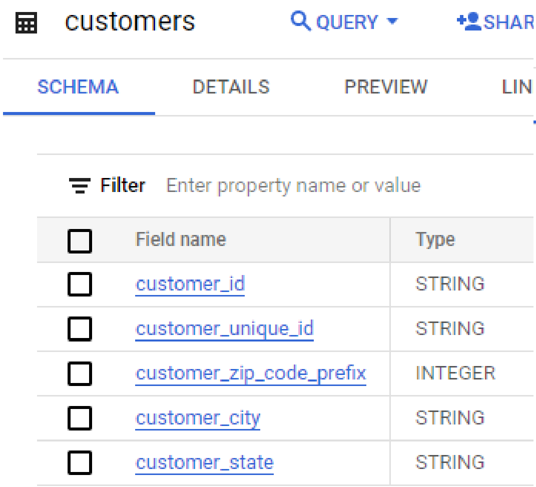
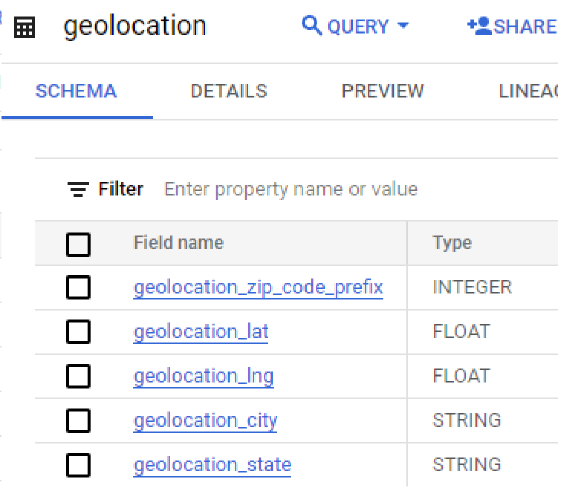
Ques 1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset**

1. Data type of columns in a table

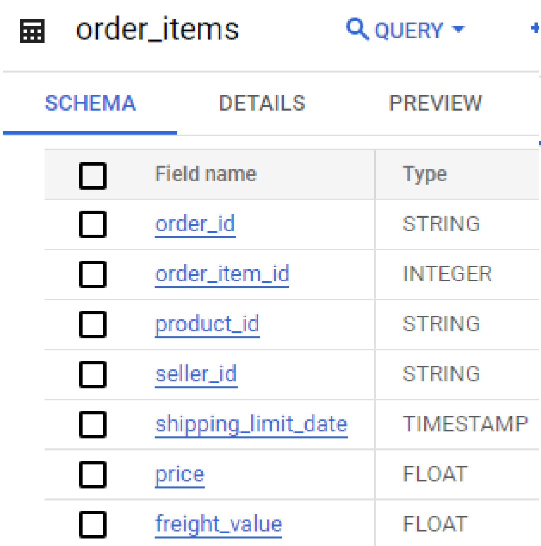
Ans:-



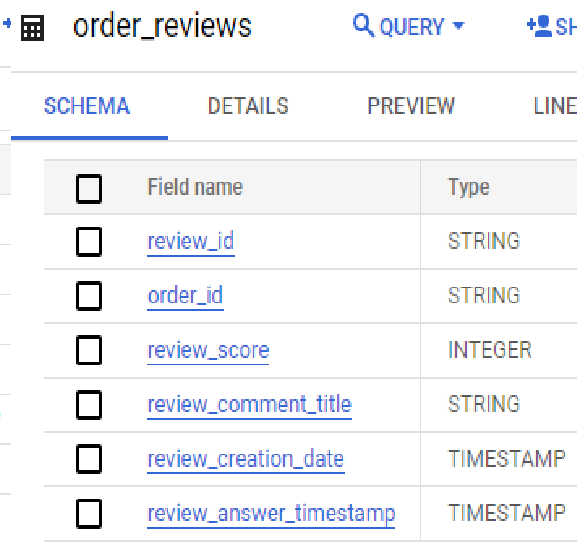
□	Field name	Type
□	customer_id	STRING
□	customer_unique_id	STRING
□	customer_zip_code_prefix	INTEGER
□	customer_city	STRING
□	customer_state	STRING



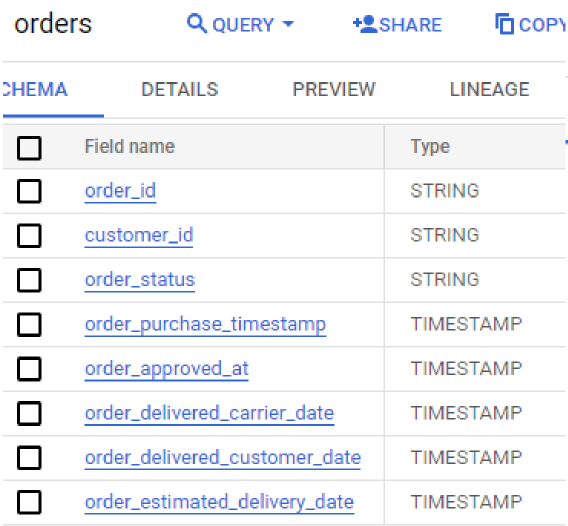
□	Field name	Type
□	geolocation_zip_code_prefix	INTEGER
□	geolocation_lat	FLOAT
□	geolocation_lng	FLOAT
□	geolocation_city	STRING
□	geolocation_state	STRING



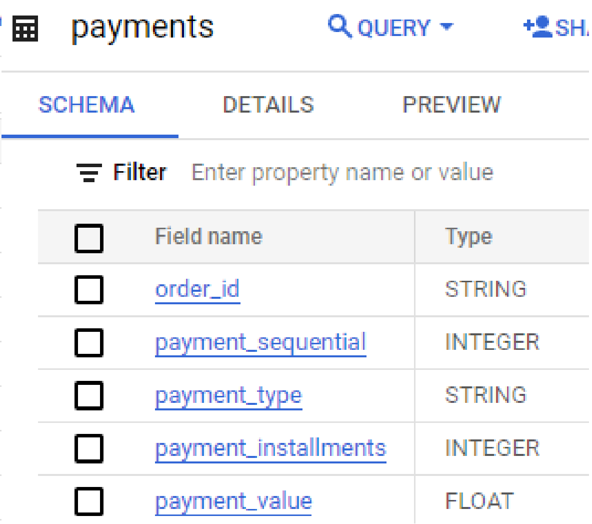
□	Field name	Type
□	order_id	STRING
□	order_item_id	INTEGER
□	product_id	STRING
□	seller_id	STRING
□	shipping_limit_date	TIMESTAMP
□	price	FLOAT
□	freight_value	FLOAT



