BUSINESS CASE USING SQL

This case study is about one of the world's most recognized brands and one of America's leading retailers. This brand makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation, and an exceptional guest experience that no other retailer can deliver.

This business case has information of 100k orders from 2016 to 2018 made by the brand in Brazil. Its features allow viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

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

    1.1 Data type of columns in a table
    1.2 Time period for which the data is given.
    1.3 Cities and States of customers ordered during the given period.

    Solution:

    We can know about the Data Type of the columns from Table SCHEMA. The following table SCHEMA shows that the whole dataset contains data types such as **STRING, INTEGER, FLOAT, TIMESTAMP**, etc.

Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

Field name	Type
geolocation_zip_code_prefix	INTEGER
geolocation_lat	FLOAT
geolocation_lng	FLOAT
geolocation_city	STRING
geolocation_state	STRING

Field name	Type
order_id	STRING
order_item_id	INTEGER
product_id	STRING
seller_id	STRING
shipping_limit_date	TIMESTAMP
price	FLOAT
freight_value	FLOAT

Field name	Type
order_id	STRING
customer_id	STRING
order_status	STRING
order_purchase_timestamp	TIMESTAMP
order_approved_at	TIMESTAMP
order_delivered_carrier_date	TIMESTAMP
order_delivered_customer_date	TIMESTAMP
order_estimated_delivery_date	TIMESTAMP

Field name	Type
order_id	STRING
payment_sequential	INTEGER
payment_type	STRING
payment_installments	INTEGER
payment_value	FLOAT

Field name	Type
product_id	STRING
product_category	STRING
product_name_length	INTEGER
product_description_length	INTEGER
product_photos_qty	INTEGER
product_weight_g	INTEGER
product_length_cm	INTEGER
product_height_cm	INTEGER
product_width_cm	INTEGER

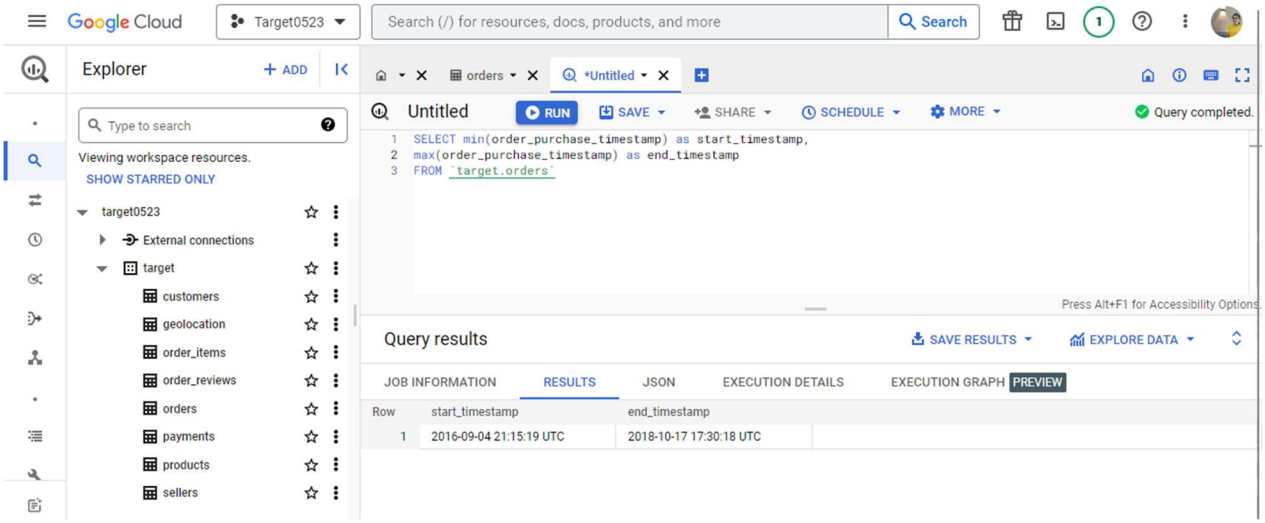
Field name	Type
seller_id	STRING
seller_zip_code_prefix	INTEGER
seller_city	STRING
seller_state	STRING

Field name	Type
review_id	STRING
order_id	STRING
review_score	INTEGER
review_comment_title	STRING
review_creation_date	TIMESTAMP
review_answer_timestamp	TIMESTAMP

The time period for which the data is given -

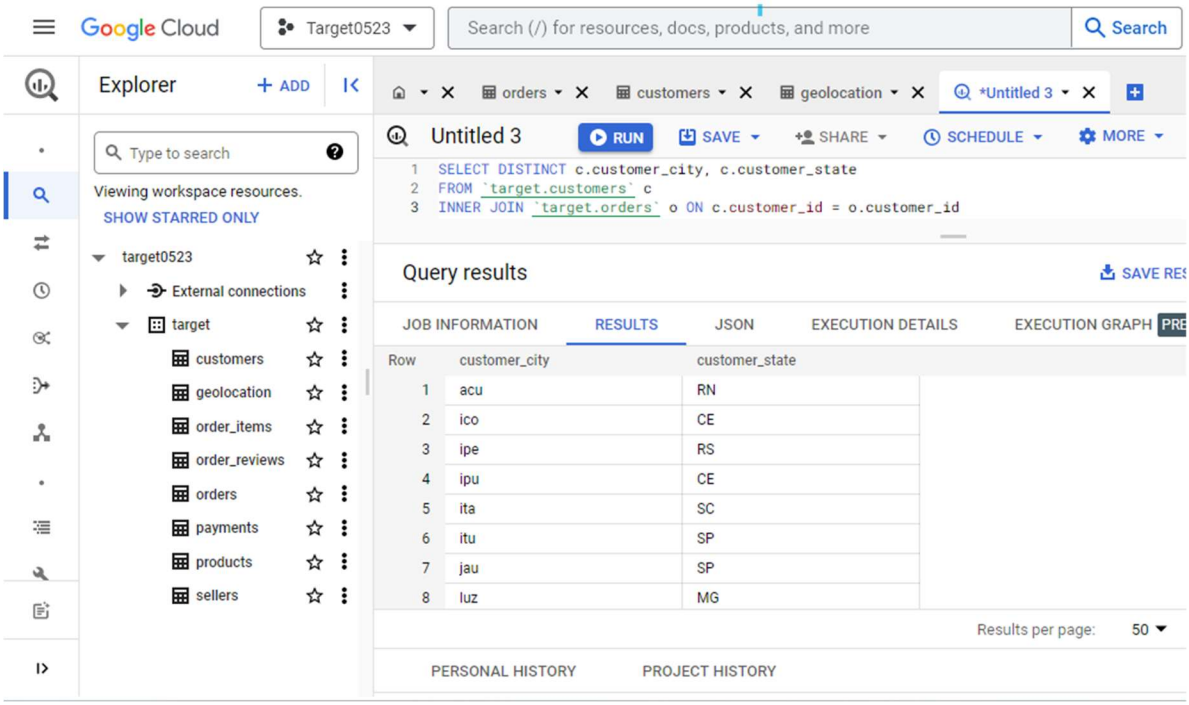
From min(order_purchase_timestamp) and max(order_purchase_timestamp), we can get the total period for which the data is given.

