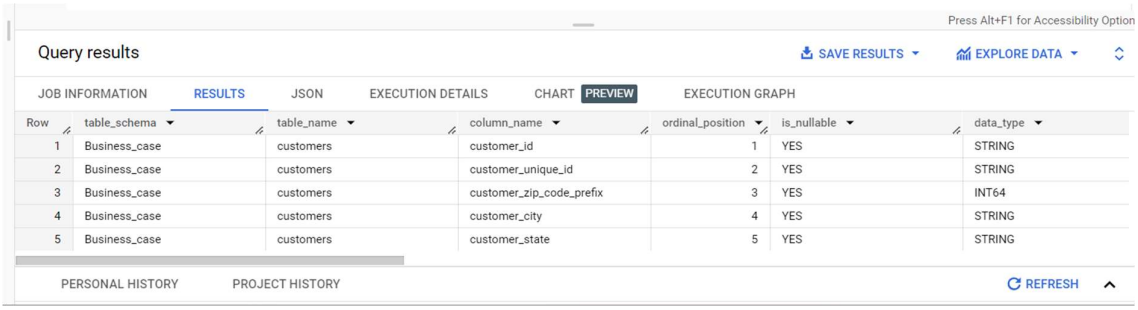
Q(1) **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

(1.1) Data type of all columns in the "customers" table.

Query :

```
SELECT * EXCEPT(table_catalog)
FROM Business_case.INFORMATION_SCHEMA.COLUMNS
where table_name = 'customers';
```

Result :



Press Alt+F1 for Accessibility Option

Query results

SAVE RESULTS EXPLORE DATA

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

Row	table_schema	table_name	column_name	ordinal_position	is_nullable	data_type
1	Business_case	customers	customer_id	1	YES	STRING
2	Business_case	customers	customer_unique_id	2	YES	STRING
3	Business_case	customers	customer_zip_code_prefix	3	YES	INT64
4	Business_case	customers	customer_city	4	YES	STRING
5	Business_case	customers	customer_state	5	YES	STRING

PERSONAL HISTORY PROJECT HISTORY REFRESH

Insights : In this example , table_catelog contains the information about database catalog or name and being excluded in the result set . Then " WHERE" condition applies filter that the restricts result to rows where the 'table_name' column is equal to 'customers'. This means that you're interested in information about columns specifically for 'customers ' table .

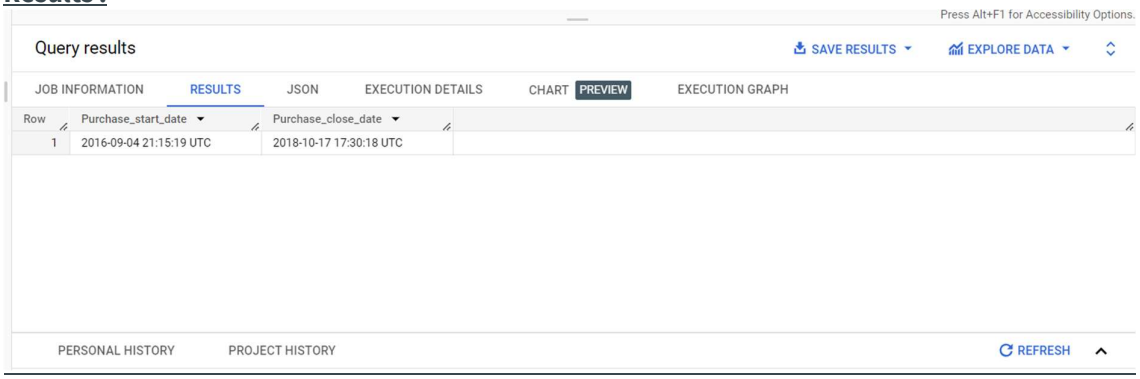
(1.2) Get the time range between which the orders were placed.

Query :

```
select

min (order_purchase_timestamp) as Purchase_start_date ,
max(order_purchase_timestamp) as Purchase_close_date
from `Business_case.orders`;
```

Results :



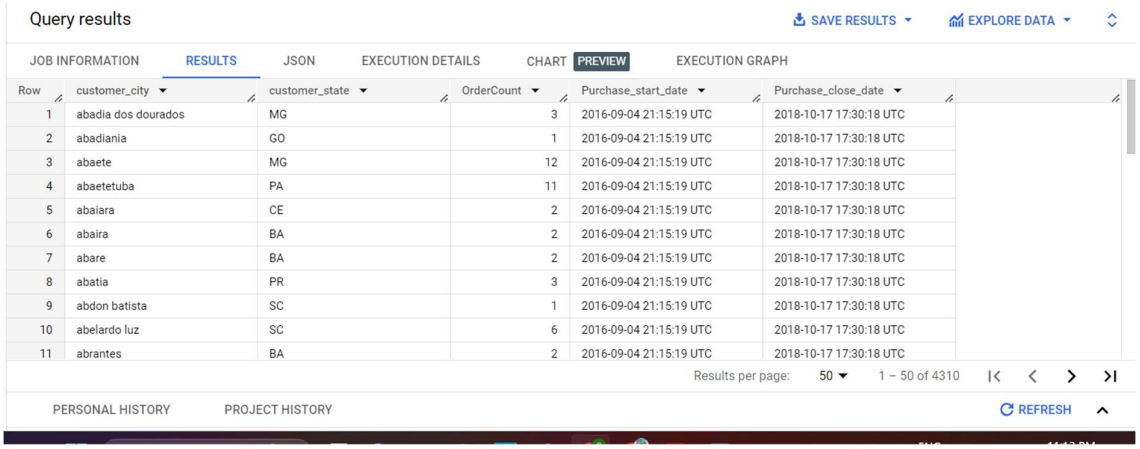
Insights : The results of this query will provide you with insights into the time frame of purchases activities within orders table by taking min() and max() function. Specifically it will give you the start date (earliest purchase) and close date (latest purchase) recorded in dataset.

(1.3) Count the Cities & States of customers who ordered during the given period.