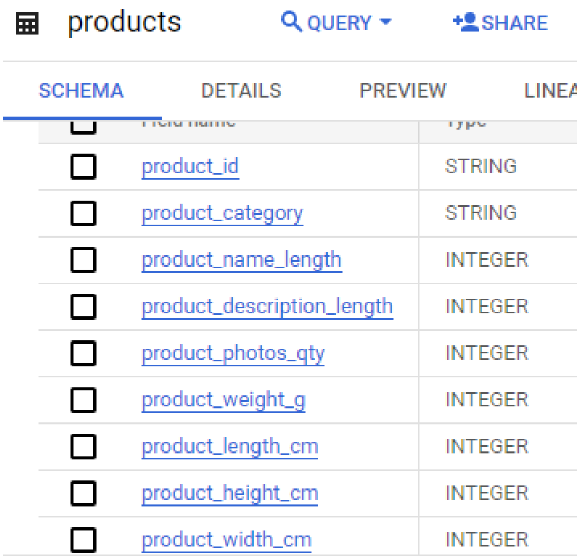
□	Field name	Type
□	review_id	STRING
□	order_id	STRING
□	review_score	INTEGER
□	review_comment_title	STRING
□	review_creation_date	TIMESTAMP
□	review_answer_timestamp	TIMESTAMP



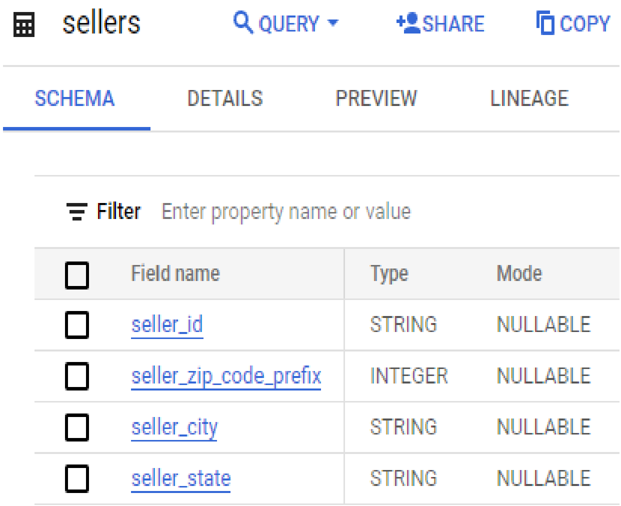
□	Field name	Type
□	order_id	STRING
□	customer_id	STRING
□	order_status	STRING
□	order_purchase_timestamp	TIMESTAMP
□	order_approved_at	TIMESTAMP
□	order_delivered_carrier_date	TIMESTAMP
□	order_delivered_customer_date	TIMESTAMP
□	order_estimated_delivery_date	TIMESTAMP



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



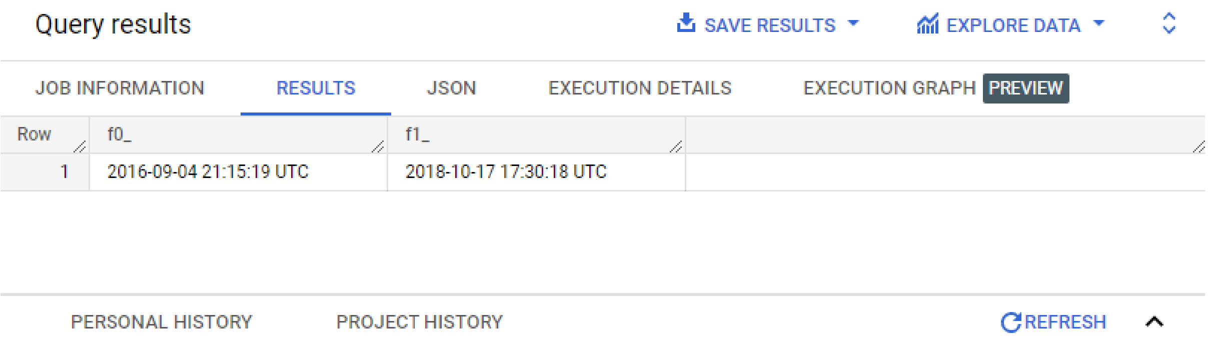
□	Field name	Type
□	product_id	STRING
□	product_category	STRING
□	product_name_length	INTEGER
□	product_description_length	INTEGER
□	product_photos_qty	INTEGER
□	product_weight_g	INTEGER
□	product_length_cm	INTEGER
□	product_height_cm	INTEGER
□	product_width_cm	INTEGER



□	Field name	Type	Mode
□	seller_id	STRING	NULLABLE
□	seller_zip_code_prefix	INTEGER	NULLABLE
□	seller_city	STRING	NULLABLE
□	seller_state	STRING	NULLABLE

2. Time period for which the data is given-(25 months)

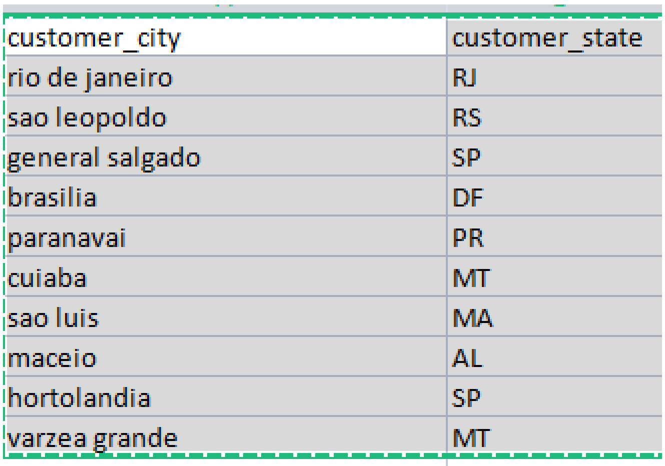
```
select min(order_purchase_timestamp),max(order_purchase_timestamp)
from `TARGET.orders`
```