The data is given from **4th September 2016 to 17th October 2018.**



Cities and States of customers ordered during the given period –

To get the cities and states of the customers who were active during the time period, let's join the customer and order table over the customer_id and get the distinct customer_city and customer_state from which we got the orders.



INSIGHT:

- The dataset consists of the data types such as **STRING, INTEGER, FLOAT, and TIMESTAMP.**
- We received the dataset from the period **September 2016 to October 2018.**

2. In-depth Exploration:

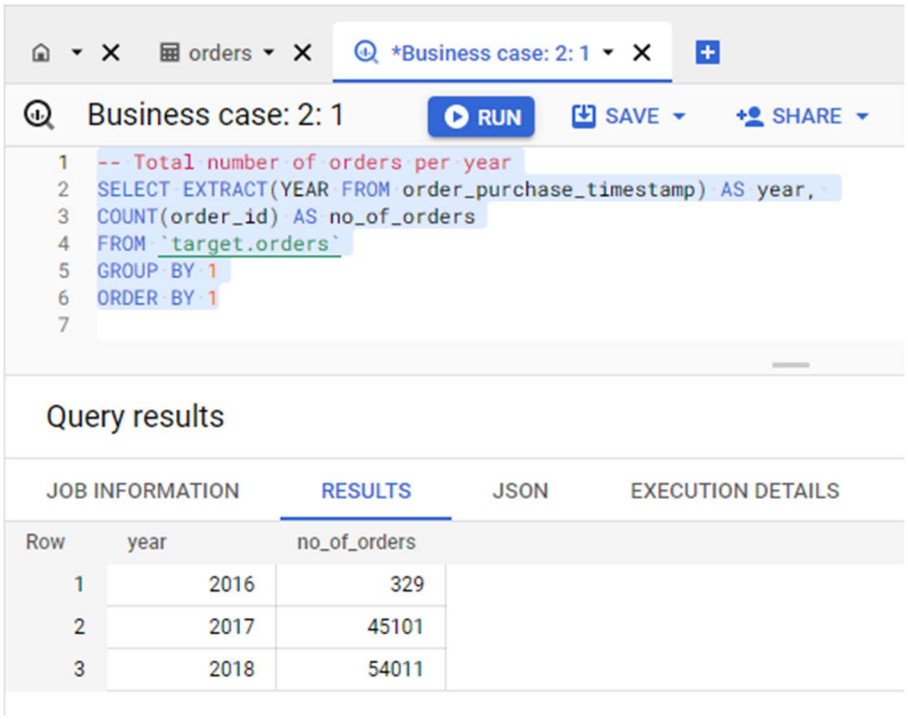
   2.1 Is there a growing trend in e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

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

   0-7: Dawn
   7-12: Morning
   12-18: Afternoon
   18-23:Night

Solution:

To see if there is a growing trend or not, we need to see if there is any increase in the number of orders over the years.



