Business Case Study - 1 (SQL)

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

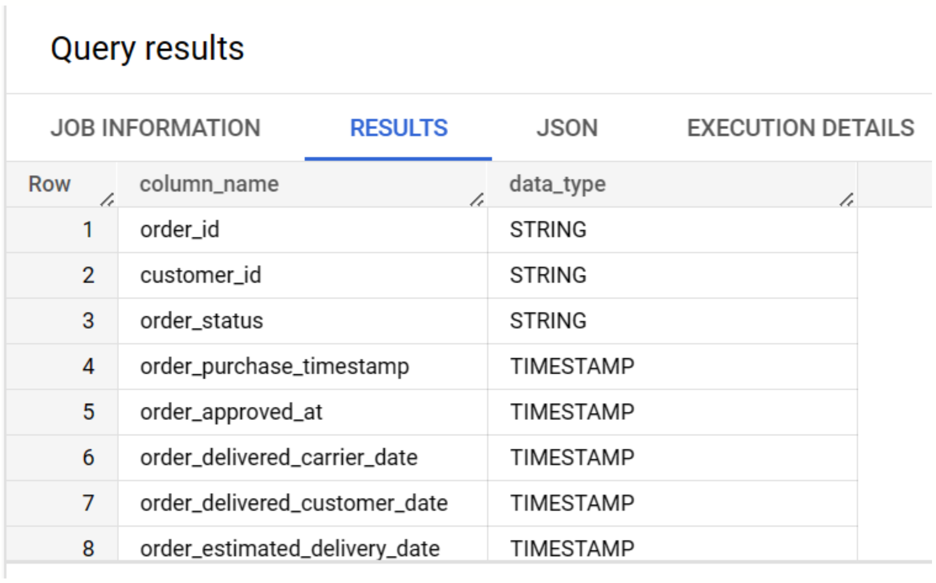
   Ans: Understanding the dataset , analyse the information given i try to solve some questions with required queries.

   1. **Datatype of columns in a table.**

   Query:-

```
SELECT column_name, data_type

FROM businesscase1sqlscaler.businesscasestudy1.INFORMATION_SCHEMA.COLUMNS

WHERE table_name = "orders";
```

Result:



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	column_name	data_type
1	order_id	STRING
2	customer_id	STRING
3	order_status	STRING
4	order_purchase_timestamp	TIMESTAMP
5	order_approved_at	TIMESTAMP
6	order_delivered_carrier_date	TIMESTAMP
7	order_delivered_customer_date	TIMESTAMP
8	order_estimated_delivery_date	TIMESTAMP

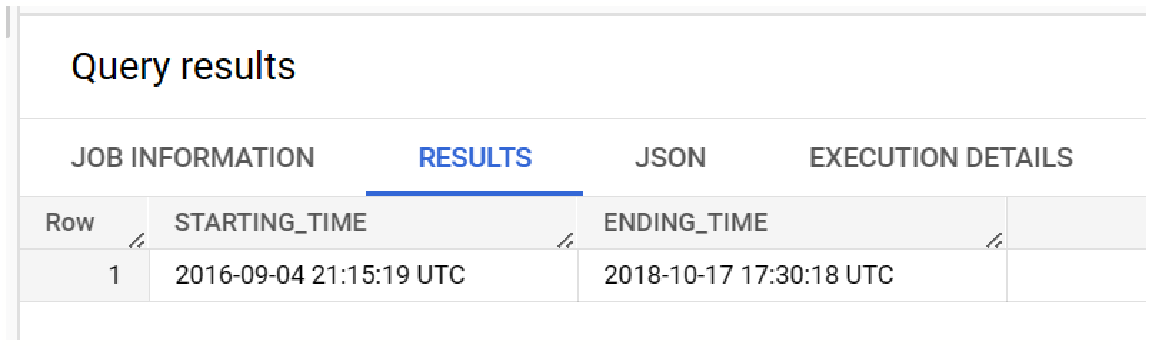
We can also use different table_name to know the datatype of the table's columns.

   2. **Time period for which the data is given.**

   Query:-

```
SELECT

   MIN(order_purchase_timestamp) AS STARTING_TIME,

   MAX(order_purchase_timestamp) As ENDING_TIME

FROM `businesscase1sqlscaler.businesscasestudy1.orders`;
```

Business Case Study - 1 (SQL)


Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

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

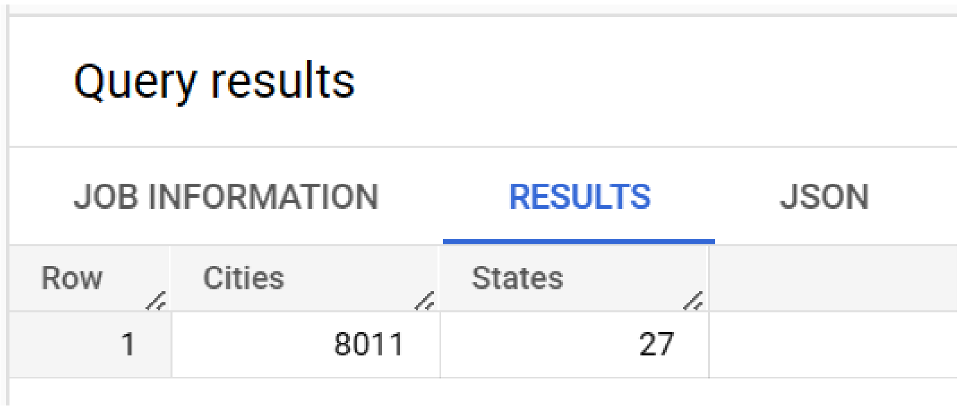
The time period of dataset is Starting from 2016-09-04 21:15:19 UTC & End in 2018-10-17 17:30:18 UTC

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

According to question i am getting two type of analysis:

Query1: Count All the distinct Cities and States.

```
SELECT
  COUNT(DISTINCT(geolocation_city)) AS Cities,
  COUNT(DISTINCT(geolocation_state)) AS States
FROM `businesscase1sqlscaler.businesscasestudy1.geolocation`
```


Query results

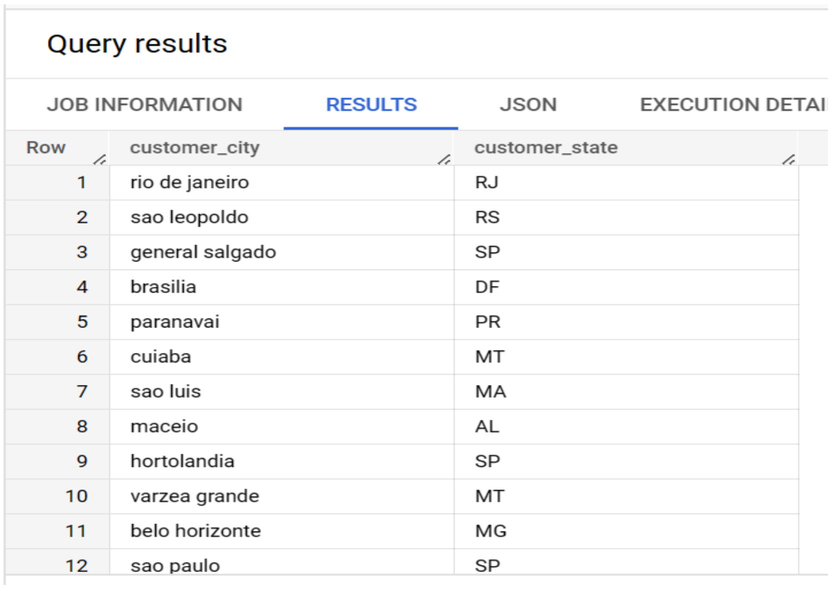
JOB INFORMATION | RESULTS | JSON

Row	Cities	States
1	8011	27

Query2: Cities & State name of the customers Order

```
SELECT
  DISTINCT customer_city,
  customer_state
FROM `businesscase1sqlscaler.businesscasestudy1.customers` AS Cus
JOIN `businesscase1sqlscaler.businesscasestudy1.orders` AS Ord
ON Cus.customer_id = Ord.customer_id
```

Business Case Study - 1 (SQL)



Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAI

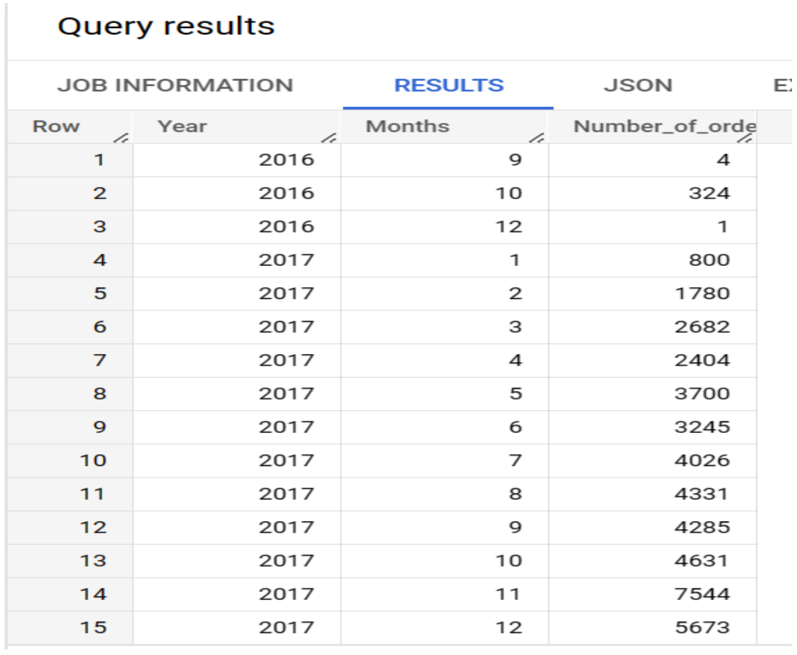
Row	customer_city	customer_state
1	rio de janeiro	RJ
2	sao leopoldo	RS
3	general salgado	SP
4	brasilia	DF
5	paranavai	PR
6	cuiaba	MT
7	sao luis	MA
8	maceio	AL
9	hortolandia	SP
10	varzea grande	MT
11	belo horizonte	MG
12	sao paulo	SP

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 ?

Query 1:- Growing trends on e-commerce

```
SELECT
  EXTRACT(year FROM order_purchase_timestamp) AS Year,
  EXTRACT(month FROM order_purchase_timestamp) AS Months,
  COUNT(*) AS Number_of_orders
FROM `businesscase1sqlscaler.businesscasestudy1.orders`
GROUP BY Year, Months
ORDER BY Year, Months
```



Query results

JOB INFORMATION | RESULTS | JSON | E

Row	Year	Months	Number_of_orde
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
14	2017	11	7544
15	2017	12	5673

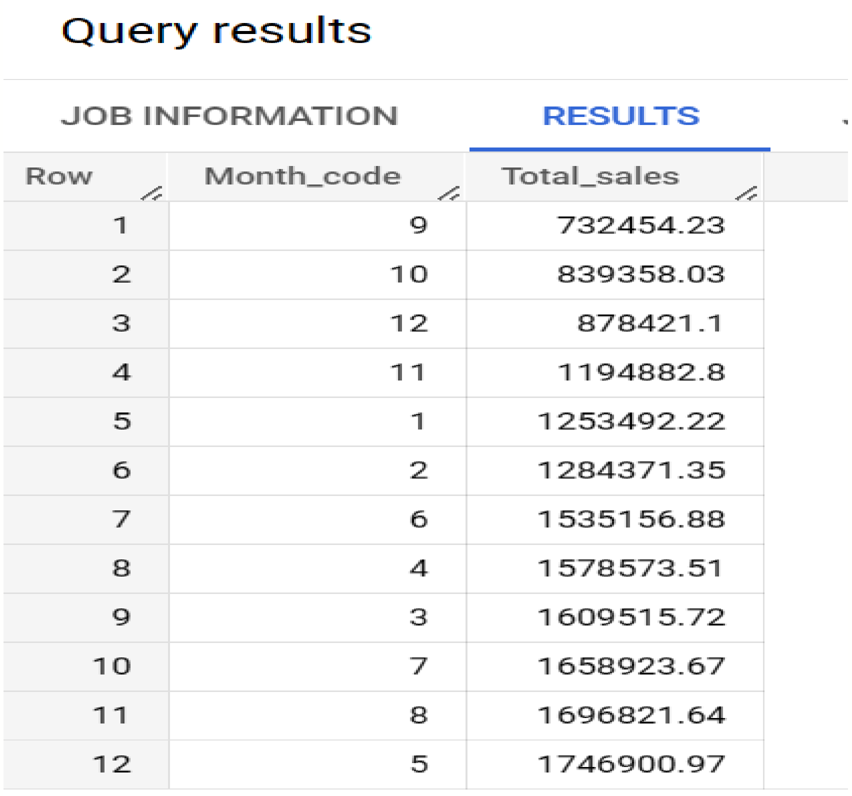
Business Case Study - 1 (SQL)

According to the result there we have 3 months data of 2016, total year of 2017 And 10 months data of 2018. Clearly showing that the trend is increasing year by year according to the number of orders.

Query 2:- **Seasonality with peaks at specific months**

According to Sales Payments:

```
SELECT
  Month_code,
  Total_sales
FROM (SELECT
      EXTRACT(MONTH FROM ord.order_purchase_timestamp) AS Month_code,
      ROUND(SUM(pay.payment_value), 2) AS Total_sales
  FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord
  JOIN `businesscase1sqlscaler.businesscasestudy1.payments` AS pay
  ON ord.order_id = pay.order_id
  GROUP BY EXTRACT(MONTH FROM ord.order_purchase_timestamp))
ORDER BY Total_sales;
```



Query results

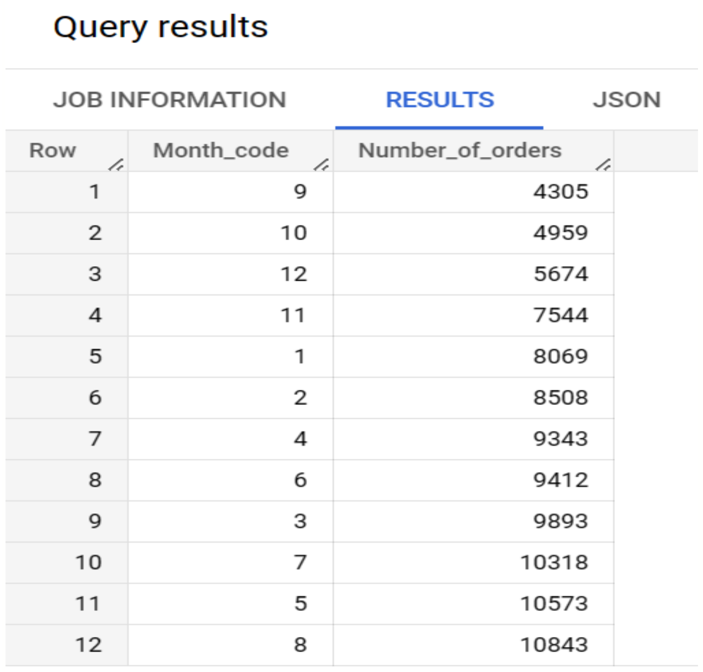
JOB INFORMATION RESULTS

Row	Month_code	Total_sales
1	9	732454.23
2	10	839358.03
3	12	878421.1
4	11	1194882.8
5	1	1253492.22
6	2	1284371.35
7	6	1535156.88
8	4	1578573.51
9	3	1609515.72
10	7	1658923.67
11	8	1696821.64
12	5	1746900.97

Business Case Study - 1 (SQL)

According to Number of orders :

```
SELECT
  EXTRACT(MONTH FROM order_purchase_timestamp) AS Month_code,
  COUNT(*) AS Number_of_orders
FROM `businesscase1sqlscaler.businesscasestudy1.orders`
GROUP BY Month_code
ORDER BY Number_of_orders;
```



Query results

Row	Month_code	Number_of_orders
1	9	4305
2	10	4959
3	12	5674
4	11	7544
5	1	8069
6	2	8508
7	4	9343
8	6	9412
9	3	9893
10	7	10318
11	5	10573
12	8	10843

In both cases we can see seasonality peak according to orders and total sales.