Row	f0_	f1_
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 distinct c.customer_city, c.customer_state
from `TARGET.customers` c
inner join `TARGET.orders` o
on c.customer_id=o.customer_id;
```



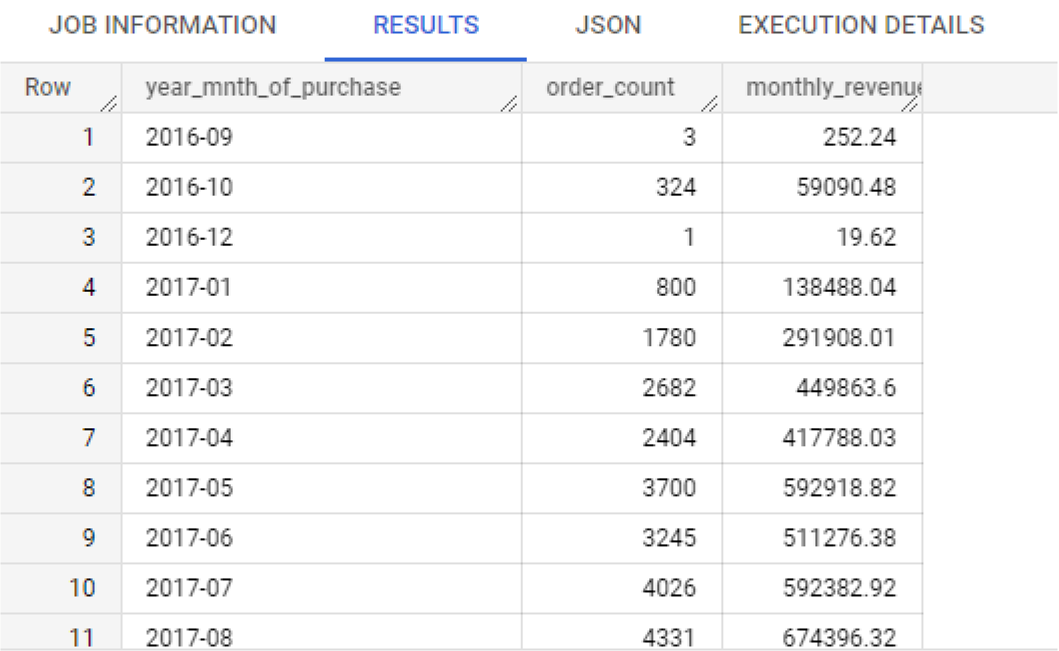
customer_city	customer_state
rio de janeiro	RJ
sao leopoldo	RS
general salgado	SP
brasilia	DF
paranavai	PR
cuiaba	MT
sao luis	MA
maceio	AL
hortolandia	SP
varzea grande	MT

Ques 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?

Ans:-

```
with monthly_purchase as (select order_id, format_date('%Y-
%m', date(order_purchase_timestamp))as year_mnth_of_purchase
from `TARGET.orders` ),

revenue as (select order_id, round(sum(payment_value),2) as total_payment
from `TARGET.payments`
group by order_id)

select mp.year_mnth_of_purchase,count(mp.order_id)as order_count,
round(sum(r.total_payment),2) as monthly_revenue
from monthly_purchase mp join
revenue r
on mp.order_id=r.order_id
group by mp.year_mnth_of_purchase
order by mp.year_mnth_of_purchase;
```



Row	year_mnth_of_purchase	order_count	monthly_revenu
1	2016-09	3	252.24
2	2016-10	324	59090.48
3	2016-12	1	19.62
4	2017-01	800	138488.04
5	2017-02	1780	291908.01
6	2017-03	2682	449863.6
7	2017-04	2404	417788.03
8	2017-05	3700	592918.82
9	2017-06	3245	511276.38
10	2017-07	4026	592382.92
11	2017-08	4331	674396.32

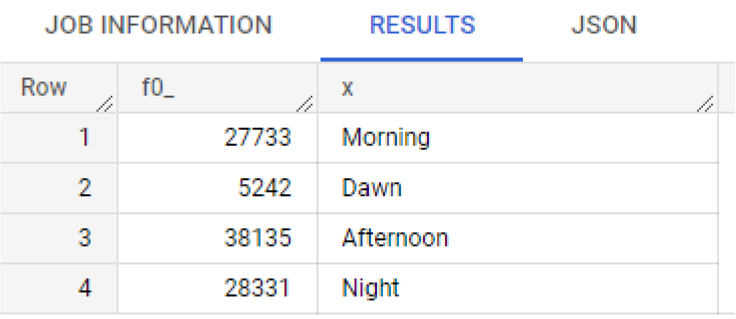
INSIGHT- As per data mentioned above, compared to 2017, revenue has increased in 2018 by almost 21%.

In May, July and August count of orders is more as compared to other months.

There is a increasing trend in orders and the same trend can be observed in monthly revenue tab.

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

```
 with extract_time as
(select extract(hour from order_purchase_timestamp)as HOUR,count(order_id)as ORDERS
from `TARGET.orders`
group by order_purchase_timestamp)