Query :

```
SELECT
    distinct c.customer_city,
    c.customer_state,
    COUNT(o.customer_id) AS OrderCount,
    x.Purchase_start_date,
    x.Purchase_close_date
FROM
    `Business_case.customers` as c
INNER JOIN
    `Business_case.orders` o ON c.customer_id = o.customer_id
INNER JOIN
    (
        SELECT
            MIN(order_purchase_timestamp) AS Purchase_start_date,
            MAX(order_purchase_timestamp) AS Purchase_close_date
        FROM
            `Business_case.orders`
    )as  x
ON
    o.order_purchase_timestamp BETWEEN x.Purchase_start_date AND
x.Purchase_close_date
GROUP BY
    c.customer_city,
    c.customer_state,
    x.Purchase_start_date,
    x.Purchase_close_date
ORDER BY
    c.customer_city;
```

Result:



Query results

SAVE RESULTS EXPLORE DATA

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

Row	customer_city	customer_state	OrderCount	Purchase_start_date	Purchase_close_date
1	abadia dos dourados	MG	3	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC
2	abadiania	GO	1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC
3	abaete	MG	12	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC
4	abaetetuba	PA	11	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC
5	abaiara	CE	2	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC
6	abaira	BA	2	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC
7	abare	BA	2	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC
8	abatia	PR	3	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC
9	abdon batista	SC	1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC
10	abelardo luz	SC	6	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC
11	abrantes	BA	2	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Results per page: 50 1 – 50 of 4310

PERSONAL HISTORY PROJECT HISTORY REFRESH

Insights : The purpose of this query appears to be provide insights into customer behavior based on their location and the time period during which they placed orders . It gives you information about the number of orders made by customers in each city and state combination within specified date range.

Q(2) **In-depth Exploration:**

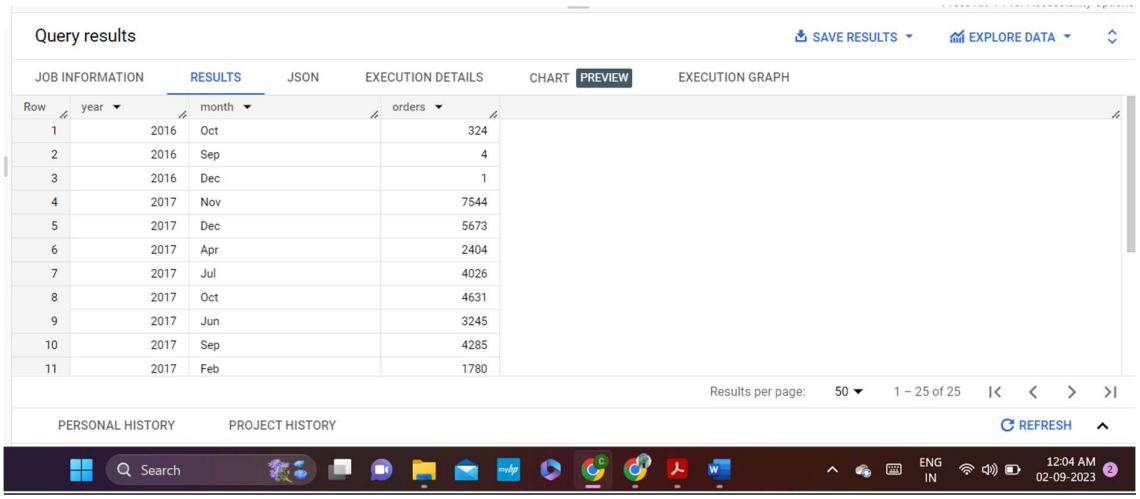
(2.1)Is there a growing trend in the no. of orders placed over the past years?

Query :

```
select

    x.year,
    x.month,
    count(x.Orderss) as orders
from
    (
        select
            order_id as Orderss,
            extract(YEAR FROM order_purchase_timestamp) as year,
            format_date('%b', order_purchase_timestamp) as month
        from
            `Business_case.orders`
    ) AS x
group by
    x.year ,
    x.month
order by
    x.year;
```

Result :



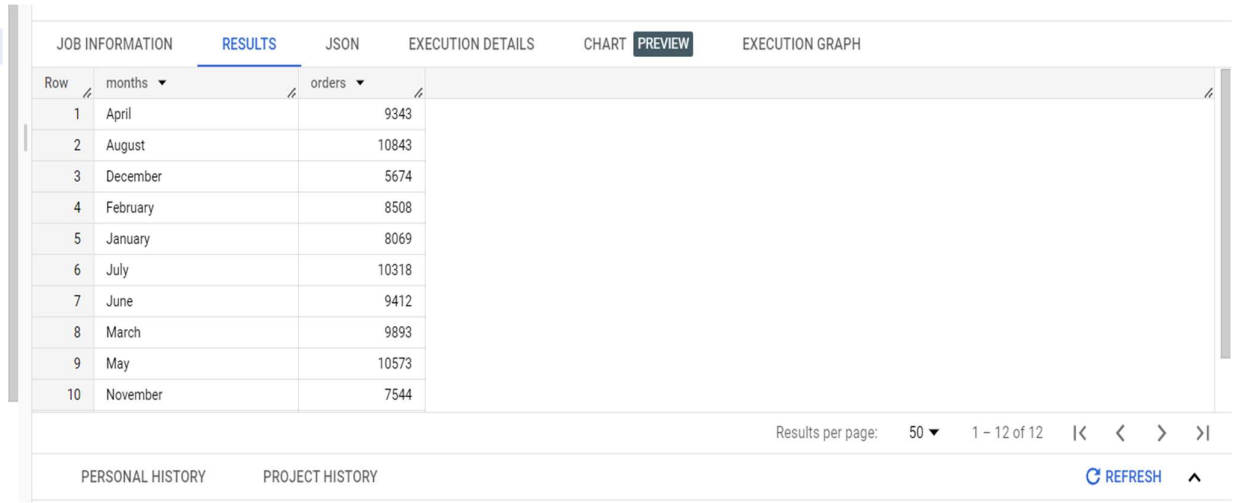
Insights : The result shows that the number of orders made in each month of each year , allowing for past analysis of order trends . from November 2017 till May 2018 the orders received are above the avg orders in given time

(2.2)Can we see some kind of monthly seasonality in terms of the no. of orders being placed ?

Query :

```
select
    x.months,
    count(x.Orderss) as orders
from
    (
        select
            order_id as Orderss,
            extract(YEAR FROM order_purchase_timestamp) as year,
            format_date('%B', order_purchase_timestamp) as months
        from
            `Business_case.orders`
    ) AS x
group by
    x.months
order by
    x.months;
```

Results :



Insights : The result shows that the number of orders in each month . This is showing that the orders starts decreasing from September to December .

