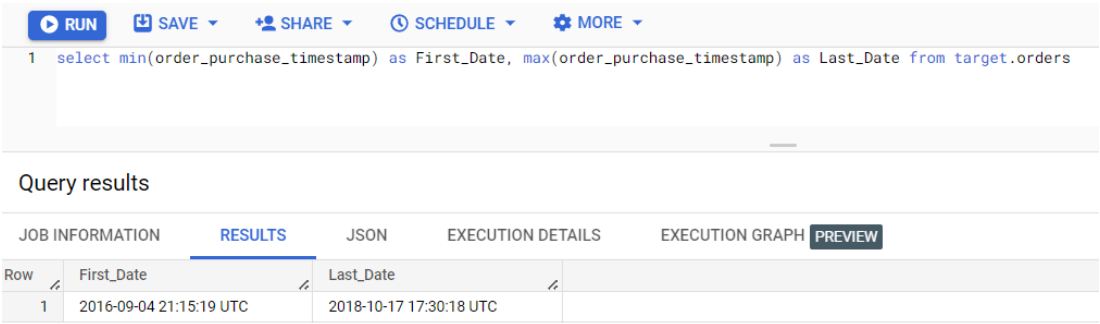
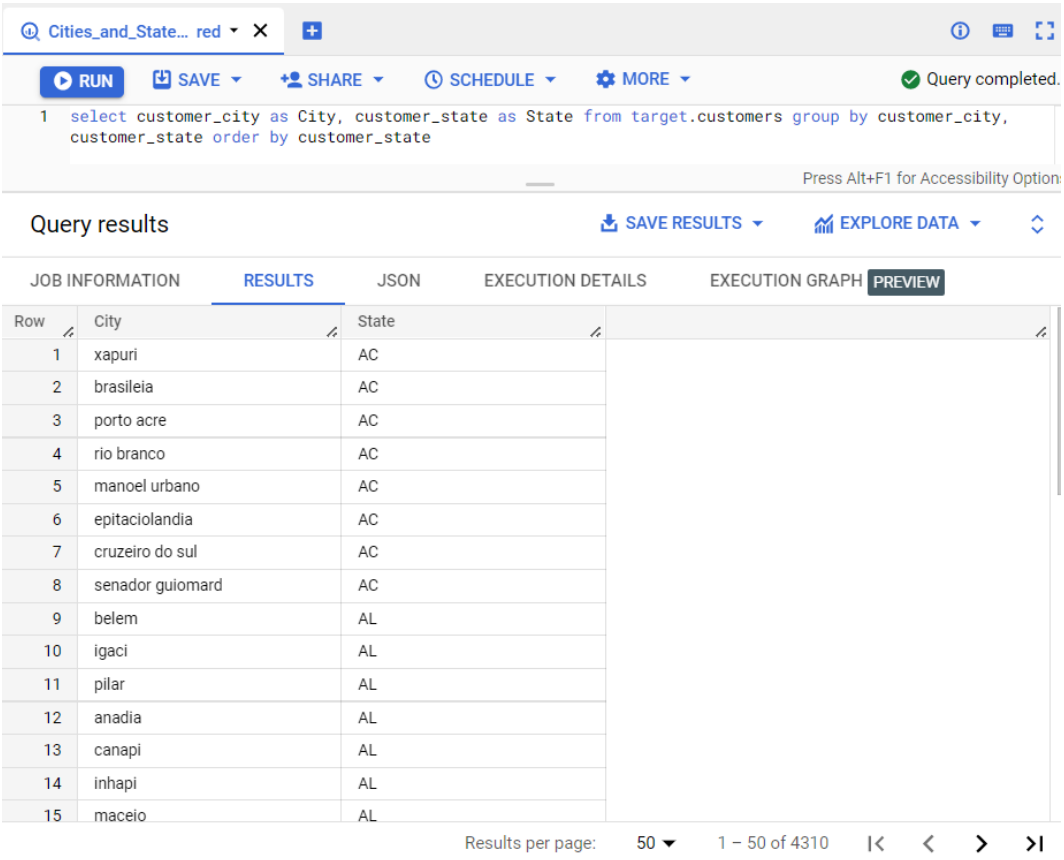
This is the Exploration of the Business Case of Target, It will help Target to increase the business and sales in the Brazil region.

1. Initial Exploration of the datasets:

- **Time Period of the Data Provided:**
  i. The data provided for the time period of 2 years approx. [04-09-2016 to 17-10-2018]
  ii. Query Used:
  ```
  select min(order_purchase_timestamp) as First_Date, max(order_purchase_timestamp) as Last_Date from target.orders
  ```



- **Cities and States covered in the provided datasets:**
  i. There are total of 4310 states and cities covered in the give datasets.
  ii. Query Used:
  ```
  select customer_city as City, customer_state as State from target.customers group by customer_city, customer_state order by customer_state
  ```



- **There were 8 Datasets provided, each dataset shares a relationship with another dataset via Foreign Key/Primary Key.**
- **Customers Dataset:**
  i. IT Contains the details of the customers.
  ii. customer_id is Foreign Key/Primary Key.
  iii. It shares relationship with Orders and Geolocation Datasets.

customers QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
customer_id	STRING	NULLABLE				
customer_unique_id	STRING	NULLABLE				
customer_zip_code_prefix	INTEGER	NULLABLE				
customer_city	STRING	NULLABLE				
customer_state	STRING	NULLABLE				

- **Orderitems Dataset:**
  i. It contains the details of the ordered items, such as products and it's price.
  ii. order_id, product_id and seller_id are the Foreign Keys / Primary Keys.
  iii. It shares relationship with Orders, Products and Sellers Datasets.

orderitems QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
order_id	STRING	NULLABLE				
order_item_id	INTEGER	NULLABLE				
product_id	STRING	NULLABLE				
seller_id	STRING	NULLABLE				
shipping_limit_date	TIMESTAMP	NULLABLE				
price	FLOAT	NULLABLE				
freight_value	FLOAT	NULLABLE				

- **Orders Dataset:**
  i. It contains the orders details, such as which customer ordered what and what is the expected delivery and when it got delivered.
  ii. order_is and customer_id are the Foreign Key / Primary Key.
  iii. It shares relationship with Customers, Orderreviews, Orderitems and Payments.