select sum(ORDERS),
case
when HOUR between 0 AND 6 then 'Dawn'
when HOUR between 7 and 12 then 'Morning'
when HOUR between 13 and 18 then 'Afternoon'
else 'Night'
end as x
from extract_time
group by (case
when HOUR between 0 AND 6 then 'Dawn'
when HOUR between 7 and 12 then 'Morning'
when HOUR between 13 and 18 then 'Afternoon'
else 'Night'
end);
```



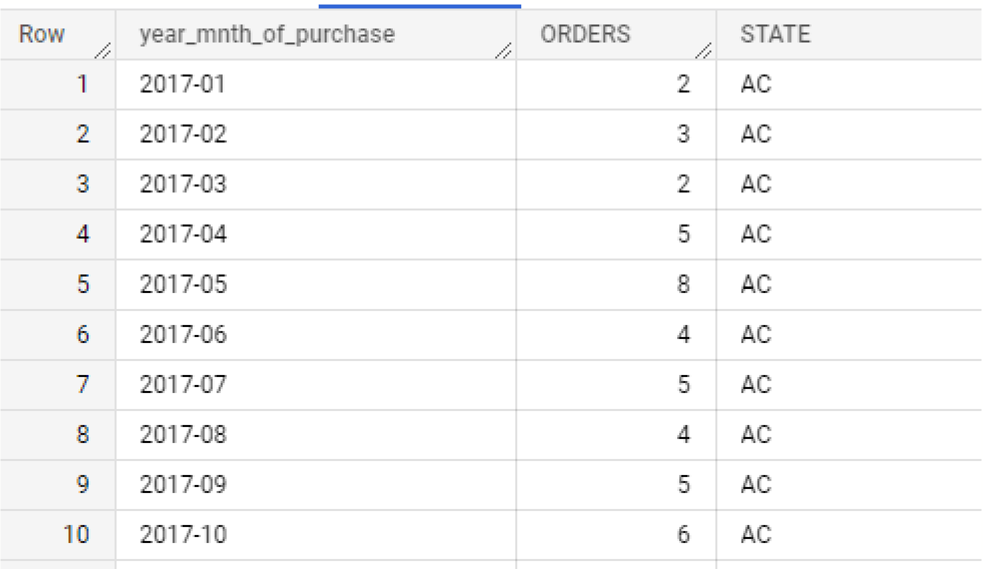
Row	f0_	x
1	27733	Morning
2	5242	Dawn
3	38135	Afternoon
4	28331	Night

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

1. Get month on month orders by states

Ans:-

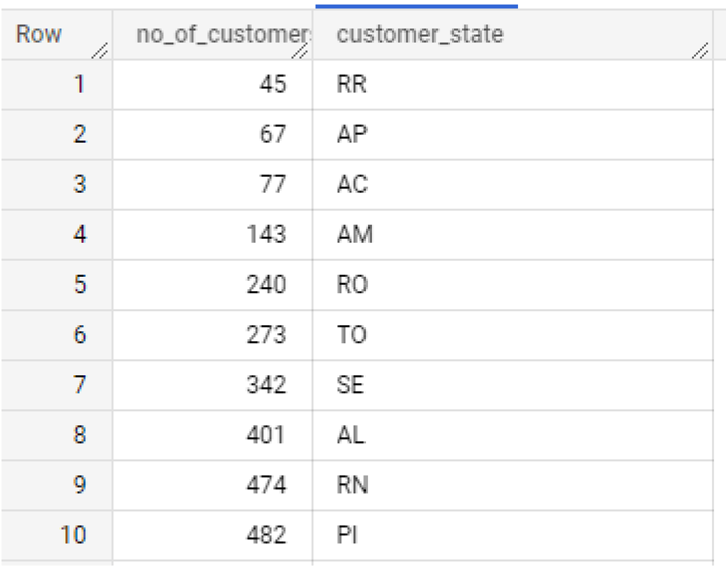
```
select format_date('%Y-%m', date(o.order_purchase_timestamp))as year_mnth_of_purchase,
count(o.order_id) as ORDERS, c.customer_state as STATE
from `TARGET.orders` o
inner join `TARGET.customers` c
on o.customer_id=c.customer_id
group by year_mnth_of_purchase, STATE
ORDER BY STATE,year_mnth_of_purchase ;
```



Row	year_mnth_of_purchase	ORDERS	STATE
1	2017-01	2	AC
2	2017-02	3	AC
3	2017-03	2	AC
4	2017-04	5	AC
5	2017-05	8	AC
6	2017-06	4	AC
7	2017-07	5	AC
8	2017-08	4	AC
9	2017-09	5	AC
10	2017-10	6	AC

2. Distribution of customers across the states in Brazil

```
SELECT count(distinct customer_unique_id) as no_of_customers, customer_state
from `TARGET.customers`
group by customer_state
order by no_of_customers;
```



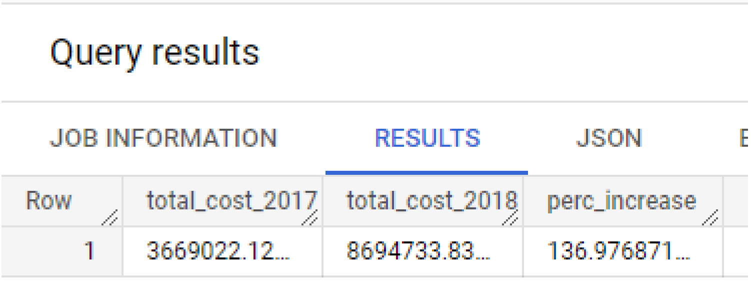
Row	no_of_customer	customer_state
1	45	RR
2	67	AP
3	77	AC
4	143	AM
5	240	RO
6	273	TO
7	342	SE
8	401	AL
9	474	RN
10	482	PI

Ques 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**

Ans:-

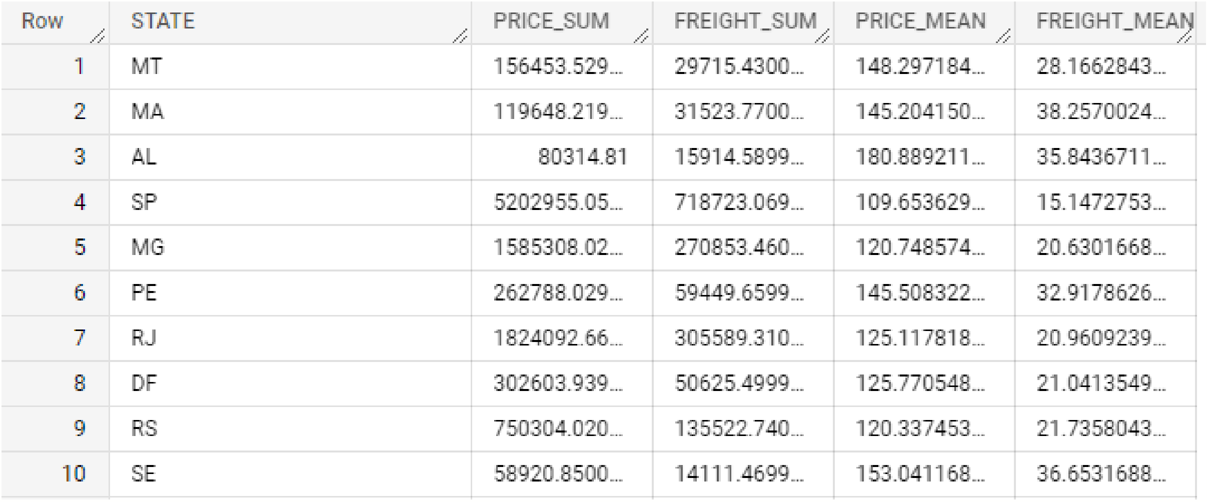
```
with orders_2017 as(select sum(p.payment_value) as payment_2017
from `TARGET.payments` p
inner join `TARGET.orders` o
on p.order_id=o.order_id
where extract (year from o.order_purchase_timestamp)=2017 and
extract (month from o.order_purchase_timestamp) between 1 and 8),

orders_2018 as
(select sum(p.payment_value)as payment_2018
from `TARGET.payments` p
inner join `TARGET.orders` o
on p.order_id=o.order_id
where extract (year from o.order_purchase_timestamp)=2018 and
extract (month from o.order_purchase_timestamp) between 1 and 8)

select orders_2017.payment_2017 as total_cost_2017,orders_2018.payment_2018 as total_cost_2018,
((orders_2018.payment_2018-orders_2017.payment_2017)/orders_2017.payment_2017)*100 as perc_increase
from orders_2017, orders_2018;
```



Row	total_cost_2017	total_cost_2018	perc_increase
1	3669022.12...	8694733.83...	136.976871...

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

