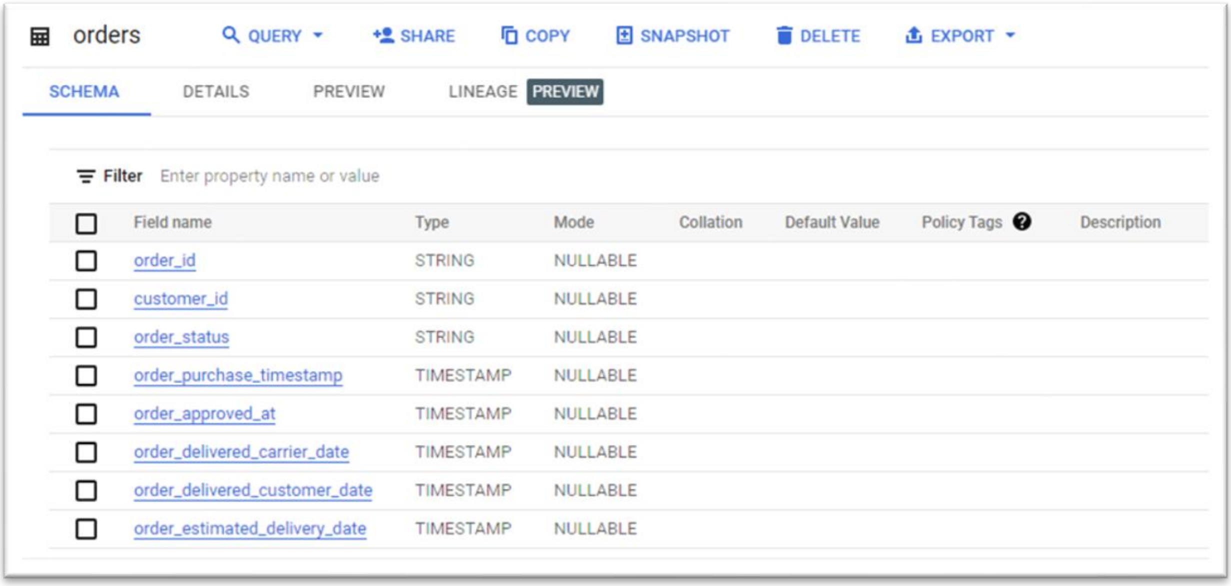
Business Case : Target SQL

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

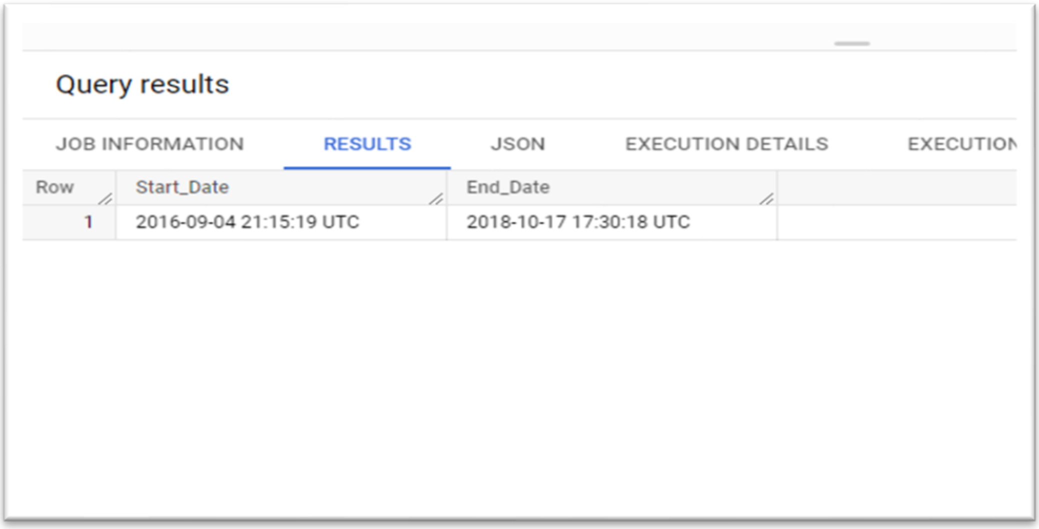
1. Data type of columns in a table



□	Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
□	order_id	STRING	NULLABLE				
□	customer_id	STRING	NULLABLE				
□	order_status	STRING	NULLABLE				
□	order_purchase_timestamp	TIMESTAMP	NULLABLE				
□	order_approved_at	TIMESTAMP	NULLABLE				
□	order_delivered_carrier_date	TIMESTAMP	NULLABLE				
□	order_delivered_customer_date	TIMESTAMP	NULLABLE				
□	order_estimated_delivery_date	TIMESTAMP	NULLABLE				

2. Time period for which the data is given

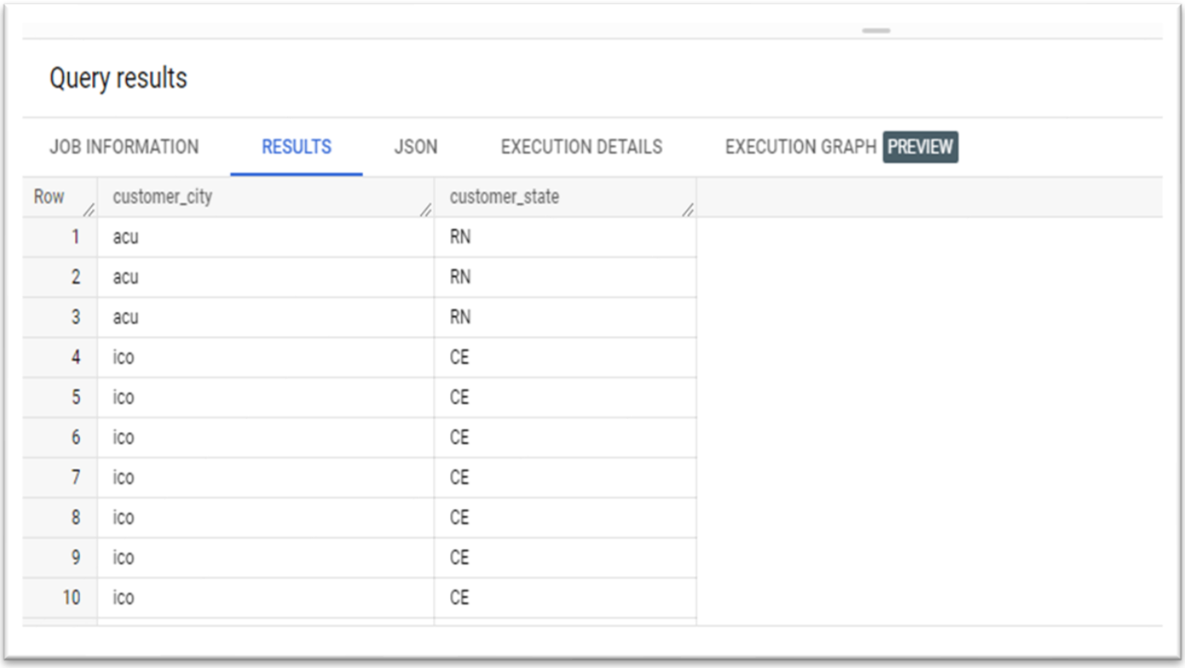
```
SELECT min(order_purchase_timestamp)AS
Start_Date,max(order_purchase_timestamp) AS End_Date

FROM `Target_SQL_Business_Case.orders`

;
```



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

3. Cities and States of customers ordered during the given period

```
SELECT customer_city,customer_state

FROM `Target_SQL_Business_Case.customers` c INNER JOIN
`Target_SQL_Business_Case.orders` o

ON c.customer_id =o.customer_id

WHERE order_purchase_timestamp BETWEEN '2016-09-04' AND '2018-10-17'

LIMIT 100

;
```