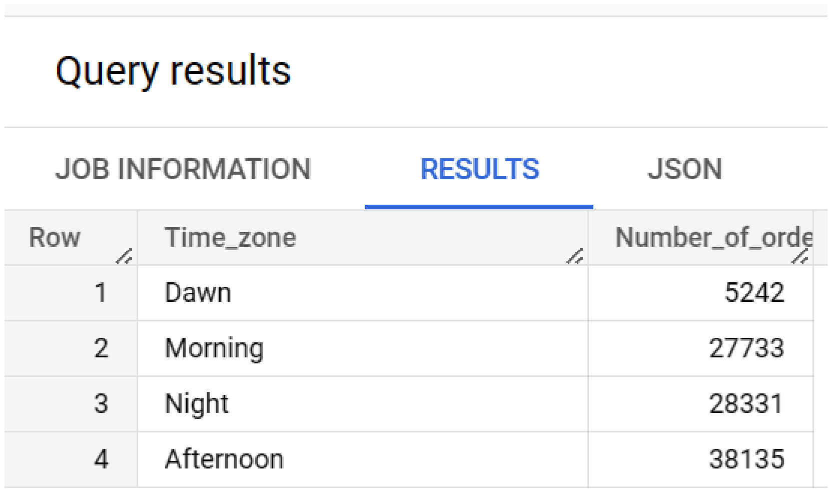
2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night) ?

Extracting the hour hand Considering there are 4 Time zone that is:

a) 0 to 6 As Dawn
b) 7 to 12 As Morning
c) 13 to 18 As Afternoon
d) 19 to 23 As Night

Business Case Study - 1 (SQL)

```
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 Time_zone,
COUNT(DISTINCT order_id) AS Number_of_orders
FROM `businesscase1sqlscaler.businesscasestudy1.orders`
GROUP BY Time_zone
ORDER BY Number_of_orders;
```



Query results

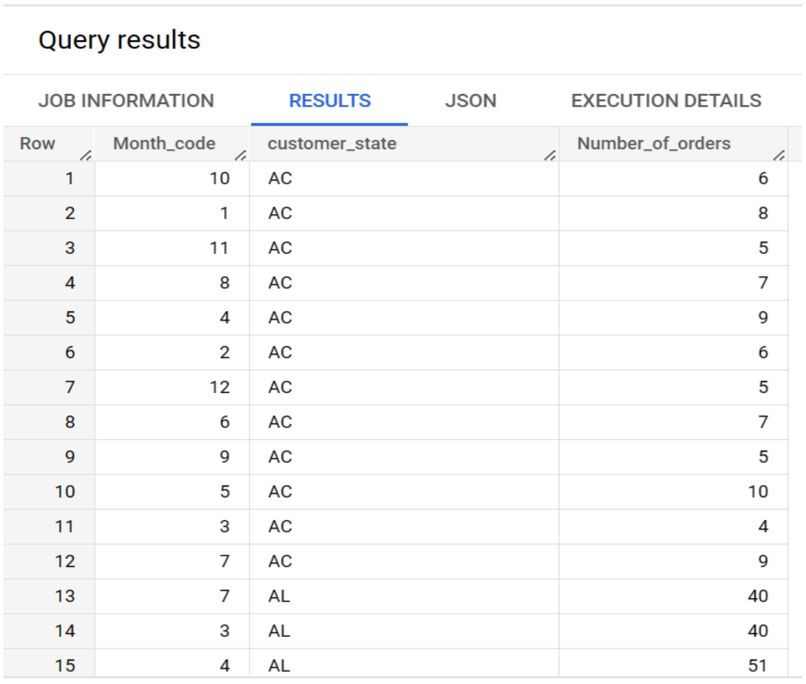
Row	Time_zone	Number_of_orde
1	Dawn	5242
2	Morning	27733
3	Night	28331
4	Afternoon	38135

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

1. Get Month on Month orders by states

```
SELECT
  EXTRACT(MONTH FROM order_purchase_timestamp) AS Month_code,
  customer_state,
  COUNT(*) AS Number_of_orders,
FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ords
JOIN `businesscase1sqlscaler.businesscasestudy1.customers` AS cust
ON ords.customer_id = cust.customer_id
GROUP BY customer_state,Month_code
ORDER BY customer_state;
```

Business Case Study - 1 (SQL)



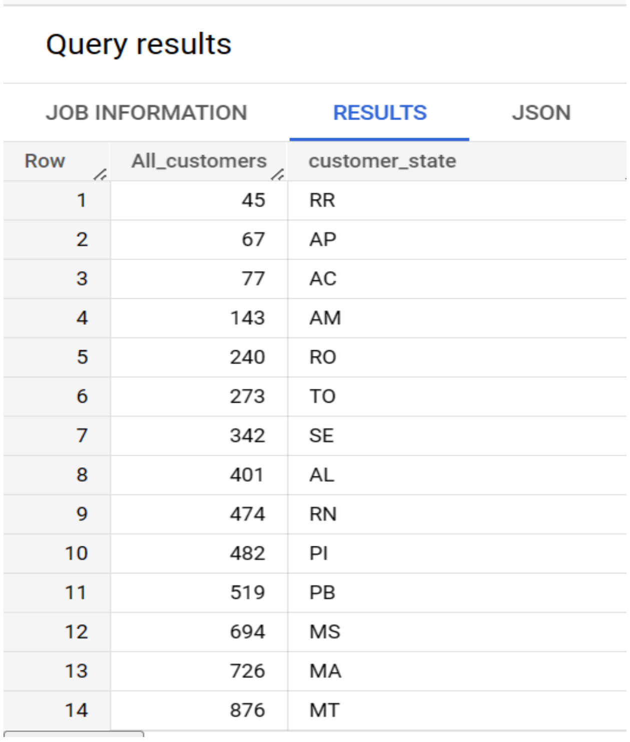
Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	Month_code	customer_state	Number_of_orders
1	10	AC	6
2	1	AC	8
3	11	AC	5
4	8	AC	7
5	4	AC	9
6	2	AC	6
7	12	AC	5
8	6	AC	7
9	9	AC	5
10	5	AC	10
11	3	AC	4
12	7	AC	9
13	7	AL	40
14	3	AL	40
15	4	AL	51

2. **Distribution of customers across the states in Brazil.**

```
SELECT
  COUNT(DISTINCT(customer_unique_id)) AS All_customers,
  customer_state
from `businesscase1sqlscaler.businesscasestudy1.customers`
GROUP BY customer_state
ORDER BY All_customers;
```



Query results

JOB INFORMATION RESULTS JSON

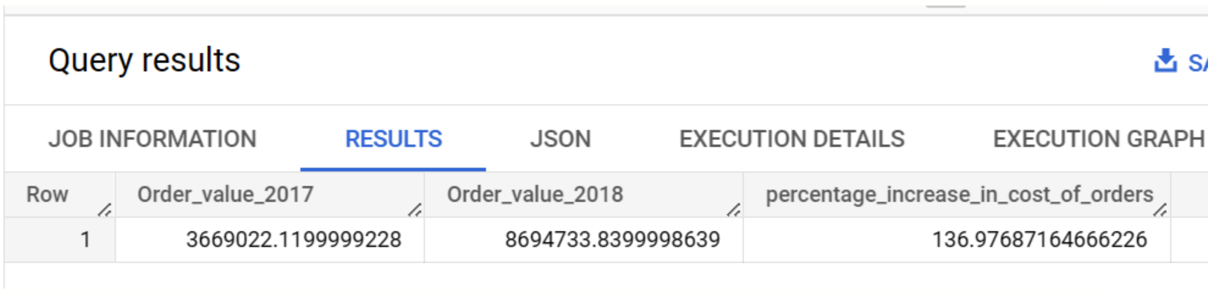
Row	All_customers	customer_state
1	45	RR
2	67	AP
3	77	AC
4	143	AM
5	240	RO
6	273	TO
7	342	SE
8	401	AL
9	474	RN
10	482	PI
11	519	PB
12	694	MS
13	726	MA
14	876	MT

Business Case Study - 1 (SQL)

4. Impact on Economy: Analyse 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

  Order_value_2017,

  Order_value_2018,

  (((Order_value_2018 - Order_value_2017)/ Order_value_2017)* 100) As
percentage_increase_in_cost_of_orders

