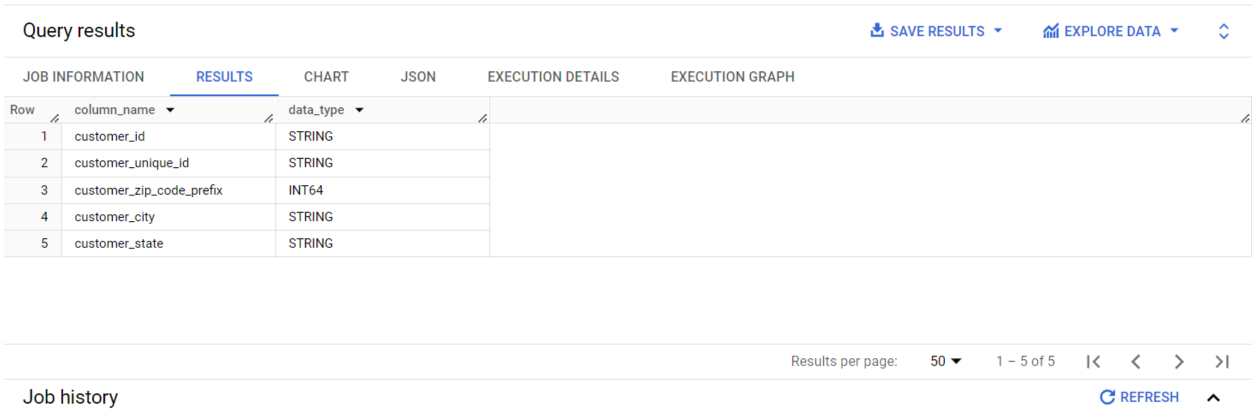
Q1.1 Data type of all columns in the "customers" table.

Answer

1st Approach

```
Select column_name,data_type
From SQL_Project_Target.INFORMATION_SCHEMA.COLUMNS
Where table_name = 'customers'
```



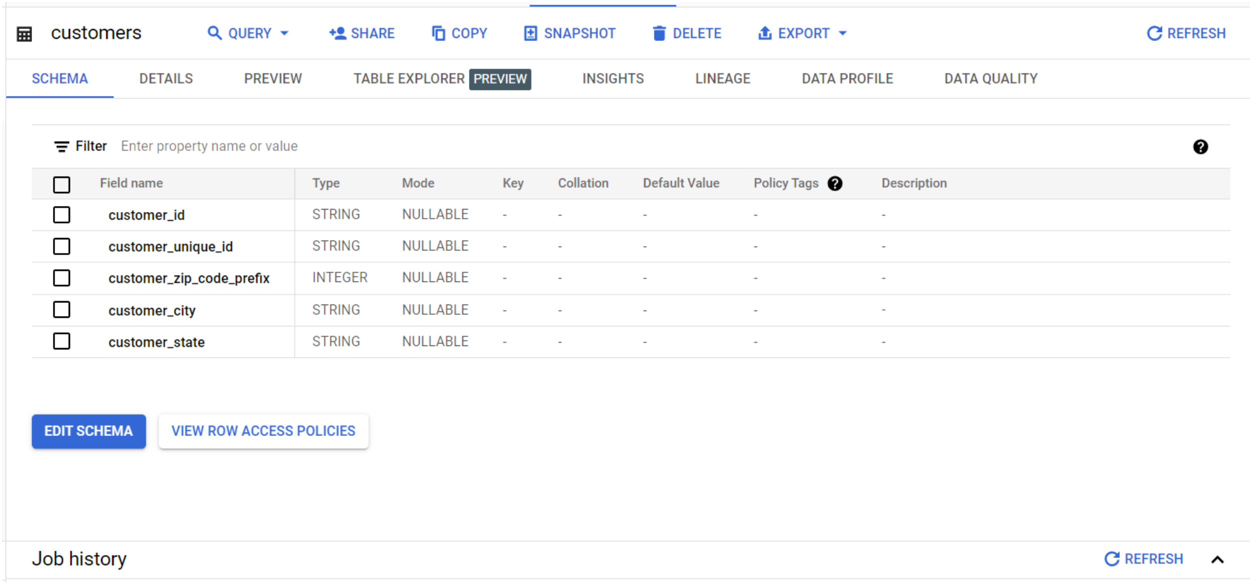
Query results

Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

Results per page: 50 1 – 5 of 5

Job history

2nd Approach



customers

Field name	Type	Mode	Key	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	-	-	-	-	-

Job history

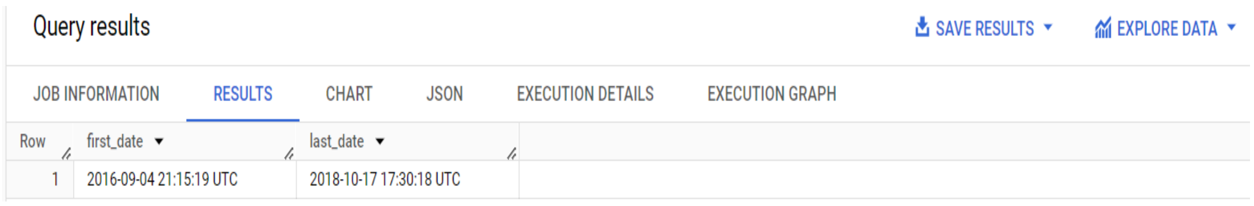
Data insights – we can see the datatypes for each column of the table here.

Recommendation – we must be aware of the datatypes of each column to perform the queries on these columns.

Q1.2 Get the time range between which the orders were placed

Answer

```
select min(order_purchase_timestamp) as first_date,
max(order_purchase_timestamp) as last_date
from `SQL_Project_Target.orders`
```



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

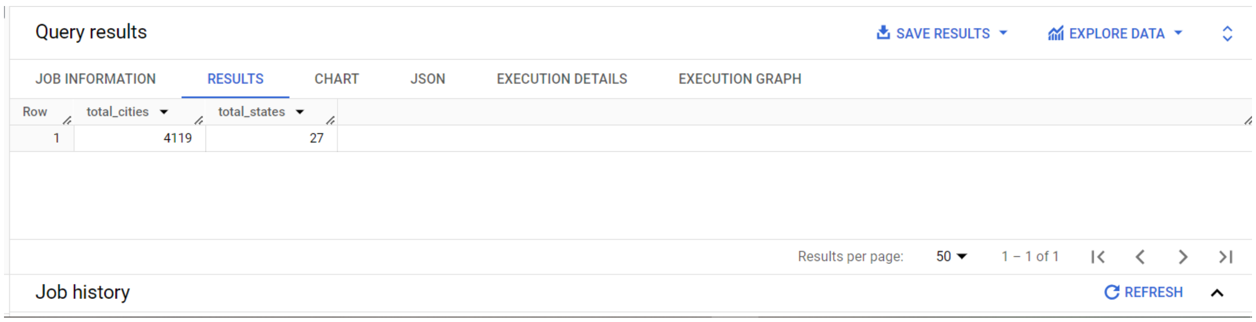
Data insights – we get the first date and last date of the orders where we have to perform the operations.