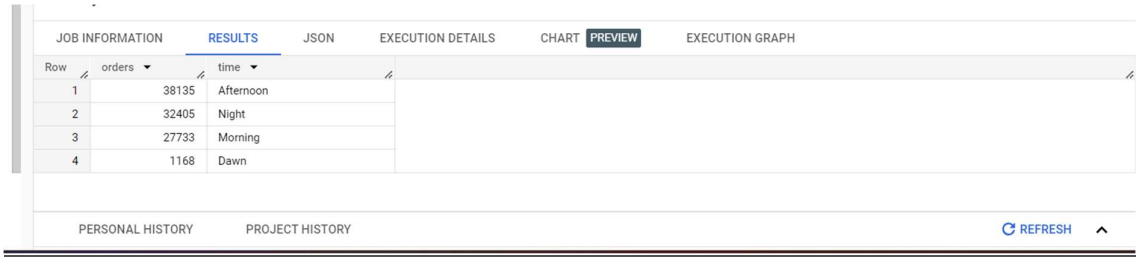
(2.3) During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

- 0-6 hrs : Dawn
- 7-12 hrs : Mornings
- 13-18 hrs : Afternoon
- 19-23 hrs : Night

Query:

```
with cte as
(
 select
 order_id,
 order_purchase_timestamp,
 extract(hour from order_purchase_timestamp) as hour
 from `Business_case.orders`
),
get_time as
(
    select
    order_id,
    cte.hour,
    case
        when hour>=3 and hour<=6
        then ‘Dawn’
        when hour>6 and hour<=12
        then ‘Morning’
        when hour>=12 and hour<=18
        then ‘Afternoon’
        else ‘Night’
        end as Time
    from cte
)
select
count(order_id) as orders,
time
from get_time
group by time
order by orders desc;
```

Results :



Insights : In this example , we can say that the Brazilians mostly placed their orders in Afternoon from 12 pm to 6 pm and after that they also prefer the night time to place their orders.

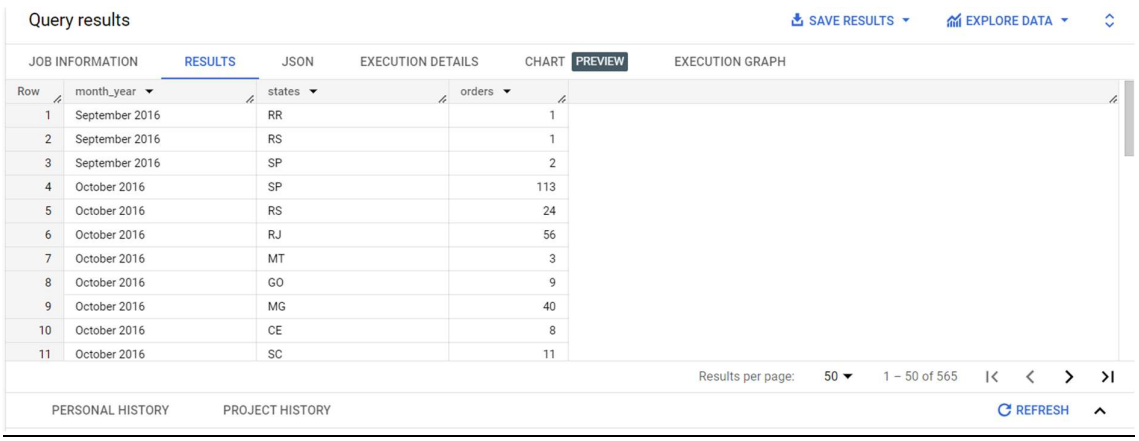
Q3 . Evolution of E-commerce orders in the Brazil region:

(3.1) Get the month on month no. of orders placed in each state.

Query :

```
with state_months as (
select
o.order_id,
extract(YEAR from order_purchase_timestamp) as year,
extract(MONTH from order_purchase_timestamp) as month_number,
format_date('%B',o.order_purchase_timestamp) AS months,
c.customer_state as states
from `Business_case.orders` as o join `Business_case.customers` as c
on o.customer_id = c.customer_id )
select
concat(months,' ',year) as month_year,
states,
count(order_id) orders
from state_months
group by states, months, year, month_number
order by year, month_number;
```

Result :



Insights : The report show us that , number of orders made in different states , categorized by month and year .

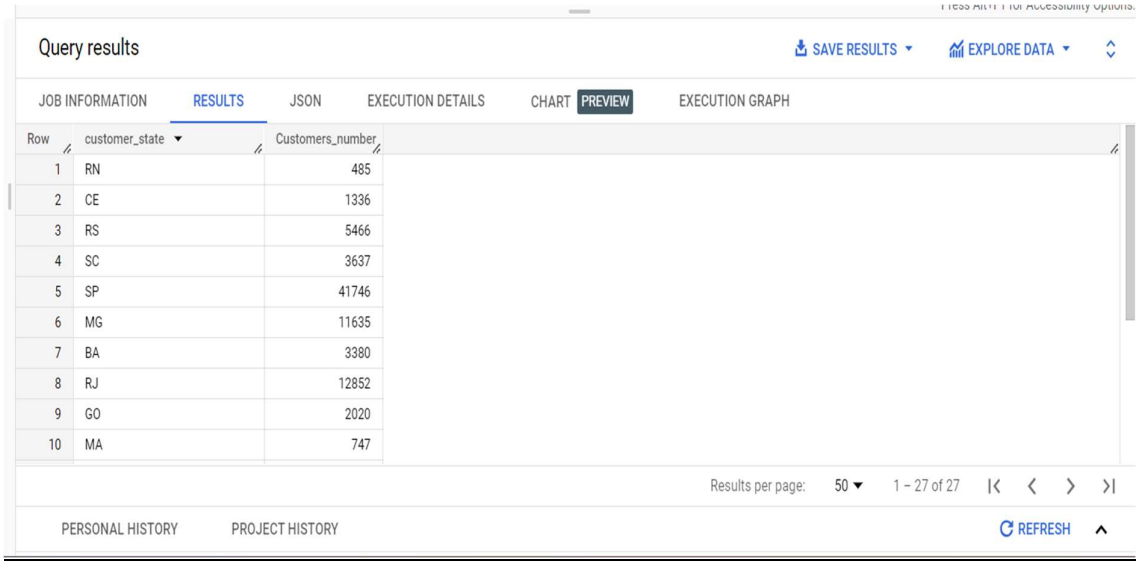
Each row represents that specific state, specific month and specific year along with the count of orders placed during that month .

(3.2) How are the customers distributed across all the states?

Query :

```
select
customer_state,
count(customer_id) as Customers_number
from `Business_case.customers`
group by customer_state ;
```

Results :



Insights : The output shows that the geographic distribution of your customers . In ' SP ' state the highest customers are there with number of 41746 customers and followed by the state 'RG' and 'MG'.

Q4. Impact on Economy: Analyse the money movement by e-commerce by looking at order prices, freight and others.