FROM (SELECT

      SUM(IF(EXTRACT(year FROM ord.order_purchase_timestamp) = 2017,
pay.payment_value,0)) AS Order_value_2017,

      SUM(IF(EXTRACT(year FROM ord.order_purchase_timestamp) = 2018,
pay.payment_value,0)) AS Order_value_2018

    FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord

    INNER JOIN `businesscase1sqlscaler.businesscasestudy1.payments` AS pay

    ON ord.order_id = pay.order_id

    WHERE EXTRACT(month FROM ord.order_purchase_timestamp) BETWEEN 1 AND 8);
```



Query results

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

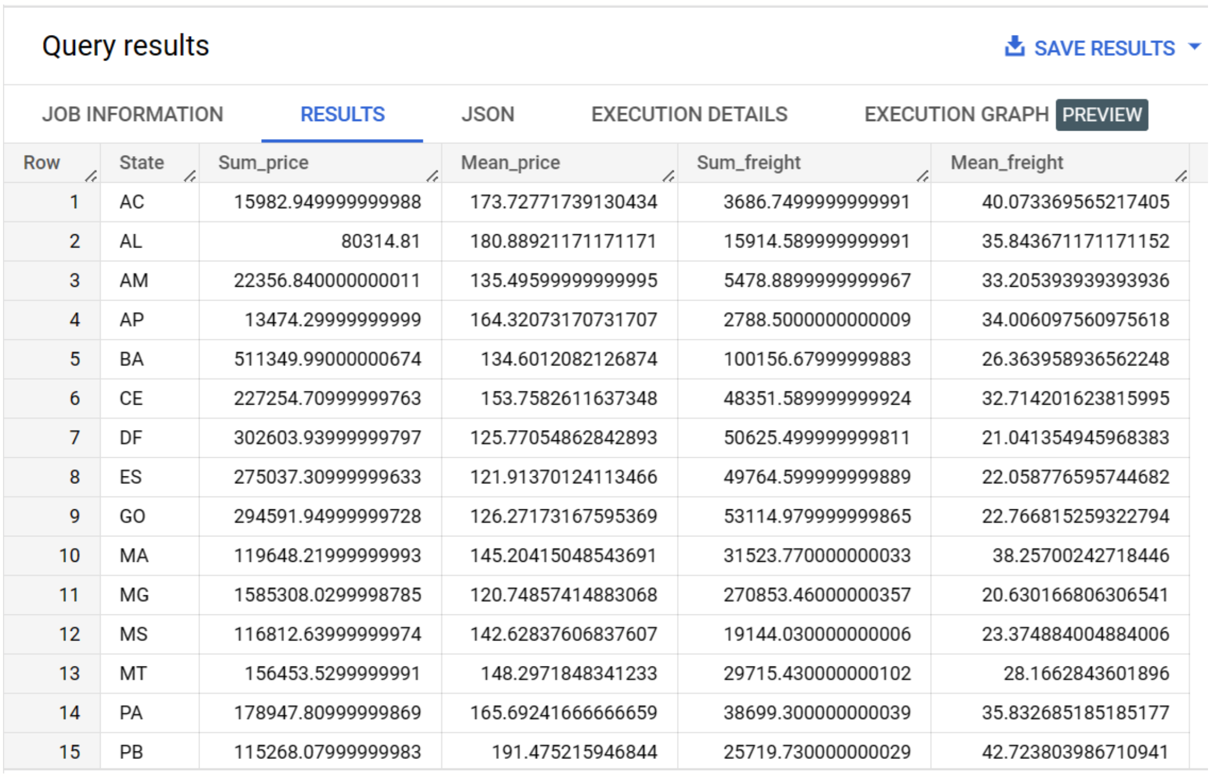
Row	Order_value_2017	Order_value_2018	percentage_increase_in_cost_of_orders
1	3669022.1199999228	8694733.8399998639	136.97687164666226

Here %increase is 136.98 Approx.

Business Case Study - 1 (SQL)

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

```
SELECT
  co.customer_state AS State,
  SUM(ordit.price) AS Sum_price,
  AVG(ordit.price) AS Mean_price,
  SUM(ordit.freight_value) AS Sum_freight,
  AVG(ordit.freight_value) AS Mean_freight
FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord
JOIN `businesscase1sqlscaler.businesscasestudy1.order_items` AS ordit
ON ord.order_id = ordit.order_id
JOIN `businesscase1sqlscaler.businesscasestudy1.customers` AS co
ON ord.customer_id = co.customer_id
GROUP BY co.customer_state
ORDER BY co.customer_state;
```



Row	State	Sum_price	Mean_price	Sum_freight	Mean_freight
1	AC	15982.949999999988	173.72771739130434	3686.7499999999991	40.073369565217405
2	AL	80314.81	180.88921171171171	15914.589999999991	35.843671171171152
3	AM	22356.840000000011	135.49599999999995	5478.8899999999967	33.205393939393936
4	AP	13474.29999999999	164.32073170731707	2788.5000000000009	34.006097560975618
5	BA	511349.99000000674	134.6012082126874	100156.67999999883	26.363958936562248
6	CE	227254.70999999763	153.7582611637348	48351.589999999924	32.714201623815995
7	DF	302603.93999999797	125.77054862842893	50625.499999999811	21.041354945968383
8	ES	275037.30999999633	121.91370124113466	49764.599999999889	22.058776595744682
9	GO	294591.94999999728	126.27173167595369	53114.979999999865	22.766815259322794
10	MA	119648.21999999993	145.20415048543691	31523.770000000033	38.25700242718446
11	MG	1585308.0299998785	120.74857414883068	270853.46000000357	20.630166806306541
12	MS	116812.63999999974	142.62837606837607	19144.030000000006	23.374884004884006
13	MT	156453.5299999991	148.2971848341233	29715.430000000102	28.1662843601896
14	PA	178947.80999999869	165.69241666666659	38699.300000000039	35.832685185185177
15	PB	115268.07999999983	191.475215946844	25719.730000000029	42.723803986710941

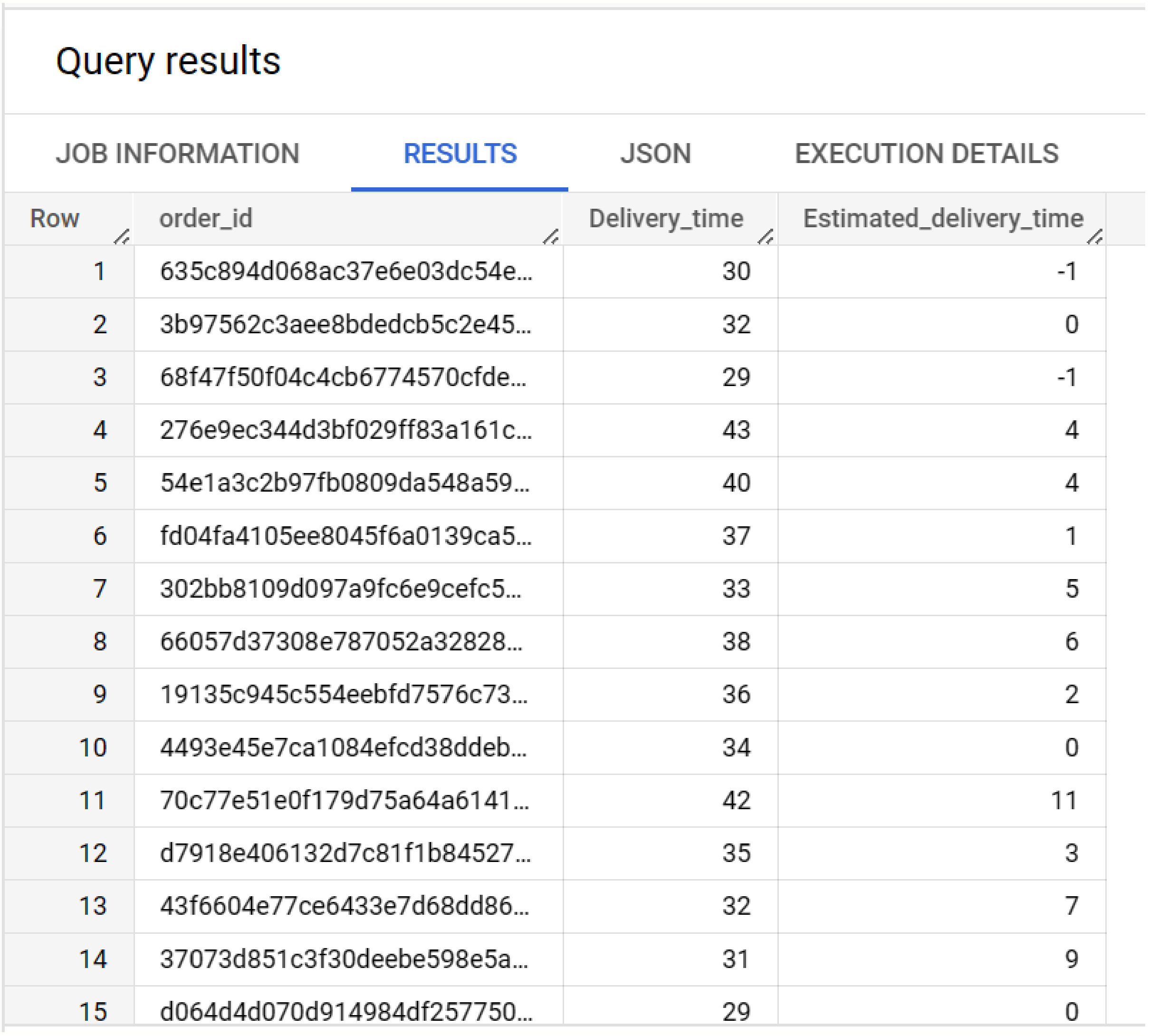
We can now see sum_price, Mean_price, Sum_freight, Mean_freight values According to their customer state.

Business Case Study - 1 (SQL)

5. Analysis on Sales, freight and delivery time:

1. **Calculate days between purchasing, delivering and estimated delivery**