orders QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

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				

- **Payments Dataset:**
  i. It contains the details of the payments and mode of payment made for each order.
  ii. order_id is the foreign Key / Primary Key.
  iii. It shares relationship with orders dataset.

payments QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
order_id	STRING	NULLABLE				
payment_sequential	INTEGER	NULLABLE				
payment_type	STRING	NULLABLE				
payment_installments	INTEGER	NULLABLE				
payment_value	FLOAT	NULLABLE				

- **Products Dataset:**
  i. It contains the details of the products and it's prices.
  ii. product_id is the foreign Key / Primary Key.
  iii. It shares the relationship with orderitems dataset.

products QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
product_id	STRING	NULLABLE				
product_category	STRING	NULLABLE				
product_name_length	INTEGER	NULLABLE				
product_description_length	INTEGER	NULLABLE				
product_photos_qty	INTEGER	NULLABLE				
product_weight_g	INTEGER	NULLABLE				
product_length_cm	INTEGER	NULLABLE				
product_height_cm	INTEGER	NULLABLE				
product_width_cm	INTEGER	NULLABLE				

- **Sellers Dataset:**
  i. It contains the details of the sellers who sells the products.
  ii. seller_id is the foreign Key / Primary Key.
  iii. It shares the relationship with the orderitems dataset.

sellers QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW

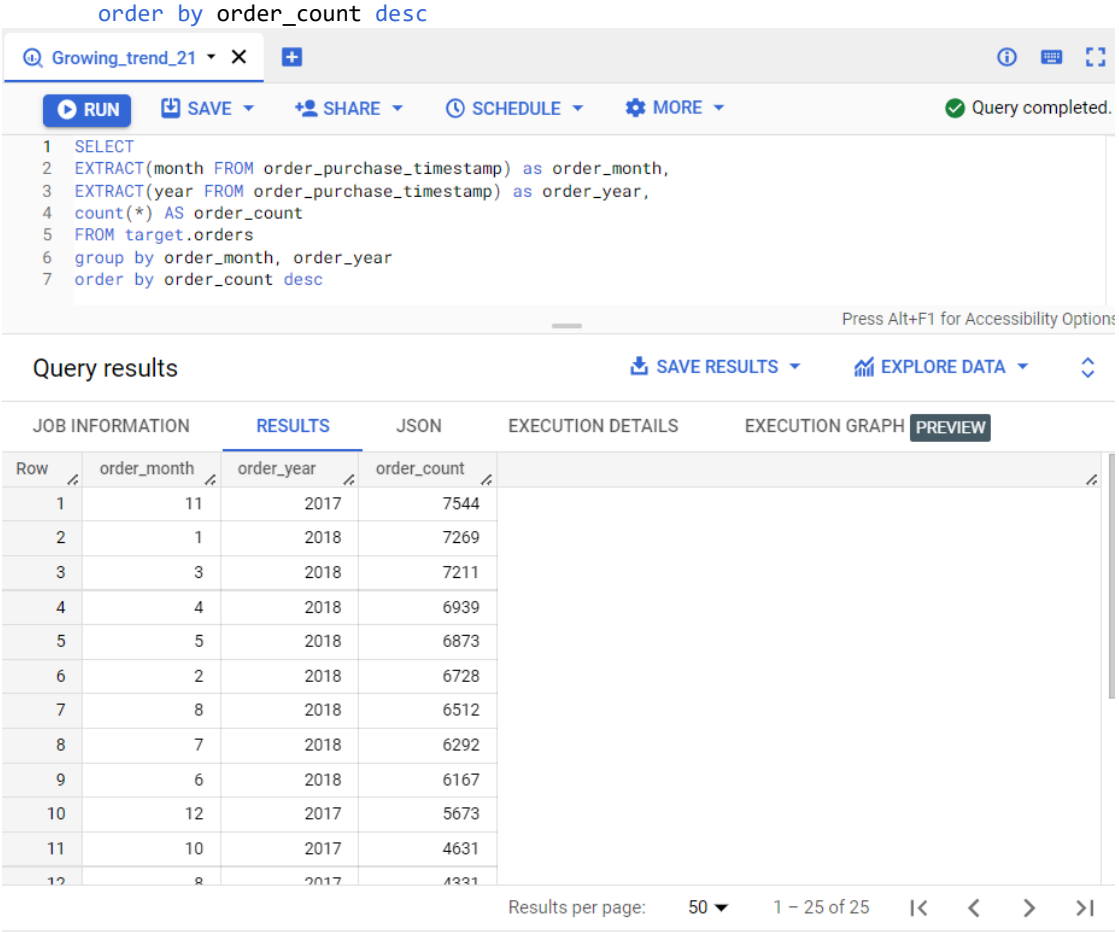
Filter Enter property name or value

Field name	Type	Mode	Collation	Default Value	Policy Tags	Description
seller_id	STRING	NULLABLE				
seller_zip_code_prefix	INTEGER	NULLABLE				
seller_city	STRING	NULLABLE				
seller_state	STRING	NULLABLE				

2. In-depth Exploration of the data.

- **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?**
  i. From 9th month of 2016 to 8th month of 2018 there is a gradual growth in number of orders in the e-commerce. In between there are some ups and downs in the number of orders.
  ii. In the 11th Month of 2017, there was a huge number of orders recorded, which is 7544. Which is the heights among all month from the provided data.
  iii. In the year 2018, there is gradual decrease in the number of orders every month, with an average order count of 5401.
  iv. In the year 2017, there is gradual increase in the number of orders every month, with an average order count of 3758.
  v. In the year 2016, there were only 3 months data were provided. With few hundred order count.
  vi. Query Used:

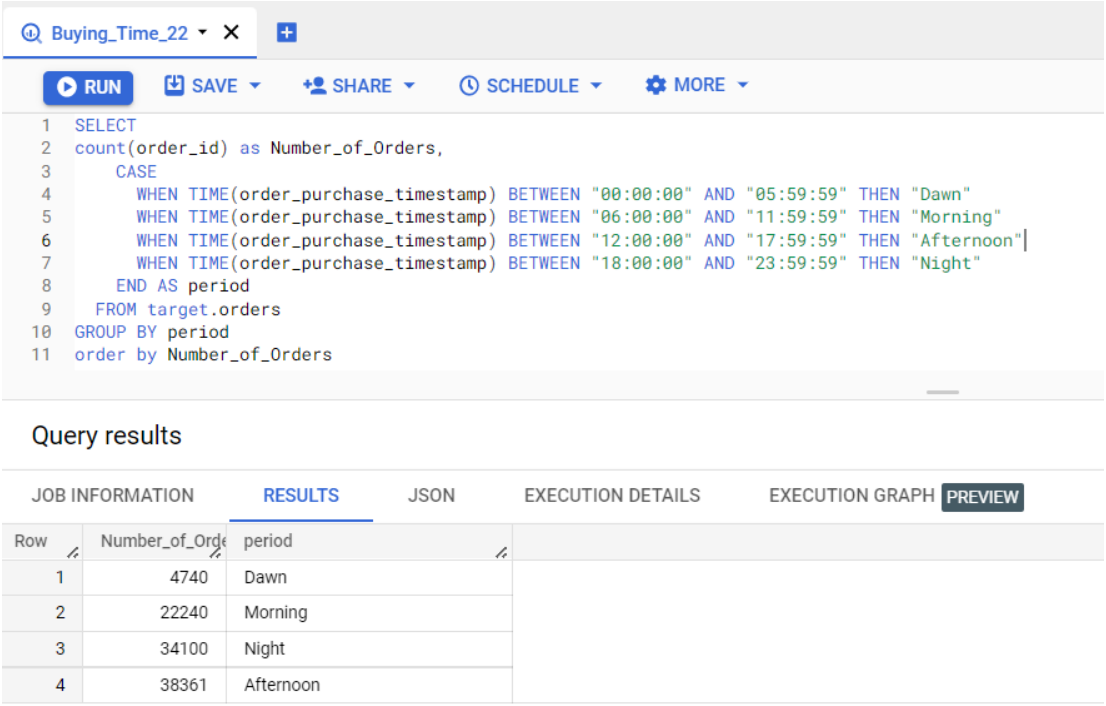
```
SELECT
EXTRACT(month FROM order_purchase_timestamp) as order_month,
EXTRACT(year FROM order_purchase_timestamp) as order_year,
count(*) AS order_count
FROM target.orders
group by order_month,order_year
order by order_count desc
```



Row	order_month	order_year	order_count
1	11	2017	7544
2	1	2018	7269
3	3	2018	7211
4	4	2018	6939
5	5	2018	6873
6	2	2018	6728
7	8	2018	6512
8	7	2018	6292
9	6	2018	6167
10	12	2017	5673
11	10	2017	4631
12	8	2017	4331

- **What time do Brazilian customers tend to buy ?**
    i. Based on the provided data, Brazilian customers tend to buy things during the Afternoon Time.
    ii. Query Used:

```
SELECT
count(order_id) as Number_of_Orders,
CASE
WHEN TIME(order_purchase_timestamp) BETWEEN "00:00:00" AND "05:59:59"
 THEN "Dawn"
WHEN TIME(order_purchase_timestamp) BETWEEN "06:00:00" AND "11:59:59"
 THEN "Morning"
WHEN TIME(order_purchase_timestamp) BETWEEN "12:00:00" AND "17:59:59"
 THEN "Afternoon"
WHEN TIME(order_purchase_timestamp) BETWEEN "18:00:00" AND "23:59:59"
 THEN "Night"
END AS period
FROM target.orders
GROUP BY period
order by Number_of_Orders
```



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

- **Month on Month orders by states:**
    i. With the below query, the number of orders happened in every month of every year in each states has been found
    ii. Query Used:

```
select
EXTRACT(MONTH from order_purchase_timestamp) as month,
EXTRACT(YEAR from order_purchase_timestamp) as year,
count(ord.order_id) as Number_of_orders,
cus.customer_state from target.orders as ord
join target.customers as cus
on cus.customer_id=ord.customer_id
group by month,year,cus.customer_state
order by customer_state, month, year
```

Orders_state_Mo... _31

RUN SAVE SHARE SCHEDULE MORE Query completed.

```
select EXTRACT(MONTH from order_purchase_timestamp) as month,
EXTRACT(YEAR from order_purchase_timestamp) as year,
count(ord.order_id) as Number_of_orders,
cus.customer_state from target.orders as ord
join target.customers as cus
on cus.customer_id=ord.customer_id
group by month,year,cus.customer_state
order by customer_state, month, year
```

Press Alt+F1 for Accessibility Option

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	month	year	Number_of_orde	customer_state
1	1	2017	2	AC
2	1	2018	6	AC
3	2	2017	3	AC
4	2	2018	3	AC
5	3	2017	2	AC
6	3	2018	2	AC
7	4	2017	5	AC
8	4	2018	4	AC
9	5	2017	8	AC
10	5	2018	2	AC
11	6	2017	4	AC

Results per page: 50 1 – 50 of 565

- **Distribution of customers across the states in Brazil:**
  i. With the below query, the number of customers in each state has been calculated.
  ii. Query Used:

```
SELECT count(DISTINCT(c.customer_id)) AS Number_of_Customers,
c.customer_state
FROM  target.customers c
GROUP BY c.customer_state
ORDER BY c.customer_state
```

customers_states_32

RUN SAVE SHARE SCHEDULE MORE Query completed.

```
SELECT count(DISTINCT(c.customer_id)) AS Number_of_Customers,
c.customer_state
FROM  target.customers c
GROUP BY c.customer_state
ORDER BY c.customer_state
```

Press Alt+F1 for Accessibility Options

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	Number_of_Cust	customer_state
1	81	AC
2	413	AL
3	148	AM
4	68	AP
5	3380	BA
6	1336	CE
7	2140	DF
8	2033	ES
9	2020	GO
10	747	MA
11	11635	MG
12	715	MS
13	907	MT

