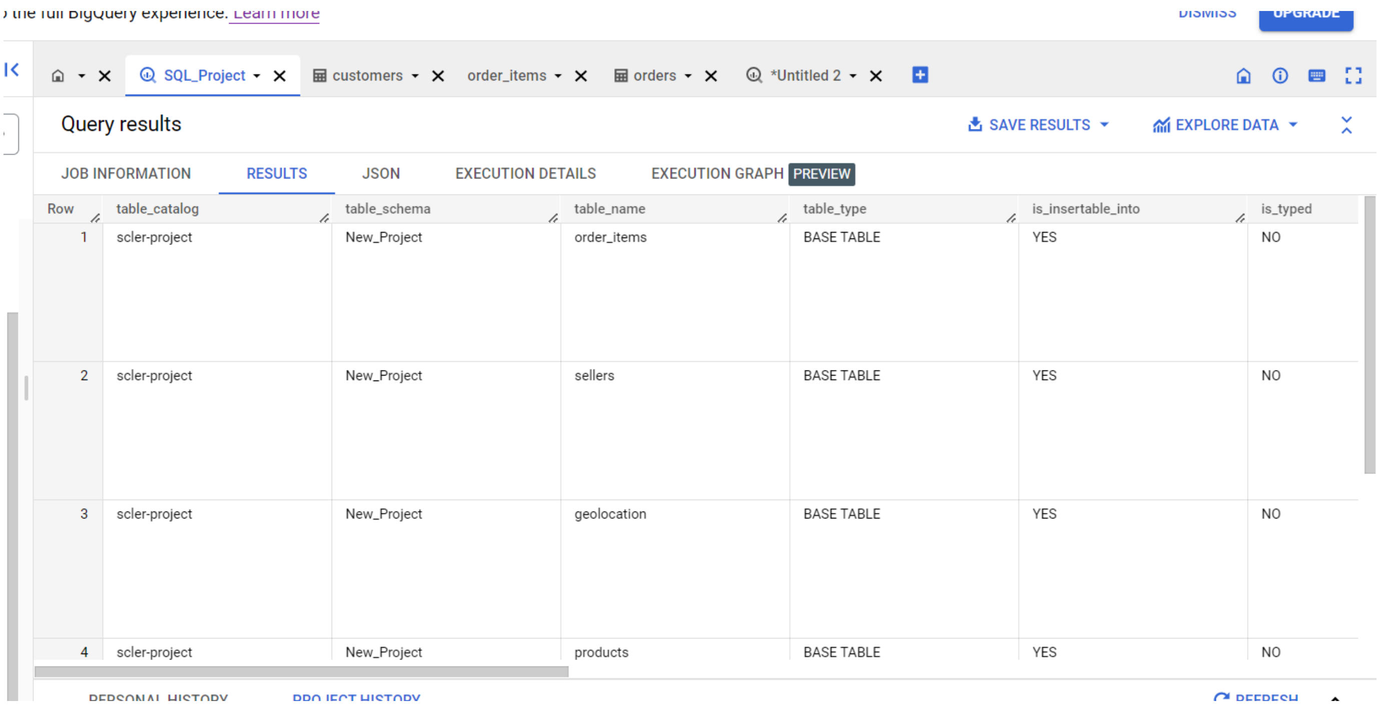
Business Case: Target SQL

Project Link

https://console.cloud.google.com/bigquery?sq=119374501015:c09bd3ab6a2e4a548bedf97f6fbd44fc

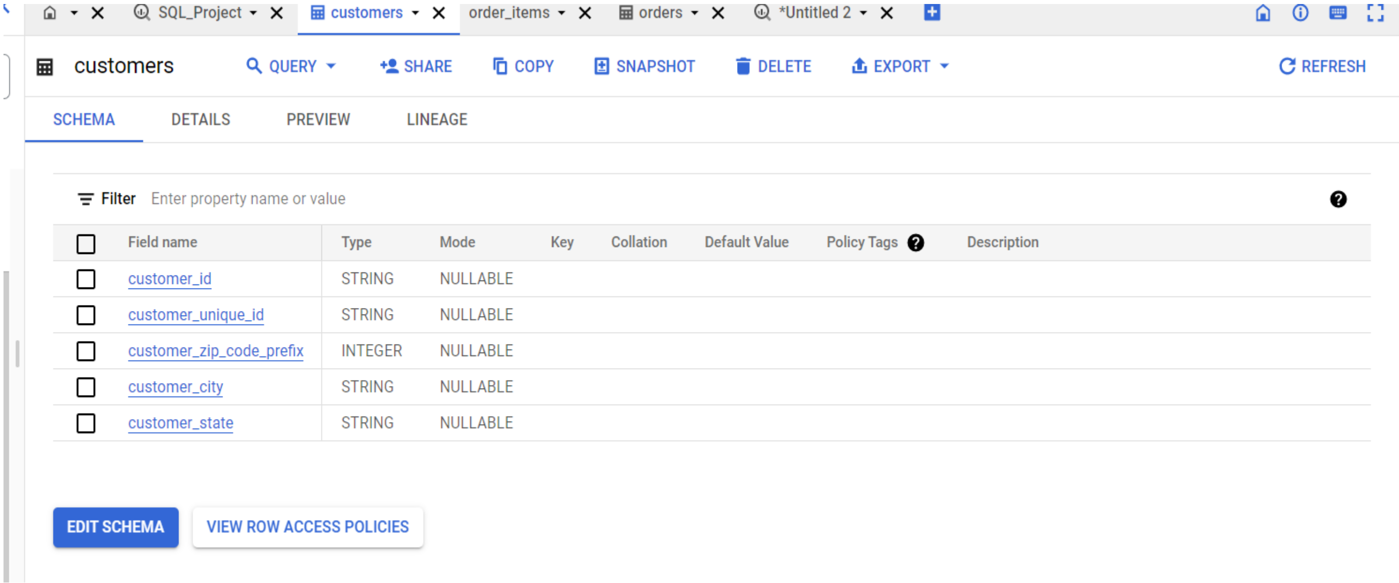
Q1 Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

```
SELECT * FROM New_Project.INFORMATION_SCHEMA.TABLES;
```