```
SELECT

  order_id,

  TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp,
day) AS Delivery_time,

  TIMESTAMP_DIFF(order_delivered_customer_date,
order_estimated_delivery_date, day) AS Estimated_delivery_time

FROM `businesscase1sqlscaler.businesscasestudy1.orders`

WHERE order_status = "delivered"
```



Query results

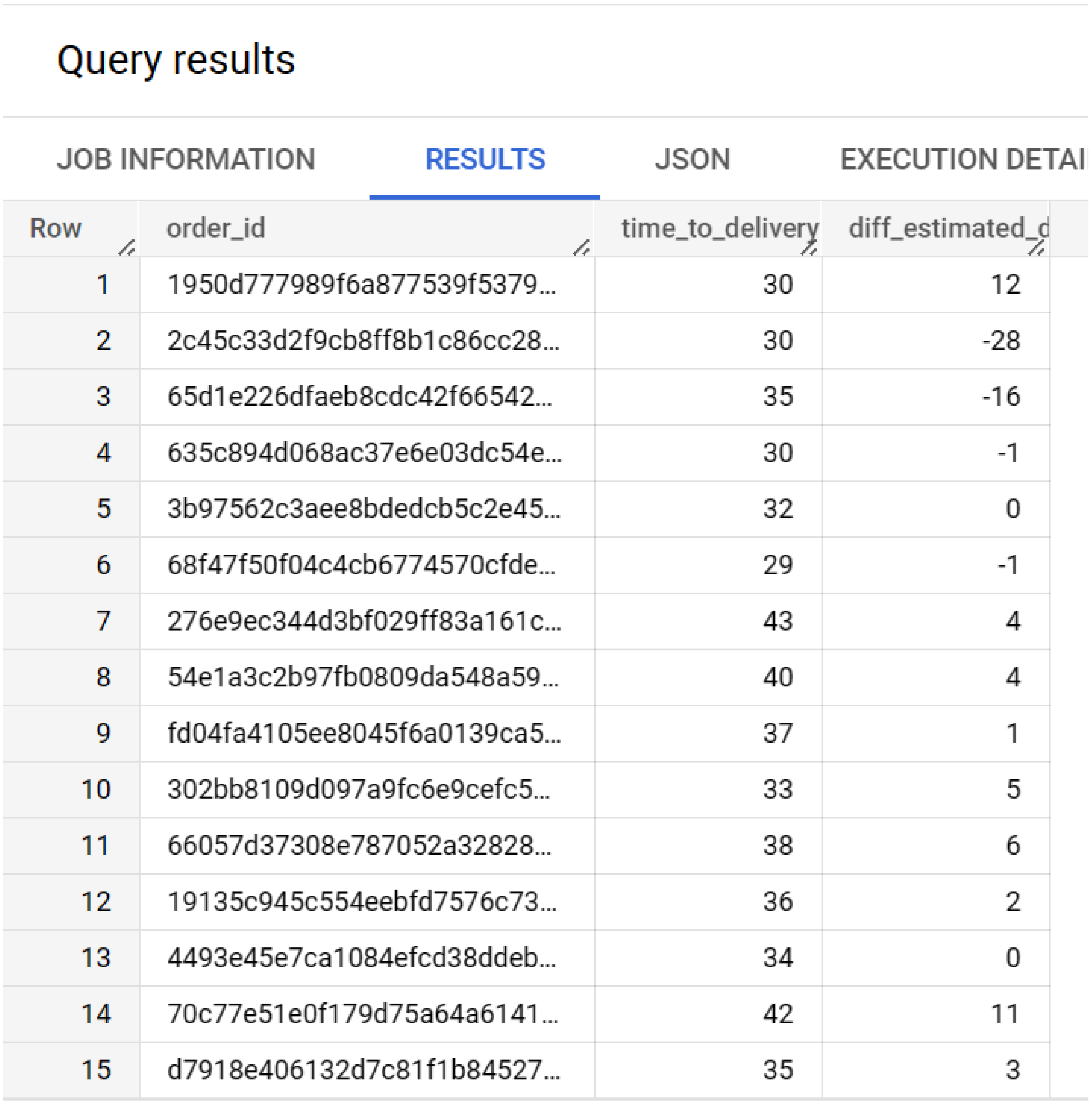
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_id	Delivery_time	Estimated_delivery_time
1	635c894d068ac37e6e03dc54e...	30	-1
2	3b97562c3aee8bdedcb5c2e45...	32	0
3	68f47f50f04c4cb6774570cfde...	29	-1
4	276e9ec344d3bf029ff83a161c...	43	4
5	54e1a3c2b97fb0809da548a59...	40	4
6	fd04fa4105ee8045f6a0139ca5...	37	1
7	302bb8109d097a9fc6e9cefc5...	33	5
8	66057d37308e787052a32828...	38	6
9	19135c945c554eebfd7576c73...	36	2
10	4493e45e7ca1084efcd38ddeb...	34	0
11	70c77e51e0f179d75a64a6141...	42	11
12	d7918e406132d7c81f1b84527...	35	3
13	43f6604e77ce6433e7d68dd86...	32	7
14	37073d851c3f30deebe598e5a...	31	9
15	d064d4d070d914984df257750...	29	0

Business Case Study - 1 (SQL)

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

```
SELECT

  order_id,

  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) AS
time_to_delivery,

  DATE_DIFF(order_delivered_customer_date, order_estimated_delivery_date, day) AS
diff_estimated_delivery