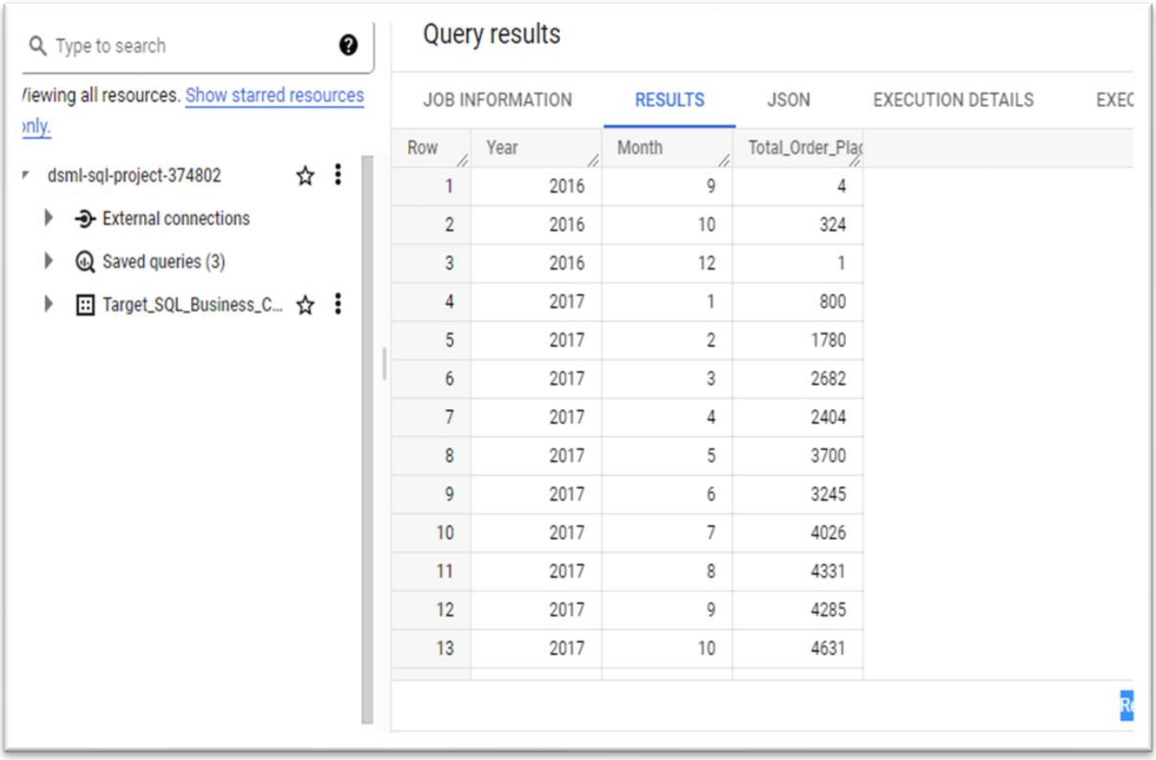
Query results

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

2. In-depth Exploration:

1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

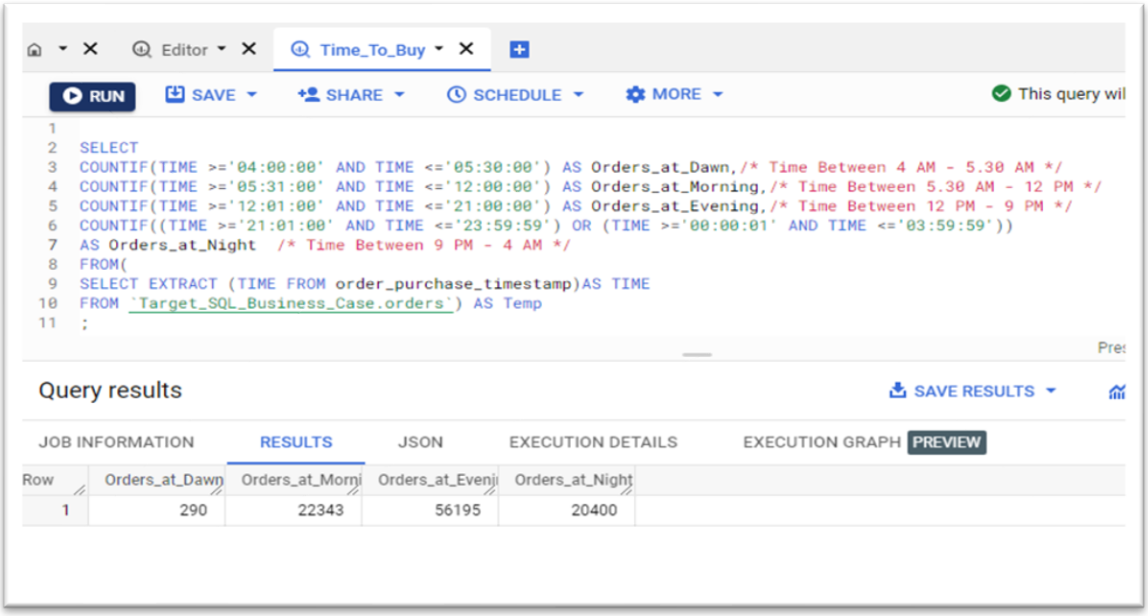
```
SELECT EXTRACT(year FROM order_purchase_timestamp) AS Year,

EXTRACT(month FROM order_purchase_timestamp) AS Month,

COUNT(order_id) AS Total_Order_Placed

FROM `Target_SQL_Business_Case.orders`

GROUP BY Month,Year

ORDER BY Year,Month

;
```



Row	Year	Month	Total_Order_Plac
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631

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

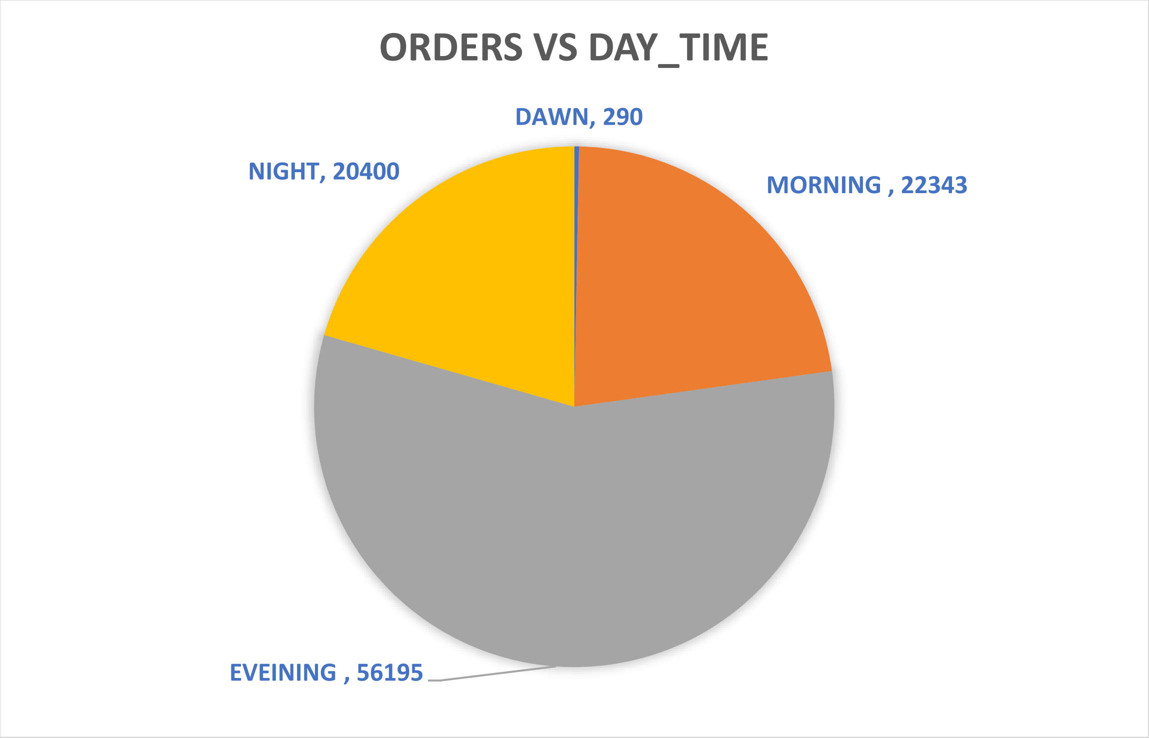
```
SELECT

COUNTIF(TIME >='04:00:00' AND TIME <='05:30:00') AS Orders_at_Dawn,/* Time Between 4 AM - 5.30 AM */

COUNTIF(TIME >='05:31:00' AND TIME <='12:00:00') AS Orders_at_Morning,/* Time Between 5.30 AM - 12 PM */

COUNTIF(TIME >='12:01:00' AND TIME <='21:00:00') AS Orders_at_Evening,/* Time Between 12 PM - 9 PM */

COUNTIF((TIME >='21:01:00' AND TIME <='23:59:59') OR (TIME >='00:00:01' AND TIME <='03:59:59'))

AS Orders_at_Night  /* Time Between 9 PM - 4 AM */

FROM(

SELECT EXTRACT (TIME FROM order_purchase_timestamp)AS TIME

FROM `Target_SQL_Business_Case.orders`) AS Temp

;
```



```sql

SELECT
COUNTIF(TIME >='04:00:00' AND TIME <='05:30:00') AS Orders_at_Dawn,/* Time Between 4 AM - 5.30 AM */
COUNTIF(TIME >='05:31:00' AND TIME <='12:00:00') AS Orders_at_Morning,/* Time Between 5.30 AM - 12 PM */
COUNTIF(TIME >='12:01:00' AND TIME <='21:00:00') AS Orders_at_Evening,/* Time Between 12 PM - 9 PM */
COUNTIF((TIME >='21:01:00' AND TIME <='23:59:59') OR (TIME >='00:00:01' AND TIME <='03:59:59'))
AS Orders_at_Night  /* Time Between 9 PM - 4 AM */
FROM(
SELECT EXTRACT (TIME FROM order_purchase_timestamp)AS TIME
FROM `Target_SQL_Business_Case.orders`) AS Temp
;
```