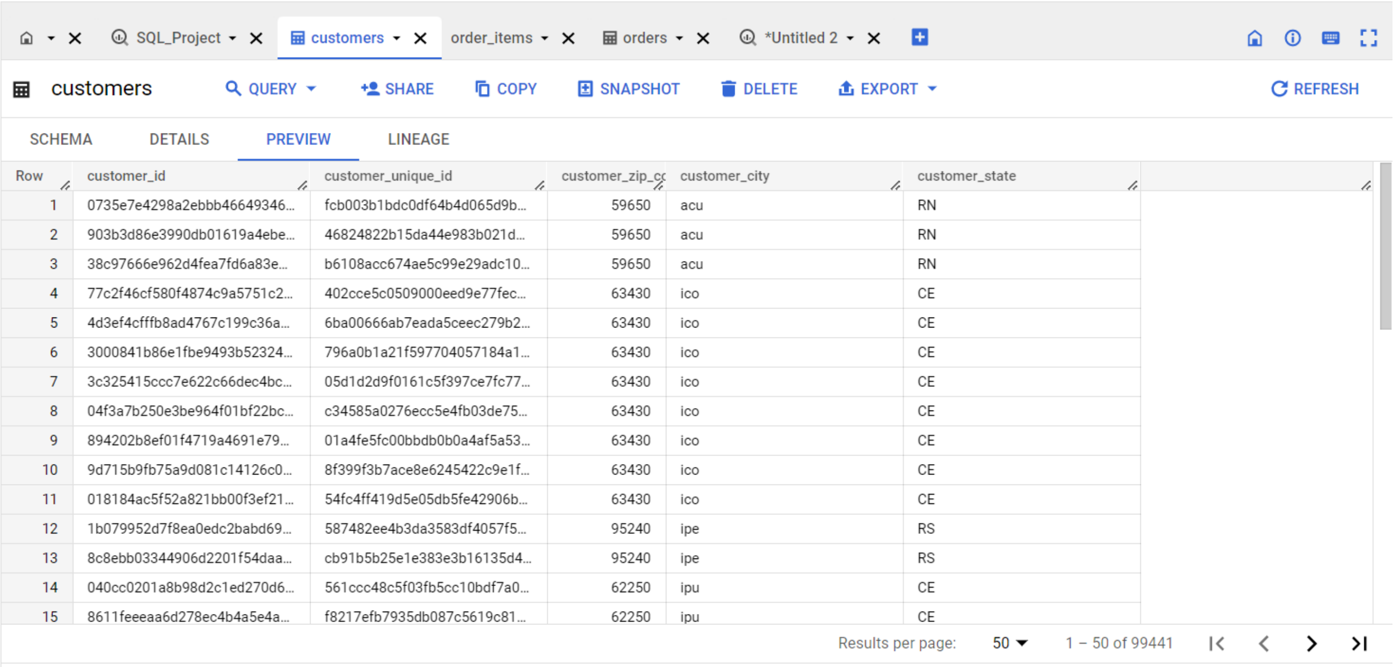


Row	table_catalog	table_schema	table_name	table_type	is_insertable_into	is_typed
1	scler-project	New_Project	order_items	BASE TABLE	YES	NO
2	scler-project	New_Project	sellers	BASE TABLE	YES	NO
3	scler-project	New_Project	geolocation	BASE TABLE	YES	NO
4	scler-project	New_Project	products	BASE TABLE	YES	NO

Customers table



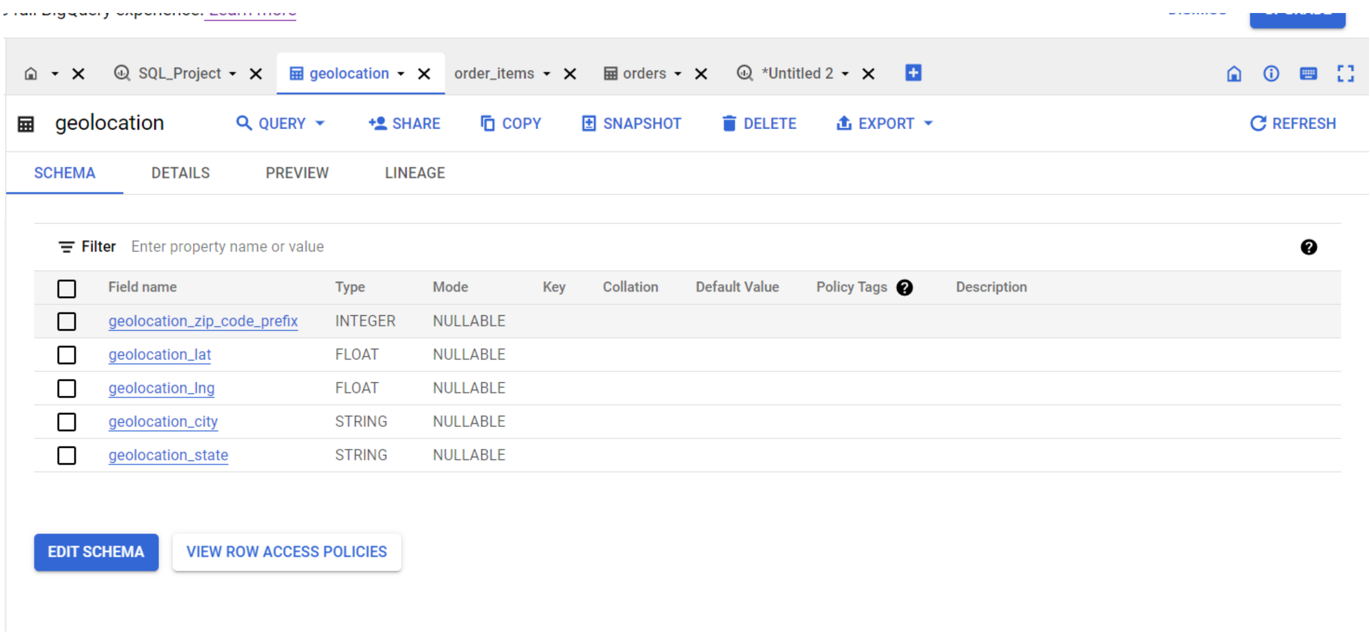
□	Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
□	customer_id	STRING	NULLABLE					
□	customer_unique_id	STRING	NULLABLE					
□	customer_zip_code_prefix	INTEGER	NULLABLE					
□	customer_city	STRING	NULLABLE					
□	customer_state	STRING	NULLABLE					



Row	customer_id	customer_unique_id	customer_zip_cc	customer_city	customer_state
1	0735e7e4298a2ebbb46649346...	fcb003b1bdc0df64b4d065d9b...	59650	acu	RN
2	903b3d86e3990db01619a4ebe...	46824822b15da44e983b021d...	59650	acu	RN
3	38c97666e962d4fea7fd6a83e...	b6108acc674ae5c99e29adc10...	59650	acu	RN
4	77c2f46cf580f4874c9a5751c2...	402cce5c0509000eed9e77fec...	63430	ico	CE
5	4d3ef4cfffb8ad4767c199c36a...	6ba00666ab7eada5ceec279b2...	63430	ico	CE
6	3000841b86e1fbe9493b52324...	796a0b1a21f597704057184a1...	63430	ico	CE
7	3c325415ccc7e622c66dec4bc...	05d1d2d9f0161c5f397ce7fc77...	63430	ico	CE
8	04f3a7b250e3be964f01bf22bc...	c34585a0276ecc5e4fb03de75...	63430	ico	CE
9	894202b8ef01f4719a4691e79...	01a4fe5fc00bbdb0b0a4af5a53...	63430	ico	CE
10	9d715b9fb75a9d081c14126c0...	8f399f3b7ace8e6245422c9e1f...	63430	ico	CE
11	018184ac5f52a821bb00f3ef21...	54fc4ff419d5e05db5fe42906b...	63430	ico	CE
12	1b079952d7f8ea0edc2babd69...	587482ee4b3da3583df4057f5...	95240	ipe	RS
13	8c8ebb03344906d2201f54daa...	cb91b5b25e1e383e3b16135d4...	95240	ipe	RS
14	040cc0201a8b98d2c1ed270d6...	561ccc48c5f03fb5cc10bdf7a0...	62250	ipu	CE
15	8611feeeaa6d278ec4b4a5e4a...	f8217efb7935db087c5619c81...	62250	ipu	CE