orders | *Business case: 2: 1

Business case: 2: 1 RUN SAVE SHARE

```sql
-- Total number of orders in a month
SELECT a.year, a.month, a.no_of_orders,
SUM(a.no_of_orders) OVER (partition by a.year) AS orders_per_ye
FROM
  (SELECT EXTRACT(YEAR FROM order_purchase_timestamp) AS year,
  EXTRACT(MONTH FROM order_purchase_timestamp) AS month,
  COUNT(order_id) AS no_of_orders
  FROM `target.orders`
  GROUP BY 1, 2) AS a
ORDER BY 1 ASC, 3 DESC
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	year	month	no_of_orders	orders_per_year
14	2017	2	1780	45101
15	2017	1	800	45101
16	2018	1	7269	54011

orders | *Business case: 2: 1

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	year	month	no_of_orders	orders_per_year
1	2016	10	324	329
2	2016	9	4	329
3	2016	12	1	329
4	2017	11	7544	45101
5	2017	12	5673	45101
6	2017	10	4631	45101
7	2017	8	4331	45101
8	2017	9	4285	45101
9	2017	7	4026	45101
10	2017	5	3700	45101
11	2017	6	3245	45101
12	2017	3	2682	45101

Query results

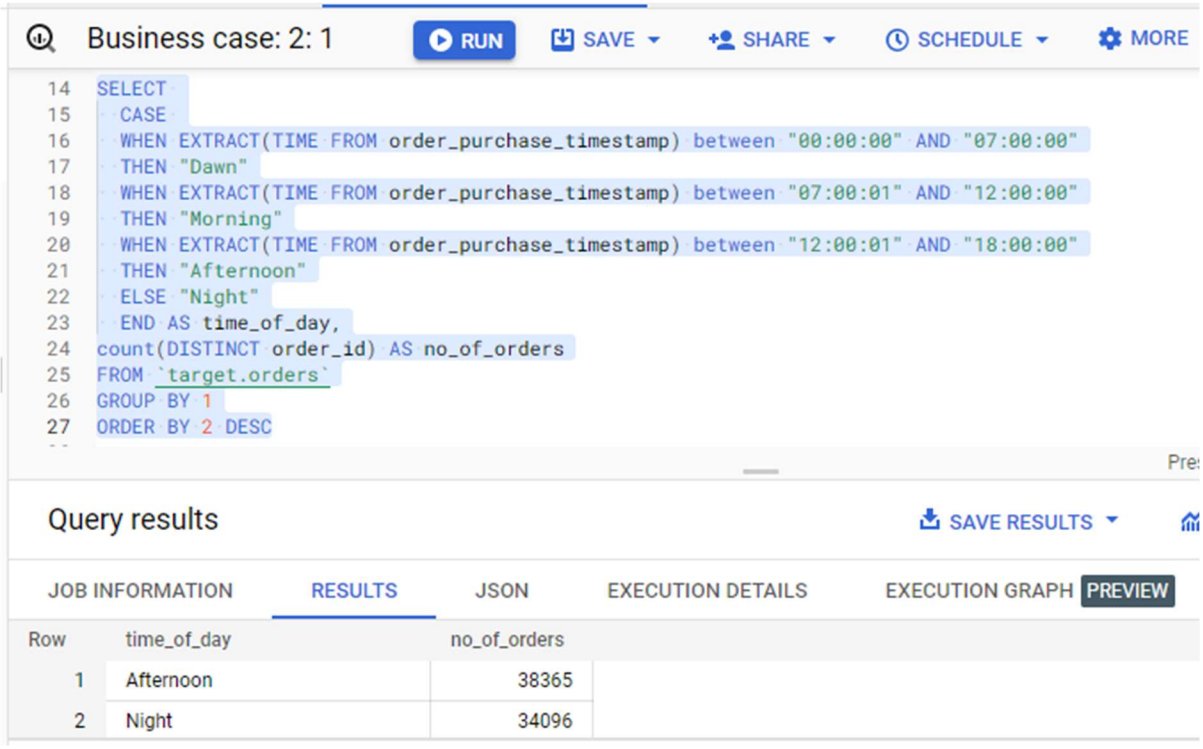
JOB INFORMATION | RESULTS | JSON | EXECUTION DET

Row	year	month	no_of_orders	orders_per_year
14	2017	2	1780	45101
15	2017	1	800	45101
16	2018	1	7269	54011
17	2018	3	7211	54011
18	2018	4	6939	54011
19	2018	5	6873	54011
20	2018	2	6728	54011
21	2018	8	6512	54011
22	2018	7	6292	54011
23	2018	6	6167	54011
24	2018	9	16	54011
25	2018	10	4	54011

Insight:

- As we do not have sufficient data for the year 2016 (data of only 3 months have been given), we cannot infer that there is growth in e-commerce, but from the overall growth of 2017 to 2018, we can say that there is growth in e-commerce in Brazil.
