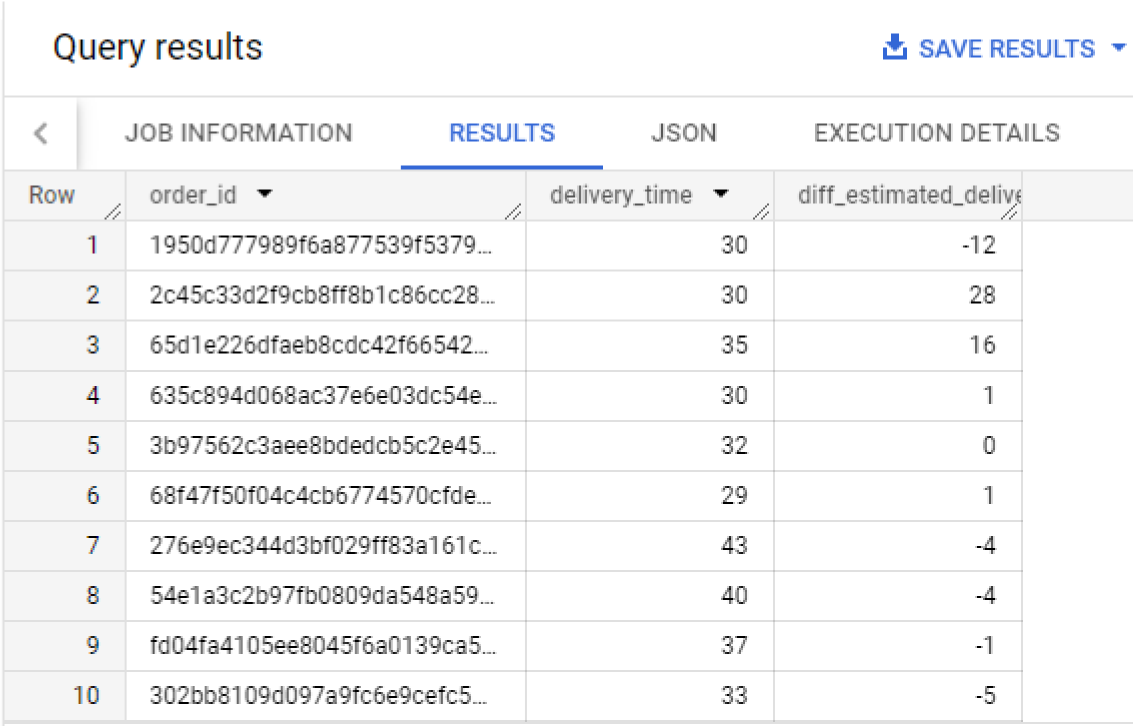
Here we get the insights about the **Total & Average value** of order freight for each **state**.

5. 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:
   a. time_to_deliver = order_delivered_customer_date - order_purchase_timestamp
   b. diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date

```
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 diff_estimated_delivery
FROM
    `scaler21-06-2023.Target_Business_case.orders`
```



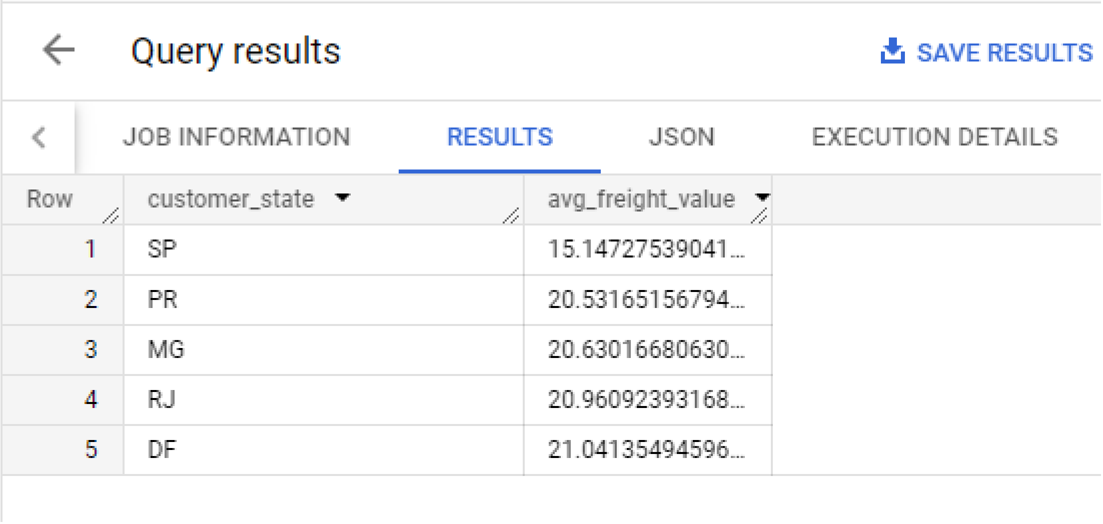
Row	order_id	delivery_time	diff_estimated_delive
1	1950d777989f6a877539f5379…	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28…	30	28
3	65d1e226dfaeb8cdc42f66542…	35	16
4	635c894d068ac37e6e03dc54e…	30	1
5	3b97562c3aee8bdedcb5c2e45…	32	0
6	68f47f50f04c4cb6774570cfde…	29	1
7	276e9ec344d3bf029ff83a161c…	43	-4
8	54e1a3c2b97fb0809da548a59…	40	-4
9	fd04fa4105ee8045f6a0139ca5…	37	-1
10	302bb8109d097a9fc6e9cefc5…	33	-5