Recommendation – we always analyze the data of a particular timeframe so we must be aware of start date and end date

Q1.3 Count the Cities & States of customers who ordered during the given period.

Answer

```
select count(distinct customer_city) as total_cities, count(distinct customer_state)
as total_states from `SQL_Project_Target.customers` c join
`SQL_Project_Target.orders` o on c.customer_id=o.customer_id
where order_purchase_timestamp between '2016-09-04' and '2018-10-17'
```



Row	total_cities	total_states
1	4119	27

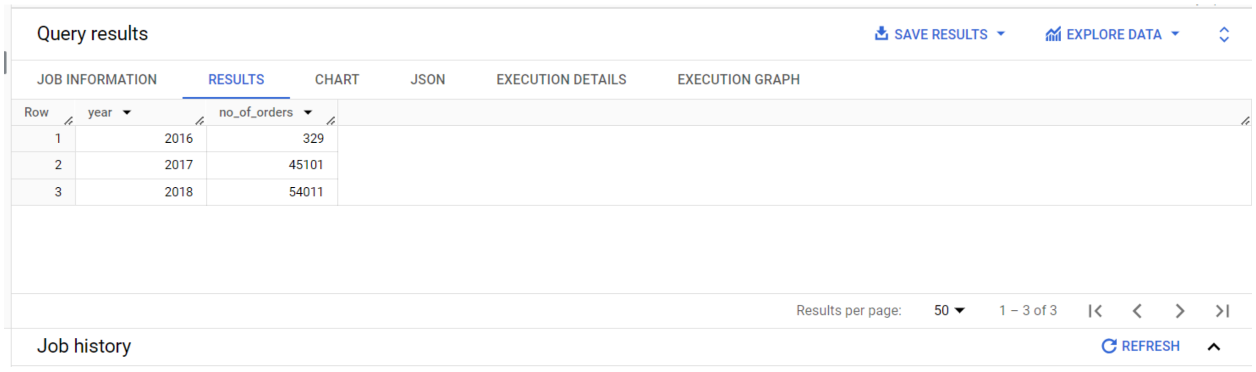
Data insights – we get the total cities and states where the orders have been placed

Recommendation – we always analyze the data of a particular city and state we must be aware of this information.

Q2.1 Is there a growing trend in the no. of orders placed over the past years?

Answer

```
select extract(year from order_purchase_timestamp) as year, count(distinct
order_id) as no_of_orders
from `SQL_Project_Target.orders`
group by 1
order by 1
```



Row	year	no_of_orders
1	2016	329
2	2017	45101
3	2018	54011

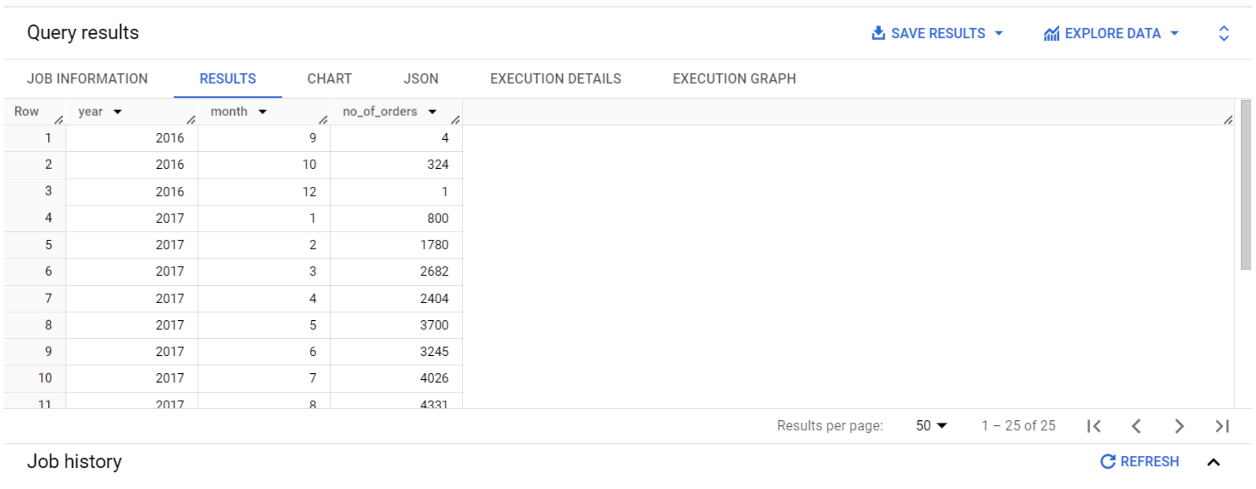
Data insights – There is growing trend here and number are significantly increasing over the years. Orders are less in 2016 because it is just a last few months data

Recommendation – Number of orders are getting increased over the past years so it's is a positive sign but still we need to check all the parameters where we can work to increase the productivity.

Q2.2 Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Answer

```
select extract(year from order_purchase_timestamp) as year, extract(month from
order_purchase_timestamp) as month, count(distinct order_id) as no_of_orders
from `SQL_Project_Target.orders`
group by 1,2
order by 1,2
```



Query results

Row	year	month	no_of_orders
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

Results per page: 50 1 – 25 of 25

Job history

Data insights – we started the operation in Sept 2016 so we can consider the data of 2017 and 2018. we can look at the data then it is significantly increasing and also company is having good number of orders during the festive seasons like Diwali, Thanksgiving and New Year while lesser sale in December

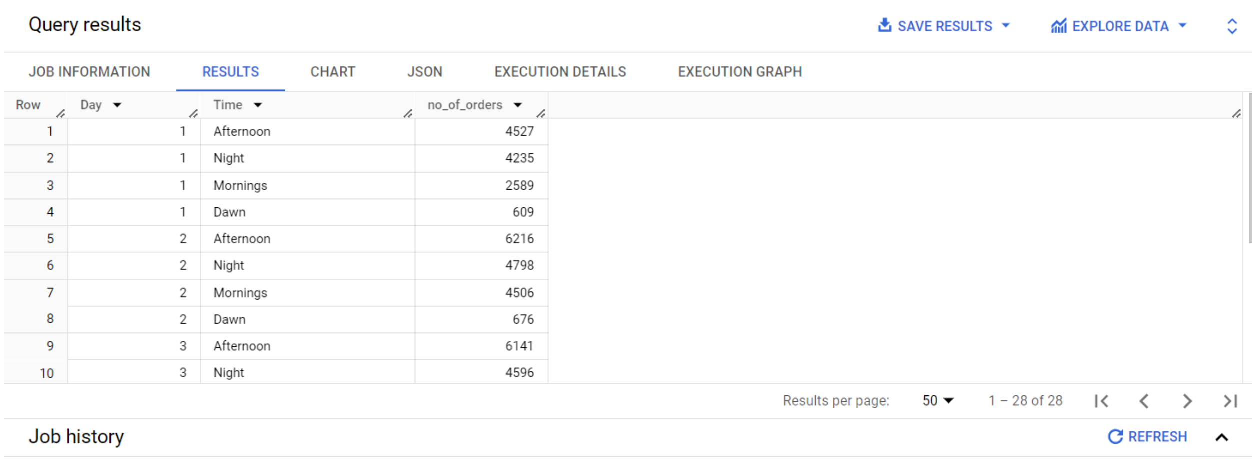
Recommendation – we have low sale in December because people always plan to buy something in the New Year instead of last month of the year. We should try to provide some offers which should be valid till 31st Dec so it can attract the buyers in the month of December.