- We can see from the 2nd query that the highest number of orders were towards the end of the year 2017 and the start of the year 2018, i.e., during the winter season.
- But in 2018, the number of orders was decreasing over the year.

2.1 What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon, or Night)?
0-7: Dawn
7-12: Morning
12-18: Afternoon
18-23:Night



Query results

JOB INFORMATION | RESULTS | JSON | EXECU

Row	time_of_day	no_of_orders
1	Afternoon	38365
2	Night	34096
3	Morning	21738
4	Dawn	5242

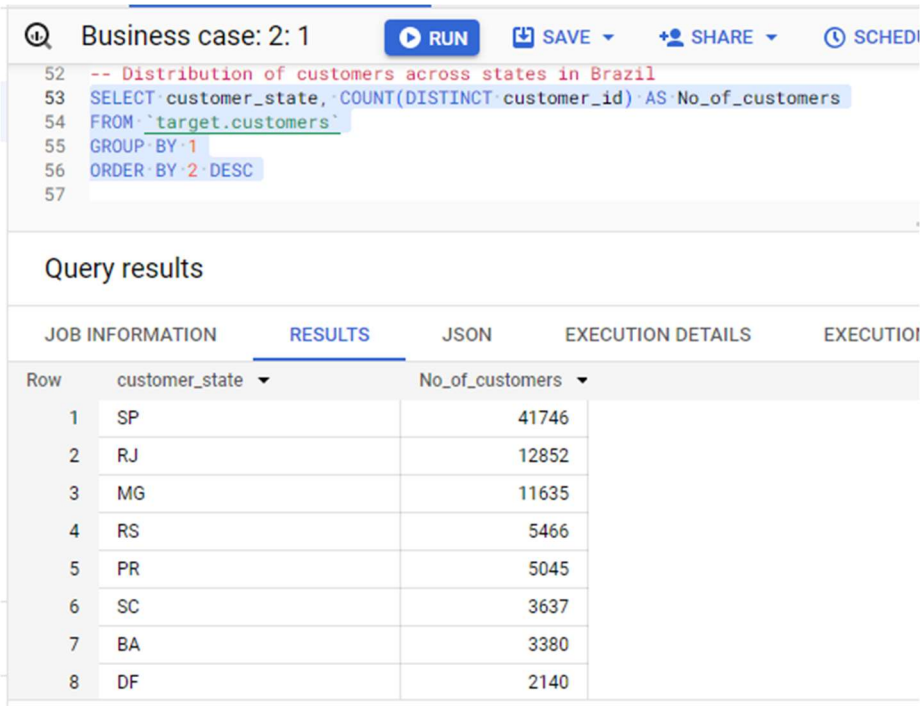
PERSONAL HISTORY | PROJECT HISTORY

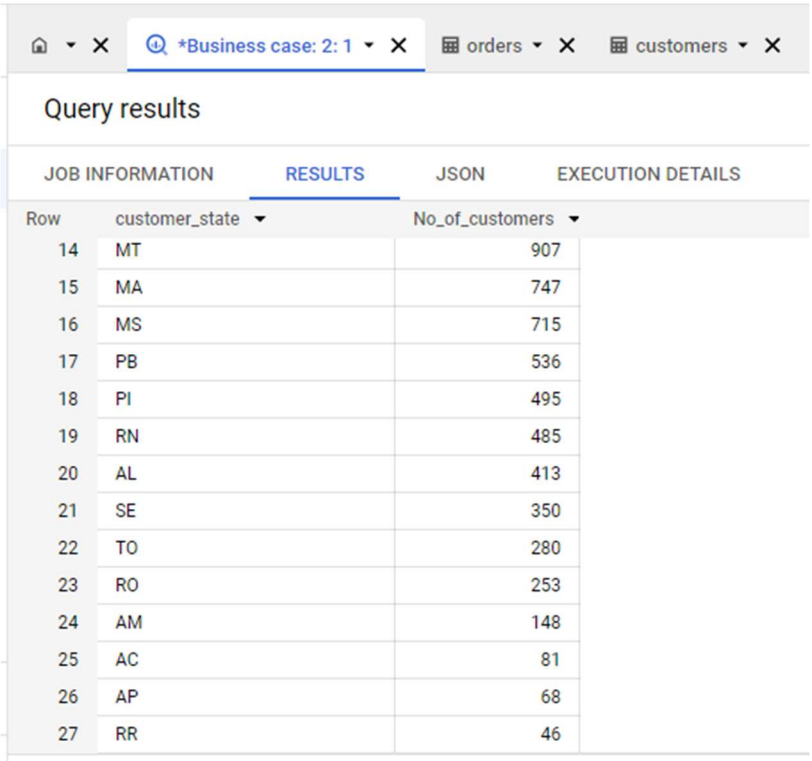
Insight:

We can clearly see from the result that the customers tend to buy more during the afternoon time.

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

    3.1 Get month-on-month orders by state.
    3.2 Distribution of customers across the states in Brazil.





*Business case: 2: 1 | orders | customers

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	No_of_customers
14	MT	907
15	MA	747
16	MS	715
17	PB	536
18	PI	495
19	RN	485
20	AL	413
21	SE	350
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46

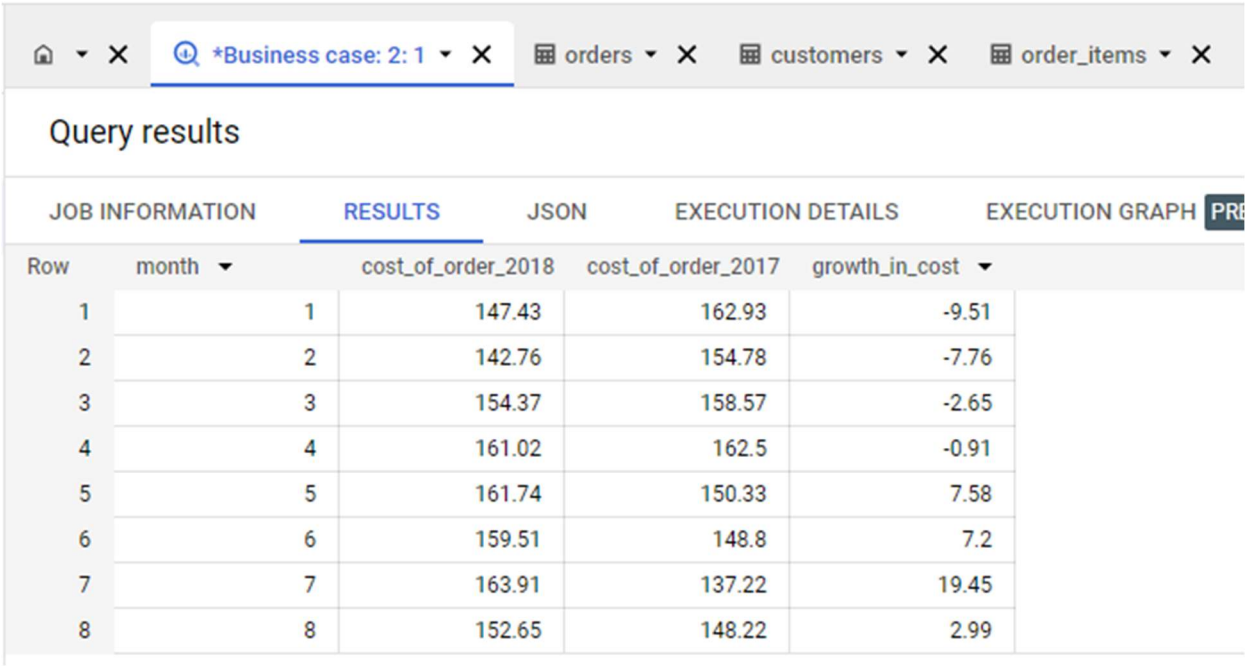
4 Impact on the Economy: Analyze the money movement by e-commerce by looking at order prices, freight, and others.

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

4.2 Mean & Sum of price and freight value by customer states.

Business case: 2: 1 — RUN | SAVE | SHARE | SCHEDULE | MORE