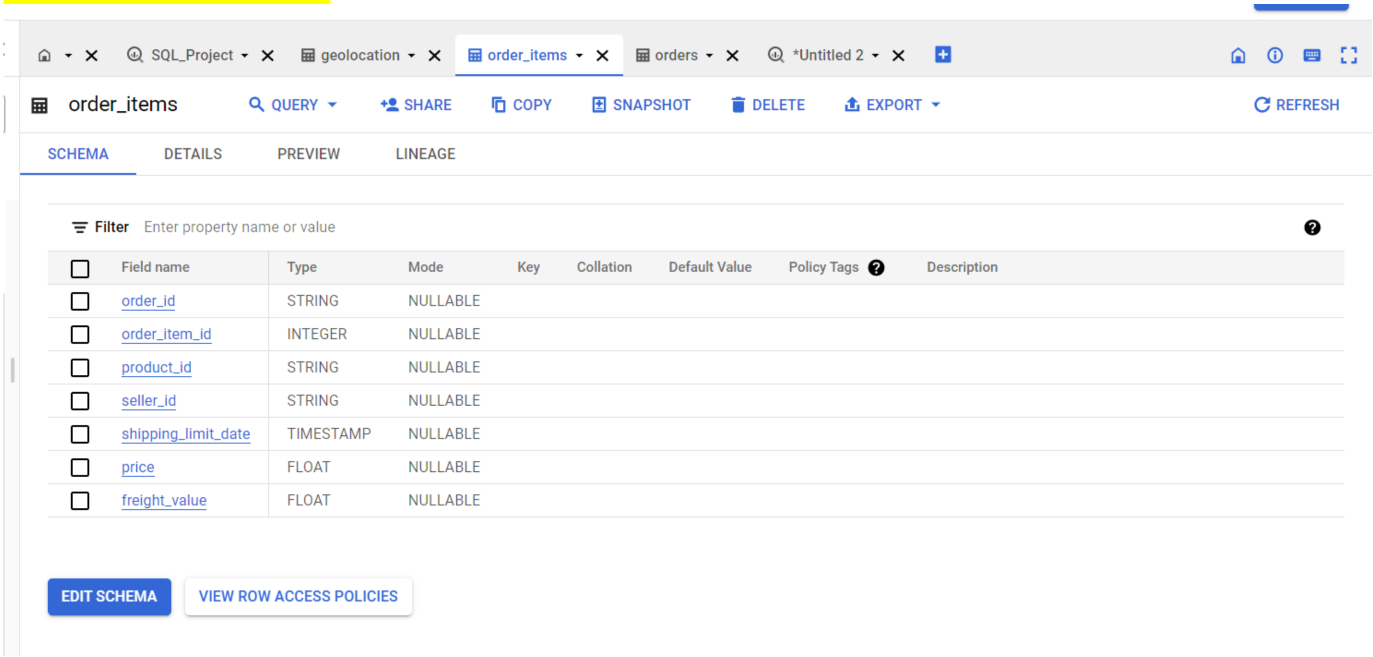
Results per page: 50 1 – 50 of 99441

Geolocation table



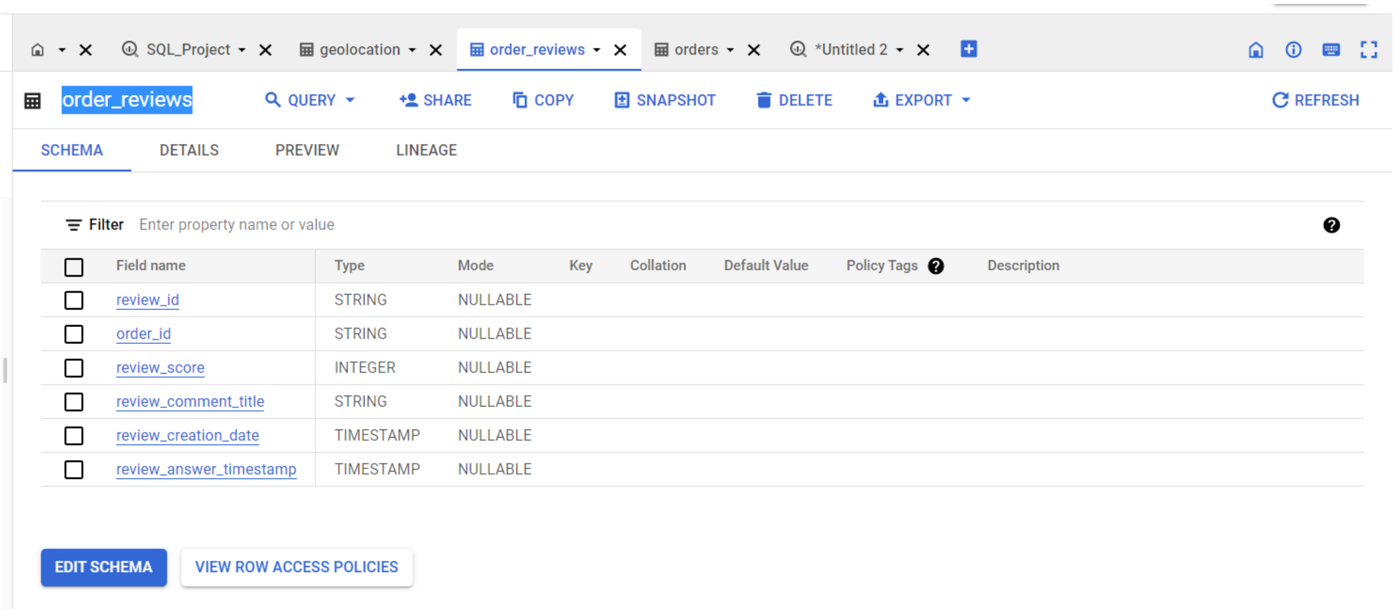
□	Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
□	geolocation_zip_code_prefix	INTEGER	NULLABLE					
□	geolocation_lat	FLOAT	NULLABLE					
□	geolocation_lng	FLOAT	NULLABLE					
□	geolocation_city	STRING	NULLABLE					
□	geolocation_state	STRING	NULLABLE					

Order_items Table



□	Field name	Type	Mode	Key	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					

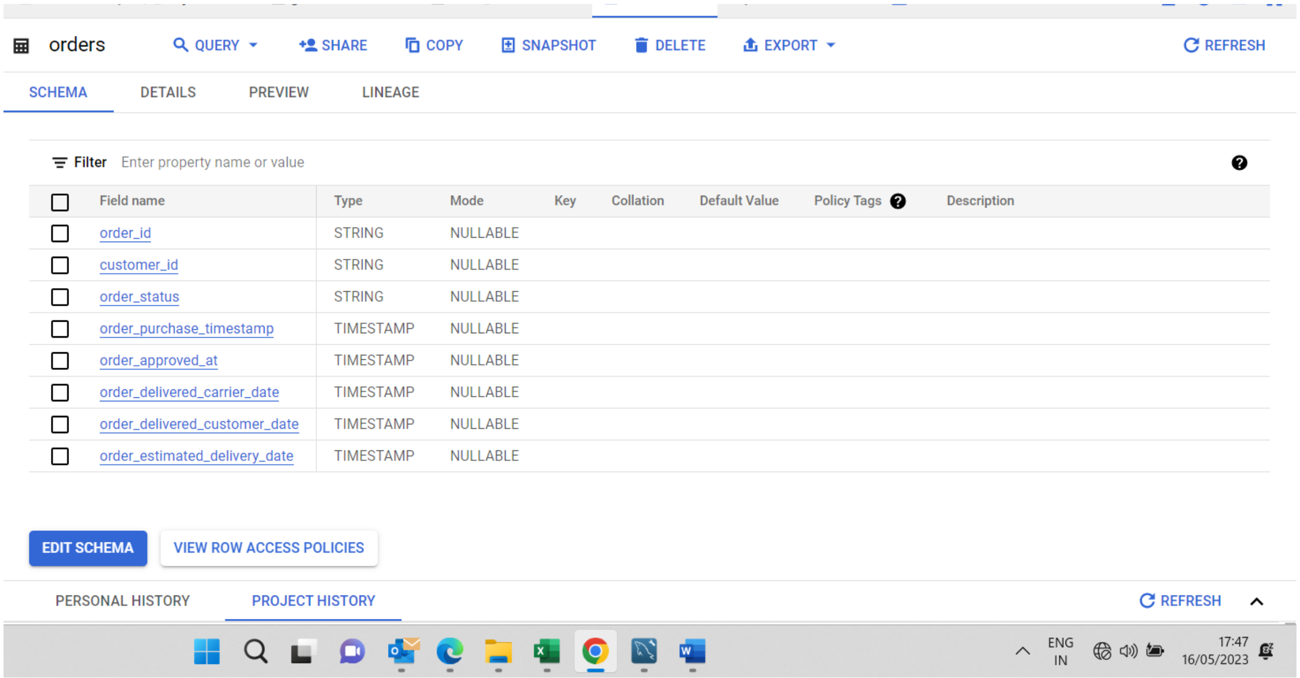
order_reviews Table



order_reviews

	Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
□	review_id	STRING	NULLABLE					
□	order_id	STRING	NULLABLE					
□	review_score	INTEGER	NULLABLE					
□	review_comment_title	STRING	NULLABLE					
□	review_creation_date	TIMESTAMP	NULLABLE					
□	review_answer_timestamp	TIMESTAMP	NULLABLE					