Q2.3 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

Answer

```
select extract(dayofweek from order_purchase_timestamp) as Day,
(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
'Mornings'
   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,
count(order_id) as no_of_orders
from `SQL_Project_Target.orders`
group by 1,2
order by 1,3 desc
```



Row	Day	Time	no_of_orders
1	1	Afternoon	4527
2	1	Night	4235
3	1	Mornings	2589
4	1	Dawn	609
5	2	Afternoon	6216
6	2	Night	4798
7	2	Mornings	4506
8	2	Dawn	676
9	3	Afternoon	6141
10	3	Night	4596

Insight - Brazilian customers mostly place their orders in Afternoon and then Night.

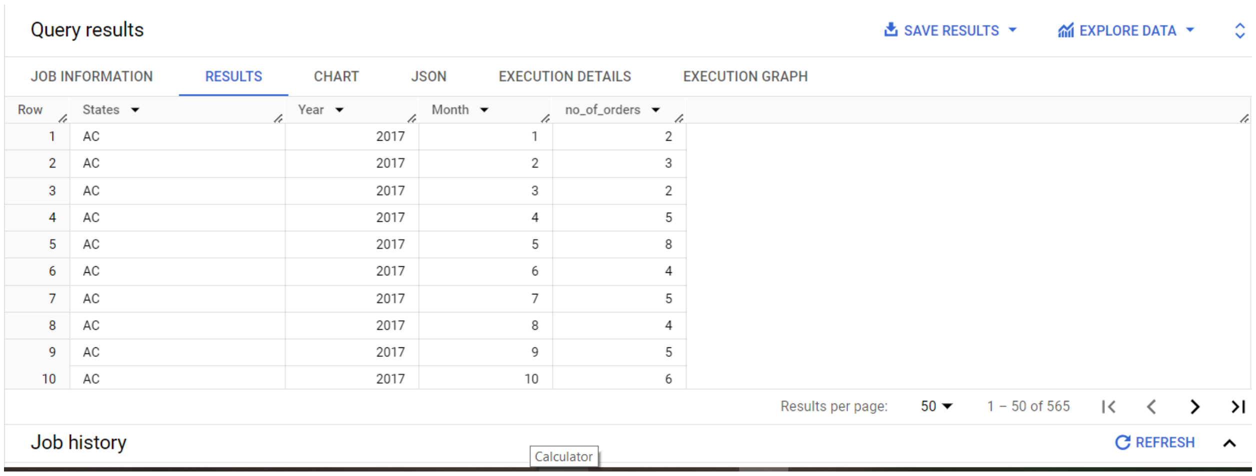
Recommendations – As per the data, we have less order in Dawn time so we should open the store early morning and try to recommend some offers that time and check if the orders are increasing in Dawn and Morning time.

Q3.1 Get the month on month no. of orders placed in each state

Answer

Approach 1

```
select c.customer_state as States,
extract(year from o.order_purchase_timestamp) as Year, extract(month from
o.order_purchase_timestamp) as Month,
count(o.order_id) as no_of_orders
from `SQL_Project_Target.orders` o join `SQL_Project_Target.customers` c
on o.customer_id=c.customer_id
group by 1,2,3
order by 1,2,3
```


Query results

Row	States	Year	Month	no_of_orders
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6

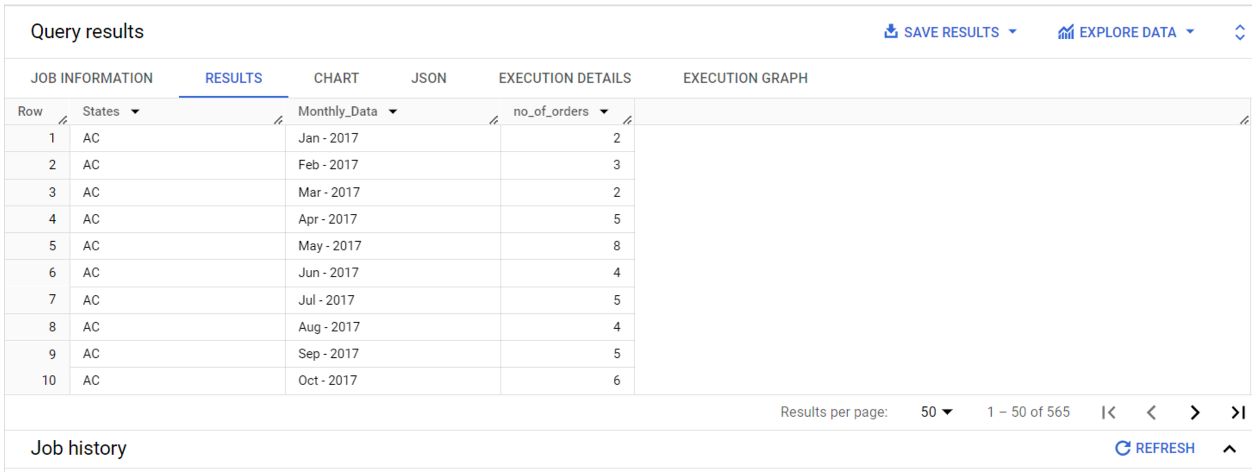
Results per page: 50   1 – 50 of 565

Job history

Approach 2

```
with cte as
(select c.customer_state as States,
format_date("%b - %Y", order_purchase_timestamp) as Monthly_Data,
extract(year from o.order_purchase_timestamp) as Year, extract(month from
o.order_purchase_timestamp) as Month,
count(o.order_id) as no_of_orders
from `SQL_Project_Target.orders` o join `SQL_Project_Target.customers` c
on o.customer_id=c.customer_id
group by 1,2,3,4
order by states,Year,Month
)

select States,Monthly_Data,no_of_orders
from cte
```