Query results

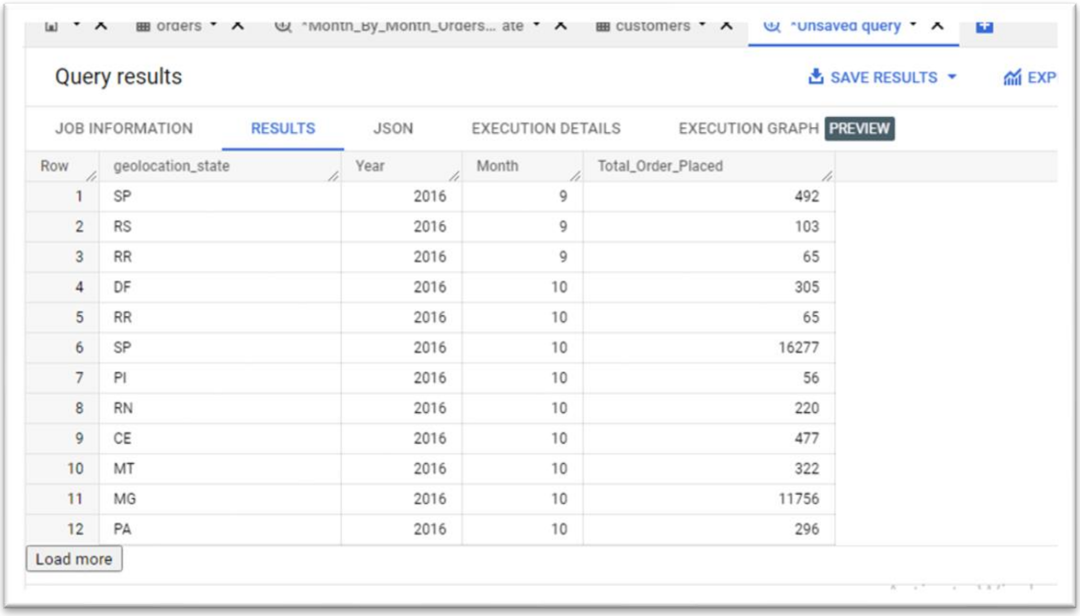
Row	Orders_at_Dawn	Orders_at_Morni	Orders_at_Eveni	Orders_at_Night
1	290	22343	56195	20400



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

1. Get month on month orders by states

```
SELECT g.geolocation_state,EXTRACT(year FROM o.order_purchase_timestamp) AS Year,

EXTRACT(month FROM o.order_purchase_timestamp) AS Month,

COUNT(DISTINCT order_id) AS Total_Order_Placed



FROM `Target_SQL_Business_Case.geolocation` g INNER JOIN

`Target_SQL_Business_Case.customers` c

ON g.geolocation_zip_code_prefix =c.customer_zip_code_prefix

INNER JOIN `Target_SQL_Business_Case.orders` o ON

c.customer_id=o.customer_id

GROUP BY Month,Year,g.geolocation_state

ORDER BY Year,Month

LIMIT 100

;
```

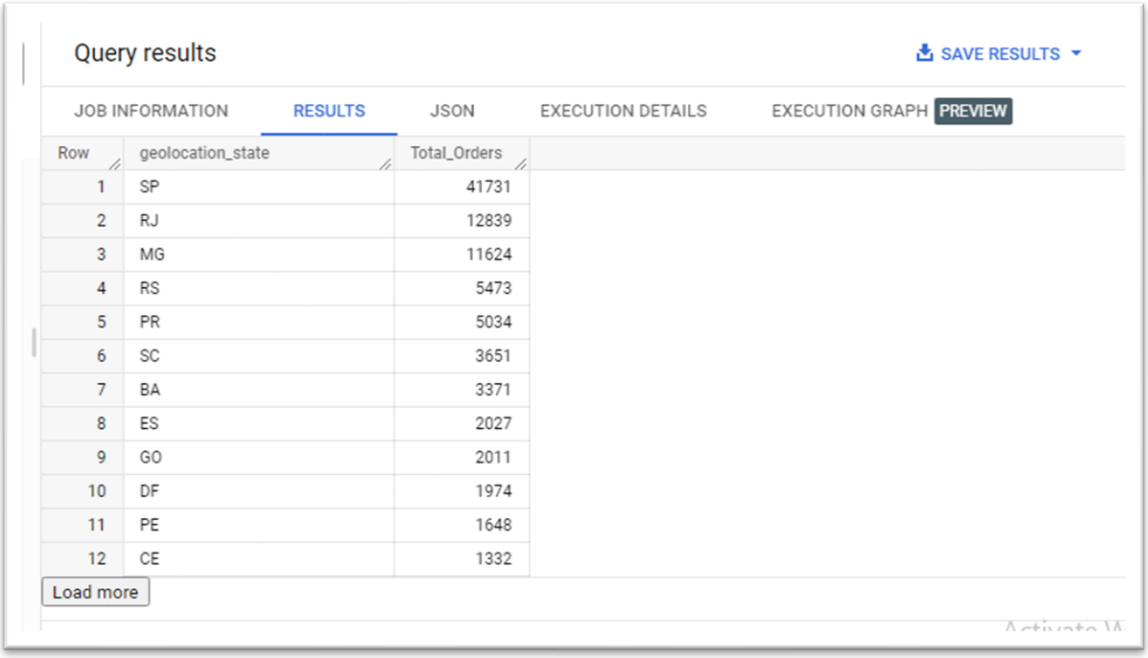


Query results

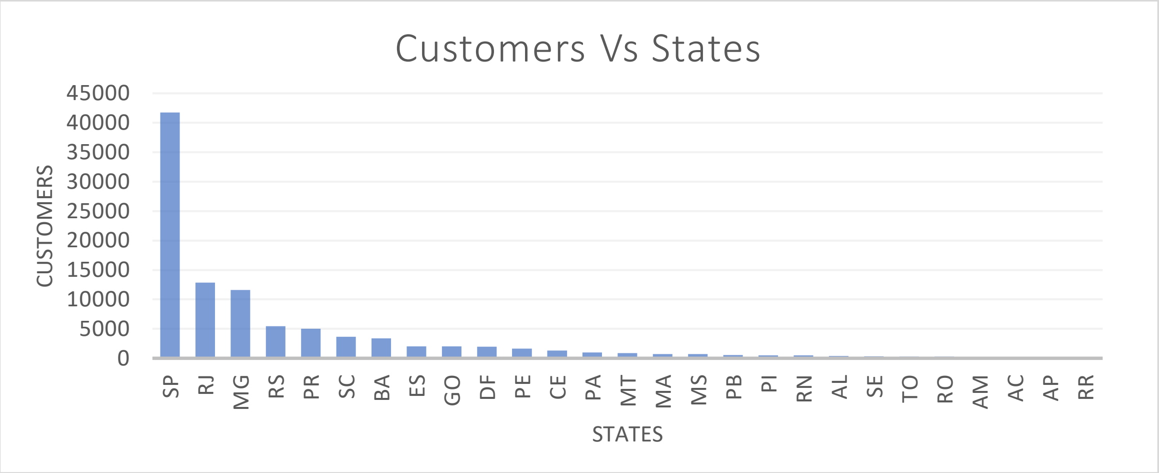
Row	geolocation_state	Year	Month	Total_Order_Placed
1	SP	2016	9	492
2	RS	2016	9	103
3	RR	2016	9	65
4	DF	2016	10	305
5	RR	2016	10	65
6	SP	2016	10	16277
7	PI	2016	10	56
8	RN	2016	10	220
9	CE	2016	10	477
10	MT	2016	10	322
11	MG	2016	10	11756
12	PA	2016	10	296

2. Distribution of customers across the states in Brazil

```
SELECT g.geolocation_state, COUNT(DISTINCT(c.customer_id))AS Total_Customers

FROM `Target_SQL_Business_Case.geolocation` g INNER JOIN
`Target_SQL_Business_Case.customers` c

ON g.geolocation_zip_code_prefix=c.customer_zip_code_prefix

GROUP BY g.geolocation_state

ORDER BY Total_Customers DESC

LIMIT 100

;
```