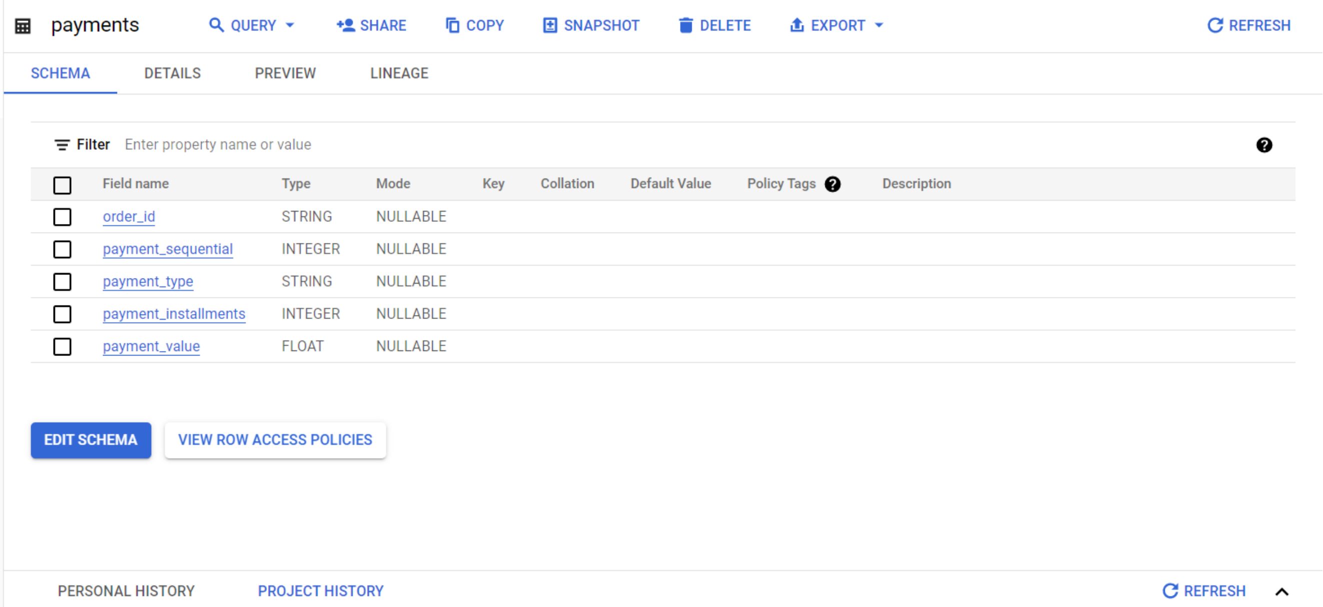
Orders Table



orders

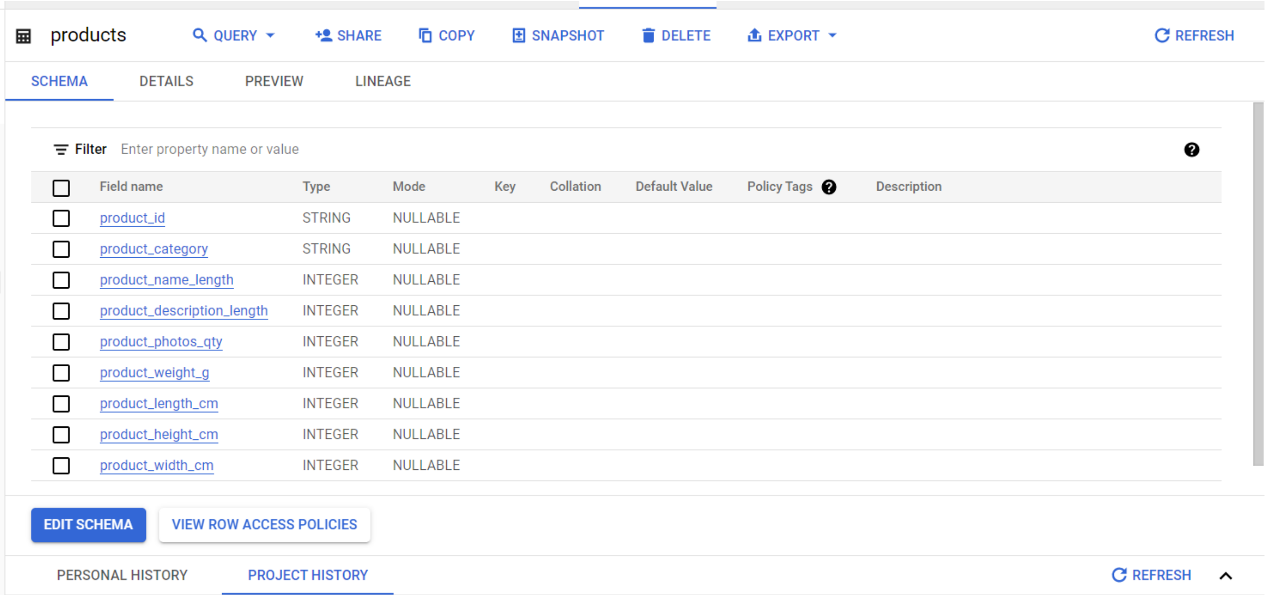
	Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
□	order_id	STRING	NULLABLE					
□	customer_id	STRING	NULLABLE					
□	order_status	STRING	NULLABLE					
□	order_purchase_timestamp	TIMESTAMP	NULLABLE					
□	order_approved_at	TIMESTAMP	NULLABLE					
□	order_delivered_carrier_date	TIMESTAMP	NULLABLE					
□	order_delivered_customer_date	TIMESTAMP	NULLABLE					
□	order_estimated_delivery_date	TIMESTAMP	NULLABLE					

Payments Table



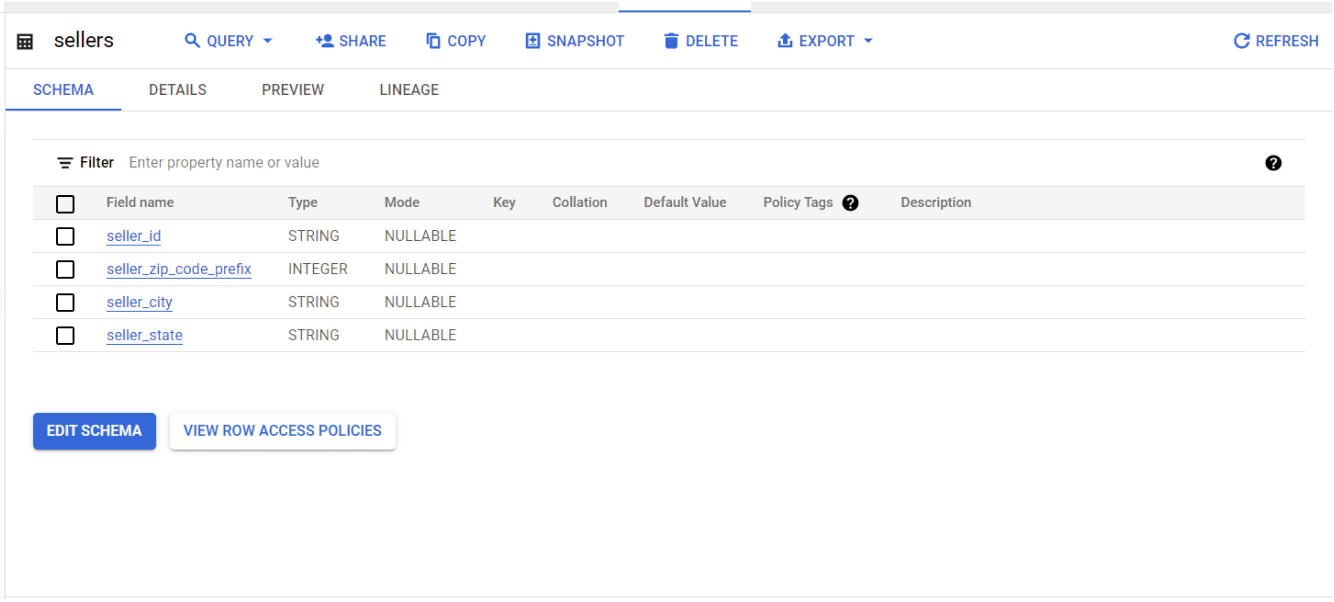
□	Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
□	order_id	STRING	NULLABLE					
□	payment_sequential	INTEGER	NULLABLE					
□	payment_type	STRING	NULLABLE					
□	payment_installments	INTEGER	NULLABLE					
□	payment_value	FLOAT	NULLABLE					