Query results

Row	States	Monthly_Data	no_of_orders
1	AC	Jan - 2017	2
2	AC	Feb - 2017	3
3	AC	Mar - 2017	2
4	AC	Apr - 2017	5
5	AC	May - 2017	8
6	AC	Jun - 2017	4
7	AC	Jul - 2017	5
8	AC	Aug - 2017	4
9	AC	Sep - 2017	5
10	AC	Oct - 2017	6

Results per page: 50 1 – 50 of 565

Job history

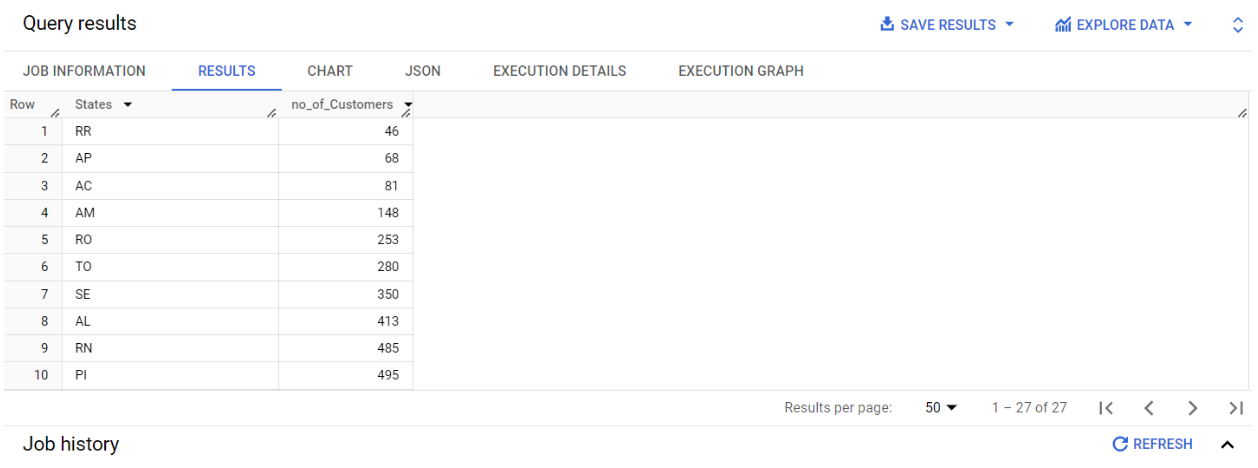
Insight – Here we have few states where we have maximum number of orders like SP/ SC/ RS/ RJ/ MG/ES/BA on month on month and few states with minimum number of orders like RR/AP/AM/AC

Recommendation- We have to analyze the strategies what we are following in states where we have highest number of orders and try to apply those strategies there where we have minimum number of orders

Q3.2 How are the customers distributed across all the states?

Answer

```
select customer_state as States,count(distinct customer_id) as no_of_Customers
from `SQL_Project_Target.customers`
group by 1
order by 2
```


Query results

Row	States	no_of_Customers
1	RR	46
2	AP	68
3	AC	81
4	AM	148
5	RO	253
6	TO	280
7	SE	350
8	AL	413
9	RN	485
10	PI	495

Results per page: 50 1 – 27 of 27

Job history

Insight – we are able to see the number of customers in every states.

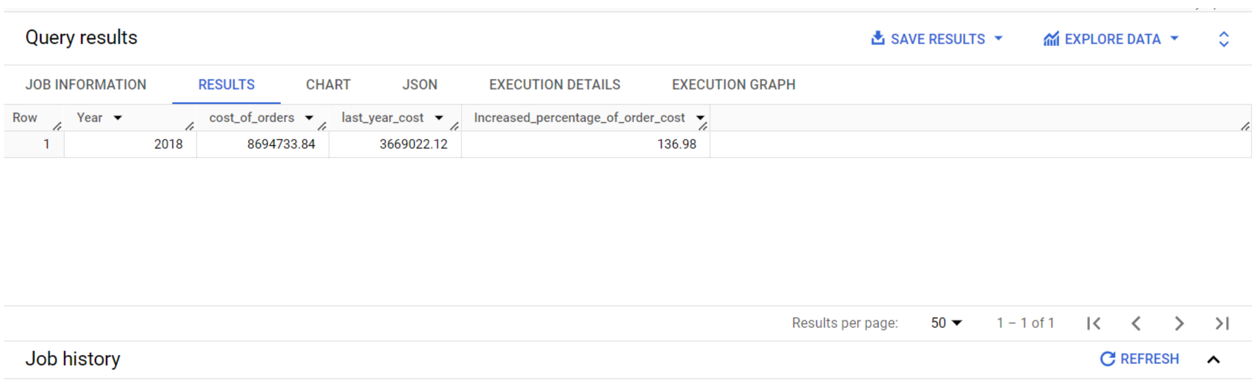
Recommendations – there are few states like RR/AP/AC where we have very less customers (below 100) so we should put our extra efforts like open more stores, we can work on rates by providing attractive offers and quality of the product. so that we can attract people from these states.

Q4.1 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.

Approach 1

```
with cte as
(select extract(year from order_purchase_timestamp) as Year,
round(sum(payment_value),2) as
cost_of_orders,lag(round(sum(payment_value),2)) over(order by
round(sum(payment_value),2)) as last_year_cost
from `SQL_Project_Target.payments` p join `SQL_Project_Target.orders` o
on p.order_id=o.order_id
where extract(month from order_purchase_timestamp) between 1 and 8
group by 1
order by 1)

select *, round((cost_of_orders-last_year_cost)/last_year_cost*100,2) as
Increased_percentage_of_order_cost from cte
where last_year_cost is not null
```



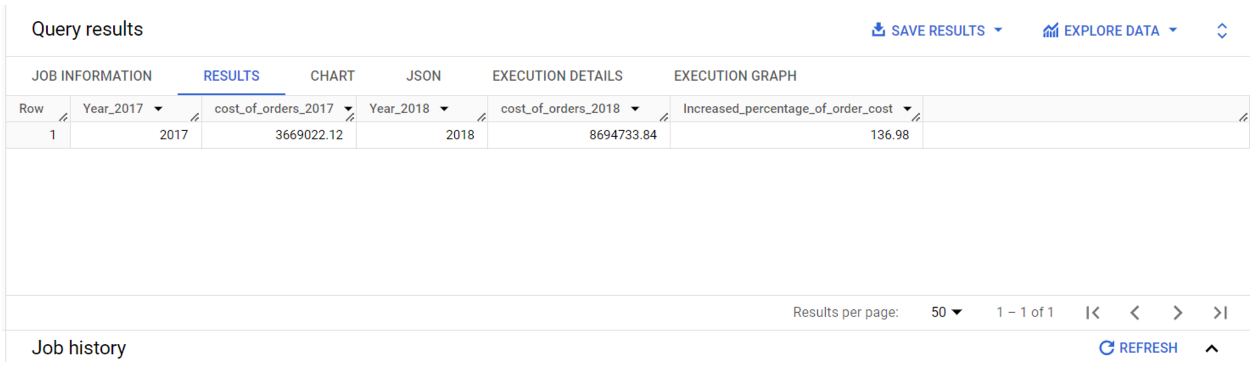
Row	Year	cost_of_orders	last_year_cost	Increased_percentage_of_order_cost
1	2018	8694733.84	3669022.12	136.98