Results per page: 50 1 – 27 of 27

4. Impact on Economy:

- **% Increase in cost of orders from 2017 to 2018 (months between Jan to Aug only):**
  i. With the below query, the % Increase in cost of orders from 2017 to 2018 is calculated
  ii. Query Used:

```
with  year_wise_order AS
(
SELECT pay.order_id,pay.payment_value,
EXTRACT(year from order_purchase_timestamp) as year
FROM target.orders AS ord
JOIN target.payments AS pay
ON ord.order_id=pay.order_id
WHERE order_purchase_timestamp BETWEEN '2017-01-01' AND '2018-10-
31'
)
SELECT
count(a.order_id) AS count_of_order,
sum(a.payment_value) as Sum_of_payment, a.year,
(sum(a.payment_value) - (LAG(sum(a.payment_value)) OVER (ORDER BY
 a.year)))/(LAG(sum(a.payment_value))OVER (ORDER BY a.year)) * 10
0 AS percentage_increase_orders
FROM year_wise_order as a GROUP BY a.year order by a.year;
```

*Unsaved query 5

RUN SAVE SHARE SCHEDULE MORE

```
with  year_wise_order AS
(
SELECT pay.order_id,pay.payment_value,
EXTRACT(year from order_purchase_timestamp) as year
FROM target.orders AS ord
JOIN target.payments AS pay
ON ord.order_id=pay.order_id
WHERE order_purchase_timestamp BETWEEN '2017-01-01' AND '2018-10-31'
)
SELECT
count(a.order_id) AS count_of_order,
sum(a.payment_value) as Sum_of_payment, a.year,
(sum(a.payment_value) - (LAG(sum(a.payment_value)) OVER (ORDER BY a.year)))/(LAG(sum(a.payment_value))
OVER (ORDER BY a.year)) * 100 AS percentage_increase_orders
FROM year_wise_order as a GROUP BY a.year order by a.year;
```

Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	count_of_order	Sum_of_paymer	year	percentage_incr
1	47525	7249746.72...	2017	null
2	56015	8699763.04...	2018	20.0009238...

- **Mean & Sum of price and freight value by customer state:**
  i. With the below query, the Sum and Mean of Freight Value and Price in each state has been calculated.
  ii. Query Used:

```
select cus.customer_state,
Round(avg(ordi.price),2) as Mean_of_Price,
Round(avg(ordi.freight_value),2) as Mean_of_Freight_Value,
Round(sum(ordi.price),2) as Sum_of_Price,
Round(sum(ordi.freight_value),2) as Sum_of_Freight_Value
from target.orderitems as ordi
join target.orders as ord
on ord.order_id=ordi.order_id
join target.customers as cuson ord.customer_id=cus.customer_id
group by cus.customer_state
```

sum_mean_price... _42

RUN SAVE SHARE SCHEDULE MORE Query completed.

```
select cus.customer_state,
Round(avg(ordi.price),2) as Mean_of_Price,
Round(avg(ordi.freight_value),2) as Mean_of_Freight_Value,
Round(sum(ordi.price),2) as Sum_of_Price,
Round(sum(ordi.freight_value),2) as Sum_of_Freight_Value
from target.orderitems as ordi
join target.orders as ord
on ord.order_id=ordi.order_id
join target.customers as cus
on ord.customer_id=cus.customer_id
group by cus.customer_state
```

Press Alt+F1 for Accessibility Options

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	customer_state	Mean_of_Price	Mean_of_Freight	Sum_of_Price	Sum_of_Freight
1	MT	148.3	28.17	156453.53	29715.43
2	MA	145.2	38.26	119648.22	31523.77
3	AL	180.89	35.84	80314.81	15914.59
4	SP	109.65	15.15	5202955.05	718723.07
5	MG	120.75	20.63	1585308.03	270853.46
6	PE	145.51	32.92	262788.03	59449.66
7	RJ	125.12	20.96	1824092.67	305589.31
8	DF	125.77	21.04	302603.94	50625.5
9	RS	120.34	21.74	750304.02	135522.74

Results per page: 50 1 – 27 of 27

5. Analysis on sales, freight and delivery time:

- **Top 5 States with Highest Average freight value**
  i. The top 5 Customer states with Highest Average freight value are PB, RR, RO, AC and PI.
  ii. Query Used:

```
with time_analysis as(
select customer_id,order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp
,day) as time_to_delivery,
timestamp_diff(order_estimated_delivery_date,order_delivered_customer
_date,day) as diff_estimated_delivery
from target.orders
where order_purchase_timestamp is not null
  and order_delivered_customer_date is not null
  and order_estimated_delivery_date is not null
)
select t2.customer_state,avg(t1.time_to_delivery) as time_to_delivery
,avg(t1.diff_estimated_delivery) as diff_estimated_delivery,avg(t3.fr
eight_value) as freight_value from target.customers as t2
join  time_analysis as t1
on t1.customer_id=t2.customer_id
join target.orderitems as t3
on t1.order_id=t3.order_id
group by t2.customer_state
order by freight_value desc
limit 5
```

*high_freight_value_541

RUN SAVE SHARE SCHEDULE MORE This query will process 16.82 MB when

```
with time_analysis as(
select customer_id,order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,
timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_estimated_delivery
from target.orders
where order_purchase_timestamp is not null
  and order_delivered_customer_date is not null
  and order_estimated_delivery_date is not null
)
select t2.customer_state,avg(t1.time_to_delivery) as time_to_delivery,avg(t1.diff_estimated_delivery)
as diff_estimated_delivery,avg(t3.freight_value) as freight_value from target.customers as t2
join  time_analysis as t1
on t1.customer_id=t2.customer_id
join target.orderitems as t3
on t1.order_id=t3.order_id
group by t2.customer_state
order by freight_value desc
limit 5
```

Press Alt+F1 for Accessibility Opti

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	customer_state	time_to_delivery	diff_estimated_c	freight_value
1	PB	20.1194539...	12.1501706...	43.0916894...
2	RR	27.8260869...	17.4347826...	43.0880434...
3	RO	19.2820512...	19.0805860...	41.3305494...
4	AC	20.3296703...	20.0109890...	40.0479120...
5	PI	18.9311663...	10.6826003...	39.1150860...