Products Table



□	Field name	Type	Mode	Key	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					

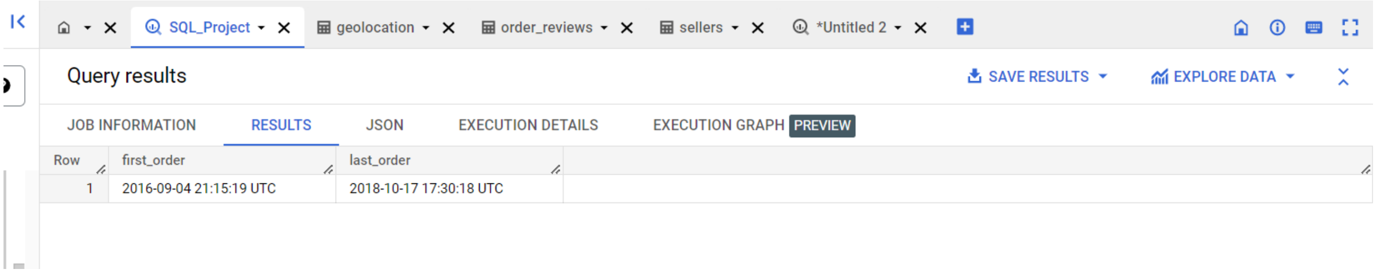
Sellers Table



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

02 Time period for which the data is given

```
SELECT
MIN(order_purchase_timestamp) AS first_order,
MAX(order_purchase_timestamp) AS last_order
from `New_Project.orders`;
```

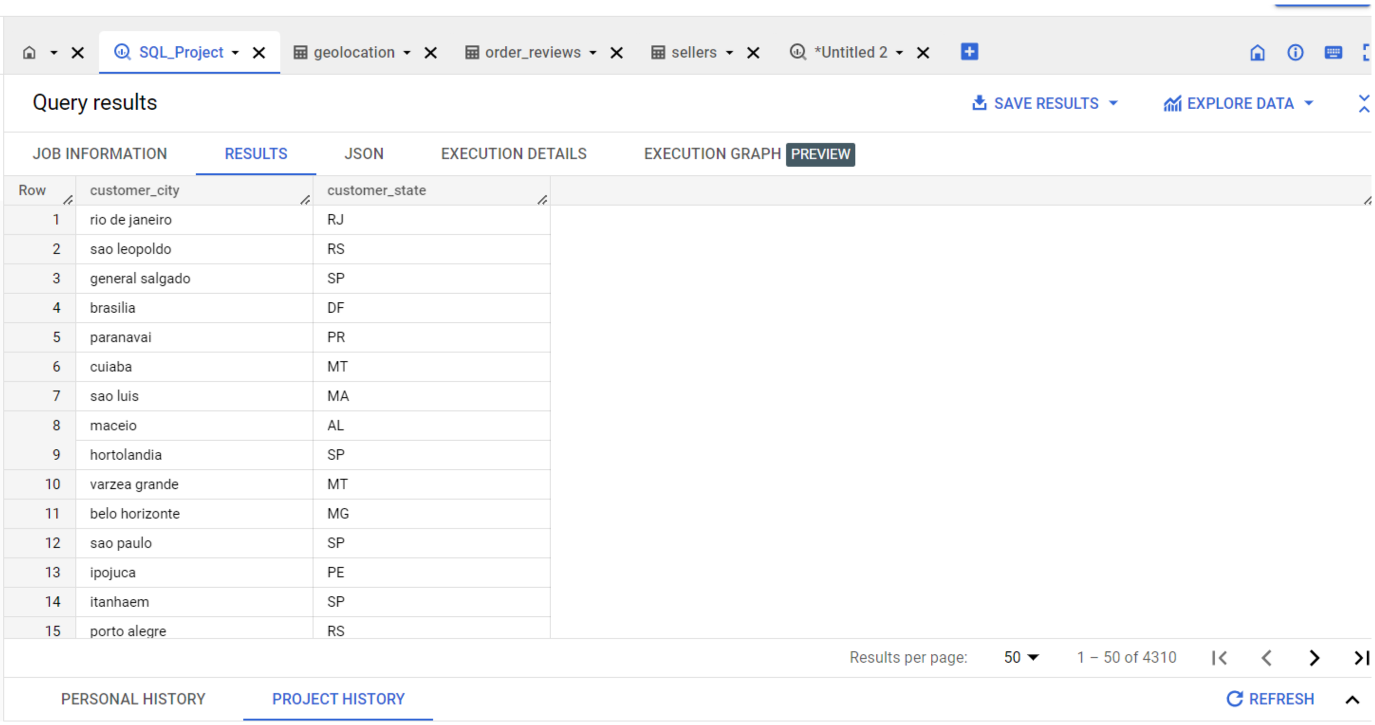


Row	first_order	last_order
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

03 Cities and States of customers ordered during the given period

```
SELECT DISTINCT
`New_Project.customers`.customer_city,
`New_Project.customers`.customer_state
FROM
`New_Project.orders`
```

```
LEFT OUTER JOIN `New_Project.customers` ON
`New_Project.orders`.customer_id = `New_Project.customers`.customer_id;
```

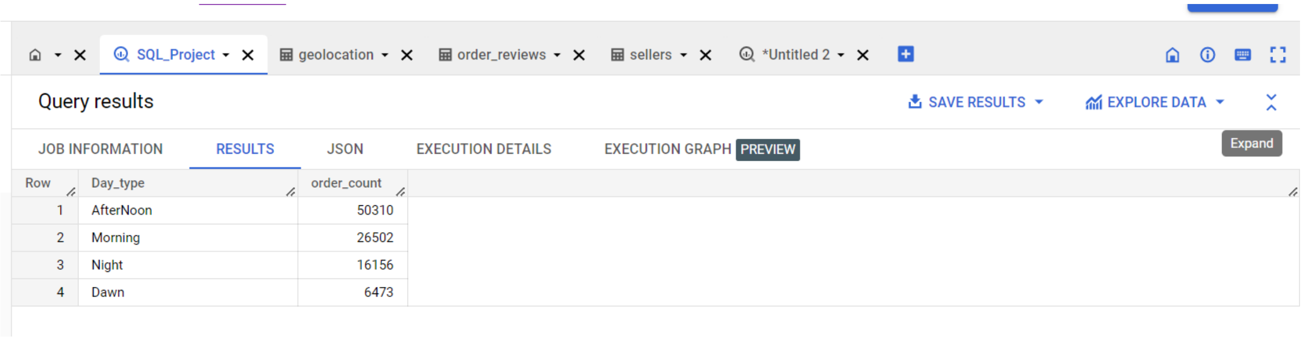


Row	customer_city	customer_state
1	rio de janeiro	RJ
2	sao leopoldo	RS
3	general salgado	SP
4	brasilia	DF
5	paranavai	PR
6	cuiaba	MT
7	sao luis	MA
8	maceio	AL
9	hortolandia	SP
10	varzea grande	MT
11	belo horizonte	MG
12	sao paulo	SP
13	ipojuca	PE
14	itanhaem	SP
15	porto alegre	RS