Approach 2

```
with cte1 as
(select extract(year from order_purchase_timestamp) as Year_2017,
round(sum(payment_value),2) as cost_of_orders_2017
from `SQL_Project_Target.payments` p join `SQL_Project_Target.orders` o
on p.order_id=o.order_id
where extract(month from order_purchase_timestamp) between 1 and 8 and
extract(year from order_purchase_timestamp) = 2017
group by 1
order by 1),

cte2 as
(select extract(year from order_purchase_timestamp) as Year_2018,
round(sum(payment_value),2) as cost_of_orders_2018
from `SQL_Project_Target.payments` p join `SQL_Project_Target.orders` o
on p.order_id=o.order_id
where extract(month from order_purchase_timestamp) between 1 and 8 and
extract(year from order_purchase_timestamp) = 2018
group by 1
order by 1)
```

```
select *, round((cost_of_orders_2018-
cost_of_orders_2017)/cost_of_orders_2017*100,2) as
Increased_percentage_of_order_cost from cte1,cte2
where cost_of_orders_2017 is not null
```



Row	Year_2017	cost_of_orders_2017	Year_2018	cost_of_orders_2018	Increased_percentage_of_order_cost
1	2017	3669022.12	2018	8694733.84	136.98

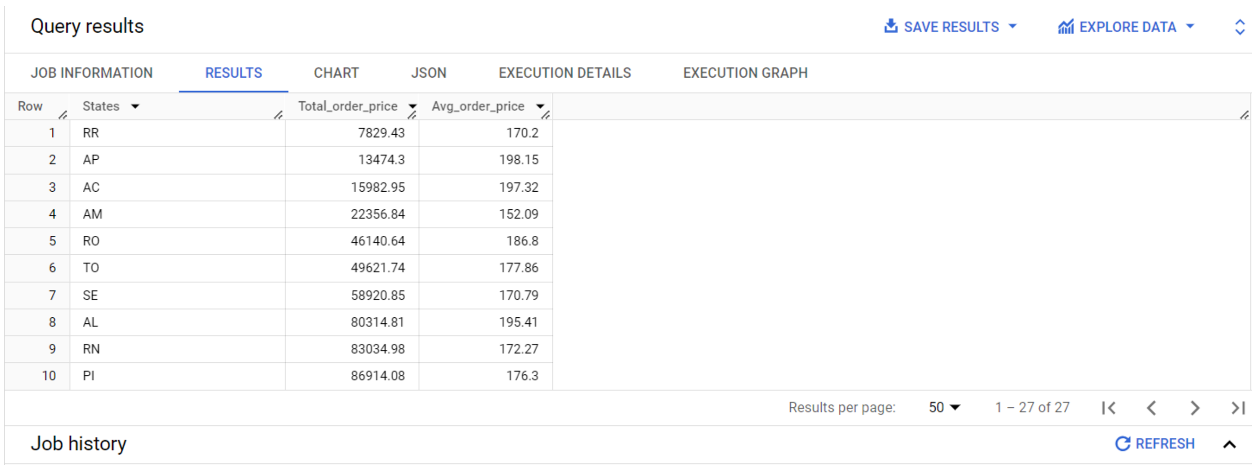
Insight – There is an increment in the cost of orders so we have 136.98% increments.

Recommendations – we can keep putting efforts in the upcoming years so that we can have significant increment for the next years.

Q4.2 Calculate the Total & Average value of order price for each state

Answer

```
select c.customer_state as States,round(sum(i.price),2) as
Total_order_price,round(sum(i.price)/count(distinct i.order_id),2) as
Avg_order_price
from `SQL_Project_Target.customers` c join `SQL_Project_Target.orders` o
on c.customer_id=o.customer_id
join `SQL_Project_Target.order_items` i
on i.order_id=o.order_id
group by 1
order by 2,3
```



Row	States	Total_order_price	Avg_order_price
1	RR	7829.43	170.2
2	AP	13474.3	198.15
3	AC	15982.95	197.32
4	AM	22356.84	152.09
5	RO	46140.64	186.8
6	TO	49621.74	177.86
7	SE	58920.85	170.79
8	AL	80314.81	195.41
9	RN	83034.98	172.27
10	PI	86914.08	176.3

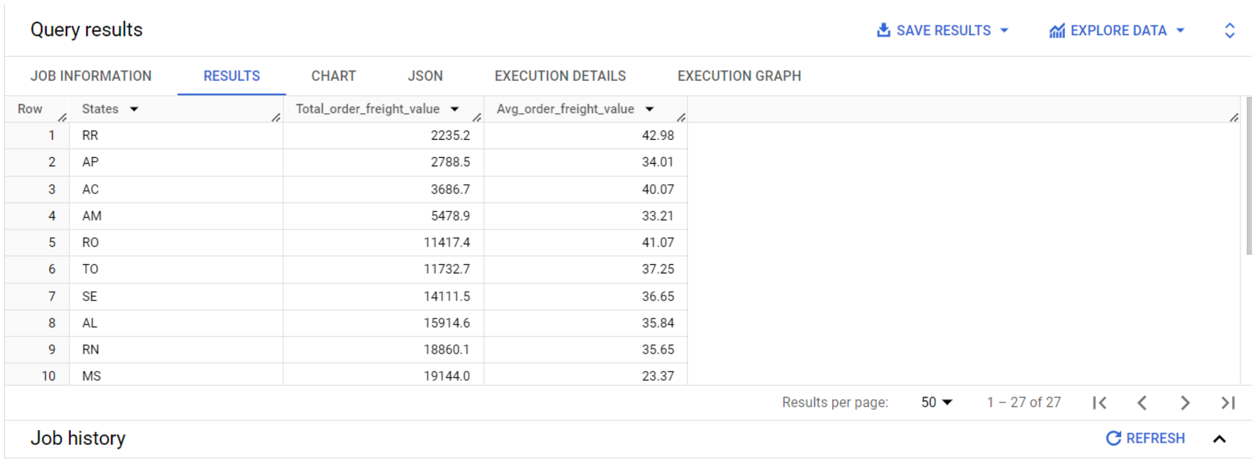
Insight – we have the total order price and the average order price for each and every state which will be helpful to make strategies for every state accordingly. AC/AP/RR are the states where we have less total order price.

Recommendation – we should take the challenges where we are having less orders and accordingly we can make our strategies.

Q4.3 Calculate the Total & Average value of order freight for each state.