```
58  -- 4.1 Get a % increase in the cost of orders from 2017 to 2018 (include months between Jan to Aug only) - Y
    payments table.
59  SELECT
60  b2.month, ROUND(b2.payment_value_2018, 2) cost_of_order_2018,
61  ROUND(b2.payment_value_2017, 2) cost_of_order_2017,
62  ROUND((((b2.payment_value_2018 - b2.payment_value_2017)/payment_value_2017)*100), 2) AS growth_in_cost
63  FROM
64  (SELECT
65  b1.month,b1.payment_value AS payment_value_2018,
66  LAG(b1.payment_value) OVER(partition by month ORDER BY year) AS payment_value_2017
67  FROM
68  (SELECT
69  EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
70  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
71  AVG(p.payment_value) payment_value
72  FROM `target.orders` o
73  INNER JOIN `target.payments` p
74  ON o.order_id = p.order_id
75  GROUP BY 1, 2
76  ORDER BY 1, 2) AS b1
77  WHERE b1.year IN (2017, 2018) AND b1.month IN (1, 2, 3, 4, 5, 6, 7, 8)
78  ORDER BY 2) AS b2
79  WHERE NOT b2.payment_value_2017 IS null
80  ORDER BY 1
```



Row	month	cost_of_order_2018	cost_of_order_2017	growth_in_cost
1	1	147.43	162.93	-9.51
2	2	142.76	154.78	-7.76
3	3	154.37	158.57	-2.65
4	4	161.02	162.5	-0.91
5	5	161.74	150.33	7.58
6	6	159.51	148.8	7.2
7	7	163.91	137.22	19.45
8	8	152.65	148.22	2.99

Insight:

- **For the first four months of 2018, the cost of orders decreased, but after that, it increased.**

Mean & Sum of price and freight value by customer states -

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_state	total_price	avg_price	total_frieght_value	avg_freight_value
1	RN	83034.98	156.97	18860.1	35.65
2	CE	227254.71	153.76	48351.59	32.71
3	RS	750304.02	120.34	135522.74	21.74

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	customer_state	total_price	avg_price	total_frieght_value	avg_freight_value
4	SC	520553.34	124.65	89660.26	21.47
5	SP	5202955.05	109.65	718723.07	15.15
6	MG	1585308.03	120.75	270853.46	20.63
7	BA	511349.99	134.6	100156.68	26.36
8	RJ	1824092.67	125.12	305589.31	20.96
9	GO	294591.95	126.27	53114.98	22.77
10	MA	119648.22	145.2	31523.77	38.26
11	PE	262788.03	145.51	59449.66	32.92
12	PB	115268.08	191.48	25719.73	42.72
13	ES	275037.31	121.91	49764.6	22.06
14	PR	683083.76	119.0	117851.68	20.53
15	RO	46140.64	165.97	11417.38	41.07
16	MS	116812.64	142.63	19144.03	23.37
17	PA	178947.81	165.69	38699.3	35.83

5. Analysis of sales, freight, and delivery time

5.1 Calculate the days between purchasing, delivery, and estimated delivery.

Business case: 2: 1 RUN SAVE SHARE SCHEDULE MORE