- **Top 5 States with Lowest Average freight value**
  i. The top 5 Customer states with Lowest Average freight value are SP, PR, MG, RJ and DF.
  ii. Query Used:

```
with time_analysis as(
select customer_id,order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_times
tamp,day) as time_to_delivery,
timestamp_diff(order_estimated_delivery_date,order_delivered_cust
omer_date,day) as diff_estimated_delivery
from target.orders
where order_purchase_timestamp is not null
  and order_delivered_customer_date is not null
  and order_estimated_delivery_date is not null
)
select t2.customer_state,avg(t1.time_to_delivery) as time_to_deli
very,avg(t1.diff_estimated_delivery) as diff_estimated_delivery,a
vg(t3.freight_value) as freight_value from target.customers as t2
join  time_analysis as t1
on t1.customer_id=t2.customer_id
join target.orderitems as t3
on t1.order_id=t3.order_id
group by t2.customer_state
order by freight_value
limit 5
```

low_freight_value_541

RUN SAVE SHARE SCHEDULE MORE

```
with time_analysis as(
select customer_id,order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,
timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_estimated_delivery
from target.orders
where order_purchase_timestamp is not null
  and order_delivered_customer_date is not null
  and order_estimated_delivery_date is not null
)
select t2.customer_state,avg(t1.time_to_delivery) as time_to_delivery,avg(t1.diff_estimated_delivery)
as diff_estimated_delivery,avg(t3.freight_value) as freight_value from target.customers as t2
join  time_analysis as t1
on t1.customer_id=t2.customer_id
join target.orderitems as t3
on t1.order_id=t3.order_id
group by t2.customer_state
order by freight_value
limit 5
```

Press Alt+F1 for Accessibility Options

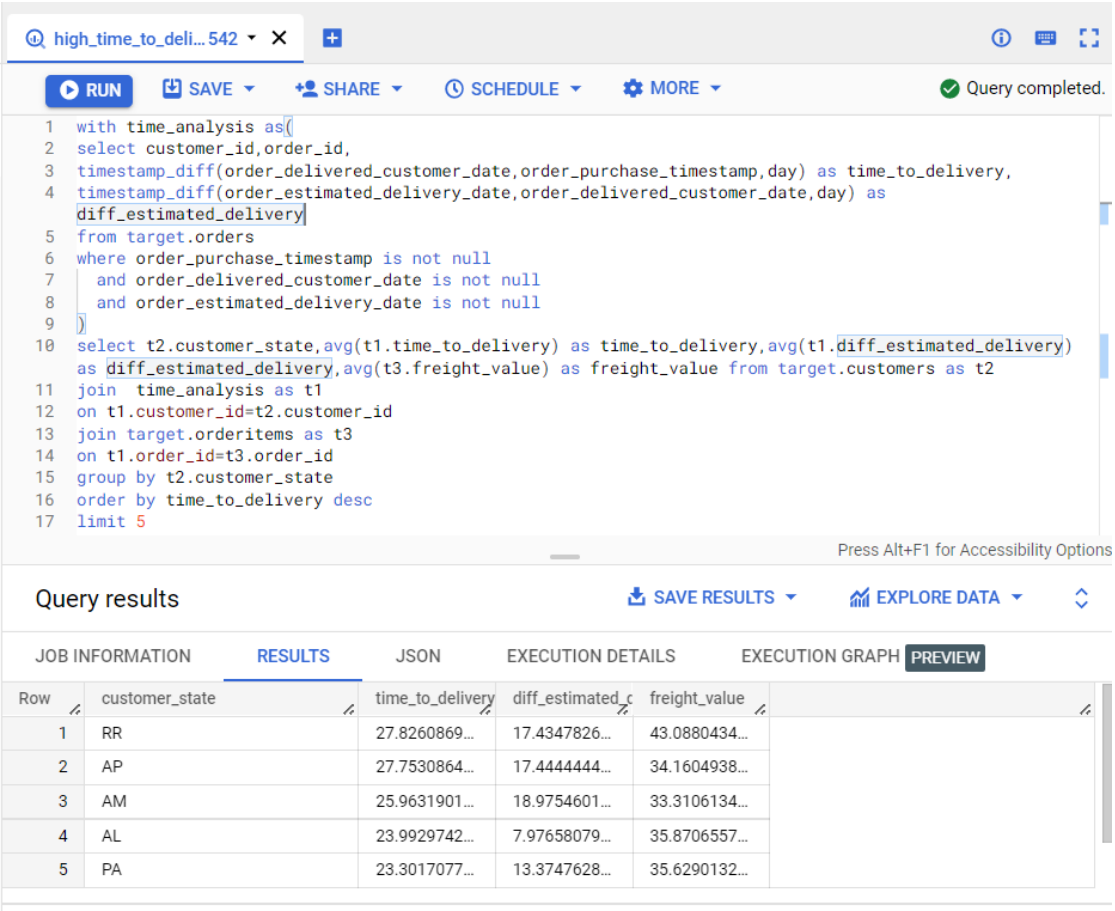
Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	customer_state	time_to_delivery	diff_estimated_c	freight_value
1	SP	8.25960855...	10.2655943...	15.1149940...
2	PR	11.4807930...	12.5338998...	20.4718162...
3	MG	11.5155221...	12.3971510...	20.6258372...
4	RJ	14.6893821...	11.1444931...	20.9097843...
5	DF	12.5014861...	11.2747346...	21.0721613...