Answer

```
select c.customer_state as States,round(sum(i.freight_value),1) as
Total_order_freight_value,round(sum(i.freight_value)/count(i.order_id),2) as
Avg_order_freight_value
from `SQL_Project_Target.customers` c join `SQL_Project_Target.orders` o
on c.customer_id=o.customer_id
join `SQL_Project_Target.order_items` i
on i.order_id=o.order_id
group by 1
order by 2,3
```



Row	States	Total_order_freight_value	Avg_order_freight_value
1	RR	2235.2	42.98
2	AP	2788.5	34.01
3	AC	3686.7	40.07
4	AM	5478.9	33.21
5	RO	11417.4	41.07
6	TO	11732.7	37.25
7	SE	14111.5	36.65
8	AL	15914.6	35.84
9	RN	18860.1	35.65
10	MS	19144.0	23.37

Insight – we have the total order freight value and the average order freight value for each and every state which will be helpful to make strategies for every state accordingly. AC/AP/RR/AM are the states where we have less total order freight value (less than 10000).

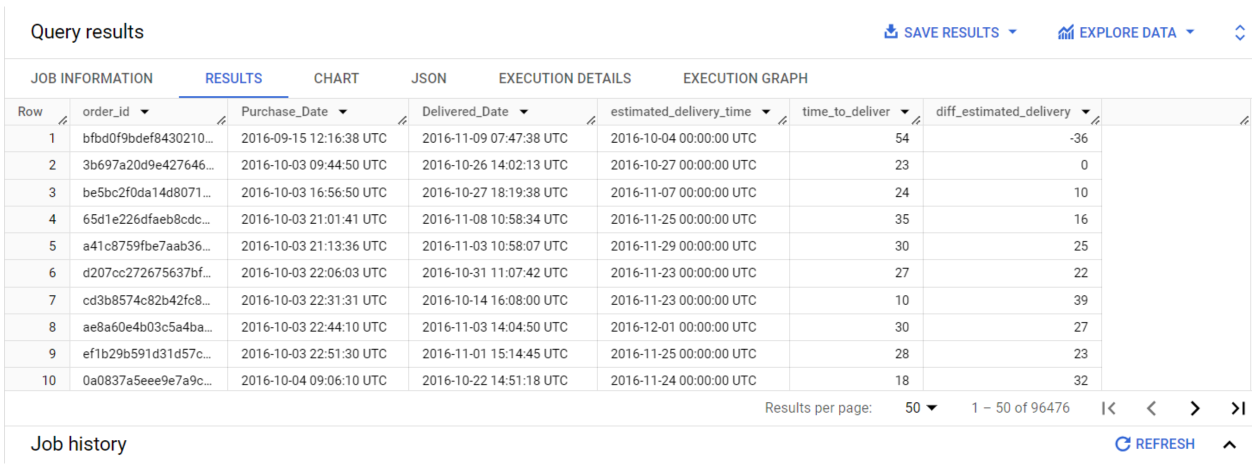
Recommendation – we should take the challenges where we are having less orders freight value and accordingly we can make our strategies.

Q5.1 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.

Answer

```
select order_id, order_purchase_timestamp as
Purchase_Date,order_delivered_customer_date as
Delivered_Date,order_estimated_delivery_date as estimated_delivery_time,
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 `SQL_Project_Target.orders`
where order_delivered_customer_date is not null
```

```
order by 2
```



Query results

Row	order_id	Purchase_Date	Delivered_Date	estimated_delivery_time	time_to_deliver	diff_estimated_delivery
1	bfbd0f9bdef8430210...	2016-09-15 12:16:38 UTC	2016-11-09 07:47:38 UTC	2016-10-04 00:00:00 UTC	54	-36
2	3b697a20d9e427646...	2016-10-03 09:44:50 UTC	2016-10-26 14:02:13 UTC	2016-10-27 00:00:00 UTC	23	0
3	be5bc2f0da14d8071...	2016-10-03 16:56:50 UTC	2016-10-27 18:19:38 UTC	2016-11-07 00:00:00 UTC	24	10
4	65d1e226dfaeb8cdc...	2016-10-03 21:01:41 UTC	2016-11-08 10:58:34 UTC	2016-11-25 00:00:00 UTC	35	16
5	a41c8759fbe7aab36...	2016-10-03 21:13:36 UTC	2016-11-03 10:58:07 UTC	2016-11-29 00:00:00 UTC	30	25
6	d207cc272675637bf...	2016-10-03 22:06:03 UTC	2016-10-31 11:07:42 UTC	2016-11-23 00:00:00 UTC	27	22
7	cd3b8574c82b42fc8...	2016-10-03 22:31:31 UTC	2016-10-14 16:08:00 UTC	2016-11-23 00:00:00 UTC	10	39
8	ae8a60e4b03c5a4ba...	2016-10-03 22:44:10 UTC	2016-11-03 14:04:50 UTC	2016-12-01 00:00:00 UTC	30	27
9	ef1b29b591d31d57c...	2016-10-03 22:51:30 UTC	2016-11-01 15:14:45 UTC	2016-11-25 00:00:00 UTC	28	23
10	0a0837a5eee9e7a9c...	2016-10-04 09:06:10 UTC	2016-10-22 14:51:18 UTC	2016-11-24 00:00:00 UTC	18	32

Results per page: 50   1 – 50 of 96476

Job history

Insight – we have the data where we have the delivery time that we have taken to delivery an order and how efficient we are to deliver the product before an estimated delivery date.

For Example – for 1st order we have taken so much time (54 days) to deliver this and also there is a delay of 36 days after the estimated delivery time.

Recommendation – we must have a discussion with that customer and should apologies and provide some discount on the next order because of late delivery so that we will not lose that customer.

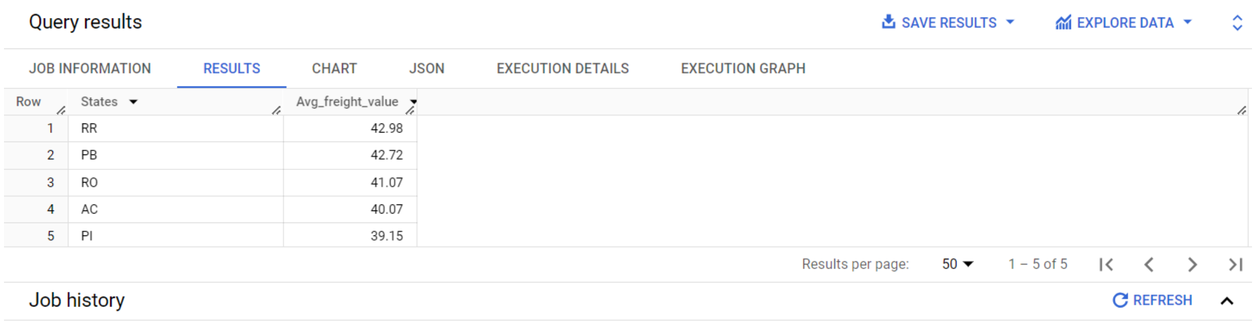
Q5.2 Find out the top 5 states with the highest & lowest average freight value.