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

Query :

```
with MonthlyData as (
    select
        format_date('%B', order_purchase_timestamp) as month,
        extract(MONTH FROM order_purchase_timestamp) as month_number,
        extract(YEAR FROM order_purchase_timestamp) as year,
        count(o.order_id) as no_of_orders,
        round(SUM(payment_value), 2) as total_payment
    from
        `Business_case.orders` as o inner join `Business_case.payments` as p
        on o.order_id = p.order_id
    where
        extract(YEAR from order_purchase_timestamp) in (2017, 2018) and
        extract(MONTH from order_purchase_timestamp) BETWEEN 1 AND 8
    group by month, month_number, year
),
MonthlyComparison as (
    select
        m1.month,
        m1.month_number,
        m1.no_of_orders as order_2017,
        m2.no_of_orders as order_2018,
        m1.total_payment as payment_2017,
        m2.total_payment as payment_2018
    from
        MonthlyData as m1 left join MonthlyData as m2
        on m1.month_number = m2.month_number and m1.year = 2017 and m2.year = 2018
)
select
    month,
    payment_2017,
    payment_2018,
    ROUND(((order_2018 - order_2017) / order_2017) * 100, 2) as
percent_increase_order,
    ROUND(((payment_2018 - payment_2017) / payment_2017) * 100, 2) as
percent_increase_payment
from
    MonthlyComparison
where
    payment_2018 is not null
order by
    month_number;
```

Result :

Query results

SAVE RESULTS EXPLORE DATA

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

Row	month	payment_2017	payment_2018	percent_increase_orc	percent_increase_pa
1	January	138488.04	1115004.18	789.76	705.13
2	February	291908.01	992463.34	268.61	239.99
3	March	449863.6	1159652.12	164.79	157.78
4	April	417788.03	1160785.48	180.4	177.84
5	May	592918.82	1153982.15	80.91	94.63
6	June	511276.38	1023880.5	86.82	100.26
7	July	592382.92	1066540.75	50.73	80.04
8	August	674396.32	1022425.32	47.21	51.61

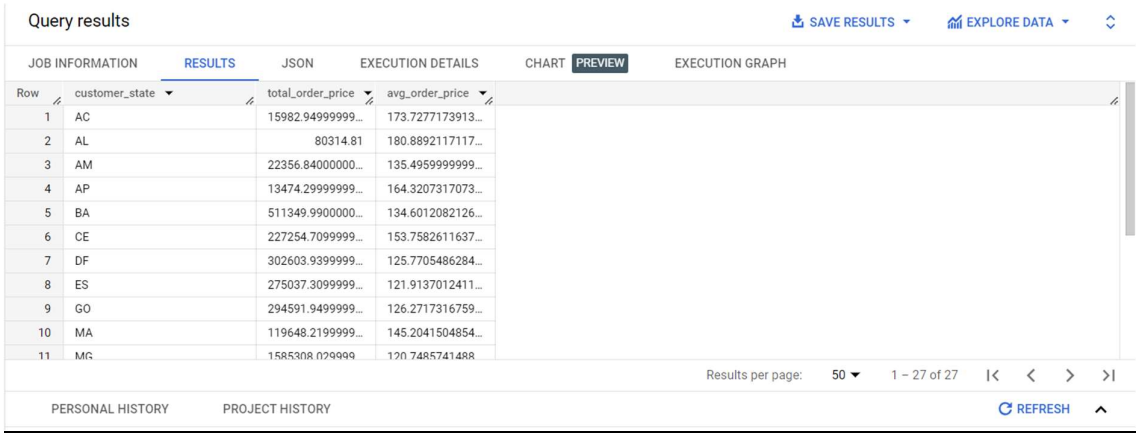
Insights : 1. We can say that there is increase in the % increase in the cost of orders from Jan to March and then there is sudden fall till August .
2. We can say that this fall is not because of any particular product as % increase in orders as more or less same as the % increase in the cost of product.

(4.2) Calculate the Total & Average value of order price for each state.

Query :

```
select
    c.customer_state,
    sum(o2.price) as total_order_price,
    avg(o2.price) as avg_order_price
from `Business_case.order_items` as o2 inner join `Business_case.orders` as o1
on o2.order_id = o1.order_id
inner join `Business_case.customers` as c
on o1.customer_id = c.customer_id
group by c.customer_state  ;
```

Results :



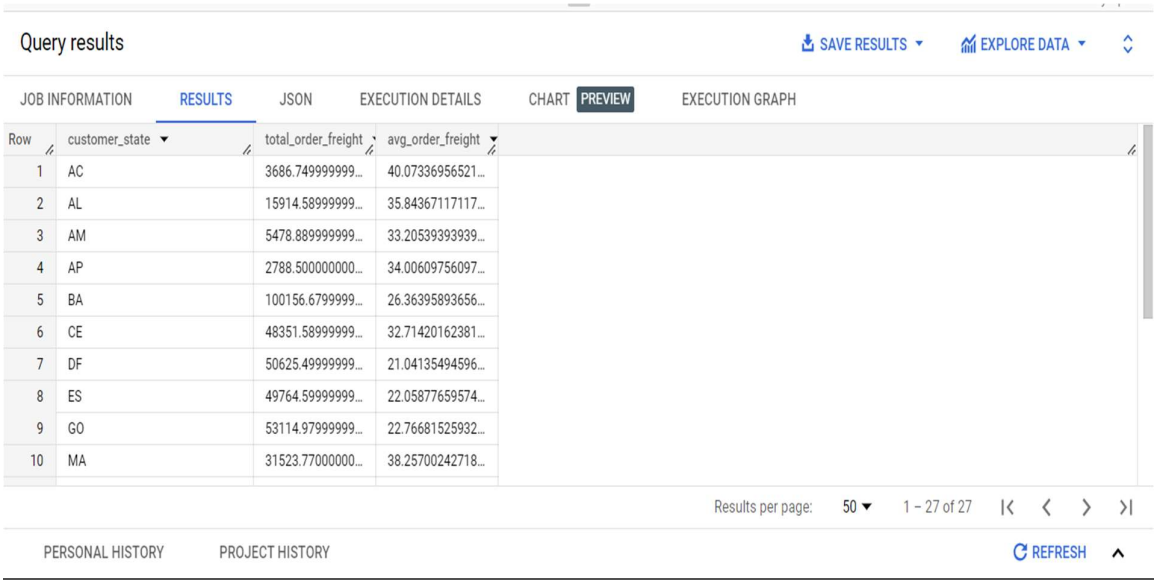
Insights : The output of this query provide you the insights on spending behavior of customers in different states with respect to orders price .

(4.3) Calculate the Total & Average value of order freight for each state.

Query :

```
select
    c.customer_state,
    sum(o1.freight_value) as total_order_freight,
    avg(o1.freight_value) as avg_order_freight
from `Business_case.order_items` as o1 inner join `Business_case.orders` as o2
on o2.order_id = o1.order_id
inner join `Business_case.customers` as c
on o2.customer_id = c.customer_id
group by c.customer_state
order by c.customer_state ;
```

Results :



Insights : The result of this query shows you the insights on total and average of orders freights. The state ‘AC’ has the highest total and average value as compared to others.