FROM `businesscase1sqlscaler.businesscasestudy1.orders`
```



Query results

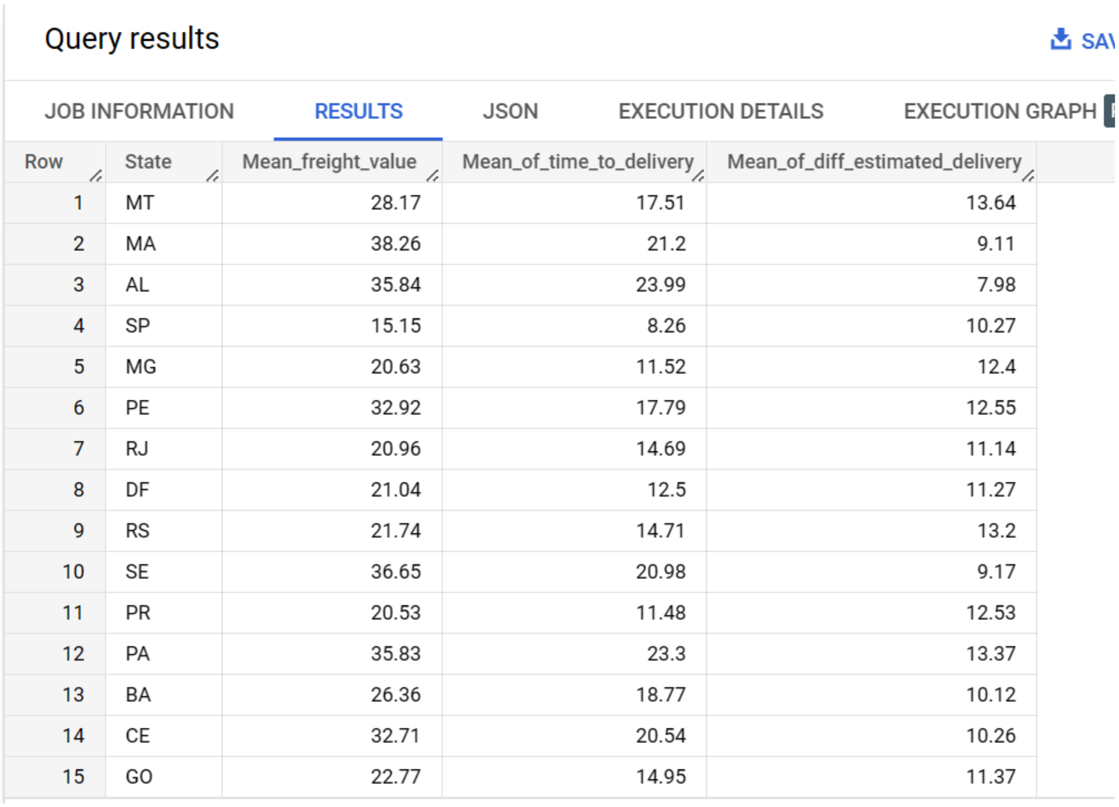
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAI

Row	order_id	time_to_delivery	diff_estimated_d
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
11	66057d37308e787052a32828...	38	6
12	19135c945c554eebfd7576c73...	36	2
13	4493e45e7ca1084efcd38ddeb...	34	0
14	70c77e51e0f179d75a64a6141...	42	11
15	d7918e406132d7c81f1b84527...	35	3

Business Case Study - 1 (SQL)

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

```
SELECT

  co.customer_state AS State,

  ROUND(AVG(ordit.freight_value),2) AS Mean_freight_value,

  ROUND(AVG(TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp,
DAY)),2) AS Mean_of_time_to_delivery,

  ROUND(AVG(TIMESTAMP_DIFF( order_estimated_delivery_date,
order_delivered_customer_date, day)),2) AS Mean_of_diff_estimated_delivery

FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord

JOIN `businesscase1sqlscaler.businesscasestudy1.customers` AS co

ON ord.customer_id = co.customer_id

JOIN `businesscase1sqlscaler.businesscasestudy1.order_items` AS ordit

ON ord.order_id = ordit.order_id

GROUP BY co.customer_state;
```



Row	State	Mean_freight_value	Mean_of_time_to_delivery	Mean_of_diff_estimated_delivery
1	MT	28.17	17.51	13.64
2	MA	38.26	21.2	9.11
3	AL	35.84	23.99	7.98
4	SP	15.15	8.26	10.27
5	MG	20.63	11.52	12.4
6	PE	32.92	17.79	12.55
7	RJ	20.96	14.69	11.14
8	DF	21.04	12.5	11.27
9	RS	21.74	14.71	13.2
10	SE	36.65	20.98	9.17
11	PR	20.53	11.48	12.53
12	PA	35.83	23.3	13.37
13	BA	26.36	18.77	10.12
14	CE	32.71	20.54	10.26
15	GO	22.77	14.95	11.37

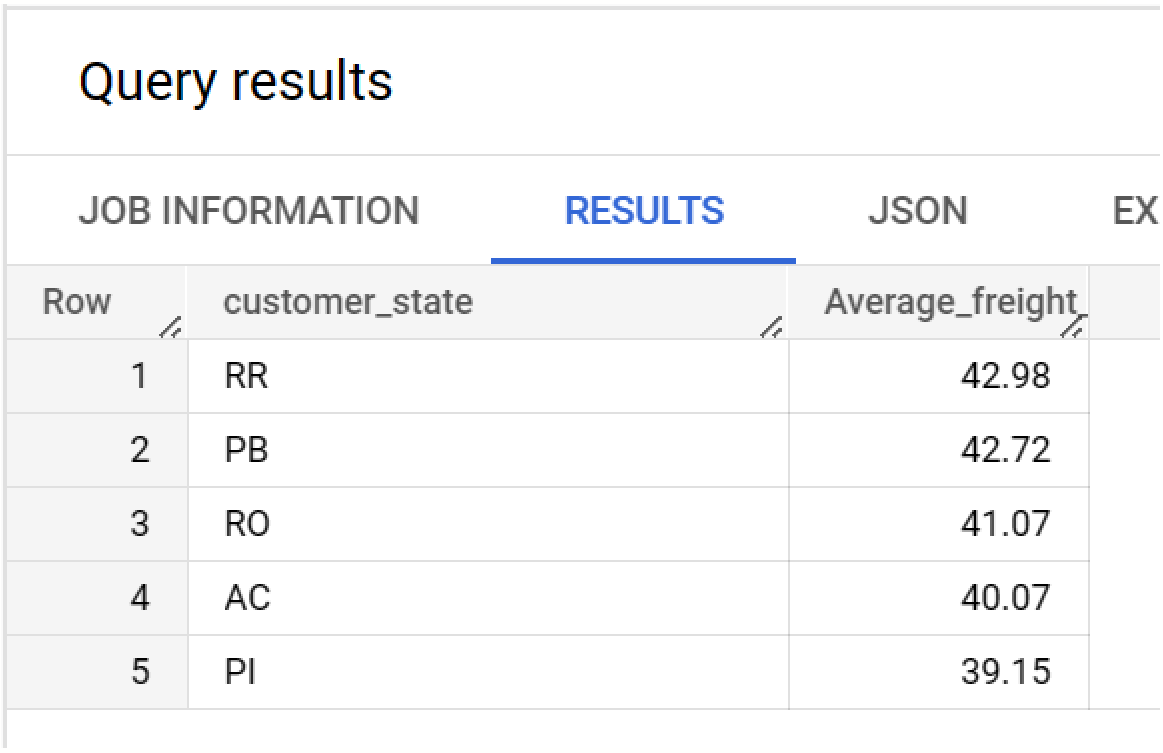
Business Case Study - 1 (SQL)

4. Sort the data to get the following:

5. TOP 5 State with highest/lowest average freight value - sort in desc/asc limit 5

a. Top 5 Highest with Average freight value

```
SELECT
  co.customer_state,
  ROUND(AVG(ordit.freight_value),2) AS Average_freight_value
FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord
JOIN `businesscase1sqlscaler.businesscasestudy1.customers` AS co
ON ord.customer_id = co.customer_id
JOIN `businesscase1sqlscaler.businesscasestudy1.order_items` AS ordit
ON ord.order_id = ordit.order_id
GROUP BY co.customer_state
ORDER BY Average_freight_value DESC
LIMIT 5
```

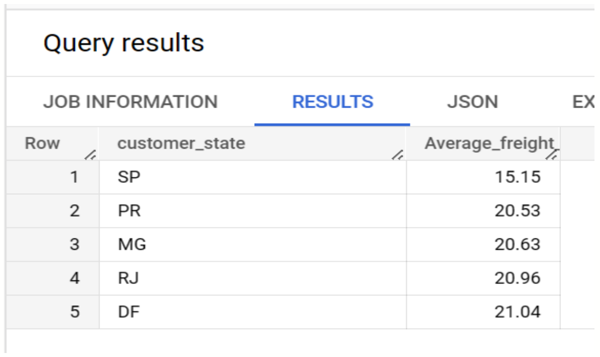


Row	customer_state	Average_freight
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

Business Case Study - 1 (SQL)

b. **Bottom 5 lowest with Average freight value**