Answer:-

(A) Top 5 states with Highest average freight value

```
with cte as
(select customer_state as States,round(avg(freight_value),2) as Avg_freight_value
from `SQL_Project_Target.customers` c join `SQL_Project_Target.orders` o
on c.customer_id=o.customer_id
join `SQL_Project_Target.order_items` i
on i.order_id=o.order_id
group by 1
order by 2 desc)

select * from cte
limit 5
```



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

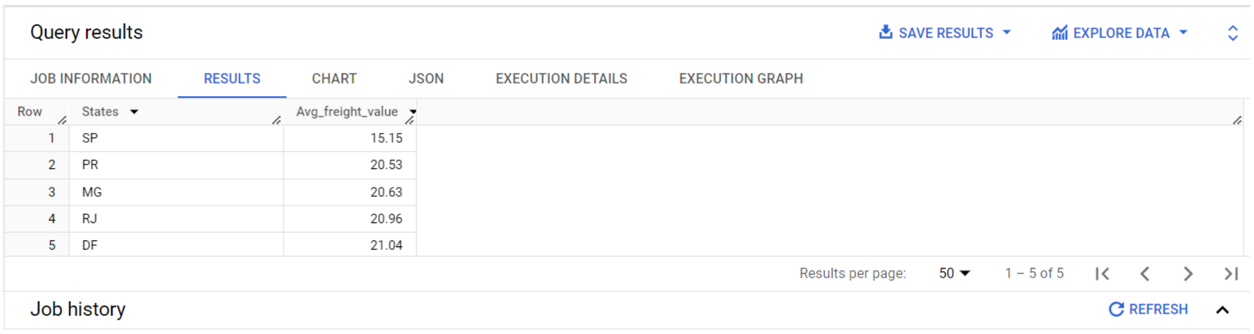
Insight – we have top 5 states where we have highest average freight value.

Recommendation – Freight value must be lower so we should open more stores in multiple locations so we can deliver the product from the nearest stores and make delivery charges minimum

(B) Top 5 states with Lowest average freight value

```
with cte as
(select customer_state as States,round(avg(freight_value),2) as Avg_freight_value
from `SQL_Project_Target.customers` c join `SQL_Project_Target.orders` o
on c.customer_id=o.customer_id
join `SQL_Project_Target.order_items` i
on i.order_id=o.order_id
group by 1
order by 2)

select * from cte
limit 5
```



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

Insight – we have top 5 states where we have lowest average freight value.

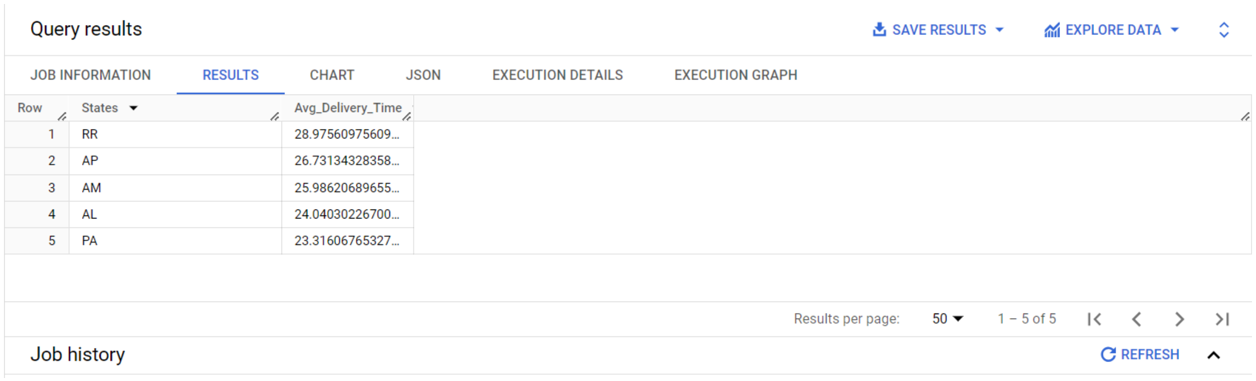
Recommendation – Freight value must be lower so that customers are happy to get the product in low delivery charges.

Q5.3 Find out the top 5 states with the highest & lowest average delivery time.

Answer

(A) Top 5 states with the highest average delivery time

```
select customer_state as States,
avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))
as Avg_Delivery_Time
from `SQL_Project_Target.customers` c join `SQL_Project_Target.orders` o
on c.customer_id=o.customer_id
group by 1
order by 2 desc
limit 5
```



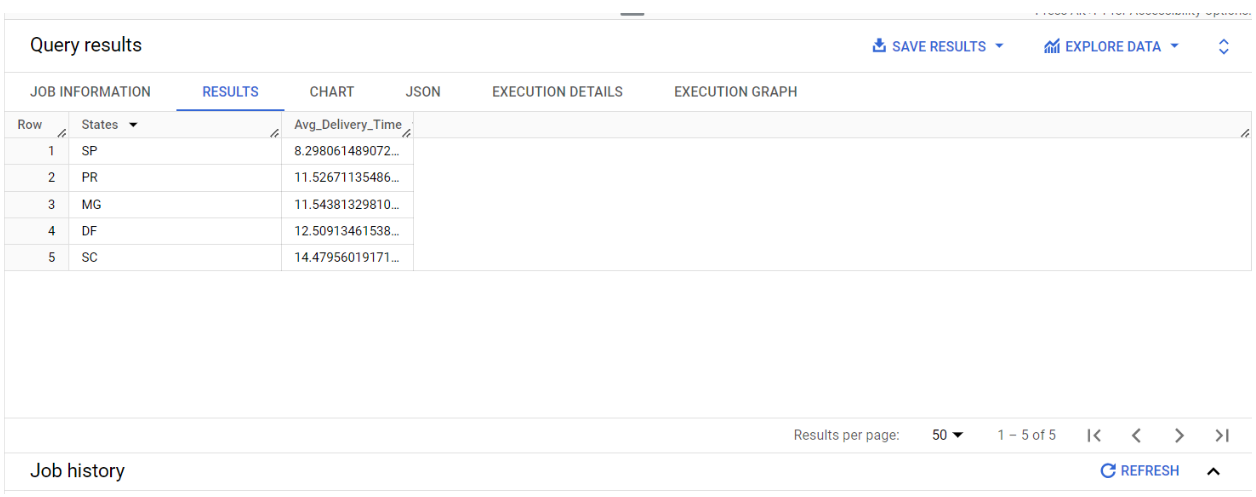
Row	States	Avg_Delivery_Time
1	RR	28.97560975609...
2	AP	26.73134328358...
3	AM	25.98620689655...
4	AL	24.04030226700...
5	PA	23.31606765327...