Q5.Analysis based on sales, freight and delivery time.

(5.1) Find the no. of days taken to deliver each order from the order's purchase date as delivery time.
Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
Do this in a single query.

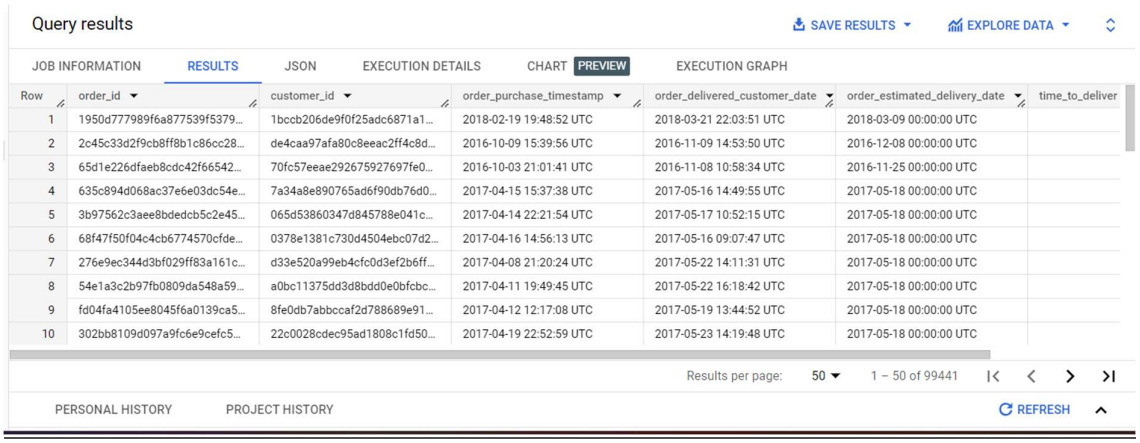
You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- **time_to_deliver** = order_delivered_customer_date - order_purchase_timestamp
- **diff_estimated_delivery** = order_estimated_delivery_date - order_delivered_customer_date

Query :

```
select
order_id,
customer_id,
order_purchase_timestamp,
order_delivered_customer_date,
order_estimated_delivery_date,
date_diff(order_delivered_customer_date, order_purchase_timestamp, day) as
time_to_deliver,
date_diff(order_estimated_delivery_date, order_delivered_customer_date, day) as
diff_estimated_delivery
from `Business_case.orders'
```

Result :


Query results

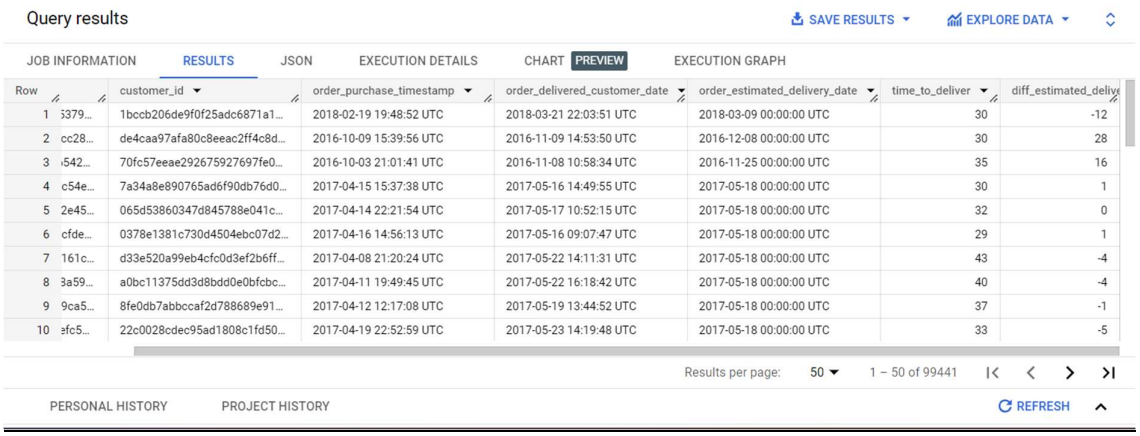
SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS CHART PREVIEW EXECUTION GRAPH

Row	order_id	customer_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_deliver
1	1950d777989f6a877539f5379...	1bccb206de9f0f25adc6871a1...	2018-02-19 19:48:52 UTC	2018-03-21 22:03:51 UTC	2018-03-09 00:00:00 UTC	
2	2c45c33d2f9cb8ff8b1c86cc28...	de4caa97afa80c8eeac2ff4c8d...	2016-10-09 15:39:56 UTC	2016-11-09 14:53:50 UTC	2016-12-08 00:00:00 UTC	
3	65d1e226dfaeb8cdc42f66542...	70fc57eeae292675927697fe0...	2016-10-03 21:01:41 UTC	2016-11-08 10:58:34 UTC	2016-11-25 00:00:00 UTC	
4	635c894d068ac37e6e03dc54e...	7a34a8e890765ad6f90db76d0...	2017-04-15 15:37:38 UTC	2017-05-16 14:49:55 UTC	2017-05-18 00:00:00 UTC	
5	3b97562c3aee8bdedcb5c2e45...	065d53860347d845788e041c...	2017-04-14 22:21:54 UTC	2017-05-17 10:52:15 UTC	2017-05-18 00:00:00 UTC	
6	68f47f50f04c4cb6774570cfde...	0378e1381c730d4504ebc07d2...	2017-04-16 14:56:13 UTC	2017-05-16 09:07:47 UTC	2017-05-18 00:00:00 UTC	
7	276e9ec344d3bf029ff83a161c...	d33e520a99eb4cfc0d3ef2b6ff...	2017-04-08 21:20:24 UTC	2017-05-22 14:11:31 UTC	2017-05-18 00:00:00 UTC	
8	54e1a3c2b97fb0809da548a59...	a0bc11375dd3d8bdd0e0bfcbc...	2017-04-11 19:49:45 UTC	2017-05-22 16:18:42 UTC	2017-05-18 00:00:00 UTC	
9	fd04fa4105ee8045f6a0139ca5...	8fe0db7abbccaf2d788689e91...	2017-04-12 12:17:08 UTC	2017-05-19 13:44:52 UTC	2017-05-18 00:00:00 UTC	
10	302bb8109d097a9fc6e9cefc5...	22c0028cdec95ad1808c1fd50...	2017-04-19 22:52:59 UTC	2017-05-23 14:19:48 UTC	2017-05-18 00:00:00 UTC	