Results per page: 50 1 – 50 of 4310

Q2 In-depth Exploration

```
SELECT
case
when hour_of_day>=0 and hour_of_day <=7 then 'Dawn'
when hour_of_day>=7 and hour_of_day <=12 then 'Morning'
when hour_of_day>=12 and hour_of_day <=20 then 'AfterNoon'
when hour_of_day>20 then 'Night' end AS Day_type,
count(*) AS order_count
from(select order_purchase_timestamp,
extract(hour from order_purchase_timestamp)AS hour_of_day from `New_Project.orders`)x
group by Day_type
order by count(*) DESC;
```

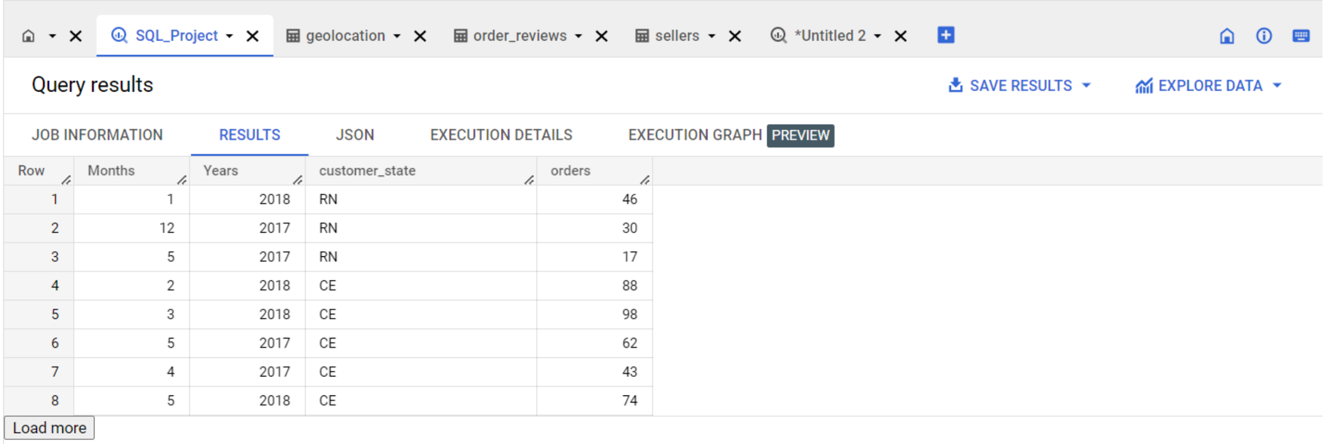


Row	Day_type	order_count
1	AfterNoon	50310
2	Morning	26502
3	Night	16156
4	Dawn	6473

Q3 Evolution of E-commerce orders in the Brazil region

1 Get month on month orders by states

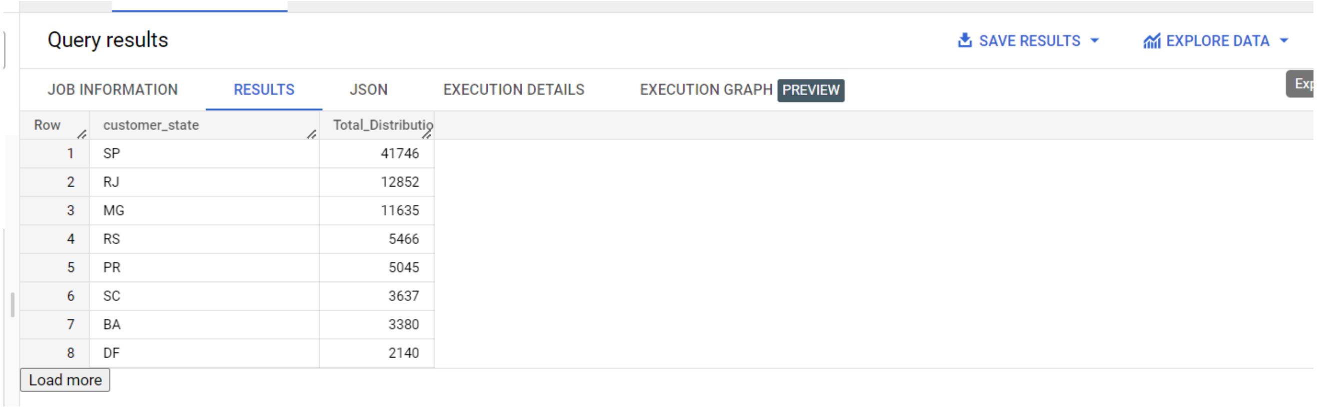
```
select
extract(month from o.order_purchase_timestamp) AS Months,
extract(year from o.order_purchase_timestamp) AS Years,
c.customer_state, count(o.order_id) AS orders
from `New_Project.customers` c
JOIN `New_Project.orders` o ON c.customer_id = o.customer_id
group by c.customer_state,Months,years;
```



Row	Months	Years	customer_state	orders
1	1	2018	RN	46
2	12	2017	RN	30
3	5	2017	RN	17
4	2	2018	CE	88
5	3	2018	CE	98
6	5	2017	CE	62
7	4	2017	CE	43
8	5	2018	CE	74

2 Distribution of customers across the states in Brazil

```
select customer_state, count(*) AS Total_Distribution
from `New_Project.customers`
group by customer_state
order by count(*) DESC;
```

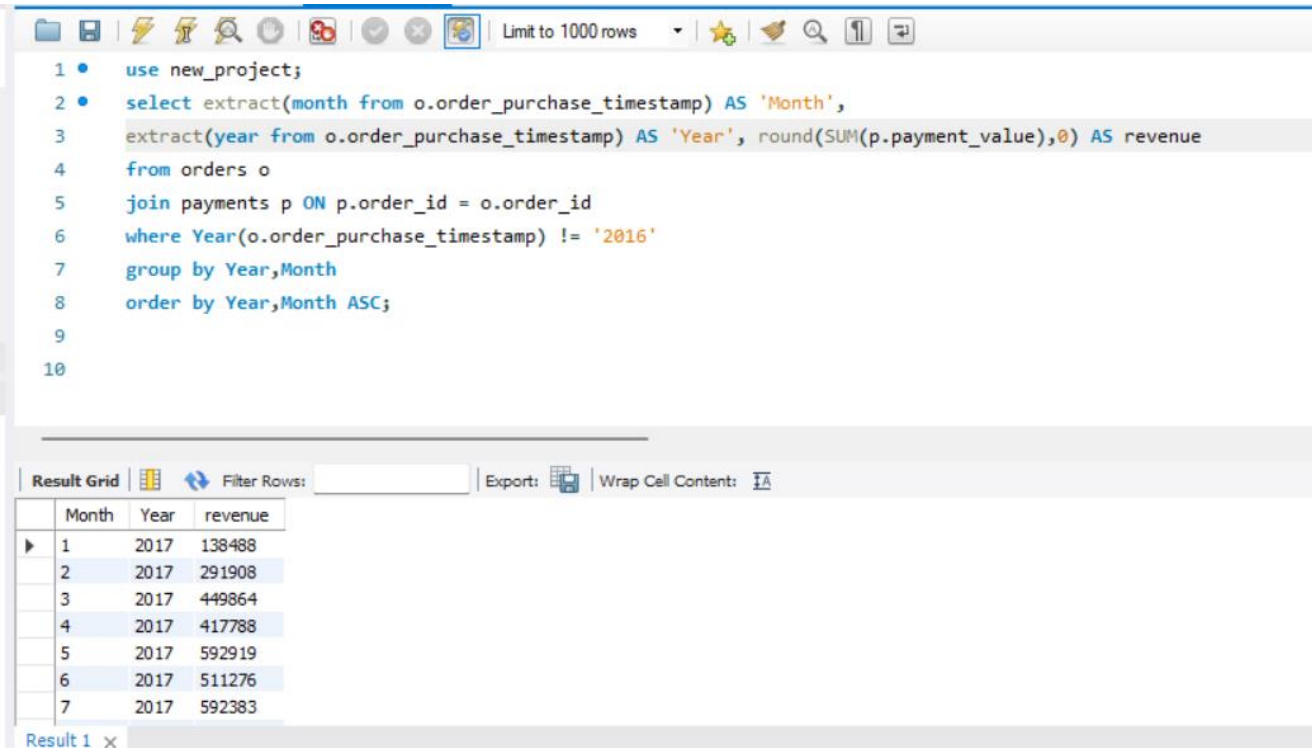


Row	customer_state	Total_Distributio
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140