Insight – we have top 5 states where we have highest average delivery time.

Recommendation – Delivery time must be minimal as it can be so we should get delivery the product in 7-15 days. We should open the more stores where we have more customers so the product can be delivered in less delivery time.

(B)Top 5 states with the lowest average delivery time

```
select customer_state as States,
avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day))
as Avg_Delivery_Time
from `SQL_Project_Target.customers` c join `SQL_Project_Target.orders` o
on c.customer_id=o.customer_id
group by 1
order by 2
limit 5
```


Query results

Row	States	Avg_Delivery_Time
1	SP	8.298061489072...
2	PR	11.52671135486...
3	MG	11.54381329810...
4	DF	12.50913461538...
5	SC	14.47956019171...

Results per page: 50 1 – 5 of 5

Job history

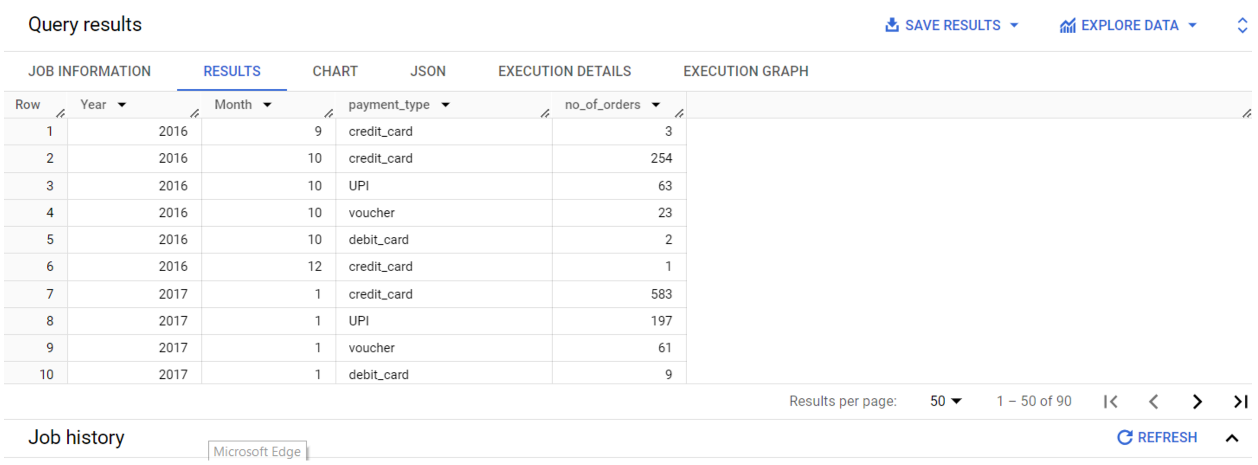
Insight – we have top 5 states where we have lowest average delivery time.

Recommendation – Delivery time must be minimal as it can be so we should get delivery the product in 7-10 days.

Q6.1 Find the month on month no. of orders placed using different payment types.

Answer

```
select extract(year from order_purchase_timestamp) as Year, extract(month from
order_purchase_timestamp) as Month, payment_type,count(o.order_id) as
no_of_orders
from `SQL_Project_Target.customers` c join `SQL_Project_Target.orders` o
on c.customer_id=o.customer_id
join `SQL_Project_Target.payments` p
on o.order_id=p.order_id
group by 1,2,3
order by 1,2,4 desc
```



Row	Year	Month	payment_type	no_of_orders
1	2016	9	credit_card	3
2	2016	10	credit_card	254
3	2016	10	UPI	63
4	2016	10	voucher	23
5	2016	10	debit_card	2
6	2016	12	credit_card	1
7	2017	1	credit_card	583
8	2017	1	UPI	197
9	2017	1	voucher	61
10	2017	1	debit_card	9

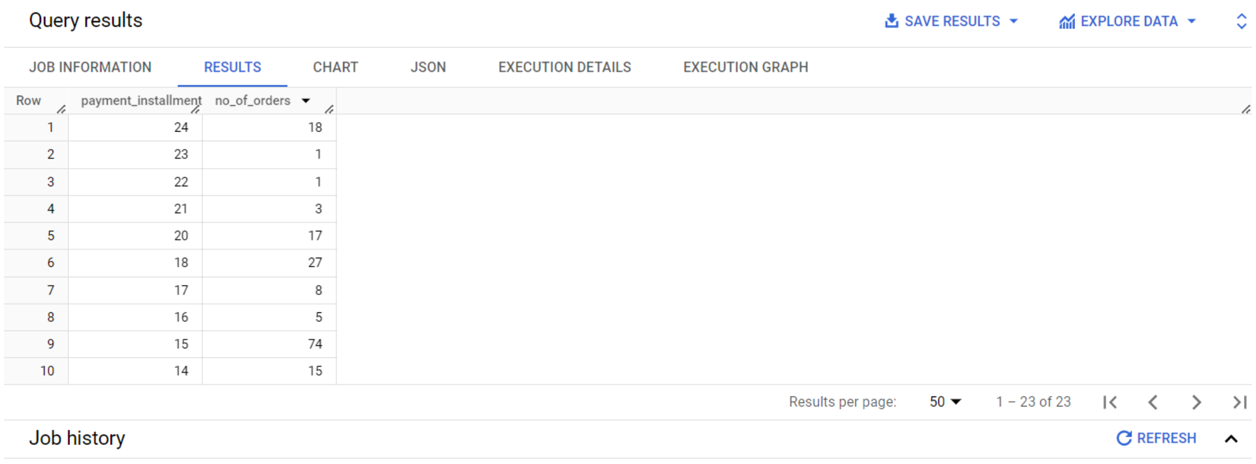
Insight – we can see we are having maximum number of orders through credit card every month.

Recommendations – we should check which credit card is mostly getting used so accordingly we can provide some discount and offers for the card users.

Q6.2 Find the no. of orders placed on the basis of the payment installments that have been paid.

Answer

```
select payment_installments, count(o.order_id) as no_of_orders
from `SQL_Project_Target.orders` o join `SQL_Project_Target.payments` p
on o.order_id=p.order_id
where payment_installments>=1
group by 1
order by 1 desc
```



Row	payment_installment	no_of_orders
1	24	18
2	23	1
3	22	1
4	21	3
5	20	17
6	18	27
7	17	8
8	16	5
9	15	74
10	14	15

Results per page: 50 1 – 23 of 23

Insight – we have total number of orders as per the number of installments have been paid where at least 1 installment has been paid.

Recommendation – we should provide the offers to the customers whose installments are about to finish so they can purchase something else accordingly and also it will help to make the strategies to provide the no cost EMI for 12/24/36 months