```
SELECT
  co.customer_state,
  ROUND(AVG(ordit.freight_value),2) AS Average_freight_value
FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord
JOIN `businesscase1sqlscaler.businesscasestudy1.customers` AS co
ON ord.customer_id = co.customer_id
JOIN `businesscase1sqlscaler.businesscasestudy1.order_items` AS ordit
ON ord.order_id = ordit.order_id
GROUP BY co.customer_state
ORDER BY Average_freight_value ASC
LIMIT 5
```



Query results

JOB INFORMATION | RESULTS | JSON | EX

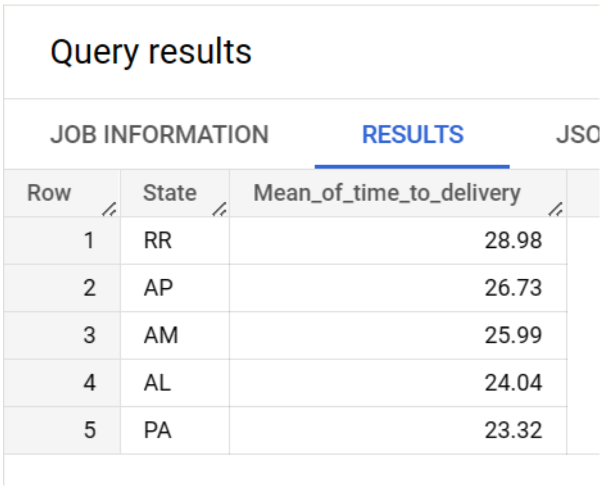
Row	customer_state	Average_freight
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

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

a. **Top 5 highest acc. To average time to delivery**

```
SELECT
  co.customer_state As State,
  ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)),2) AS Mean_of_time_to_delivery,
FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord
JOIN `businesscase1sqlscaler.businesscasestudy1.customers` AS co
ON ord.customer_id = co.customer_id
GROUP BY co.customer_state
ORDER BY Mean_of_time_to_delivery DESC
LIMIT 5
```

Business Case Study - 1 (SQL)



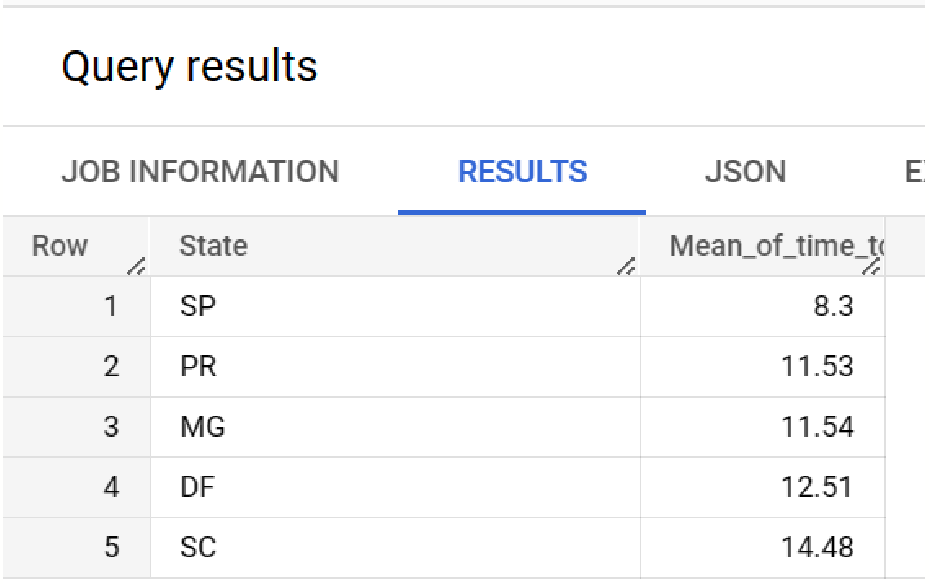
Query results

JOB INFORMATION RESULTS JSO

Row	State	Mean_of_time_to_delivery
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

b. bottom 5 lowest acc. To average time to delivery

```
SELECT
  co.customer_state As State,
  ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)),2) AS Mean_of_time_to_delivery,
FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord
JOIN `businesscase1sqlscaler.businesscasestudy1.customers` AS co
ON ord.customer_id = co.customer_id
GROUP BY co.customer_state
ORDER BY Mean_of_time_to_delivery ASC
LIMIT 5
```



Query results

JOB INFORMATION RESULTS JSON E

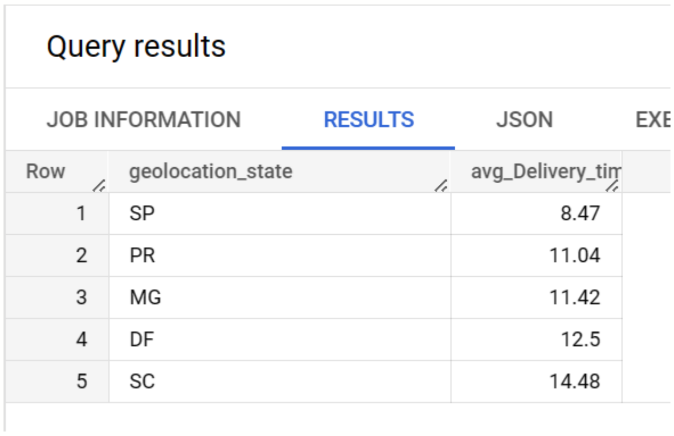
Row	State	Mean_of_time_t
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

Business Case Study - 1 (SQL)

7. Top 5 state where delivery is really fast/ not so fast compared to estimated date

a. Top 5 not so fast compared to estimated date

```
SELECT
  geolocation_state,
  ROUND(AVG(DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp, DAY)),2) AS avg_Delivery_time
  FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord
  JOIN `businesscase1sqlscaler.businesscasestudy1.customers` As co
  ON ord.customer_id = co.customer_id
  JOIN `businesscase1sqlscaler.businesscasestudy1.geolocation` AS geo
  ON co.customer_zip_code_prefix = geo.geolocation_zip_code_prefix
WHERE order_status = "delivered"
GROUP BY geolocation_state
ORDER BY avg_Delivery_time ASC
LIMIT 5;
```



Query results

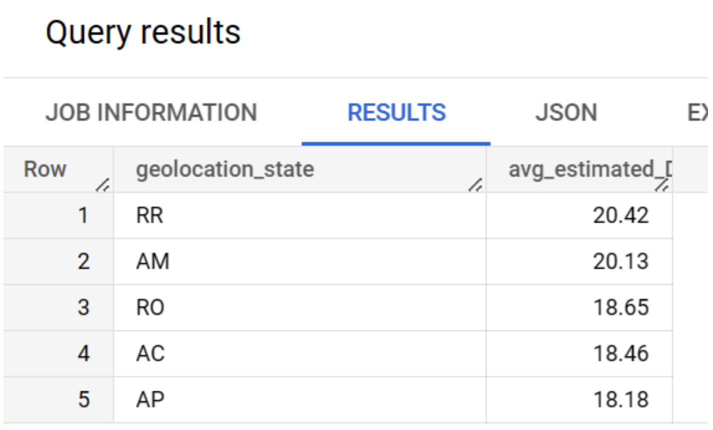
JOB INFORMATION | RESULTS | JSON | EXE

Row	geolocation_state	avg_Delivery_tir
1	SP	8.47
2	PR	11.04
3	MG	11.42
4	DF	12.5
5	SC	14.48

b. Top 5 really fast compared to estimated date