Here we get the insights about **difference between** the **estimated** and **actual delivery date**

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

```
WITH AvgFreight AS (
  SELECT
    c.customer_state,
    AVG(oi.freight_value) AS avg_freight_value
  FROM
    `scaler21-06-2023.Target_Business_case.order_items` AS oi
  JOIN
    `scaler21-06-2023.Target_Business_case.orders` AS o ON oi.order_id = o.order_id
  JOIN
    `scaler21-06-2023.Target_Business_case.customers` AS c ON c.customer_id =
o.customer_id
  GROUP BY
    c.customer_state
)
SELECT
  customer_state,
  avg_freight_value
FROM
  AvgFreight
ORDER BY
  avg_freight_value DESC
LIMIT 5;

WITH AvgFreight AS (
  SELECT
    c.customer_state,
    AVG(oi.freight_value) AS avg_freight_value
  FROM
    `scaler21-06-2023.Target_Business_case.order_items` AS oi
  JOIN
    `scaler21-06-2023.Target_Business_case.orders` AS o ON oi.order_id = o.order_id
  JOIN
    `scaler21-06-2023.Target_Business_case.customers` AS c ON c.customer_id =
o.customer_id
  GROUP BY
    c.customer_state
)
SELECT
  customer_state,
  avg_freight_value
FROM
  AvgFreight
ORDER BY
  avg_freight_value ASC
LIMIT 5
```



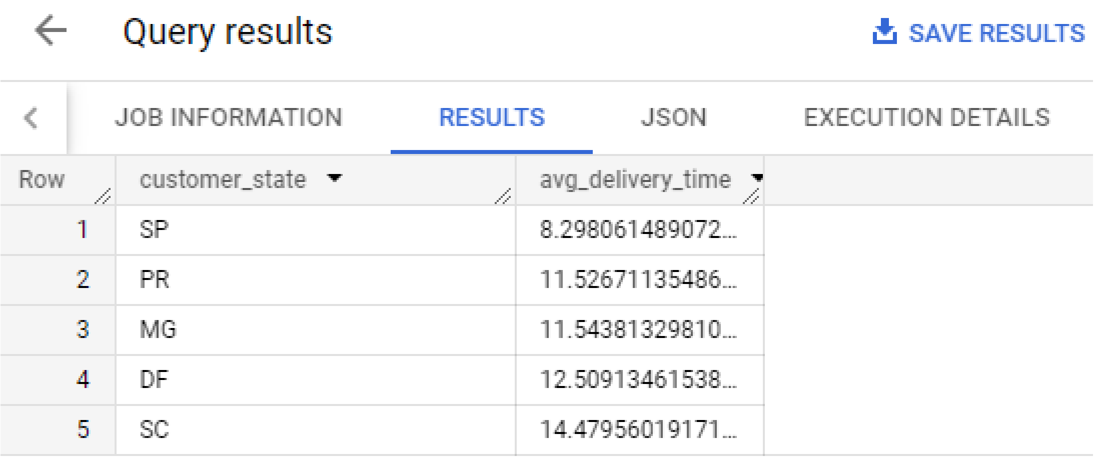
Row	customer_state	avg_freight_value
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...