Q4 Impact on Economy

1 Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only

```
use new_project;
select extract(month from o.order_purchase_timestamp) AS 'Month',
extract(year from o.order_purchase_timestamp) AS 'Year',
round(SUM(p.payment_value),0) AS revenue
from orders o
join payments p ON p.order_id = o.order_id
where Year(o.order_purchase_timestamp) != '2016'
group by Year,Month
order by Year,Month ASC;
```

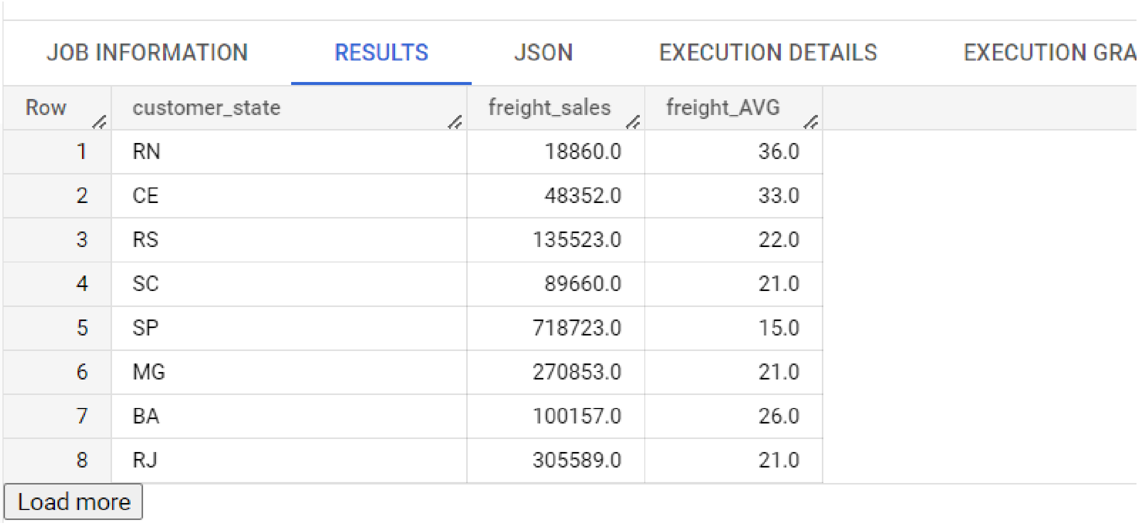


```sql
use new_project;
select extract(month from o.order_purchase_timestamp) AS 'Month',
extract(year from o.order_purchase_timestamp) AS 'Year', round(SUM(p.payment_value),0) AS revenue
from orders o
join payments p ON p.order_id = o.order_id
where Year(o.order_purchase_timestamp) != '2016'
group by Year,Month
order by Year,Month ASC;
```

Month	Year	revenue
1	2017	138488
2	2017	291908
3	2017	449864
4	2017	417788
5	2017	592919
6	2017	511276
7	2017	592383

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

```sql
select c.customer_state, round(sum(p.freight_value)) AS freight_sales,
round(AVG(p.freight_value)) AS freight_AVG
from `New_Project.customers` c
JOIN `New_Project.orders` o ON c.customer_id = o.customer_id
JOIN `New_Project.order_items` p ON p.order_id = o.order_id
group by c.customer_state;
```

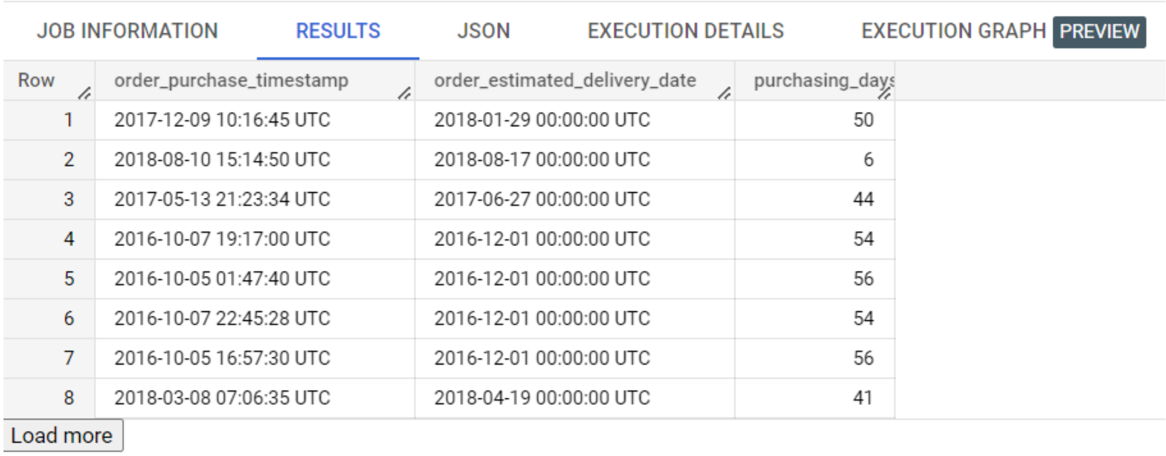


Row	customer_state	freight_sales	freight_AVG
1	RN	18860.0	36.0
2	CE	48352.0	33.0
3	RS	135523.0	22.0
4	SC	89660.0	21.0
5	SP	718723.0	15.0
6	MG	270853.0	21.0
7	BA	100157.0	26.0
8	RJ	305589.0	21.0

Q5 Analysis on sales, freight and delivery time

1 Calculate days between purchasing, delivering and estimated delivery

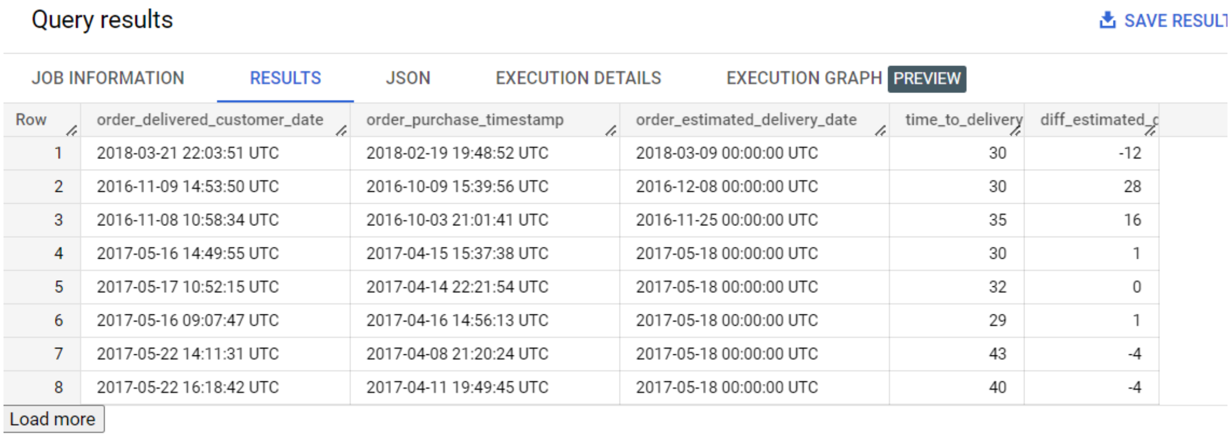
```
SELECT order_purchase_timestamp, order_estimated_delivery_date,
TIMESTAMP_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY) AS
purchasing_days
FROM `New_Project.orders`;
```