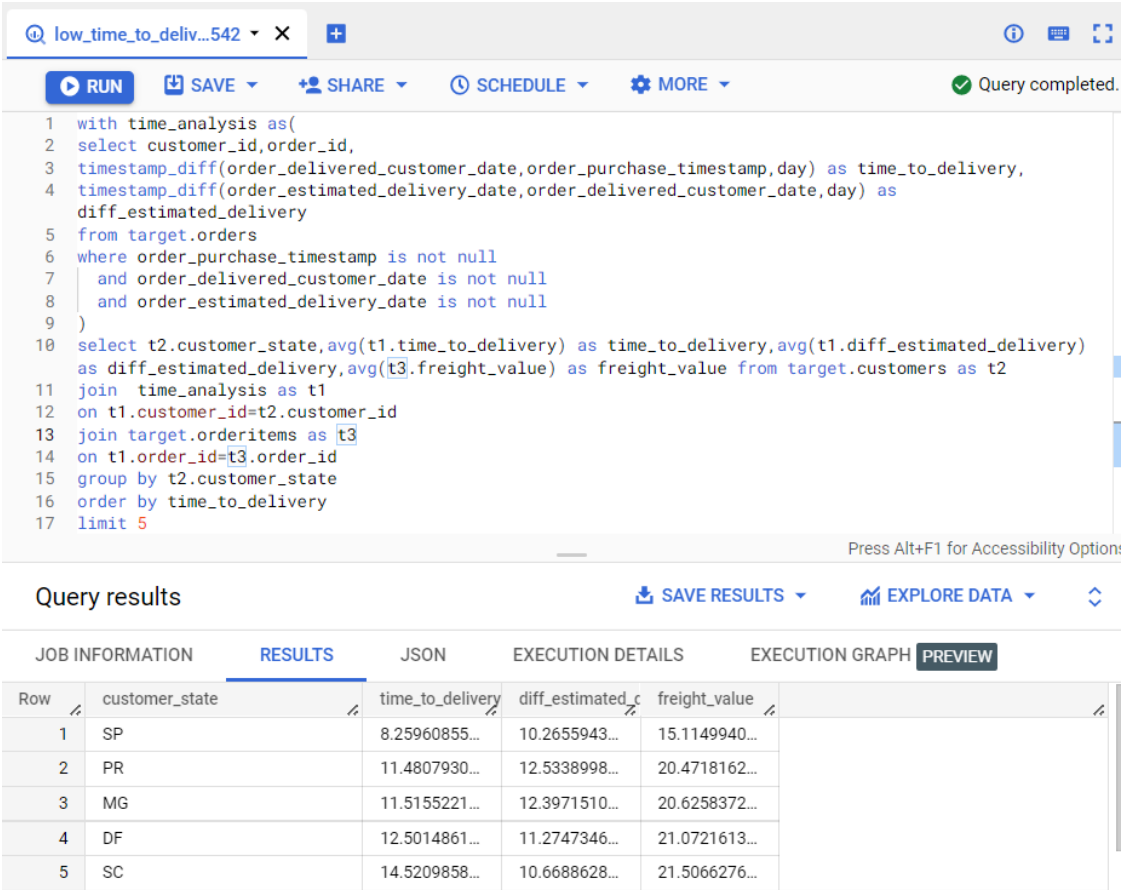
- **Top 5 States with Highest Average Time to delivery**
  i. The top 5 Customer states with Highest Average Time to Delivery are RR, AP, AM, AL and PA
  ii. Query Used:

```
with time_analysis as(
select customer_id,order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_times
tamp,day) as time_to_delivery,
timestamp_diff(order_estimated_delivery_date,order_delivered_cust
omer_date,day) as diff_estimated_delivery
from target.orders
where order_purchase_timestamp is not null
  and order_delivered_customer_date is not null
  and order_estimated_delivery_date is not null
)
select t2.customer_state,avg(t1.time_to_delivery) as time_to_deli
very,avg(t1.diff_estimated_delivery) as diff_estimated_delivery,a
vg(t3.freight_value) as freight_value from target.customers as t2
join  time_analysis as t1
on t1.customer_id=t2.customer_id
join target.orderitems as t3
on t1.order_id=t3.order_id
group by t2.customer_state
order by time_to_delivery desc
limit 5
```



- **Top 5 States with Lowest Average Time to delivery**
  i. The top 5 Customer states with Lowest Average Time to Delivery are SP, PR, MG, DF and SC
  ii. Query Used:

```
with time_analysis as(
select customer_id,order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_times
tamp,day) as time_to_delivery,
timestamp_diff(order_estimated_delivery_date,order_delivered_cust
omer_date,day) as diff_estimated_delivery
from target.orders
where order_purchase_timestamp is not null
  and order_delivered_customer_date is not null
  and order_estimated_delivery_date is not null
)
select t2.customer_state,avg(t1.time_to_delivery) as time_to_deli
very,avg(t1.diff_estimated_delivery) as diff_estimated_delivery,a
vg(t3.freight_value) as freight_value from target.customers as t2
join  time_analysis as t1
on t1.customer_id=t2.customer_id
join target.orderitems as t3
on t1.order_id=t3.order_id
group by t2.customer_state
order by time_to_delivery
limit 5
```



- **Top 5 States where delivery is fast**
  i. The top 5 Customer states delivery is fast are AC, RO, AM, AP and RR.
  ii. Query Used:

```
with time_analysis as(
select customer_id,order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_times
tamp,day) as time_to_delivery,
timestamp_diff(order_estimated_delivery_date,order_delivered_cust
omer_date,day) as diff_estimated_delivery
from target.orders
where order_purchase_timestamp is not null
  and order_delivered_customer_date is not null
  and order_estimated_delivery_date is not null
)
select t2.customer_state,avg(t1.time_to_delivery) as time_to_deli
very,avg(t1.diff_estimated_delivery) as diff_estimated_delivery,a
vg(t3.freight_value) as freight_value from target.customers as t2
join  time_analysis as t1
on t1.customer_id=t2.customer_id
join target.orderitems as t3
on t1.order_id=t3.order_id
group by t2.customer_state
order by diff_estimated_delivery desc
limit 5
```

high_estimated_d...543

RUN SAVE SHARE SCHEDULE MORE Query completed.

```
with time_analysis as(
select customer_id,order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,
timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_estimated_delivery
from target.orders
where order_purchase_timestamp is not null
  and order_delivered_customer_date is not null
  and order_estimated_delivery_date is not null
)
select t2.customer_state,avg(t1.time_to_delivery) as time_to_delivery,avg(t1.diff_estimated_delivery)
as diff_estimated_delivery,avg(t3.freight_value) as freight_value from target.customers as t2
join  time_analysis as t1
on t1.customer_id=t2.customer_id
join target.orderitems as t3
on t1.order_id=t3.order_id
group by t2.customer_state
order by diff_estimated_delivery desc
limit 5
```

Press Alt+F1 for Accessibility Option

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	customer_state	time_to_delivery	diff_estimated_delivery	freight_value
1	AC	20.3296703296703...	20.010989010989018	40.04791208791...
2	RO	19.2820512820512...	19.080586080586084	41.33054945054...
3	AM	25.9631901840490...	18.975460122699381	33.31061349693...
4	AP	27.7530864197530...	17.444444444444443	34.16049382716...
5	RR	27.8260869565217...	17.434782608695652	43.08804347826...