Here we get the insights about the top 5 states with the highest & lowest average freight value.

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

```
WITH AvgDeliveryTime AS (
  SELECT
    c.customer_state,
    AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY)) AS avg_delivery_time
  FROM
    `scaler21-06-2023.Target_Business_case.orders` AS o
  JOIN
    `scaler21-06-2023.Target_Business_case.customers` AS c
ON c.customer_id = o.customer_id
  GROUP BY
    c.customer_state
)
SELECT
  customer_state,
  avg_delivery_time
FROM
  AvgDeliveryTime
ORDER BY
  avg_delivery_time DESC
LIMIT 5;

WITH AvgDeliveryTime AS (
  SELECT
    c.customer_state,
    AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
DAY)) AS avg_delivery_time
  FROM
    `scaler21-06-2023.Target_Business_case.orders` AS o
  JOIN
    `scaler21-06-2023.Target_Business_case.customers` AS c
ON c.customer_id = o.customer_id
  GROUP BY
    c.customer_state
)
SELECT
  customer_state,
  avg_delivery_time
FROM
  AvgDeliveryTime
ORDER BY
  avg_delivery_time ASC
LIMIT 5
```



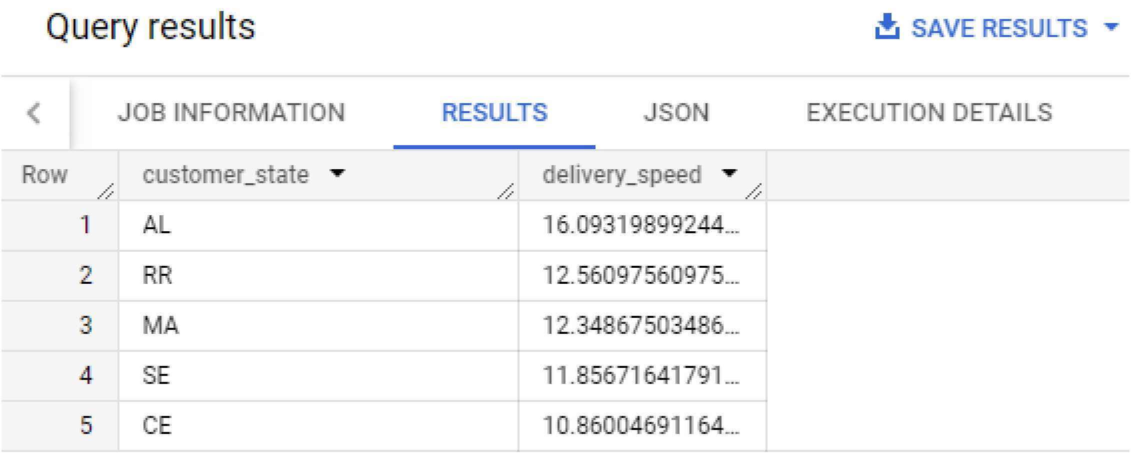
Row	customer_state	avg_delivery_time
1	SP	8.298061489072...
2	PR	11.52671135486...
3	MG	11.54381329810...
4	DF	12.50913461538...
5	SC	14.47956019171...

Here we get the insights about the top 5 states with the highest & lowest average delivery time

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.

```
WITH AvgDeliveryDifference AS (
  SELECT
    c.customer_state,
    AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, DAY)) -
AVG(TIMESTAMP_DIFF(o.order_estimated_delivery_date,
o.order_delivered_customer_date, DAY)) AS delivery_speed
  FROM
    `scaler21-06-2023.Target_Business_case.orders` AS o
  JOIN
    `scaler21-06-2023.Target_Business_case.customers` AS c ON c.customer_id =
o.customer_id
  GROUP BY
    c.customer_state
)
SELECT
  customer_state,
  delivery_speed
```

```
FROM
  AvgDeliveryDifference
ORDER BY
  delivery_speed DESC
LIMIT 5
```