Results per page: 50 1 – 50 of 99441

PERSONAL HISTORY PROJECT HISTORY REFRESH



Query results

SAVE RESULTS EXPLORE DATA

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

Row	customer_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_deliver	diff_estimated_deliv
1	3379...	1bccb206de9f0f25adc6871a1...	2018-02-19 19:48:52 UTC	2018-03-21 22:03:51 UTC	2018-03-09 00:00:00 UTC	30	-12
2	cc28...	de4caa97afa80c8eeac2ff4c8d...	2016-10-09 15:39:56 UTC	2016-11-09 14:53:50 UTC	2016-12-08 00:00:00 UTC	30	28
3	1542...	70fc57eeae292675927697fe0...	2016-10-03 21:01:41 UTC	2016-11-08 10:58:34 UTC	2016-11-25 00:00:00 UTC	35	16
4	c54e...	7a34a8e890765ad6f90db76d0...	2017-04-15 15:37:38 UTC	2017-05-16 14:49:55 UTC	2017-05-18 00:00:00 UTC	30	1
5	2e45...	065d53860347d845788e041c...	2017-04-14 22:21:54 UTC	2017-05-17 10:52:15 UTC	2017-05-18 00:00:00 UTC	32	0
6	cfde...	0378e1381c730d4504ebc07d2...	2017-04-16 14:56:13 UTC	2017-05-16 09:07:47 UTC	2017-05-18 00:00:00 UTC	29	1
7	161c...	d33e520a99eb4cfc0d3ef2b6ff...	2017-04-08 21:20:24 UTC	2017-05-22 14:11:31 UTC	2017-05-18 00:00:00 UTC	43	-4
8	3a59...	a0bc11375dd3d8bdd0e0bfcbc...	2017-04-11 19:49:45 UTC	2017-05-22 16:18:42 UTC	2017-05-18 00:00:00 UTC	40	-4
9	9ca5...	8fe0db7abbccaf2d788689e91...	2017-04-12 12:17:08 UTC	2017-05-19 13:44:52 UTC	2017-05-18 00:00:00 UTC	37	-1
10	efc5...	22c0028cdec95ad1808c1fd50...	2017-04-19 22:52:59 UTC	2017-05-23 14:19:48 UTC	2017-05-18 00:00:00 UTC	33	-5

Results per page: 50 1 – 50 of 99441

PERSONAL HISTORY PROJECT HISTORY REFRESH

Insights : The output of this query shows that, the delivery time and difference between actual delivery date and estimated delivery date .

(5.2) Find out the top 5 states with the highest & lowest average freight value.

Query :

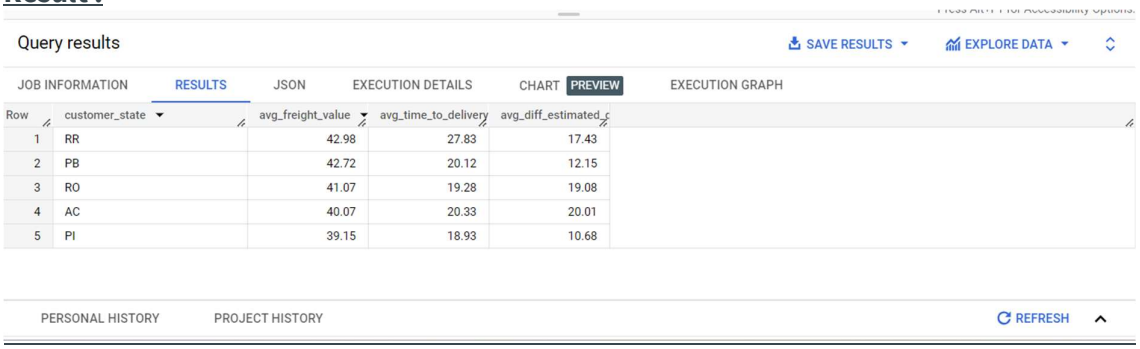
For the highest –

```
Select

c.customer_state,

round(avg(o1.freight_value),2) as avg_freight_value,
round(avg(date_diff(order_delivered_customer_date, order_purchase_timestamp,
day)),2) as avg_time_to_delivery,
round(avg(date_diff(order_estimated_delivery_date, order_delivered_customer_date,
day)),2) as avg_diff_estimated_delivery
from `Business_case.orders` as o inner join `Business_case.customers` as c
on o.customer_id = c.customer_id
inner join `Business_case.order_items` o1
on o.order_id = o1.order_id
group by c.customer_state
order by round(avg(o1.freight_value),2) desc
limit 5;
```

Result :



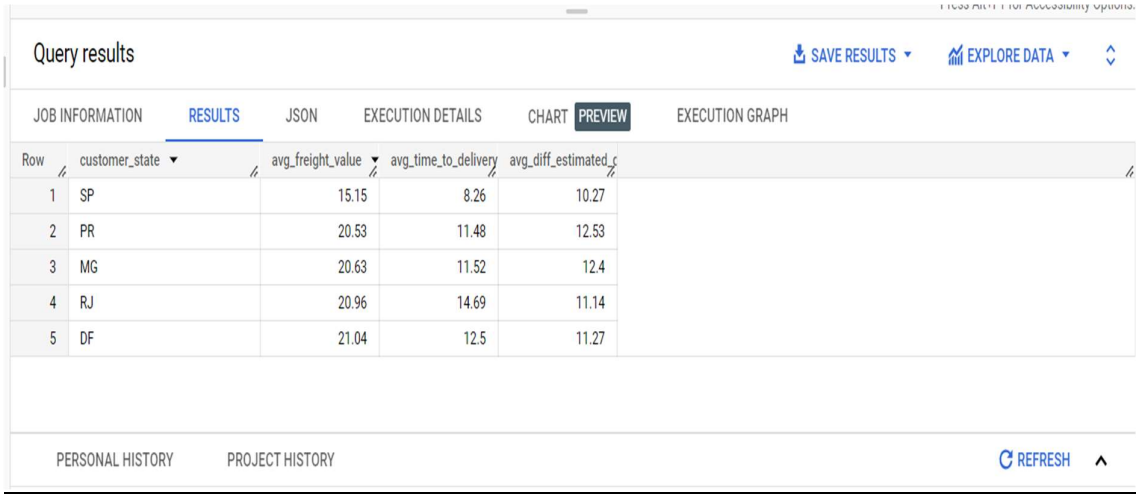
Row	customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_
1	RR	42.98	27.83	17.43
2	PB	42.72	20.12	12.15
3	RO	41.07	19.28	19.08
4	AC	40.07	20.33	20.01
5	PI	39.15	18.93	10.68

For the lowest -

Query :

```
select
c.customer_state,
round(avg(o1.freight_value),2) avg_freight_value ,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp
,DAY)),2) as avg_time_to_delivery,
round(avg(date_diff(order_estimated_delivery_date
,order_delivered_customer_date,DAY)),2) as avg_diff_estimated_delivery
from `Business_case.orders` as o join `Business_case.customers` as c
on o.customer_id = c.customer_id
join `Business_case.order_items` as o1
on o.order_id = o1.order_id
Group by c.customer_state
Order by round(avg(o1.freight_value),2)
 limit 5 ;
```

Result :



Row	customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_
1	SP	15.15	8.26	10.27
2	PR	20.53	11.48	12.53
3	MG	20.63	11.52	12.4
4	RJ	20.96	14.69	11.14
5	DF	21.04	12.5	11.27

Insights : The two results shows us that the top 5 states with highest and lowest average freight value.