- **Top 5 States where delivery is not so fast**
  i. The top 5 Customer states delivery is not so fast are AL, MA, SE, ES and BA.
  ii. Query Used:

```
with time_analysis as(
select customer_id,order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_times
tamp,day) as time_to_delivery,
timestamp_diff(order_estimated_delivery_date,order_delivered_cust
omer_date,day) as diff_estimated_delivery
from target.orders
where order_purchase_timestamp is not null
  and order_delivered_customer_date is not null
  and order_estimated_delivery_date is not null
)
select t2.customer_state,avg(t1.time_to_delivery) as time_to_deli
very,avg(t1.diff_estimated_delivery) as diff_estimated_delivery,a
vg(t3.freight_value) as freight_value from target.customers as t2
join  time_analysis as t1
on t1.customer_id=t2.customer_id
join target.orderitems as t3
on t1.order_id=t3.order_id
group by t2.customer_state
order by diff_estimated_delivery
limit 5
```

low_estimated_d... 543

RUN SAVE SHARE SCHEDULE MORE Query completed.

```
with time_analysis as(
select customer_id,order_id,
timestamp_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,
timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_estimated_delivery
from target.orders
where order_purchase_timestamp is not null
  and order_delivered_customer_date is not null
  and order_estimated_delivery_date is not null
)
select t2.customer_state,avg(t1.time_to_delivery) as time_to_delivery,avg(t1.diff_estimated_delivery)
as diff_estimated_delivery,avg(t3.freight_value) as freight_value from target.customers as t2
join  time_analysis as t1
on t1.customer_id=t2.customer_id
join target.orderitems as t3
on t1.order_id=t3.order_id
group by t2.customer_state
order by diff_estimated_delivery
limit 5
```

Press Alt+F1 for Accessibility Options

Query results SAVE RESULTS EXPLORE DATA

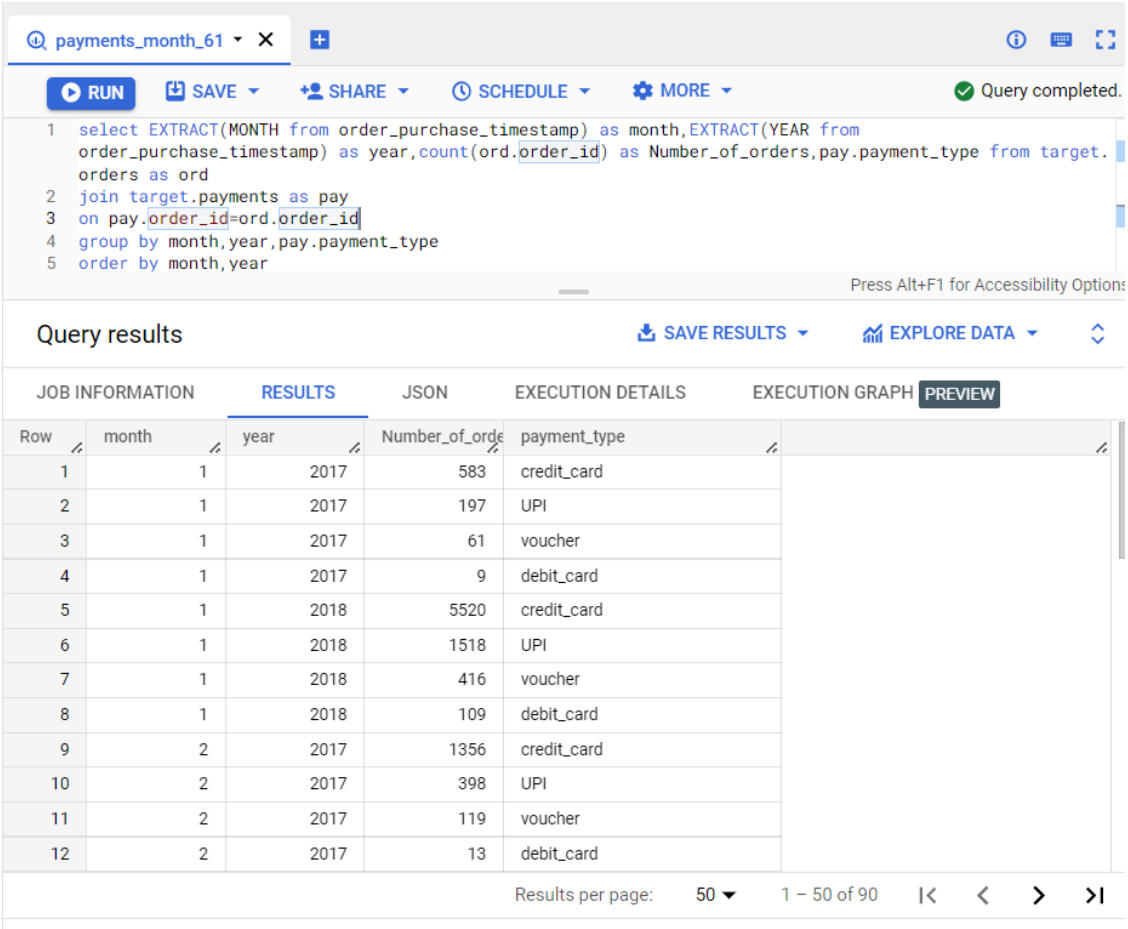
JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	customer_state	time_to_delivery	diff_estimated_c	freight_value
1	AL	23.9929742...	7.97658079...	35.8706557...
2	MA	21.2037500...	9.10999999...	38.4927125...
3	SE	20.9786666...	9.16533333...	36.5731733...
4	ES	15.1928089...	9.76853932...	22.0289797...
5	BA	18.7746402...	10.1194678...	26.4875563...