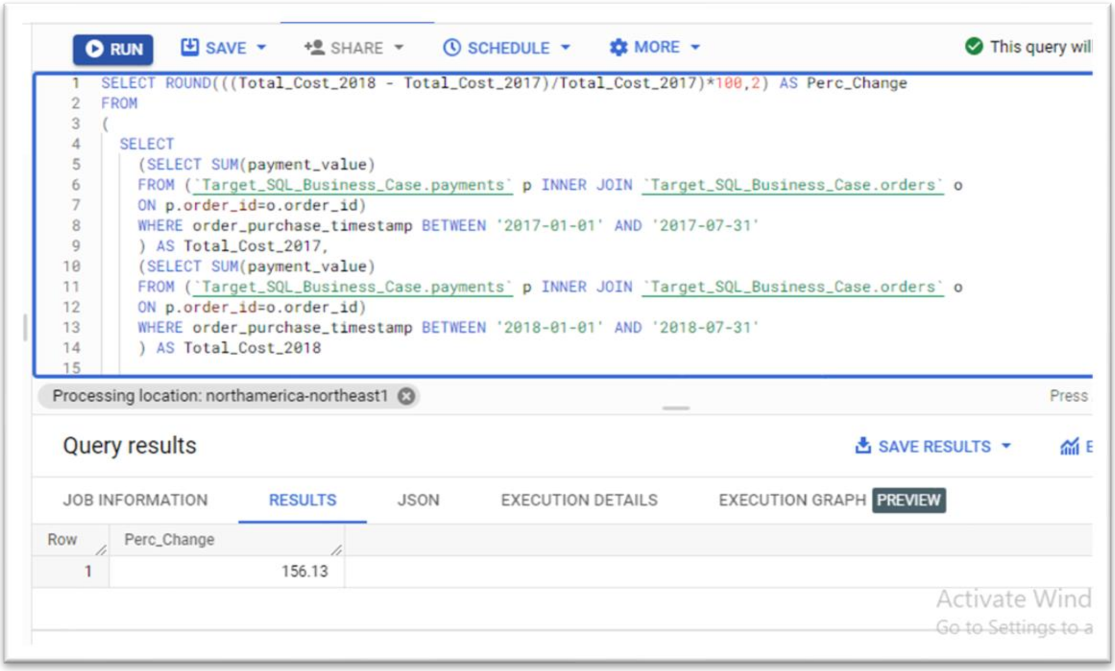
Row	geolocation_state	Total_Orders
1	SP	41731
2	RJ	12839
3	MG	11624
4	RS	5473
5	PR	5034
6	SC	3651
7	BA	3371
8	ES	2027
9	GO	2011
10	DF	1974
11	PE	1648
12	CE	1332



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

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

```
SELECT ROUND(((Total_Cost_2018 - Total_Cost_2017)/Total_Cost_2017)*100,2) AS
Perc_Change

FROM

(

 SELECT

  (SELECT SUM(payment_value)

  FROM (`Target_SQL_Business_Case.payments` p INNER JOIN
`Target_SQL_Business_Case.orders` o

  ON p.order_id=o.order_id)

  WHERE order_purchase_timestamp BETWEEN '2017-01-01' AND '2017-07-31'

  ) AS Total_Cost_2017,

  (SELECT SUM(payment_value)

  FROM (`Target_SQL_Business_Case.payments` p INNER JOIN
`Target_SQL_Business_Case.orders` o

  ON p.order_id=o.order_id)

  WHERE order_purchase_timestamp BETWEEN '2018-01-01' AND '2018-07-31'

  ) AS Total_Cost_2018

 FROM (`Target_SQL_Business_Case.payments` p INNER JOIN
`Target_SQL_Business_Case.orders` o

 ON p.order_id=o.order_id)

 LIMIT 1

) AS temp ;
```

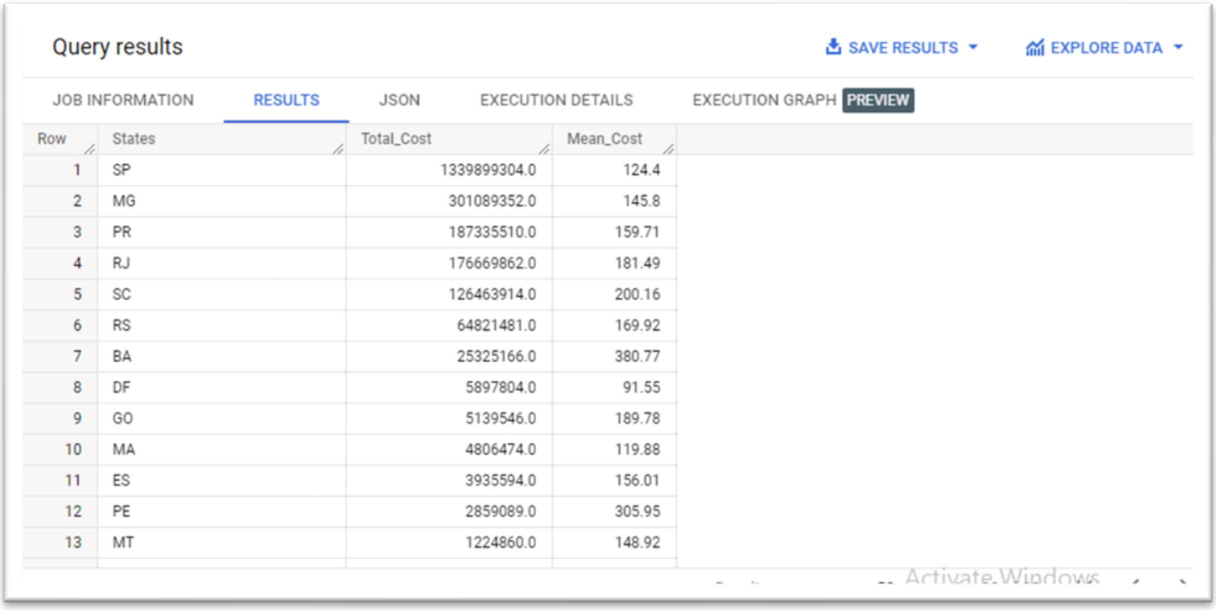


```sql
SELECT ROUND(((Total_Cost_2018 - Total_Cost_2017)/Total_Cost_2017)*100,2) AS Perc_Change
FROM
(
  SELECT
    (SELECT SUM(payment_value)
    FROM (`Target_SQL_Business_Case.payments` p INNER JOIN `Target_SQL_Business_Case.orders` o
    ON p.order_id=o.order_id)
    WHERE order_purchase_timestamp BETWEEN '2017-01-01' AND '2017-07-31'
    ) AS Total_Cost_2017,
    (SELECT SUM(payment_value)
    FROM (`Target_SQL_Business_Case.payments` p INNER JOIN `Target_SQL_Business_Case.orders` o
    ON p.order_id=o.order_id)
    WHERE order_purchase_timestamp BETWEEN '2018-01-01' AND '2018-07-31'
    ) AS Total_Cost_2018

```

Row	Perc_Change
1	156.13

2. Mean & Sum of price and freight value by customer state

```
SELECT geolocation_state AS States,ROUND(Total_Sum + Total_Freight) AS
Total_Cost,ROUND(((Total_Sum / No_of_Sum)+(Total_Freight/No_of_Sum)),2) AS
Mean_Cost

FROM (

SELECT g.geolocation_state, ROUND(SUM(o.price),2) AS
Total_Sum,ROUND(SUM(o.freight_value),2)AS Total_Freight,COUNT(o.price) AS
No_of_Sum

FROM `Target_SQL_Business_Case.order_items` o INNER JOIN
`Target_SQL_Business_Case.sellers` s

ON o.seller_id = s.seller_id INNER JOIN `Target_SQL_Business_Case.geolocation` g

ON s.seller_zip_code_prefix = g.geolocation_zip_code_prefix

GROUP BY g.geolocation_state

) AS temp


ORDER BY Total_Cost DESC;
```

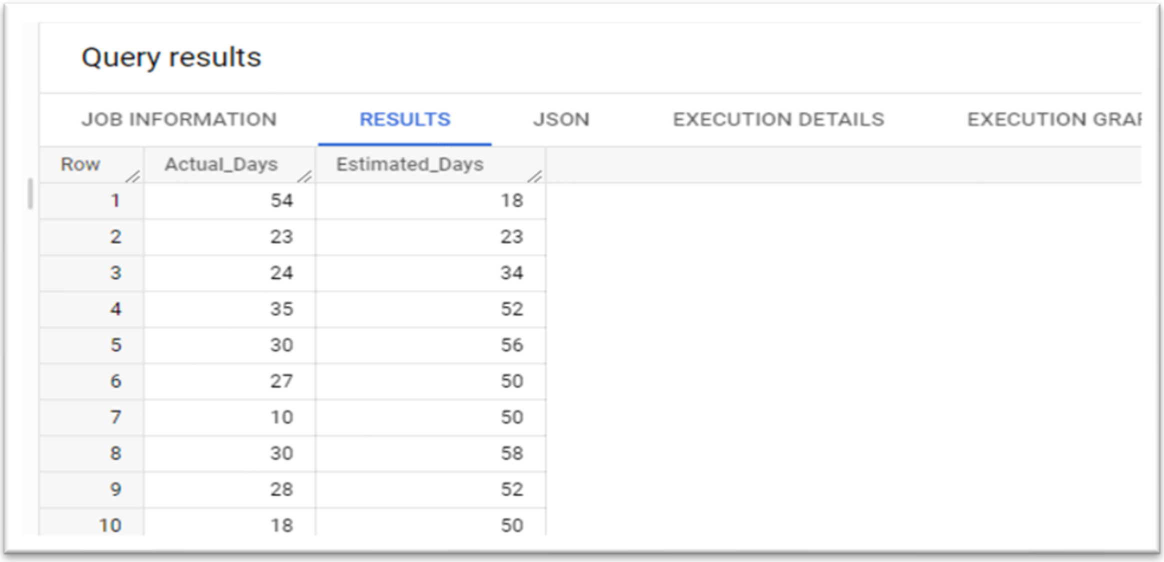


Row	States	Total_Cost	Mean_Cost
1	SP	1339899304.0	124.4
2	MG	301089352.0	145.8
3	PR	187335510.0	159.71
4	RJ	176669862.0	181.49
5	SC	126463914.0	200.16
6	RS	64821481.0	169.92
7	BA	25325166.0	380.77
8	DF	5897804.0	91.55
9	GO	5139546.0	189.78
10	MA	4806474.0	119.88
11	ES	3935594.0	156.01
12	PE	2859089.0	305.95
13	MT	1224860.0	148.92