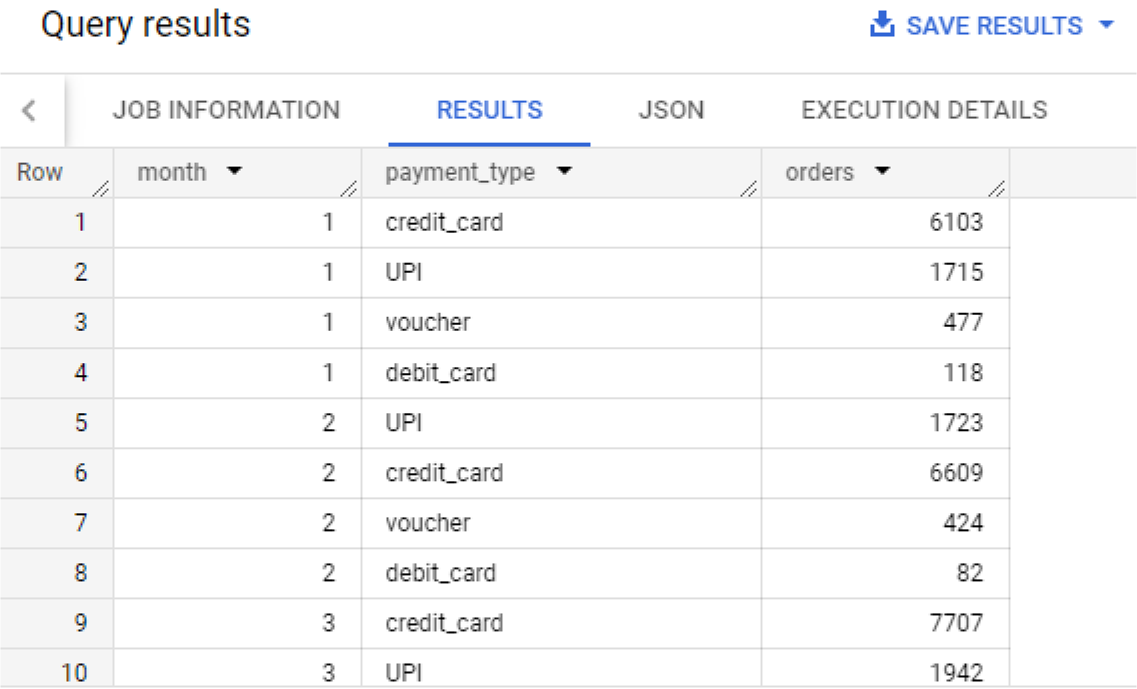
Query results

Row	customer_state	delivery_speed
1	AL	16.09319899244…
2	RR	12.56097560975…
3	MA	12.34867503486…
4	SE	11.85671641791…
5	CE	10.86004691164…

Here we get the insights about the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

6. Analysis based on the payments:
   1. Find the month on month no. of orders placed using different payment types.

```
SELECT
EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
p.payment_type,
COUNT(o.order_id) AS orders,
FROM `scaler21-06-2023.Target_Business_case.orders` AS o
JOIN `scaler21-06-2023.Target_Business_case.payments` AS p
ON o.order_id = p.order_id
GROUP BY month, p.payment_type
ORDER BY month
```