6. Payment type analysis:

- **Month over Month count of orders for different payment types:**
  i. With the below query, the number of orders happened in every month of every year with different payment types has been calculated.
  ii. Query Used:

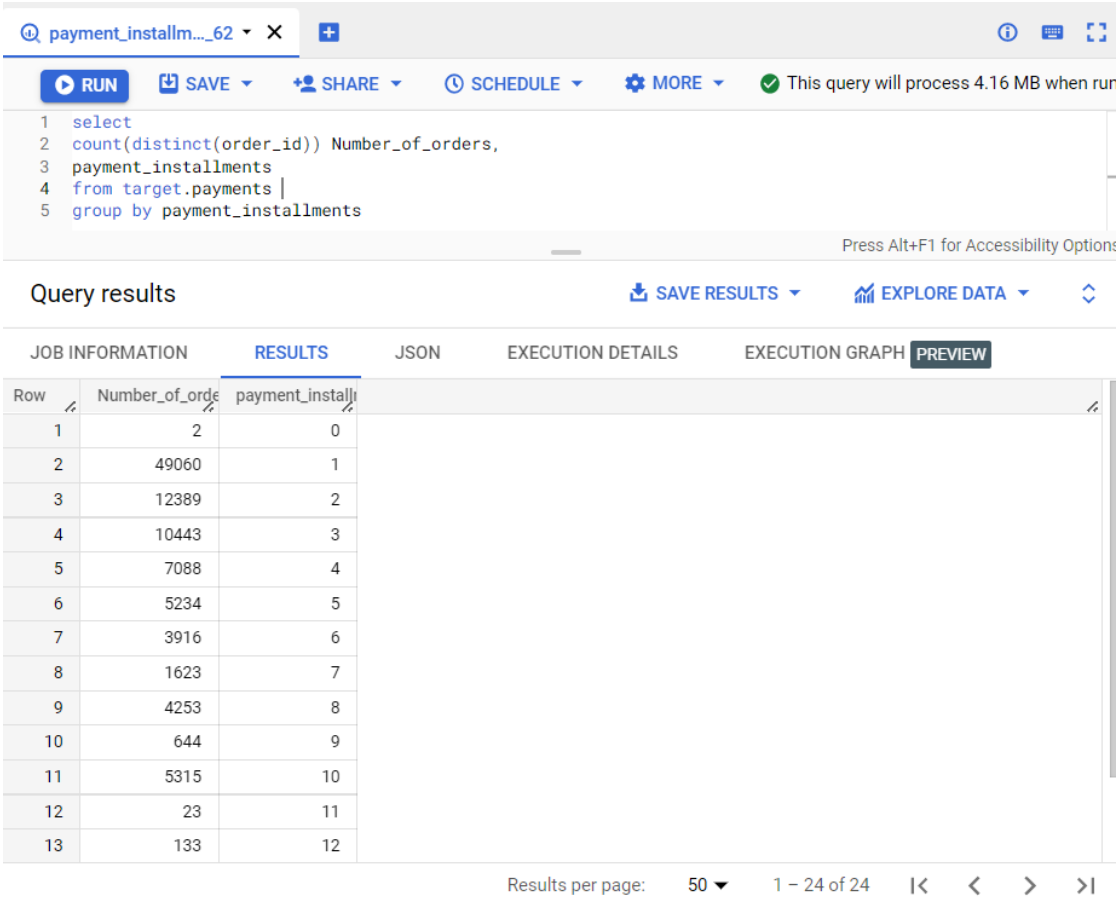
```
select
EXTRACT(MONTH from order_purchase_timestamp) as month,
EXTRACT(YEAR from order_purchase_timestamp) as year,
count(ord.order_id) as Number_of_orders,
pay.payment_type
from target.orders as ord
join target.payments as pay
on pay.order_id=ord.order_id
group by month,year,pay.payment_type
order by month,year
```



Row	month	year	Number_of_orde	payment_type
1	1	2017	583	credit_card
2	1	2017	197	UPI
3	1	2017	61	voucher
4	1	2017	9	debit_card
5	1	2018	5520	credit_card
6	1	2018	1518	UPI
7	1	2018	416	voucher
8	1	2018	109	debit_card
9	2	2017	1356	credit_card
10	2	2017	398	UPI
11	2	2017	119	voucher
12	2	2017	13	debit_card

- **Count of orders based on the no. of payment installments:**
  i. With the below query, the number of orders based on each payment instalments has been calculated .
  ii. Query Used:

```
select
count(distinct(order_id)) Number_of_orders,
payment_installments
from target.payments
group by payment_installments
```



Row	Number_of_orde	payment_installi
1	2	0
2	49060	1
3	12389	2
4	10443	3
5	7088	4
6	5234	5
7	3916	6
8	1623	7
9	4253	8
10	644	9
11	5315	10
12	23	11
13	133	12

7. Actionable Insights:

- **The data provided for the period of two years in approximate [04-09-2016 to 17-10-2018], in these periods some month details are missing and based on the remaining available data the following actionable insights have been found.**
    i. In the year 2018, there is gradual decrease in the number of orders every month, with an average order count of 5401. In the year 2017, there is gradual increase in the number of orders every month, with an average order count of 3758.
    ii. As the number of orders are getting decreased gradually in the year 2018, it would be great if some offers provided to the customers, which might increase the sales, but the average sales is higher than the previous year.
    iii. The 70% of the sales is happening during the Night and Afternoon period, so it's important to make sure that the products are available and in stock during that period. In case of any offers are announcements, it can be done during that time as it will reach to the wider customers.
    iv. The state SP is having more number of orders, where as in few states there has been no orders post 2016, need to increase the sales those states.
    v. The top 5 states with Highest Average freight value are PB, RR, RO, AC and PI.

vi. The top 5 states with Lowest Average freight value are SP, PR, MG, RJ and DF.

vii. The top 5 states with Highest Average Time to Delivery are RR, AP, AM, AL and PA.

viii. The top 5 states with Lowest Average Time to Delivery are SP, PR, MG, DF and SC.

ix. The top 5 states delivery is fast are AC, RO, AM, AP and RR.

x. The top 5 states delivery is not so fast are AL, MA, SE, ES and BA.

8. Recommendations:

- More Offers can we provided to the states and cities where the sales is very less.
- As there is more sales happening during the afternoon and night time, during that time it would be recommended to have more products available in stock.
- In the 11th month of 2017, there was huge number of sales, that method can be followed yearly once to increase the average sales.
- To the states which more time to deliver the products, we can identify the sellers nearby, because of that the time to delivery will become less and the number sales will also increase.
- In few state the number of orders are in single digits, need to advertise in those places more and build the brand in those places for more increased sales.