5. Analysis on sales, freight and delivery time

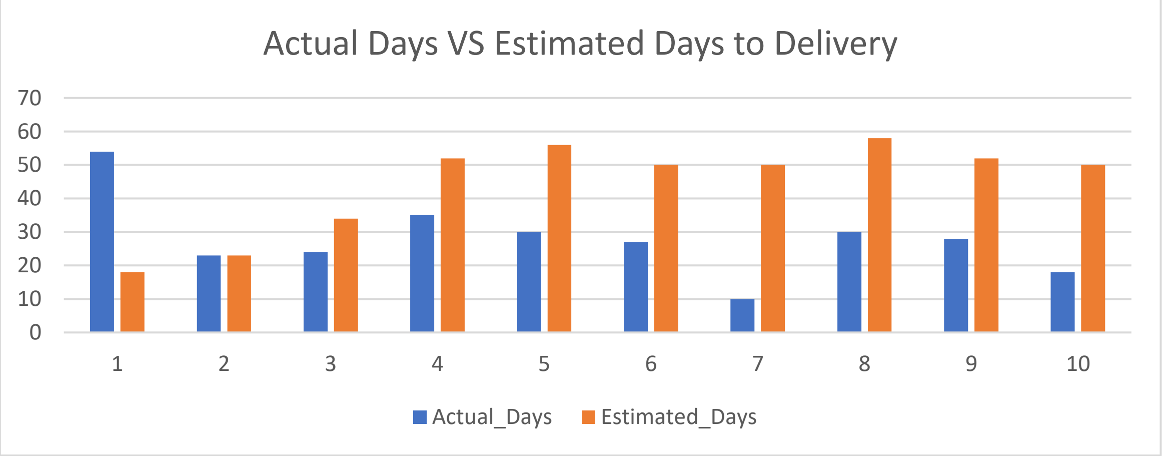
1. Calculate days between purchasing, delivering and estimated delivery

```
SELECT DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)
AS Actual_Days,

DATE_DIFF(order_estimated_delivery_date,order_purchase_timestamp,DAY) AS
Estimated_Days



FROM `Target_SQL_Business_Case.orders`

WHERE order_delivered_customer_date IS NOT null

ORDER BY order_purchase_timestamp

LIMIT 10 ;
```



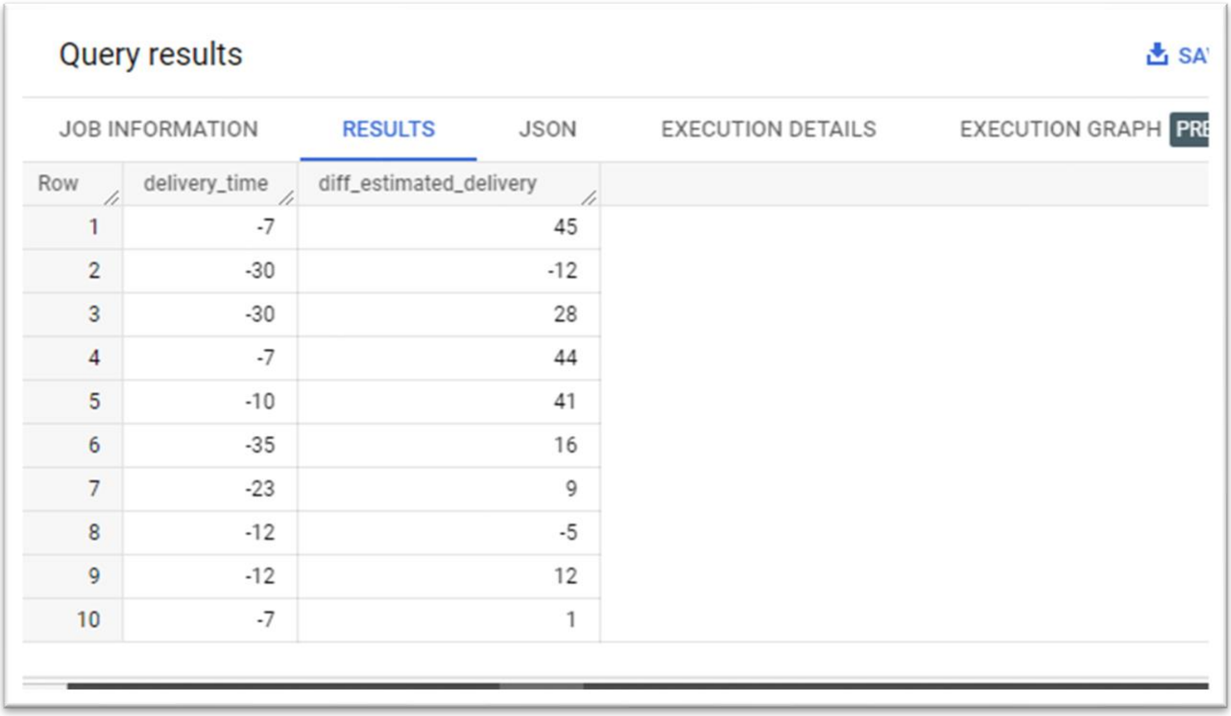
Query results

Row	Actual_Days	Estimated_Days
1	54	18
2	23	23
3	24	34
4	35	52
5	30	56
6	27	50
7	10	50
8	30	58
9	28	52
10	18	50



2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:
    - time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
    - diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

```
SELECT DATE_DIFF (order_purchase_timestamp,order_delivered_customer_date,DAY)
AS delivery_time,

DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY) AS
diff_estimated_delivery

FROM `Target_SQL_Business_Case.orders`

WHERE order_delivered_customer_date IS NOT null

LIMIT 10

;
```

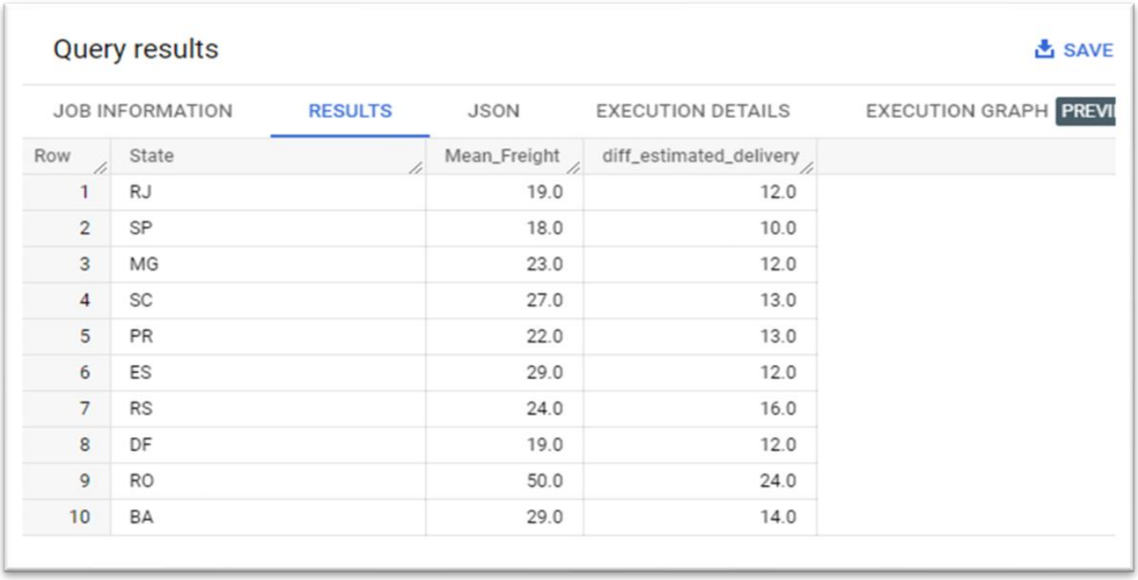


Query results

Row	delivery_time	diff_estimated_delivery
1	-7	45
2	-30	-12
3	-30	28
4	-7	44
5	-10	41
6	-35	16
7	-23	9
8	-12	-5
9	-12	12
10	-7	1

3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
SELECT geolocation_state AS State,ROUND(AVG(freight_value)) AS
Mean_Freight,ROUND(AVG (diff_estimated_delivery)) AS diff_estimated_delivery

FROM

(

  SELECT g.geolocation_state,i.freight_value,

  DATE_DIFF (o.order_purchase_timestamp,o.order_delivered_customer_date,DAY) AS
delivery_time,