```
with customer_state as (select customer_state, customer_id from `TARGET.customers`),
orders_customers as (select order_id, customer_id from `TARGET.orders`),
orders_freight as (select order_id, price, freight_value from `TARGET.order_items`)

select cs.customer_state as STATE, sum(ofr.price) as PRICE_SUM,sum(ofr.freight_value) as FREIGHT_SUM,
avg(ofr.price) as PRICE_MEAN,avg(ofr.freight_value) as FREIGHT_MEAN
from customer_state cs inner join
orders_customers oc on
cs.customer_id=oc.customer_id
inner join orders_freight ofr on
oc.order_id=ofr.order_id
group by cs.customer_state;
```



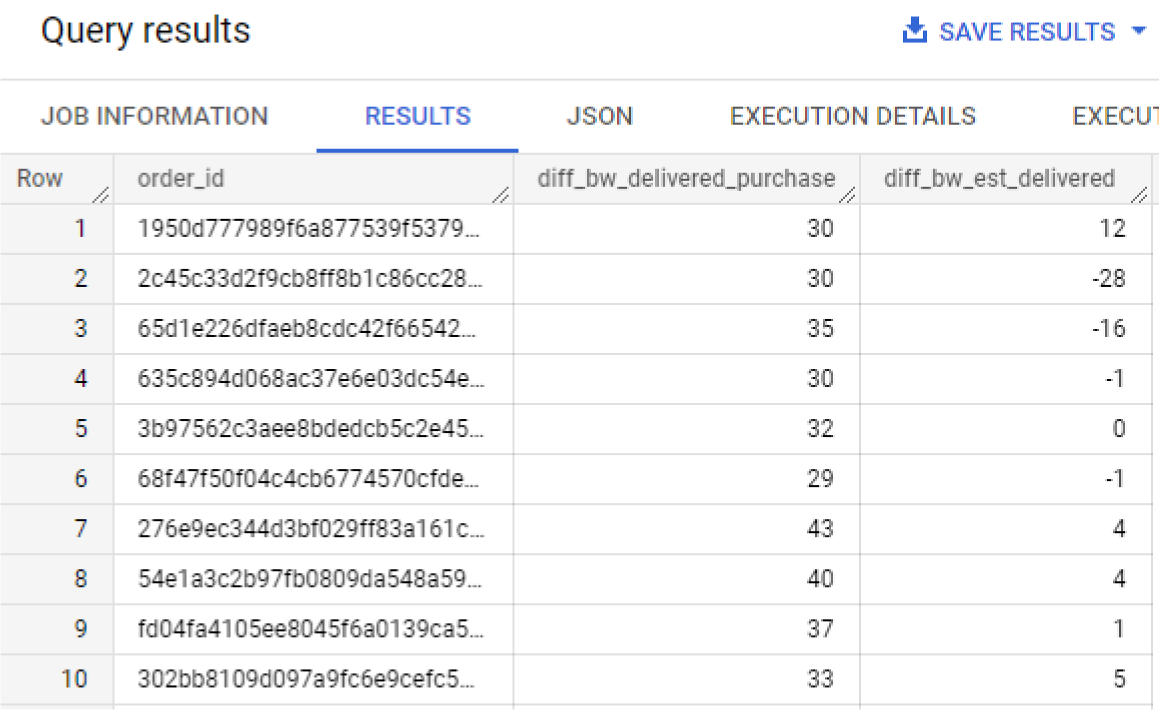
Row	STATE	PRICE_SUM	FREIGHT_SUM	PRICE_MEAN	FREIGHT_MEAN
1	MT	156453.529...	29715.4300...	148.297184...	28.1662843...
2	MA	119648.219...	31523.7700...	145.204150...	38.2570024...
3	AL	80314.81	15914.5899...	180.889211...	35.8436711...
4	SP	5202955.05...	718723.069...	109.653629...	15.1472753...
5	MG	1585308.02...	270853.460...	120.748574...	20.6301668...
6	PE	262788.029...	59449.6599...	145.508322...	32.9178626...
7	RJ	1824092.66...	305589.310...	125.117818...	20.9609239...
8	DF	302603.939...	50625.4999...	125.770548...	21.0413549...
9	RS	750304.020...	135522.740...	120.337453...	21.7358043...
10	SE	58920.8500...	14111.4699...	153.041168...	36.6531688...

Ques 5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

Ans:-

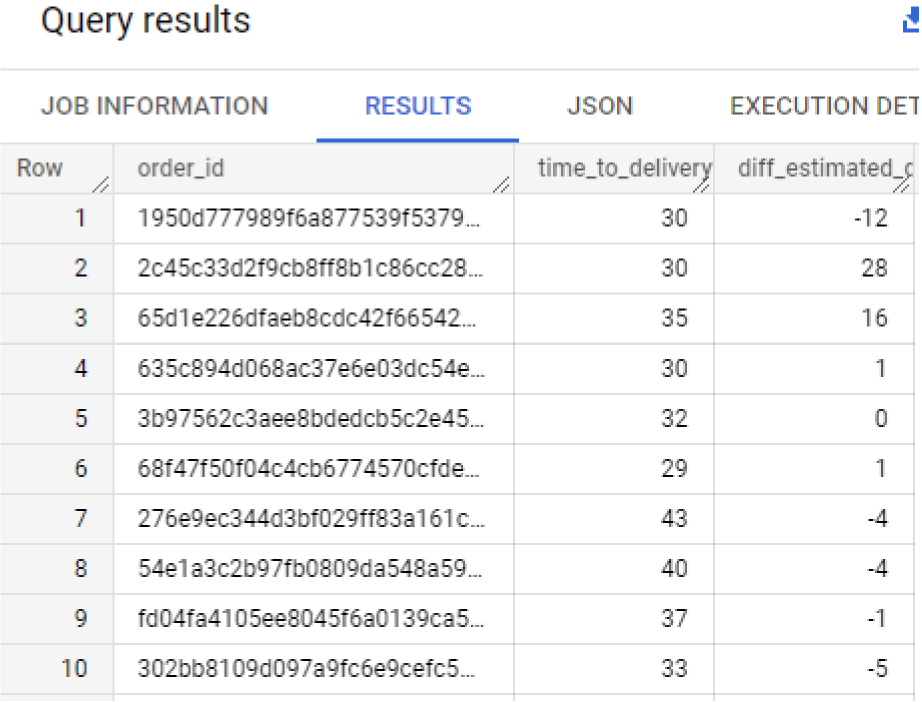
```
select order_id, date_diff(order_delivered_customer_date, order_purchase_timestamp,day) as
diff_bw_delivered_purchase, date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) as
diff_bw_est_delivered
from `TARGET.orders`
```



Row	order_id	diff_bw_delivered_purchase	diff_bw_est_delivered
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

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

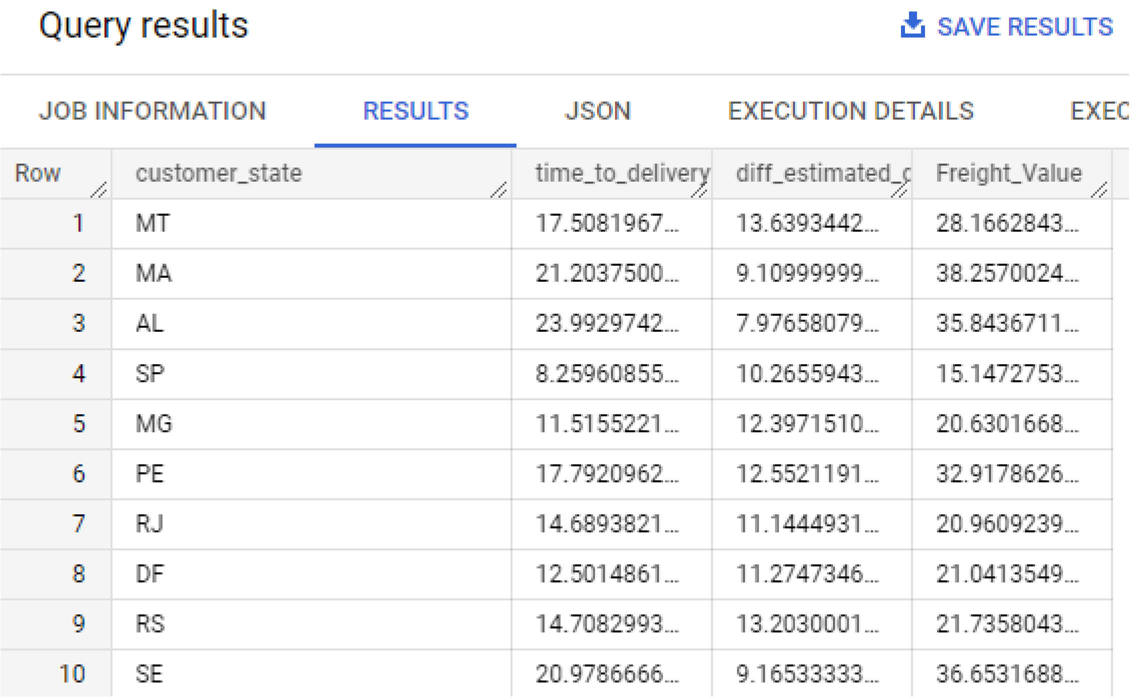
```
select  order_id, date_diff(order_delivered_customer_date, order_purchase_timestamp, day)as time_to_delivery,
date_diff(order_estimated_delivery_date, order_delivered_customer_date, day)as