```
98  -- 5.1 Calculate the days between purchasing, delivery, and estimated delivery
99  SELECT
100 order_id,
101 DATE_DIFF(DATE(order_estimated_delivery_date), DATE(order_purchase_timestamp), DAY) AS estimated_delivery_days,
102 DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp), DAY) AS actual_delivery_days
103 FROM `target.orders`
104 ORDER BY 3 DESC
```

Query results SAVE

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	order_id	estimated_delivery_c	actual_delivery_days
1	ca07593549f1816d26a572e06...	29	210
2	1b3190b2dfa9d789e1f14c05b...	20	208
3	440d0d17af552815d15a9e41a...	31	196
4	285ab9426d6982034523a855f...	29	195
5	2fb597c2f772eca01b1f5c561b...	40	195
6	0f4519c5f1c541ddec9f21b3bd...	33	194
7	47b40429ed8cce3aee9199792...	16	191

INSIGHT:

- **For some orders there is actually delivery time was much more than the estimated delivery time**

5.2 Find time_to_delivery & diff_estimated_delivery.
The formula for the same is given below:
time_to_delivery = order_delivered_customer_date-order_purchase_timestamp
diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

Business case: 2: 1 RUN SAVE SHARE SCHEDULE MORE

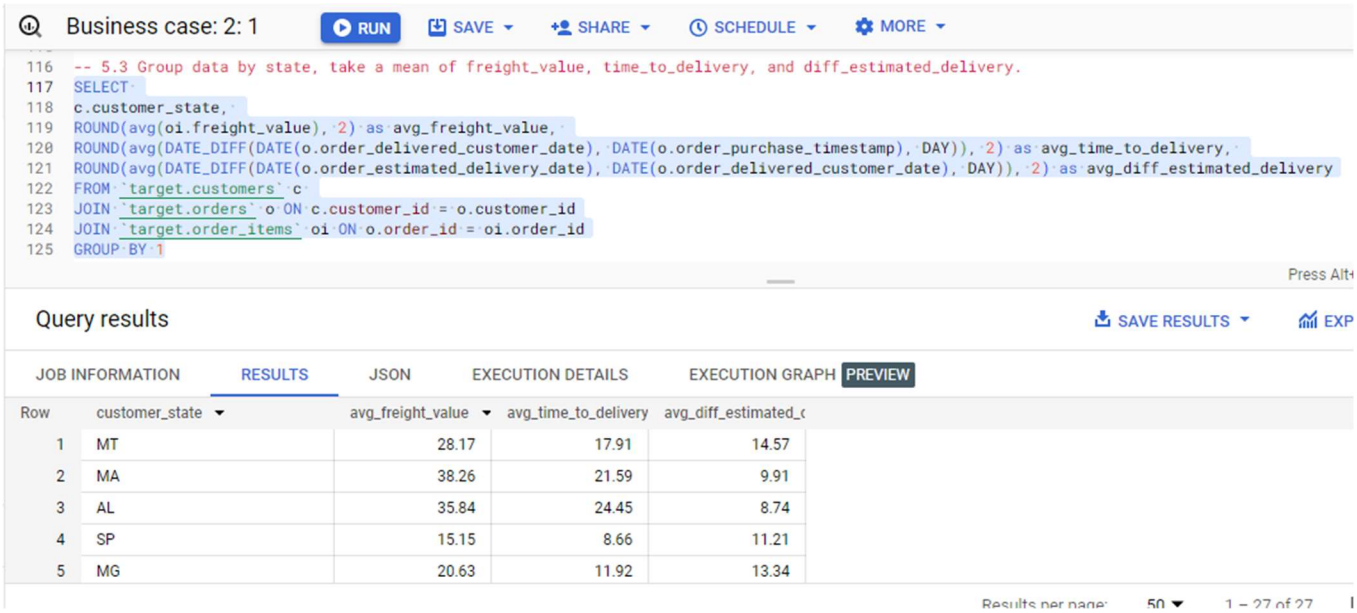
```
111 SELECT order_id,
112 DATE_DIFF(DATE(order_delivered_customer_date), DATE(order_purchase_timestamp), DAY) AS time_to_delivery,