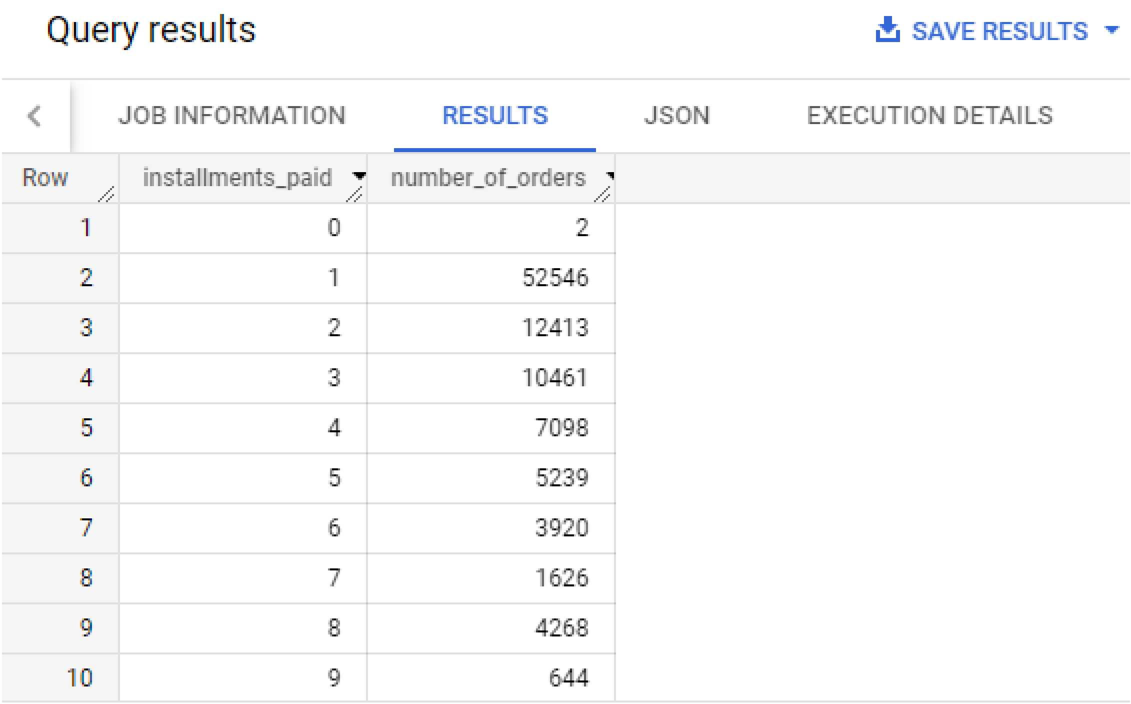
Query results

Row	month	payment_type	orders
1	1	credit_card	6103
2	1	UPI	1715
3	1	voucher	477
4	1	debit_card	118
5	2	UPI	1723
6	2	credit_card	6609
7	2	voucher	424
8	2	debit_card	82
9	3	credit_card	7707
10	3	UPI	1942

Here we get the insights about the month on month no. of orders placed using different payment types.

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

```
SELECT
  p.payment_installments AS installments_paid,
  COUNT(p.order_id) AS number_of_orders
FROM `scaler21-06-2023.Target_Business_case.payments` AS p
GROUP BY p.payment_installments
ORDER BY installments_paid
```



Row	installments_paid	number_of_orders
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

Here we get the insights about the number of orders placed on the basis of the payment installments that have been paid.

Insights and Recommendations Report: Enhancing Order Count and Customer Engagement Strategies

Order Count Analysis:

- There was a significant increase in order counts from 2016 to 2017, showcasing exceptional growth. However, the increase from 2017 to 2018 was not as substantial.
- To maintain growth momentum, Target should focus on strategic initiatives like product launches, targeted advertisements, and offering discounts during peak customer ordering times, particularly in the afternoon and at night.

Targeting Customer Preferences:

- Identifying specific customer segments that don't purchase certain times can be key. Target can employ tailored strategies such as heavy deals or special sales, particularly during dawn hours, to attract these customers.
- Analyzing pearl ordering months and regions, such as State SP, can offer insights into customer behavior for better-targeted marketing efforts.

Regional Focus and Customer Engagement:

- States with an order count lower than 200 should be a focal point for Target, providing additional discounts, expedited delivery, and enhanced customer support to increase engagement. This includes targeting areas like AM, AC, AP, and RR.
- Recognizing high-profit months, such as January and February in 2017 and 2018, should prompt strategic planning to capitalize on these periods.

Customer Service Optimization:

- Observing the customer profile of State SP can aid in tailoring services and marketing efforts for high-paying job holders, ensuring faster deliveries, and enhancing the overall customer experience.
- Limiting delivery times to under 100 days and improving partnerships with delivery services is crucial for customer satisfaction.

Payment and Purchase Experience:

- Focusing on payment methods such as debit and voucher modes can substantially enhance sales performance. Establishing more bank tie-ups can facilitate smoother transactions for customers.

- Offering installment payment options with lower interest rates can significantly boost order volume. A flexible installment plan can contribute to a better customer experience and increased sales.

By concentrating on these strategic insights and implementing the recommended actions, Target can further elevate its performance, cater to customer preferences, and sustain consistent growth in order counts and overall sales.

Conclusion:

This data analysis project delves into the E-commerce order data, providing valuable insights into various aspects of customer behavior, sales trends, and impact on the Brazilian economy. The analysis encompasses multiple dimensions, including order patterns, delivery times, payment types, and regional variations.