diff_estimated_delivery
from `TARGET.orders`;
```



Row	order_id	time_to_delivery	diff_estimated_c
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

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

```
select c.customer_state, avg(date_diff(order_delivered_customer_date, order_purchase_timestamp, day))as
time_to_delivery, avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day))as
diff_estimated_delivery, avg(oi.freight_value) as Freight_Value
from `TARGET.customers` c
inner join `TARGET.orders` o
on o.customer_id=c.customer_id
inner join `TARGET.order_items` oi
on o.order_id=oi.order_id
group by c.customer_state;
```



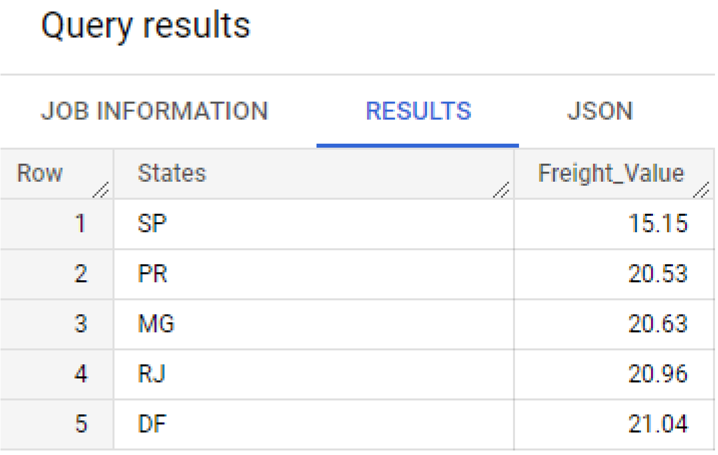
Row	customer_state	time_to_delivery	diff_estimated_c	Freight_Value
1	MT	17.5081967...	13.6393442...	28.1662843...
2	MA	21.2037500...	9.10999999...	38.2570024...
3	AL	23.9929742...	7.97658079...	35.8436711...
4	SP	8.25960855...	10.2655943...	15.1472753...
5	MG	11.5155221...	12.3971510...	20.6301668...
6	PE	17.7920962...	12.5521191...	32.9178626...
7	RJ	14.6893821...	11.1444931...	20.9609239...
8	DF	12.5014861...	11.2747346...	21.0413549...
9	RS	14.7082993...	13.2030001...	21.7358043...
10	SE	20.9786666...	9.16533333...	36.6531688...

4. Sort the data to get the following:

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

ASCENDING ORDER-

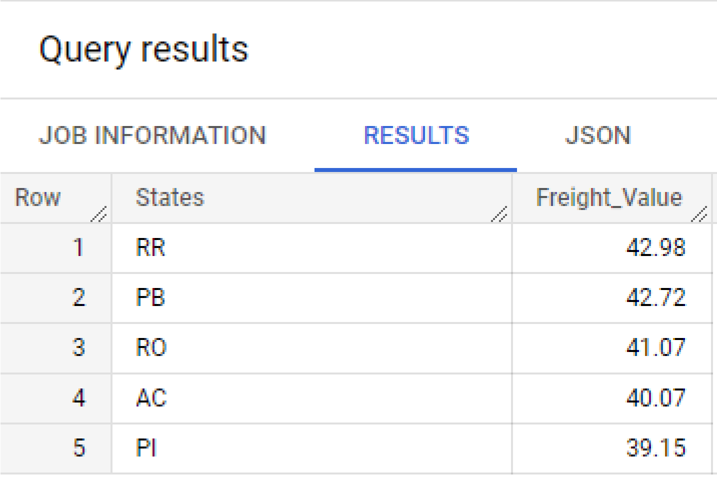
```
select c.customer_state as States,round(avg(oi.freight_value),2) as Freight_Value
from `TARGET.customers` c
inner join `TARGET.orders` o
on o.customer_id=c.customer_id
inner join `TARGET.order_items` oi
on o.order_id=oi.order_id
group by States
order by Freight_Value
limit 5;
```



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

DESCENDING ORDER-