```
SELECT
  geolocation_state,
  ROUND(AVG(TIMESTAMP_DIFF( order_estimated_delivery_date,
order_delivered_customer_date, day)),2) avg_estimated_Delivery_time
  FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord
  JOIN `businesscase1sqlscaler.businesscasestudy1.customers` As co
  ON ord.customer_id = co.customer_id
  JOIN `businesscase1sqlscaler.businesscasestudy1.geolocation` AS geo
  ON co.customer_zip_code_prefix = geo.geolocation_zip_code_prefix
WHERE order_status = "delivered"
GROUP BY geolocation_state
ORDER BY avg_estimated_Delivery_time DESC
LIMIT 5;
```

Business Case Study - 1 (SQL)



Query results

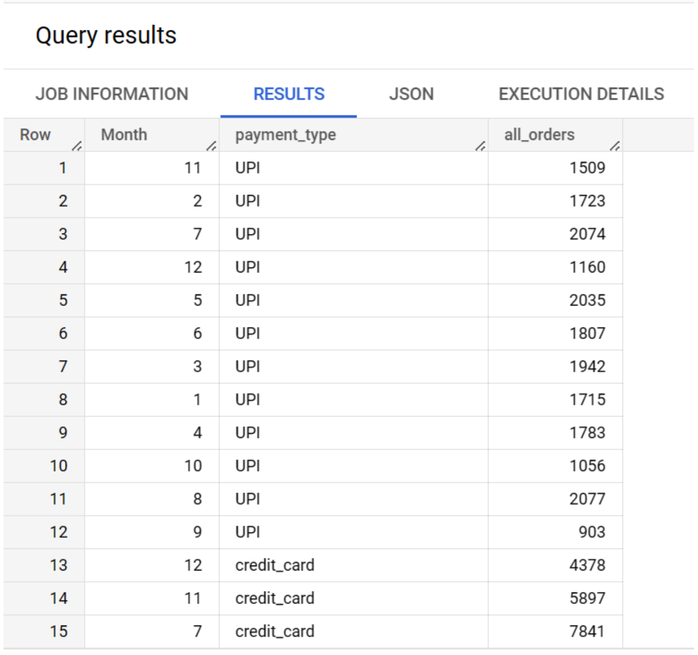
JOB INFORMATION RESULTS JSON EX

Row	geolocation_state	avg_estimated_[
1	RR	20.42
2	AM	20.13
3	RO	18.65
4	AC	18.46
5	AP	18.18

6. Payment type Analysis:

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

```
SELECT
  Month,
  payment_type,
  all_orders

FROM(SELECT COUNT(*) AS all_orders, EXTRACT(month FROM
order_purchase_timestamp) AS Month,

    pay.payment_type
    FROM `businesscase1sqlscaler.businesscasestudy1.orders` AS ord
    JOIN `businesscase1sqlscaler.businesscasestudy1.payments` AS pay
    ON ord.order_id = pay.order_id
    GROUP BY Month, pay.payment_type)
  ORDER BY payment_type;
```



Query results

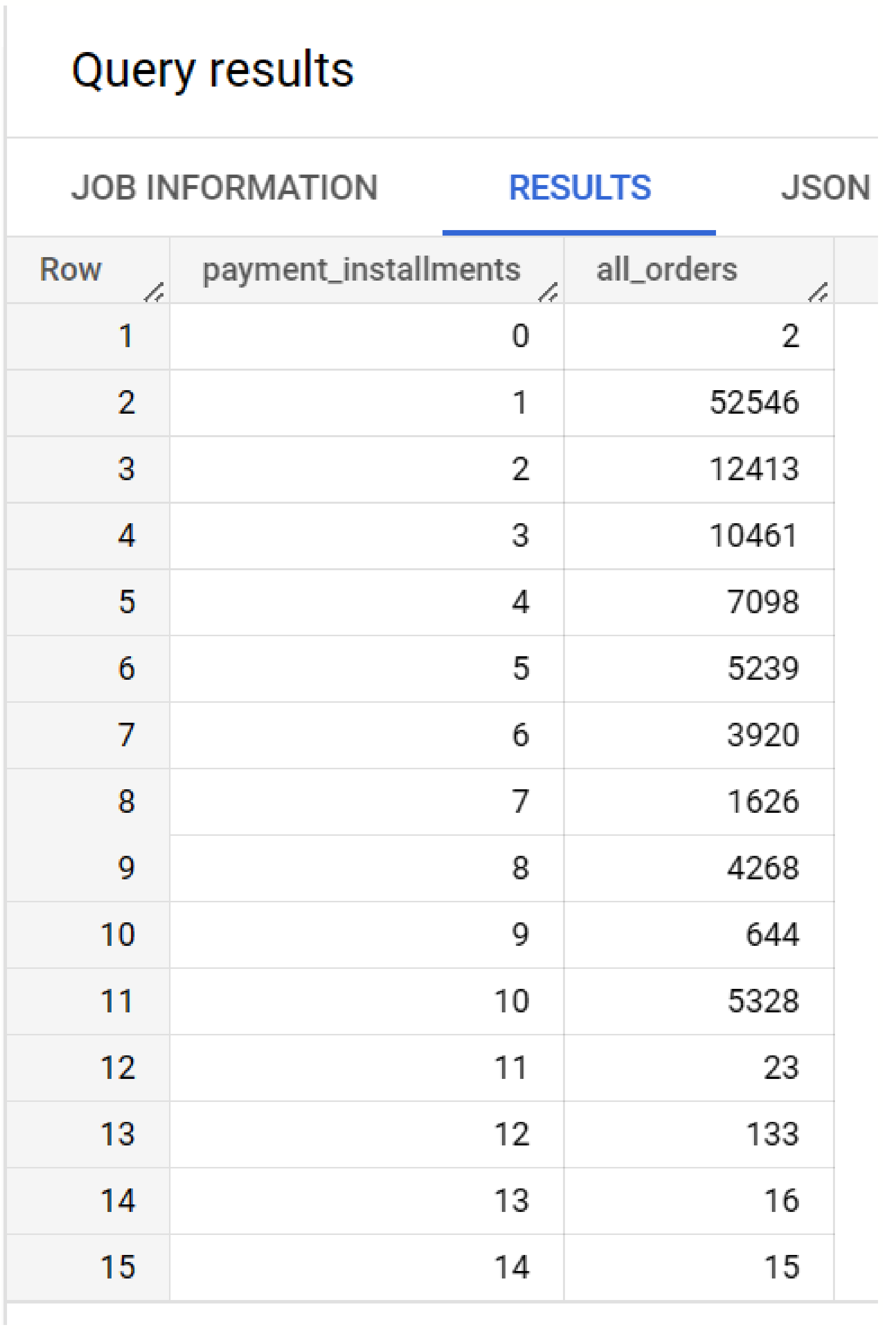
JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	Month	payment_type	all_orders
1	11	UPI	1509
2	2	UPI	1723
3	7	UPI	2074
4	12	UPI	1160
5	5	UPI	2035
6	6	UPI	1807
7	3	UPI	1942
8	1	UPI	1715
9	4	UPI	1783
10	10	UPI	1056
11	8	UPI	2077
12	9	UPI	903
13	12	credit_card	4378
14	11	credit_card	5897
15	7	credit_card	7841

Business Case Study - 1 (SQL)

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

```
SELECT

  payment_installments,

  COUNT(*) AS all_orders

FROM `businesscase1sqlscaler.businesscasestudy1.payments`

GROUP BY payment_installments;
```



Query results

Row	payment_installments	all_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
13	12	133
14	13	16
15	14	15

Business Case Study - 1 (SQL)

Actionable Insights:

- ➔ The time period of dataset is Starting from 2016-09-04 21:15:19 UTC & End in 2018-10-17 17:30:18 UTC
- ➔ There are 8011 cities and 27 states.
- ➔ According to the result there we have 3 months data of 2016, total year of 2017 And 10 months data of 2018. Clearly showing that the trend is increasing year by year according to the number of orders And sales. There is a trend going up from March(3) to August(8).
- ➔ Maximum orders in August And Minimum orders in September.
- ➔ Highest sales are in Afternoon period. Lowest sales are in Dawn period
- ➔ Order of Sales According to period:

  Dawn (5242) < Morning (27733) < Night (28331) < Afternoon (38135)

- ➔ Lowest Customers in RR(45) And Highest Customers in SP(40302).
- ➔ %increase in cost of order comparing 2017 - 2018 is 136.98.
- ➔ RR(42.98) and PB(42.72) are approximately the same in average freight value . That is the highest .
- ➔ SP has the lowest freight value that is 15.15.
- ➔ Fastest delivery in SP that takes just 8.3 days.
- ➔ Slowest delivery in RR that takes Approx 29 days.
- ➔ In this customers prefer most in 1 instalment. 52546 orders in one instalment.
- ➔ Customers pay through Credit cards in most purchases.

Recommendations:

- ☐ As per data people prefer to pay with credit card so we can provide some discounts through credit card.
- ☐ We can also research in the RR region why their customers are very low and attract customers. We can run some discounts and gift hampers also.
- ☐ Focus on delivery time so we can work on the supply chain to boost our delivery speed.
- ☐ Also focus on the highest customers in states like SP, RJ & MG we can consider to open some new stores.