  DATE_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date,DAY)
AS diff_estimated_delivery

  FROM `Target_SQL_Business_Case.orders` o INNER JOIN
`Target_SQL_Business_Case.order_items` i

  ON o.order_id =i.order_id INNER JOIN `Target_SQL_Business_Case.sellers` s

  ON i.seller_id = s.seller_id INNER JOIN `Target_SQL_Business_Case.geolocation` g

  ON s.seller_zip_code_prefix = g.geolocation_zip_code_prefix

  WHERE o.order_delivered_customer_date IS NOT null

) AS temp

GROUP BY geolocation_state

LIMIT 10 ;
```

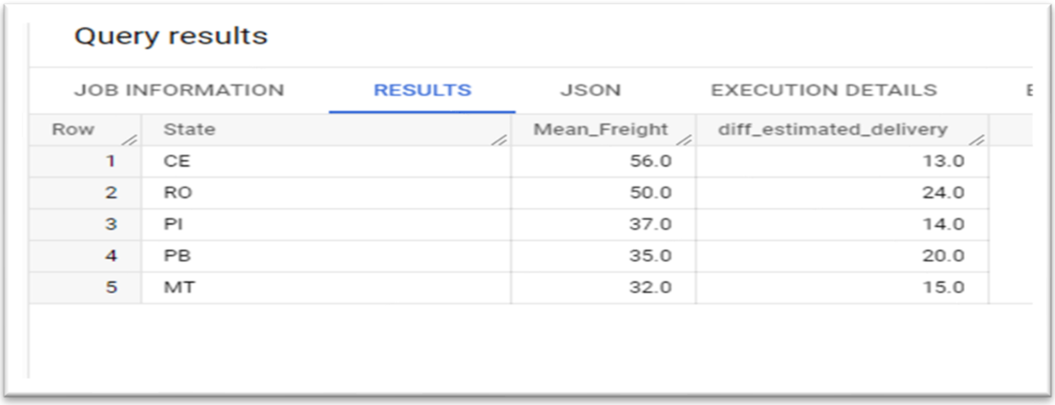


Row	State	Mean_Freight	diff_estimated_delivery
1	RJ	19.0	12.0
2	SP	18.0	10.0
3	MG	23.0	12.0
4	SC	27.0	13.0
5	PR	22.0	13.0
6	ES	29.0	12.0
7	RS	24.0	16.0
8	DF	19.0	12.0
9	RO	50.0	24.0
10	BA	29.0	14.0

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

Top 5 States

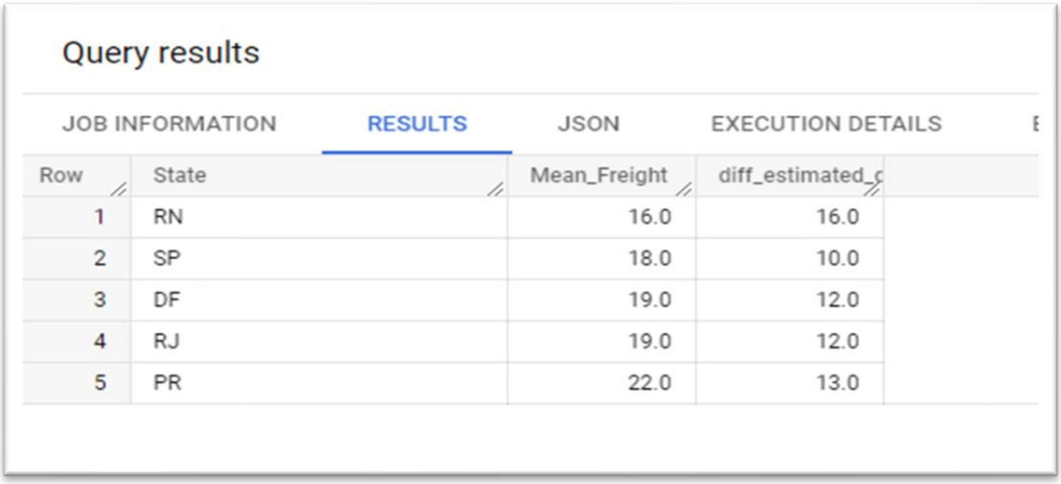
```
SELECT geolocation_state AS State,ROUND(AVG(freight_value)) AS
Mean_Freight,ROUND(AVG (diff_estimated_delivery)) AS diff_estimated_delivery

FROM   (

  SELECT g.geolocation_state,i.freight_value,

  DATE_DIFF (o.order_purchase_timestamp,o.order_delivered_customer_date,DAY) AS
delivery_time,

  DATE_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date,DAY)
AS diff_estimated_delivery

  FROM `Target_SQL_Business_Case.orders` o INNER JOIN
`Target_SQL_Business_Case.order_items` i

  ON o.order_id =i.order_id INNER JOIN `Target_SQL_Business_Case.sellers` s

  ON i.seller_id = s.seller_id INNER JOIN `Target_SQL_Business_Case.geolocation` g

  ON s.seller_zip_code_prefix = g.geolocation_zip_code_prefix

  WHERE o.order_delivered_customer_date IS NOT null

) AS temp

GROUP BY geolocation_state

ORDER BY Mean_Freight DESC

LIMIT 5

;
```



Row	State	Mean_Freight	diff_estimated_delivery
1	CE	56.0	13.0
2	RO	50.0	24.0
3	PI	37.0	14.0
4	PB	35.0	20.0
5	MT	32.0	15.0

Bottom 5 States

```
        SELECT geolocation_state AS State,ROUND(AVG(freight_value)) AS
Mean_Freight,ROUND(AVG (diff_estimated_delivery)) AS diff_estimated_delivery

FROM

(

 SELECT g.geolocation_state,i.freight_value,

 DATE_DIFF (o.order_purchase_timestamp,o.order_delivered_customer_date,DAY) AS
delivery_time,

 DATE_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date,DAY)
AS diff_estimated_delivery

 FROM `Target_SQL_Business_Case.orders` o INNER JOIN
`Target_SQL_Business_Case.order_items` i

 ON o.order_id =i.order_id INNER JOIN `Target_SQL_Business_Case.sellers` s

 ON i.seller_id = s.seller_id INNER JOIN `Target_SQL_Business_Case.geolocation` g

 ON s.seller_zip_code_prefix = g.geolocation_zip_code_prefix

 WHERE o.order_delivered_customer_date IS NOT null

) AS temp

GROUP BY geolocation_state

ORDER BY Mean_Freight

LIMIT 5

;
```



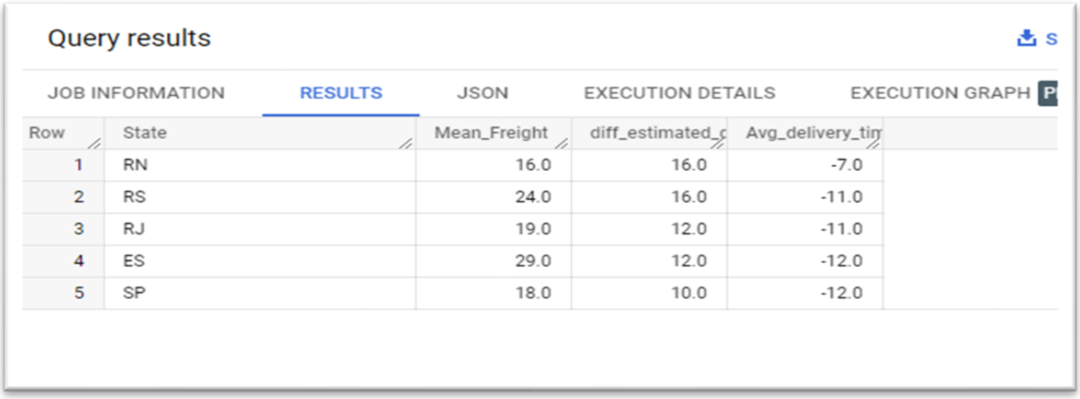
Query results

Row	State	Mean_Freight	diff_estimated_c
1	RN	16.0	16.0
2	SP	18.0	10.0
3	DF	19.0	12.0
4	RJ	19.0	12.0
5	PR	22.0	13.0

5. Top 5 states with highest/lowest average time to delivery

Top 5 States

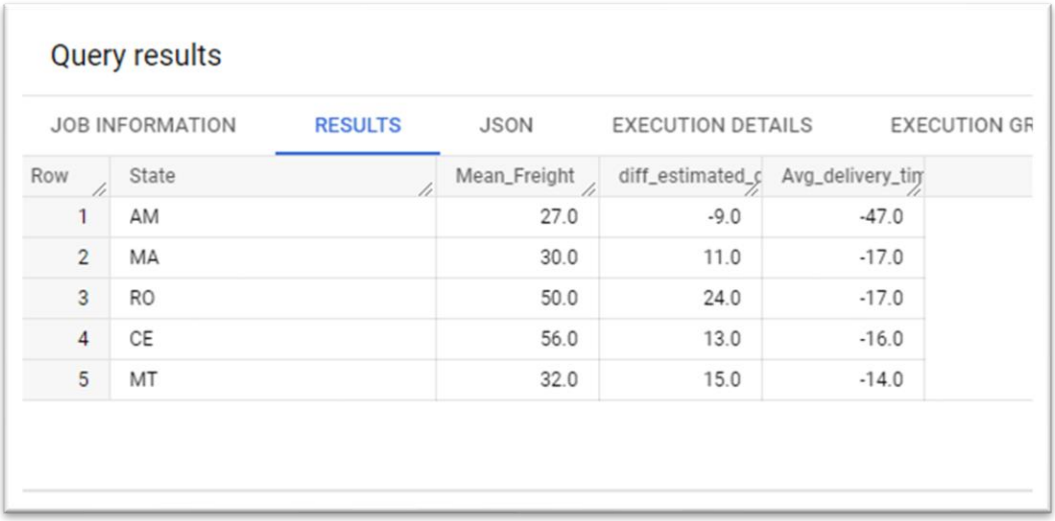
```
SELECT geolocation_state AS State,ROUND(AVG(freight_value)) AS
Mean_Freight,ROUND(AVG (diff_estimated_delivery)) AS
diff_estimated_delivery,ROUND(AVG (delivery_time)) AS Avg_delivery_time