```
select c.customer_state as States,round(avg(oi.freight_value),2) as Freight_Value
from `TARGET.customers` c
inner join `TARGET.orders` o
on o.customer_id=c.customer_id
inner join `TARGET.order_items` oi
on o.order_id=oi.order_id
group by States
order by Freight_Value desc
limit 5;
```

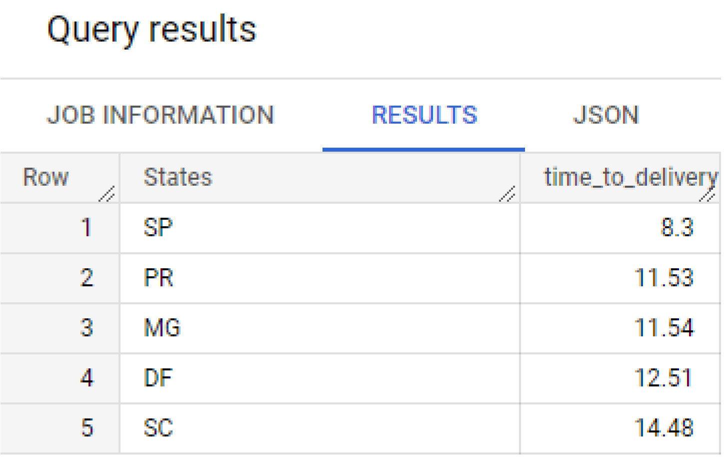


Row	States	Freight_Value
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

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

ASCENDING ORDER-

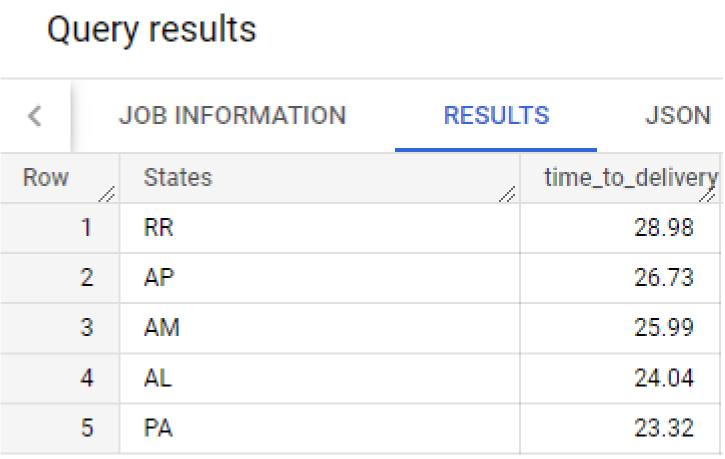
```
select c.customer_state as States,round(avg(date_diff(order_delivered_customer_date,
order_purchase_timestamp, day)),2)as time_to_delivery
from `TARGET.customers` c
inner join `TARGET.orders` o
on o.customer_id=c.customer_id
group by States
order by time_to_delivery
limit 5;
```



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

DESCENDING ORDER-

```
select c.customer_state as States,round(avg(date_diff(order_delivered_customer_date,
order_purchase_timestamp, day)),2)as time_to_delivery
from `TARGET.customers` c
inner join `TARGET.orders` o
on o.customer_id=c.customer_id
group by States
order by time_to_delivery desc
limit 5;
```



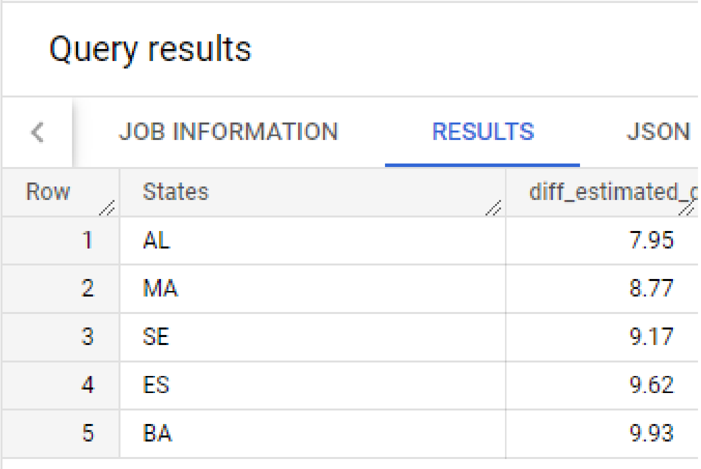
Row	States	time_to_delivery
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

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

ASCENDING ORDER-

```
select c.customer_state as
States,round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date, day)),2)as
diff_estimated_delivery
from `TARGET.customers` c
inner join `TARGET.orders` o
on o.customer_id=c.customer_id
group by States
order by diff_estimated_delivery
```

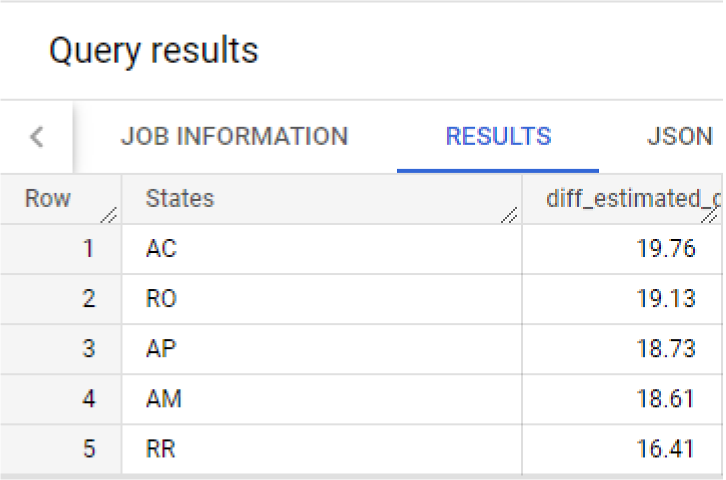
```
limit 5;
```



Row	States	diff_estimated_c
1	AL	7.95
2	MA	8.77
3	SE	9.17
4	ES	9.62
5	BA	9.93

DESCENDING ORDER-