113 DATE_DIFF(DATE(order_estimated_delivery_date), DATE(order_delivered_customer_date), DAY) AS diff_estimated_delivery
114 FROM `target.orders`
```

Query results SAVE R

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH PREVIEW

Row	order_id	time_to_delivery	diff_estimated_deliv
1	1950d777989f6a877539f5379...	30	-12
2	2c45c33d2f9cb8ff8b1c86cc28...	31	29
3	65d1e226dfaeb8cdc42f66542...	36	17
4	635c894d068ac37e6e03dc54e...	31	2
5	3b97562c3aee8bdedcb5c2e45...	33	1
6	68f47f50f04c4cb6774570cfde...	30	2
7	276e9ec344d3bf029ff83a161c...	44	-4
8	54e1a3c2b97fb0809da548a59...	41	-4

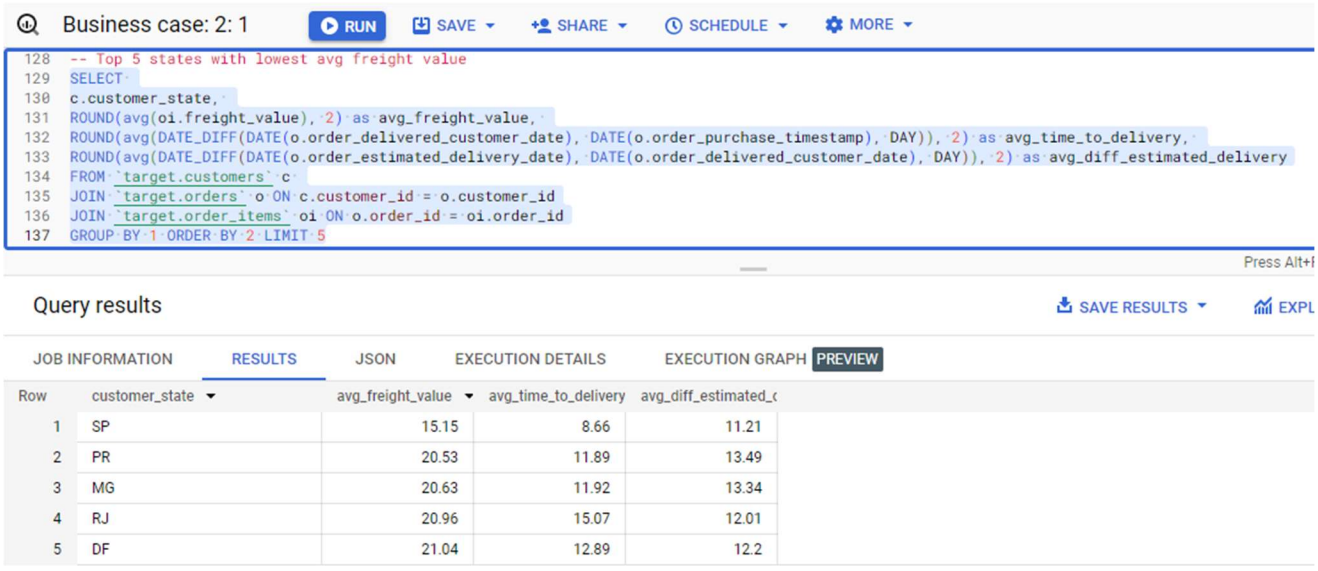
5.3 Group data by state, take a mean of freight_value, time_to_delivery, and diff_estimated_delivery.



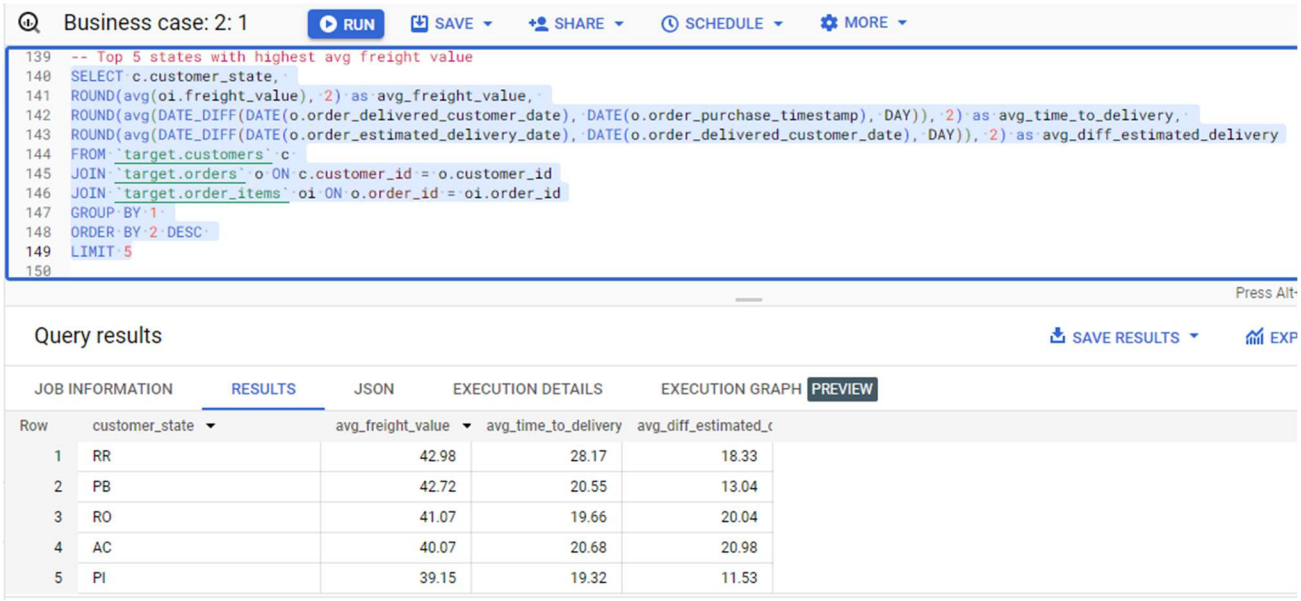
5.4 Sort the data to get the following:

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

States with lowest average freight value -

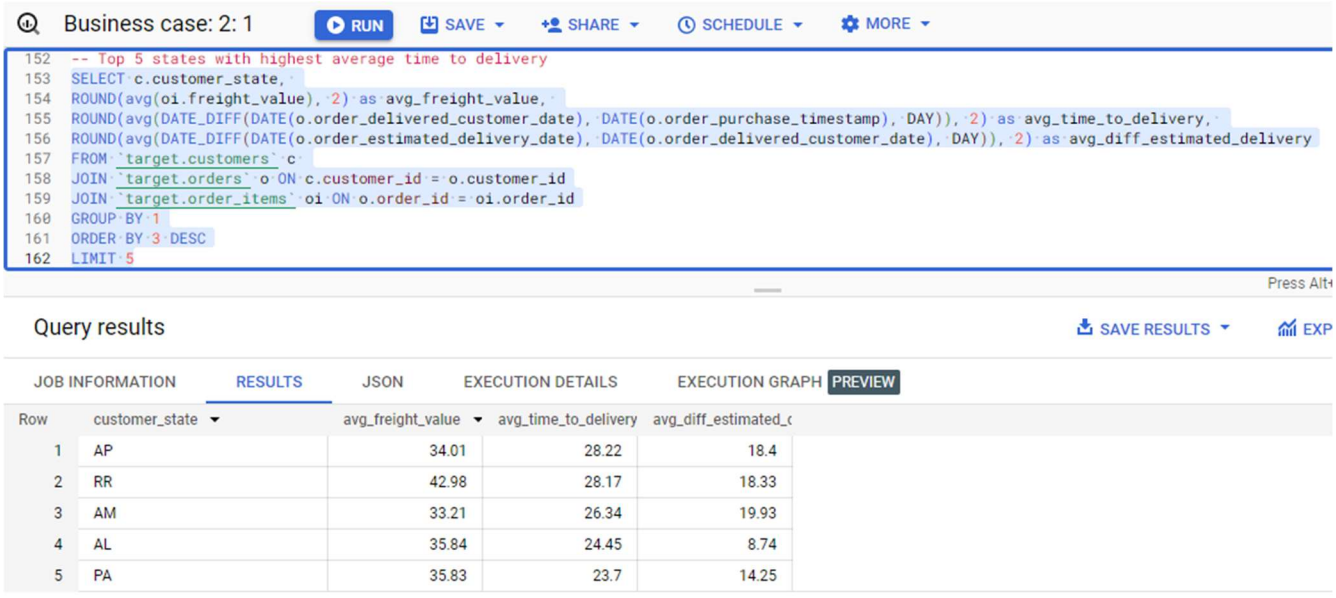


States with highest average freight value –

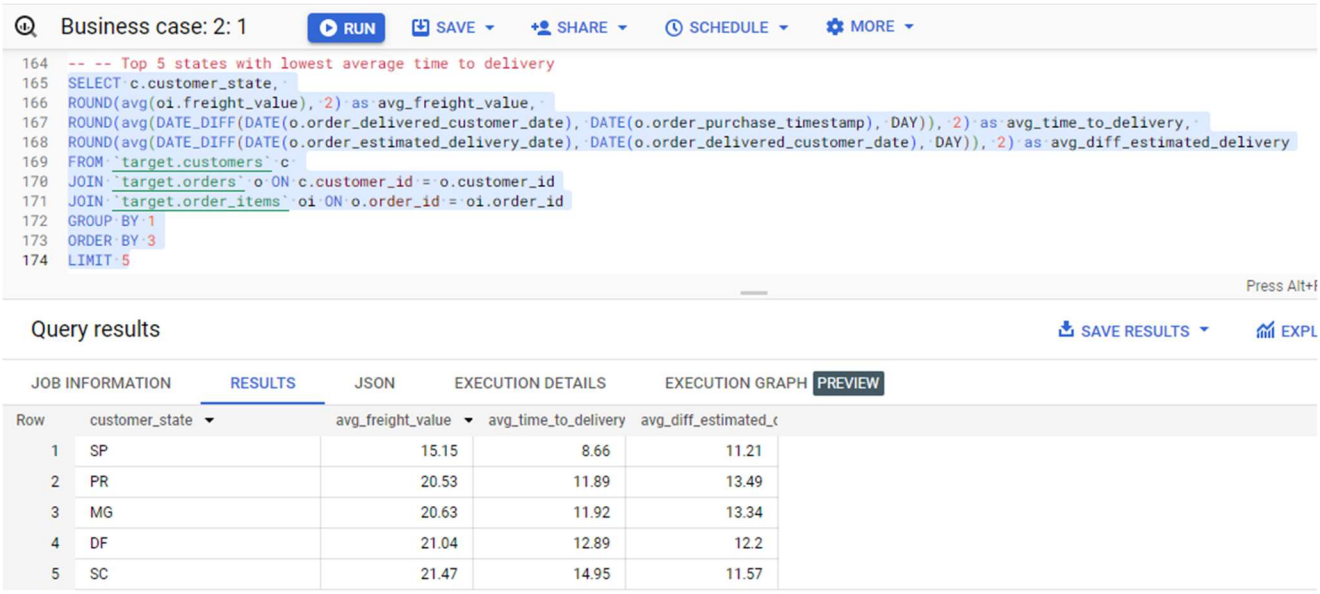


ii. Top 5 states with highest/lowest average time to delivery

States with the highest average time to delivery –



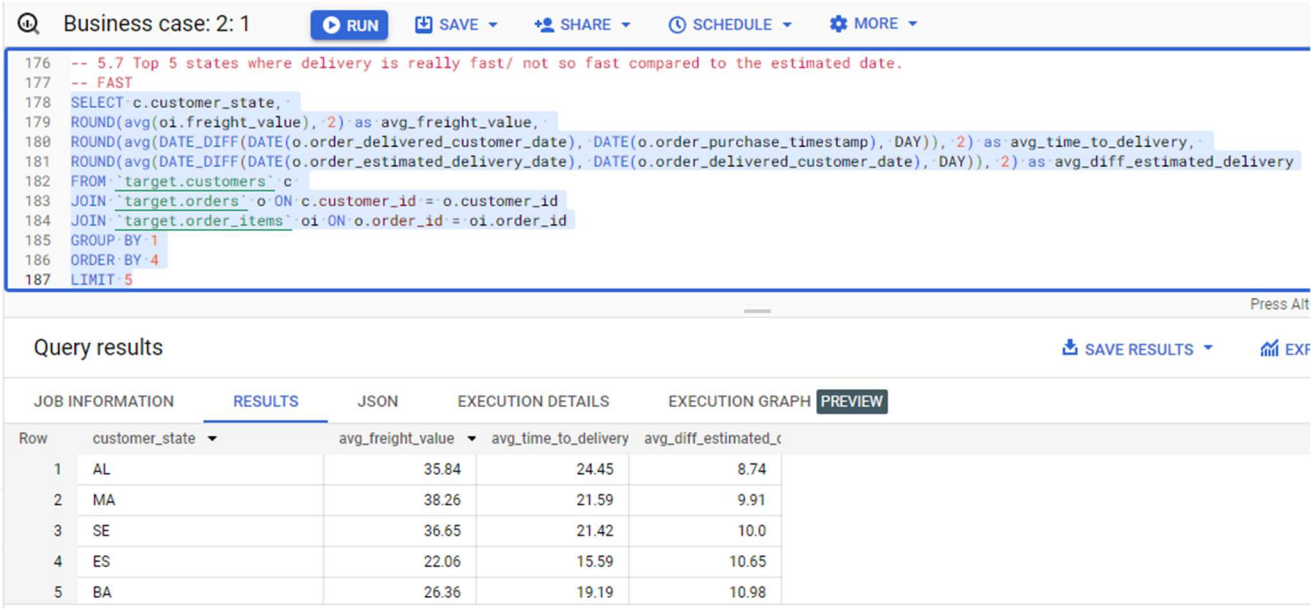
States with the lowest average time to delivery –