As ‘RR’ has the highest average freight value and ‘SP’ has lowest average freight value.

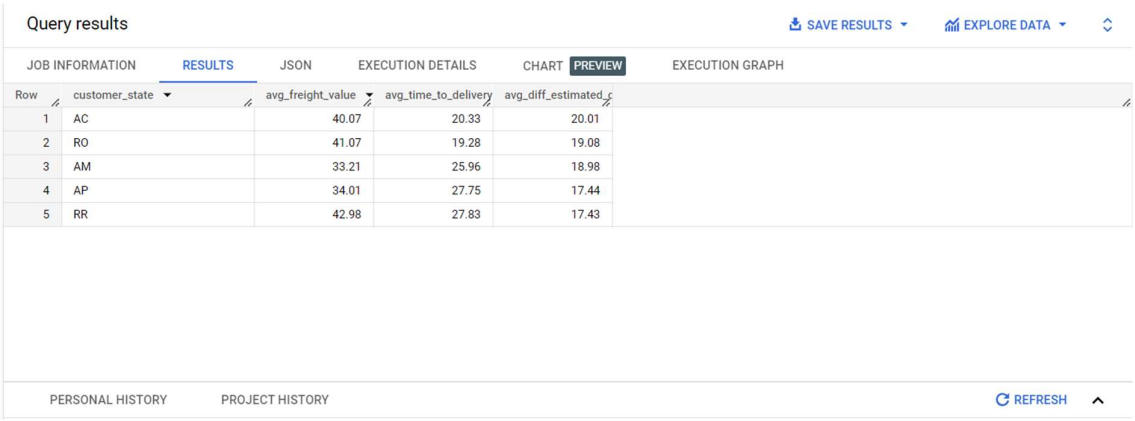
(5.3) Find out the top 5 states with the highest & lowest average delivery time.

For the highest –

Query :

```
select
c.customer_state,
round(avg(oi.freight_value),2) avg_freight_value ,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp
,DAY)),2) as avg_time_to_delivery,
round(avg(date_diff(order_estimated_delivery_date
,order_delivered_customer_date,DAY)),2) as avg_diff_estimated_delivery
from `Business_case.orders` as o join `Business_case.customers` as c
on o.customer_id = c.customer_id
join `Business_case.order_items` as oi
on o.order_id = oi.order_id
Group by c.customer_state
Order by round(avg(date_diff(order_estimated_delivery_date,
order_delivered_customer_date,DAY)),2) desc
limit 5;
```

Result :



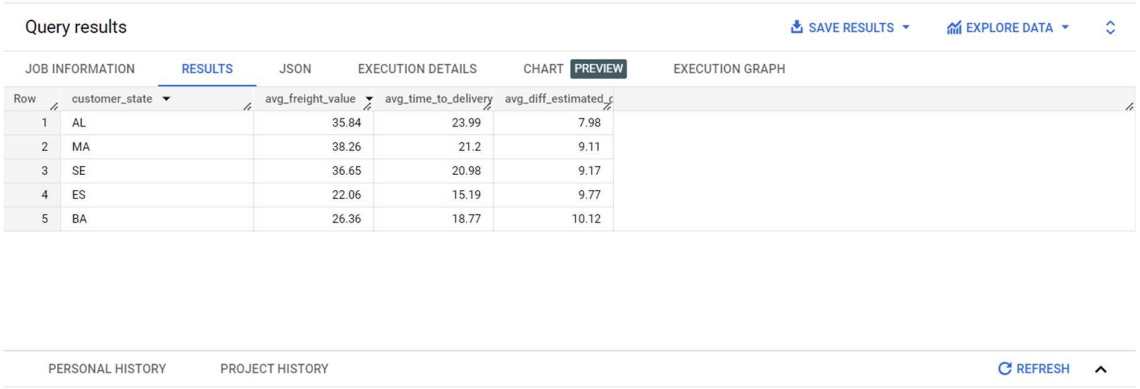
Row	customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_c
1	AC	40.07	20.33	20.01
2	RO	41.07	19.28	19.08
3	AM	33.21	25.96	18.98
4	AP	34.01	27.75	17.44
5	RR	42.98	27.83	17.43

For the lowest –

Query :

```
select
c.customer_state,
round(avg(oi.freight_value),2) avg_freight_value ,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp
,day)),2) as  avg_time_to_delivery,
round(avg(date_diff(order_estimated_delivery_date
,order_delivered_customer_date,day)),2) as avg_diff_estimated_delivery
from `Business_case.orders` as o join `Business_case.customers` as c
on o.customer_id = c.customer_id
join `Business_case.order_items` as oi
on o.order_id = oi.order_id
Group by c.customer_state
Order by round(avg(date_diff(order_estimated_delivery_date,
order_delivered_customer_date,DAY)),2)
limit 5;
```

Result :



Row	customer_state	avg_freight_value	avg_time_to_delivery	avg_diff_estimated_c
1	AL	35.84	23.99	7.98
2	MA	38.26	21.2	9.11
3	SE	36.65	20.98	9.17
4	ES	22.06	15.19	9.77
5	BA	26.36	18.77	10.12

Insights : The two results shows us that the top 5 states with highest and lowest average of delivery time .

The AC state has the highest average delivery time .