FROM  (

  SELECT g.geolocation_state,i.freight_value,

  DATE_DIFF (o.order_purchase_timestamp,o.order_delivered_customer_date,DAY) AS
delivery_time,

  DATE_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date,DAY)
AS diff_estimated_delivery

  FROM `Target_SQL_Business_Case.orders` o INNER JOIN
`Target_SQL_Business_Case.order_items` i

  ON o.order_id =i.order_id INNER JOIN `Target_SQL_Business_Case.sellers` s

  ON i.seller_id = s.seller_id INNER JOIN `Target_SQL_Business_Case.geolocation` g

  ON s.seller_zip_code_prefix = g.geolocation_zip_code_prefix

  WHERE o.order_delivered_customer_date IS NOT null

) AS temp

GROUP BY geolocation_state

ORDER BY  Avg_delivery_time DESC

LIMIT 5

;
```



Row	State	Mean_Freight	diff_estimated_c	Avg_delivery_tim
1	RN	16.0	16.0	-7.0
2	RS	24.0	16.0	-11.0
3	RJ	19.0	12.0	-11.0
4	ES	29.0	12.0	-12.0
5	SP	18.0	10.0	-12.0

Bottom 5 States

```
        SELECT geolocation_state AS State,ROUND(AVG(freight_value)) AS
Mean_Freight,ROUND(AVG (diff_estimated_delivery)) AS
diff_estimated_delivery,ROUND(AVG (delivery_time)) AS Avg_delivery_time

FROM  (

  SELECT g.geolocation_state,i.freight_value,

  DATE_DIFF (o.order_purchase_timestamp,o.order_delivered_customer_date,DAY) AS
delivery_time,

  DATE_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date,DAY)
AS diff_estimated_delivery

  FROM `Target_SQL_Business_Case.orders` o INNER JOIN
`Target_SQL_Business_Case.order_items` i

  ON o.order_id =i.order_id INNER JOIN `Target_SQL_Business_Case.sellers` s

  ON i.seller_id = s.seller_id INNER JOIN `Target_SQL_Business_Case.geolocation` g

  ON s.seller_zip_code_prefix = g.geolocation_zip_code_prefix

  WHERE o.order_delivered_customer_date IS NOT null

) AS temp

GROUP BY geolocation_state

ORDER BY  Avg_delivery_time

LIMIT 5

;
```

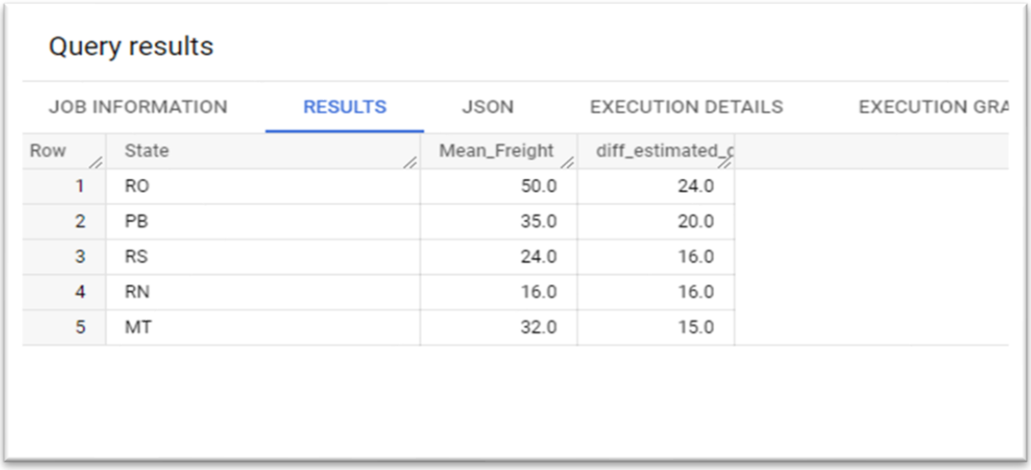


Row	State	Mean_Freight	diff_estimated_c	Avg_delivery_tim
1	AM	27.0	-9.0	-47.0
2	MA	30.0	11.0	-17.0
3	RO	50.0	24.0	-17.0
4	CE	56.0	13.0	-16.0
5	MT	32.0	15.0	-14.0

6. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Top 5 States

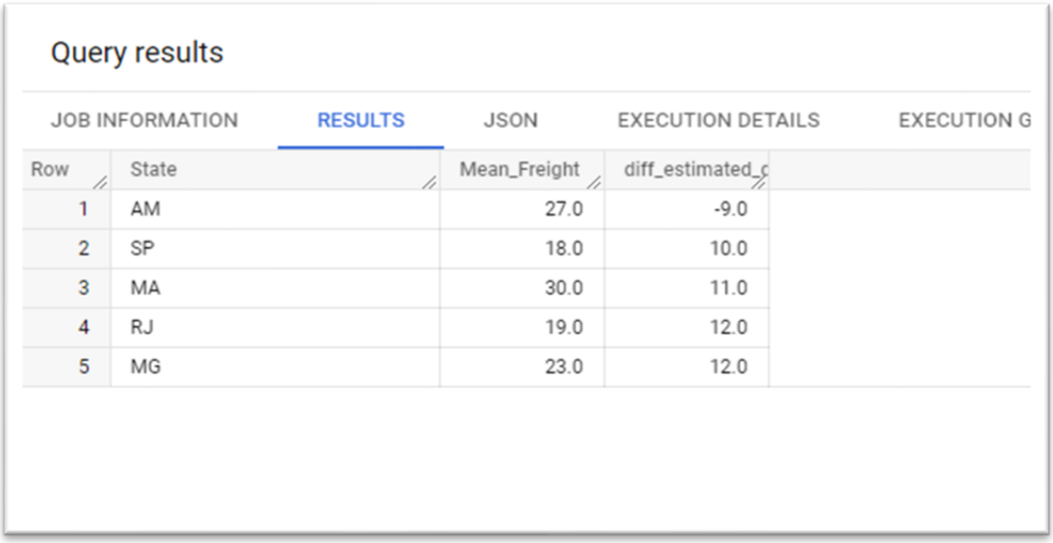
```
SELECT geolocation_state AS State,ROUND(AVG(freight_value)) AS
Mean_Freight,ROUND(AVG (diff_estimated_delivery)) AS diff_estimated_delivery

FROM (

 SELECT g.geolocation_state,i.freight_value,

 DATE_DIFF (o.order_purchase_timestamp,o.order_delivered_customer_date,DAY) AS
delivery_time,

 DATE_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date,DAY)
AS diff_estimated_delivery

 FROM `Target_SQL_Business_Case.orders` o INNER JOIN
`Target_SQL_Business_Case.order_items` i

 ON o.order_id =i.order_id INNER JOIN `Target_SQL_Business_Case.sellers` s

 ON i.seller_id = s.seller_id INNER JOIN `Target_SQL_Business_Case.geolocation` g

 ON s.seller_zip_code_prefix = g.geolocation_zip_code_prefix

 WHERE o.order_delivered_customer_date IS NOT null

) AS temp

GROUP BY geolocation_state

ORDER BY  diff_estimated_delivery DESC

LIMIT 5  ;
```



Row	State	Mean_Freight	diff_estimated_c
1	RO	50.0	24.0
2	PB	35.0	20.0
3	RS	24.0	16.0
4	RN	16.0	16.0
5	MT	32.0	15.0

Bottom 5 States

```
SELECT geolocation_state AS State,ROUND(AVG(freight_value)) AS Mean_Freight,ROUND(AVG (diff_estimated_delivery)) AS diff_estimated_delivery

FROM  (

  SELECT g.geolocation_state,i.freight_value,

  DATE_DIFF (o.order_purchase_timestamp,o.order_delivered_customer_date,DAY) AS delivery_time,

  DATE_DIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date,DAY) AS diff_estimated_delivery