```
select c.customer_state as
States,round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date, day)),2)as
diff_estimated_delivery
from `TARGET.customers` c
inner join `TARGET.orders` o
on o.customer_id=c.customer_id
group by States
order by diff_estimated_delivery desc
limit 5;
```



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

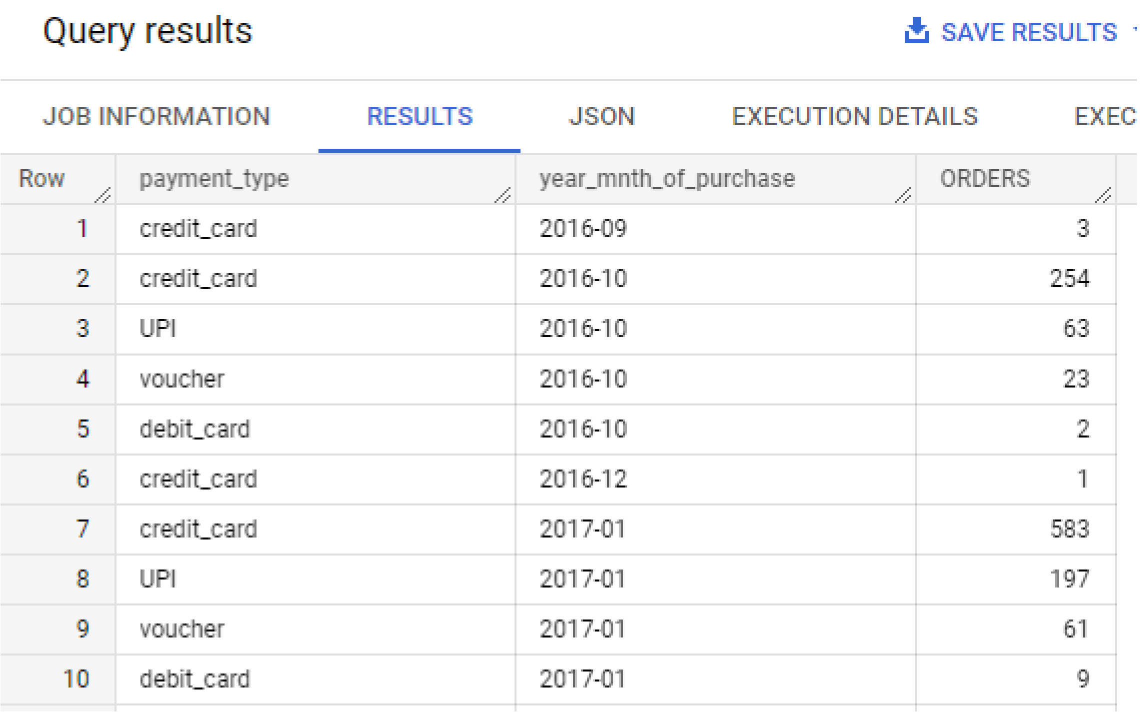
Ques 6. Payment type analysis:

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

Ans:-

```
select p.payment_type, format_date('%Y-%m', date(o.order_purchase_timestamp))as year_mnth_of_purchase,
count(o.order_id) as ORDERS
from `TARGET.orders` o
inner join `TARGET.payments` p
```

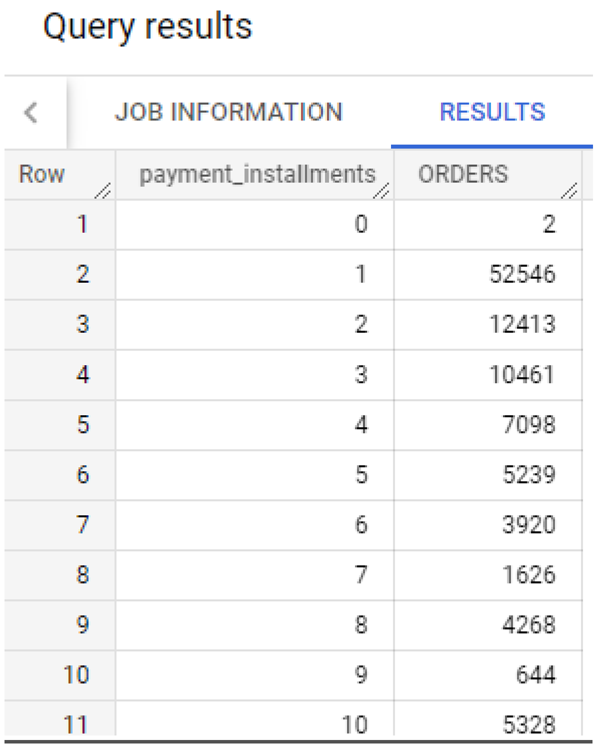
```
on o.order_id=p.order_id
group by year_mnth_of_purchase,p.payment_type
order by year_mnth_of_purchase;
```



Row	payment_type	year_mnth_of_purchase	ORDERS
1	credit_card	2016-09	3
2	credit_card	2016-10	254
3	UPI	2016-10	63
4	voucher	2016-10	23
5	debit_card	2016-10	2
6	credit_card	2016-12	1
7	credit_card	2017-01	583
8	UPI	2017-01	197
9	voucher	2017-01	61
10	debit_card	2017-01	9

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

```
select payment_installments, count(o.order_id) as ORDERS
from `TARGET.orders` o
inner join `TARGET.payments` p
on o.order_id=p.order_id
group by payment_installments
order by payment_installments;
```



Row	payment_installments	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

INSIGHTS-

We have 99,441 customers of data available with 96096 number of Unique Customers ids and 14994 different locations of customers

Customers are from different 4119 cities and 27 states from Brazil.

From total 99441 orders, 1107 are shipped ,625 were cancelled, 96478 are delivered.

Time period for which the data is given is 25 months.

São Paulo state has the highest numbers of sellers in country.

In products Data, total 32951 different products available in Target with 73 different product categories.

RECOMMENDATIONS-

As per analysis from the data, it can be seen that average time that is taken to complete the delivery is relatively very high almost double than the normal time taken to complete the delivery, so it should be reduced due to high competition rising in the e-commerce market.

If delivery becomes faster, them it will lead to rise in more customers which will result in more revenue.

Also, during October and January, sales are decreasing probably after Festival Sales.

So, to overcome this, discounts can be implemented on not so running products or whose demand is less as compared to other products so that it can attract more customers and also increase the sales revenue by selling more products during those low going months.