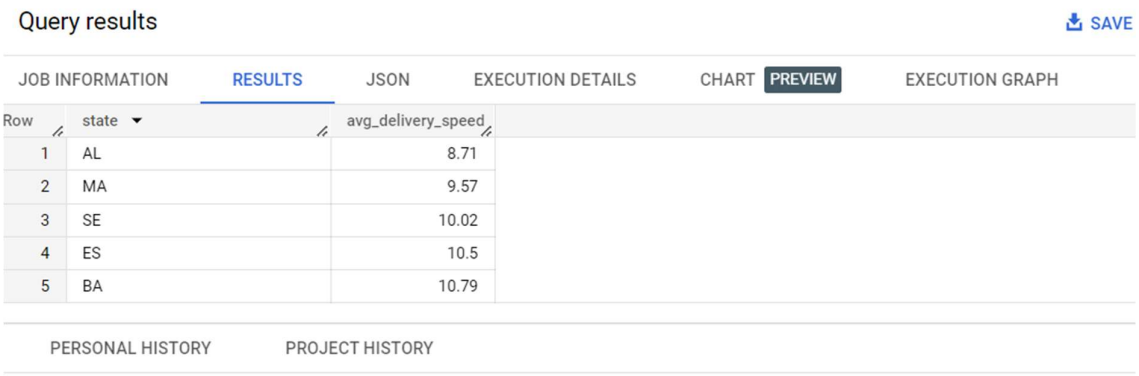
The AL state has the lowest average delivery time.

(5.4) Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

Query :

```
with std_delivery_speed as (
  select c.customer_state as state,
  round(avg(date_diff(date(o.order_estimated_delivery_date),date(o.order_del
ivered_customer_date),day)),2)as avg_delivery_speed
  from `Business_case.orders` as o inner join `Business_case.customers` as c
  on o.customer_id = c.customer_id
  group by state)
  select std_delivery_speed.state,std_delivery_speed.avg_delivery_speed from
std_delivery_speed
  order by std_delivery_speed.avg_delivery_speed
  limit 5 ;
```

Results :



Row	state	avg_delivery_speed
1	AL	8.71
2	MA	9.57
3	SE	10.02
4	ES	10.5
5	BA	10.79

Insights :

By looking at the data for Average Estimate date diff for top 5 and last 5 states we can say that where the delivery is really fast top 5 accounts also have the more or less date average delivery time , to which we can conclude that the estimated time was non stringently define which is why we are getting good figures

Q(6) Analysis based on the payments:

(6.1) Find the month on month no. of orders placed using different payment types.

Query :

```
select
a.Year_Month,
a.payment_type,
count(a.no_of_orders) as no_of_orders
from
(
select
extract(year from o.order_purchase_timestamp ) as year,
extract (month from o.order_purchase_timestamp ) as month_no,
concat (format_date('%B',o.order_purchase_timestamp)," ",
extract (year from o.order_purchase_timestamp)) as year_month,
p.payment_type,
o.order_id as no_of_orders
from `Business_case.orders` as o inner join `Business_case.payments` as p
on o.order_id = p.order_id
) as a
group by a.Year_Month, a.payment_type, a.year , a.month_no
order by a.year,a.month_no;
```

Result :

Query results

SAVE RESULTS EXPLORE DATA

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

Row	Year_Month	payment_type	no_of_orders
1	September 2016	credit_card	3
2	October 2016	credit_card	254
3	October 2016	UPI	63
4	October 2016	voucher	23
5	October 2016	debit_card	2
6	December 2016	credit_card	1
7	January 2017	credit_card	583
8	January 2017	UPI	197
9	January 2017	voucher	61
10	January 2017	debit_card	9

Results per page: 50 1 – 50 of 90

PERSONAL HISTORY PROJECT HISTORY REFRESH

Insights :

1.By looking at the above data we can clearly say that the people in Brazil are more like to use Credit card as their payment option and the 2nd payment option they like is UPI .

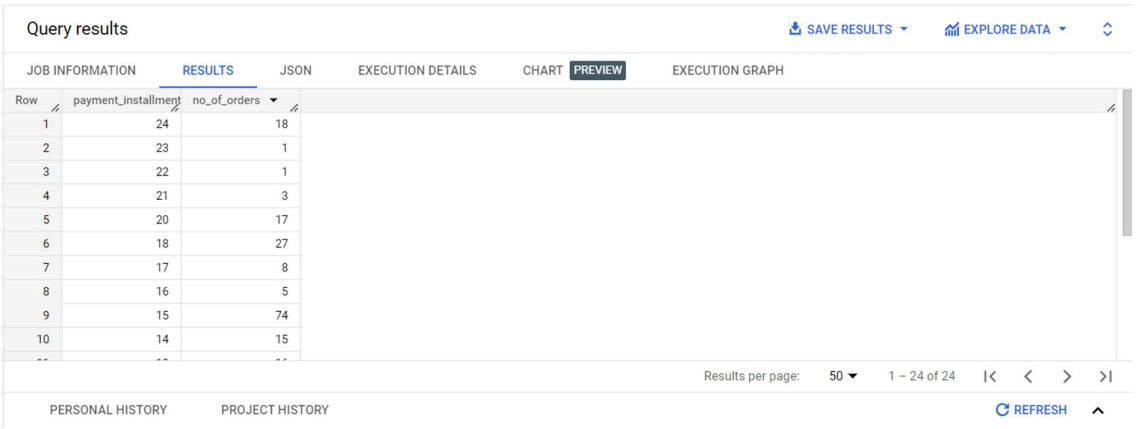
2. The transactions through the vouchers are more less same throughput the timeline.

(6.2) Find the no. of orders placed on the basis of the payment installments that have been paid.

Query :

```
select
p.payment_installments,
count(o.order_id) as no_of_orders
FROM `Business_case.orders` as o inner join `Business_case.payments` as p
on o.order_id = p.order_id
Group by p.payment_installments
order by p.payment_installments desc;
```

Results :



Insights : By looking at the above data we can say that 80% of the payments comes to the company in first 5 instalments in which 50% of total payments is single one shot payment.

Recommendations for Target Corp. :-

1. Looking at the given data if we compare the Total Number of orders per state we can say by applying Pareto's rule the 80 % of the orders are coming from nearly 7 states out of 28 states, So the Target Corporation can start with these 7 states initially with full fledge capacity and with all the product which are SP, RJ, MG ,RS , PR ,SC. Look at the table below to get the numbers

States	Sum of orders	Total Orders	% of total order
SP	41746	99441	41.98
RJ	12852	99441	12.9
MG	11635	99441	11.70
RS	5466	99441	5.49
PR	5045	99441	5.07
SC	3637	99441	3.65

2. As most of the orders are coming from these 5 States, they should have a good logistics system implemented in those to reduce the Avg. time to deliver the product as this will increase the customer satisfaction.

3. When we check the data for Average price vs Total Price we came to know the Avg. price is less in all the states where the total price is at top . From this we can say that the ticket price is less and the number of orders are big , the suggestion will be they should not keep less hight price product to not block the more money into that or they should be focusing on the average price product while doing marketing so increase the sales .
4. When we look at the data for no of orders per payment type we saw that people from Brazil likes to use the credit card, so for any festive season they can market for the offers on the credit card to increase the sales .
5. As we can see that the Top 5 or 7 States with big success we need to plan the manpower to run the business , as we cannot afford the loss of sales due to the less manpower in any department .