  FROM `Target_SQL_Business_Case.orders` o INNER JOIN `Target_SQL_Business_Case.order_items` i

  ON o.order_id =i.order_id INNER JOIN `Target_SQL_Business_Case.sellers` s

  ON i.seller_id = s.seller_id INNER JOIN `Target_SQL_Business_Case.geolocation` g

  ON s.seller_zip_code_prefix = g.geolocation_zip_code_prefix

  WHERE o.order_delivered_customer_date IS NOT null

) AS temp

GROUP BY geolocation_state

ORDER BY  diff_estimated_delivery

LIMIT 5

;
```

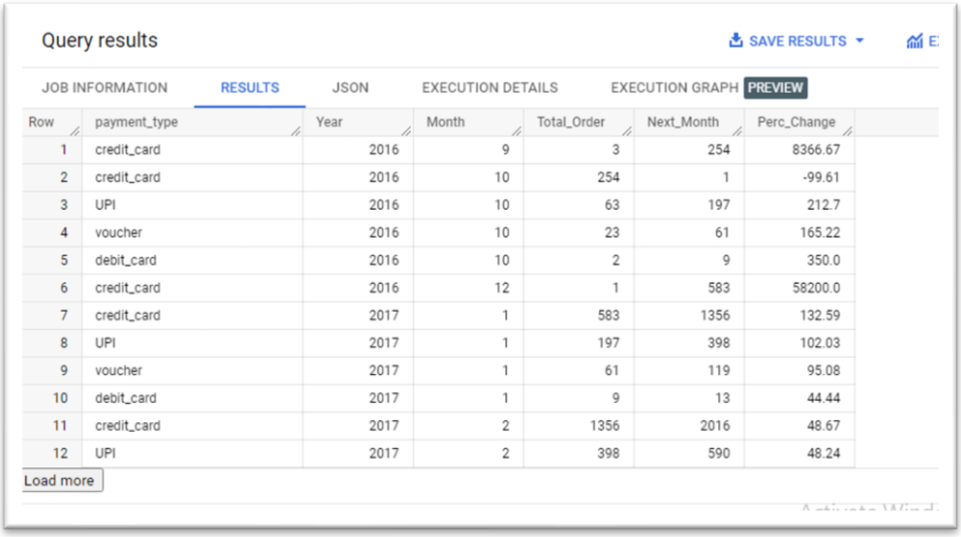


Row	State	Mean_Freight	diff_estimated_c
1	AM	27.0	-9.0
2	SP	18.0	10.0
3	MA	30.0	11.0
4	RJ	19.0	12.0
5	MG	23.0	12.0

6. Payment type analysis:

1. Month over Month count of orders for different payment types

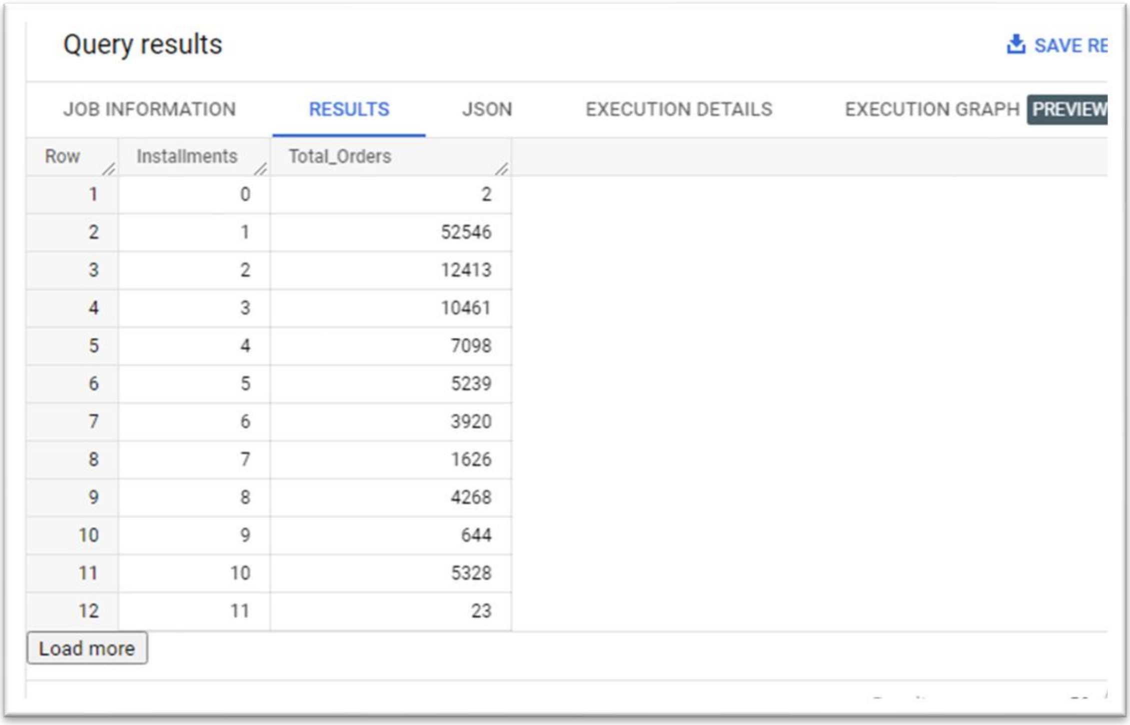
```
SELECT *, ROUND(((Next_Month-Total_Order)/Total_Order)*100,2) AS Perc_Change

FROM (

 SELECT *,LEAD(Total_Order,1,1) OVER(PARTITION BY payment_type ORDER BY
Year,Month) AS Next_Month

 FROM

 (

  SELECT payment_type,EXTRACT(year FROM order_purchase_timestamp) AS Year,

  EXTRACT(month FROM order_purchase_timestamp) AS Month,

  COUNT ( o.order_id) AS Total_Order

  FROM `Target_SQL_Business_Case.orders` o INNER JOIN
`Target_SQL_Business_Case.payments` p

  ON o.order_id = p.order_id

  GROUP BY payment_type,Month,Year

 ) AS tem

 ORDER BY Year,Month

) AS temp ;
```



Row	payment_type	Year	Month	Total_Order	Next_Month	Perc_Change
1	credit_card	2016	9	3	254	8366.67
2	credit_card	2016	10	254	1	-99.61
3	UPI	2016	10	63	197	212.7
4	voucher	2016	10	23	61	165.22
5	debit_card	2016	10	2	9	350.0
6	credit_card	2016	12	1	583	58200.0
7	credit_card	2017	1	583	1356	132.59
8	UPI	2017	1	197	398	102.03
9	voucher	2017	1	61	119	95.08
10	debit_card	2017	1	9	13	44.44
11	credit_card	2017	2	1356	2016	48.67
12	UPI	2017	2	398	590	48.24

2. Count of orders based on the no. of payment installments

```
SELECT payment_installments AS Installments,COUNT(order_id) AS Total_Orders

FROM `Target_SQL_Business_Case.payments`

GROUP BY payment_installments

ORDER BY payment_installments

;
```



Row	Installments	Total_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
11	10	5328
12	11	23

Insights :

1. In Brazil up to the year 2017 end E-Commerce Business is at its peak and then settled stable till the year 2018.
2. Nov-Dec these are the months were business on its peak because of Christmas and New year.
3. Delivery and Transport department doing well as the Actual Delivery Time is less than the Estimated Delivery to Customers.
4. It is a good sign to further grow in companies business
5. States in the southern region have plenty no of customers than the northern region
6. Ceará, Rondônia, Piauí, Paraíba, and Mato Grosso are the name of the states where freight value is more.
7. Rio Grande do Norte, São Paulo, Distrito Federal, Rio de Janeiro, Paraná are the states where freight rates are minimum.
8. Rio Grande do Norte, Rio Grande do Sul, Rio de Janeiro, Espírito Santo, São Paulo states where average delivery time is less.
9. Amazonas, Maranhão, Rondônia, Ceará, Mato Grosso states where actual delivery time is more
10. Rondônia, Paraíba, Rio Grande do Sul, Rio Grande do Norte, Mato Grosso are the states where estimated delivery time is more.
11. Amazonas, São Paulo, Maranhão, Rio de Janeiro, Minas Gerais are the states where estimated delivery time is less.
12. Most of the peoples uses minimum no of installments for their items.

Recommendations :

1. Before Nov-Dec-Jan company should increase the number of employees, and increase the inventory and goods.
2. The company should take special focus on freight, transport, and delivery during these months.
3. In the northern region too many opportunities to build a business.
4. Ceará, Rondônia, Piauí, Paraíba, and Mato Grosso We should lower the freight rates, so we can reach more customers to grow business in this state.
5. Amazonas, Maranhão, Rondônia, Ceará, Mato Grosso In these states actual delivery time is more so need to work on that
6. We need to focus on how to minimize estimated delivery time by making necessary changes in the department.
7. As we lower the interest rates and increase installments make more convenient and flexible way to buy
8. We need to offer modern payment options which benefits our customer.
9. Providing a variety of payment options opens the door for more customers and could lead business long term