Row	customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_(
1	SP	15.15	8.66	11.21
2	PR	20.53	11.89	13.49
3	MG	20.63	11.92	13.34
4	DF	21.04	12.89	12.2
5	SC	21.47	14.95	11.57

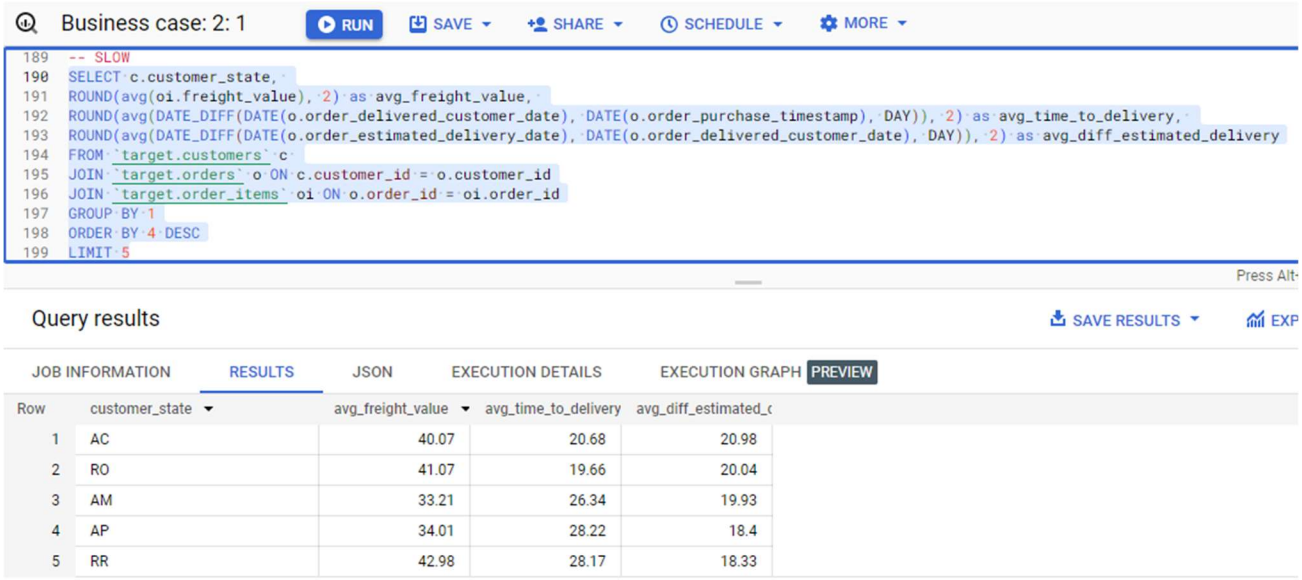
iii. Top 5 states where delivery is really fast/ not so fast compared to the estimated date.

States with fast delivery –



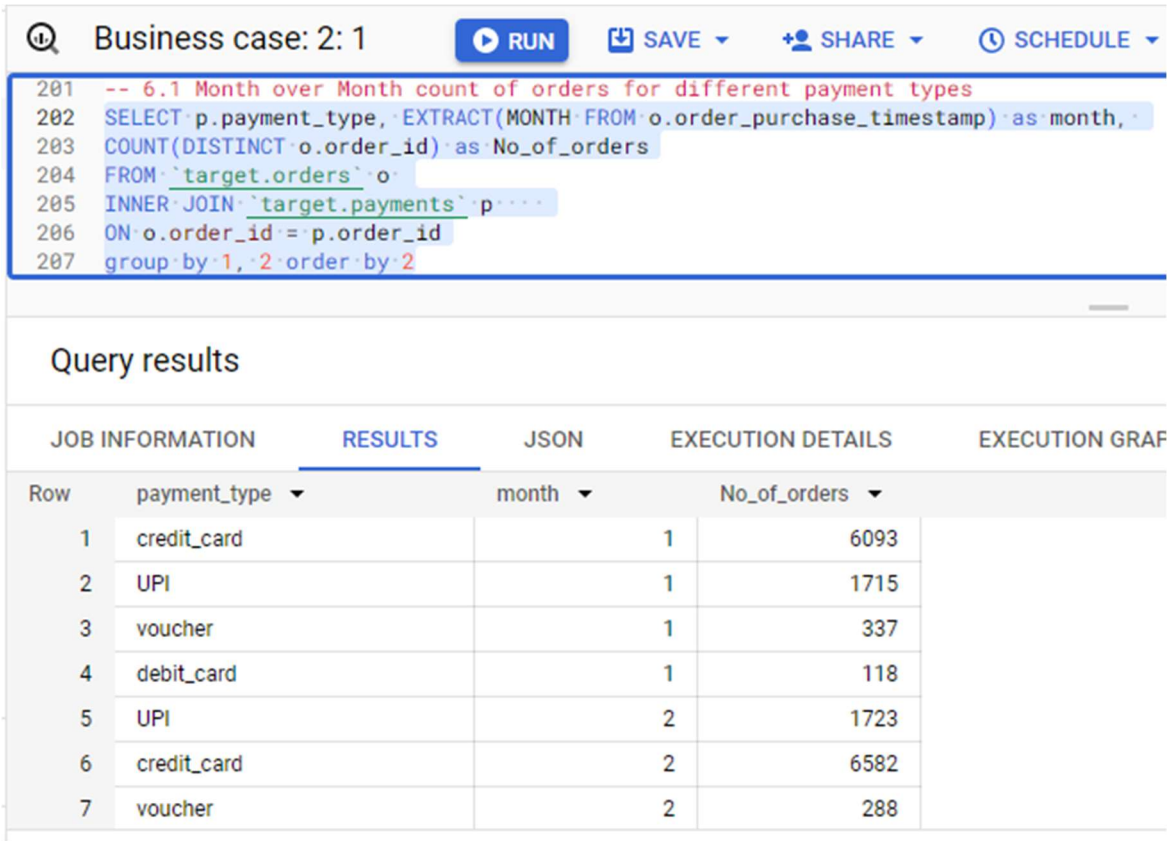
Row	customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_(
1	AL	35.84	24.45	8.74
2	MA	38.26	21.59	9.91
3	SE	36.65	21.42	10.0
4	ES	22.06	15.59	10.65
5	BA	26.36	19.19	10.98

States with slow delivery –



6. Payment type analysis:

6.1 Month over Month count of orders for different payment types



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

Business case: 2: 1 RUN SAVE SHARE

```
209  -- 6.2 Count of orders based on the no. of payment installments.
210  SELECT payment_installments, COUNT(DISTINCT order_id) as No_of_orders
211  FROM `target.payments`
212  GROUP BY 1
213  ORDER BY 1
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXE

Row	payment_installment	No_of_orders
1	0	2
2	1	49060
3	2	12389
4	3	10443
5	4	7088
6	5	5234
7	6	3916
8	7	1623

Insight:

- **Most people tend to buy from credit cards.**
- **Most of the orders are with payment installment - 1.**