Row	order_purchase_timestamp	order_estimated_delivery_date	purchasing_days
1	2017-12-09 10:16:45 UTC	2018-01-29 00:00:00 UTC	50
2	2018-08-10 15:14:50 UTC	2018-08-17 00:00:00 UTC	6
3	2017-05-13 21:23:34 UTC	2017-06-27 00:00:00 UTC	44
4	2016-10-07 19:17:00 UTC	2016-12-01 00:00:00 UTC	54
5	2016-10-05 01:47:40 UTC	2016-12-01 00:00:00 UTC	56
6	2016-10-07 22:45:28 UTC	2016-12-01 00:00:00 UTC	54
7	2016-10-05 16:57:30 UTC	2016-12-01 00:00:00 UTC	56
8	2018-03-08 07:06:35 UTC	2018-04-19 00:00:00 UTC	41

2 Find time_to_delivery & diff_estimated_delivery. Formula for the same given below

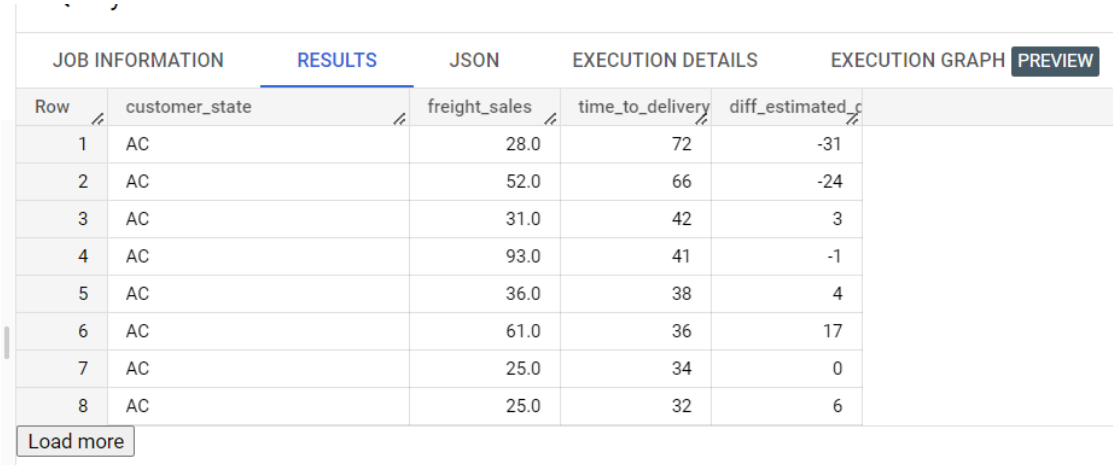
```
SELECT order_delivered_customer_date,
order_purchase_timestamp,order_estimated_delivery_date,
TIMESTAMP_DIFF(order_delivered_customer_date, order_purchase_timestamp,DAY) AS
time_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS
diff_estimated_delivery
FROM `New_Project.orders`;
```



Row	order_delivered_customer_date	order_purchase_timestamp	order_estimated_delivery_date	time_to_delivery	diff_estimated_
1	2018-03-21 22:03:51 UTC	2018-02-19 19:48:52 UTC	2018-03-09 00:00:00 UTC	30	-12
2	2016-11-09 14:53:50 UTC	2016-10-09 15:39:56 UTC	2016-12-08 00:00:00 UTC	30	28
3	2016-11-08 10:58:34 UTC	2016-10-03 21:01:41 UTC	2016-11-25 00:00:00 UTC	35	16
4	2017-05-16 14:49:55 UTC	2017-04-15 15:37:38 UTC	2017-05-18 00:00:00 UTC	30	1
5	2017-05-17 10:52:15 UTC	2017-04-14 22:21:54 UTC	2017-05-18 00:00:00 UTC	32	0
6	2017-05-16 09:07:47 UTC	2017-04-16 14:56:13 UTC	2017-05-18 00:00:00 UTC	29	1
7	2017-05-22 14:11:31 UTC	2017-04-08 21:20:24 UTC	2017-05-18 00:00:00 UTC	43	-4
8	2017-05-22 16:18:42 UTC	2017-04-11 19:49:45 UTC	2017-05-18 00:00:00 UTC	40	-4

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

```
select c.customer_state, round(sum(p.freight_value)) AS freight_sales,
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 `New_Project.customers` c
JOIN `New_Project.orders` o ON c.customer_id = o.customer_id
JOIN `New_Project.order_items` p ON p.order_id = o.order_id
group by c.customer_state,time_to_delivery,diff_estimated_delivery
order by c.customer_state,time_to_delivery DESC,diff_estimated_delivery DESC;
```

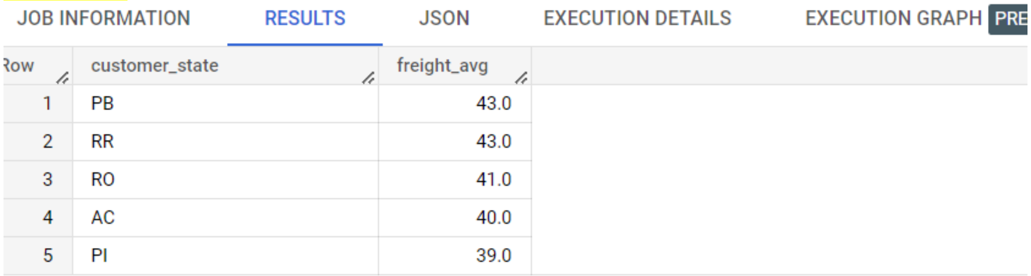


Row	customer_state	freight_sales	time_to_delivery	diff_estimated_c
1	AC	28.0	72	-31
2	AC	52.0	66	-24
3	AC	31.0	42	3
4	AC	93.0	41	-1
5	AC	36.0	38	4
6	AC	61.0	36	17
7	AC	25.0	34	0
8	AC	25.0	32	6

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

```
select c.customer_state, round(AVG(p.freight_value)) AS freight_avg
from `New_Project.customers` c
JOIN `New_Project.orders` o ON c.customer_id = o.customer_id
JOIN `New_Project.order_items` p ON p.order_id = o.order_id
group by c.customer_state
order by freight_avg DESC
limit 5;
```

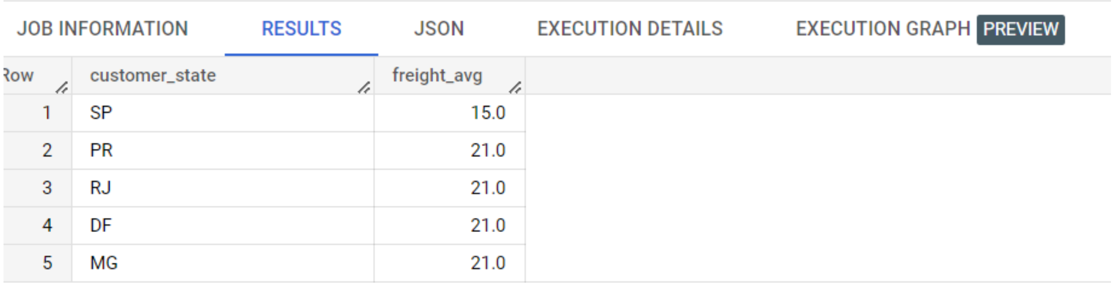
Highest



Row	customer_state	freight_avg
1	PB	43.0
2	RR	43.0
3	RO	41.0
4	AC	40.0
5	PI	39.0

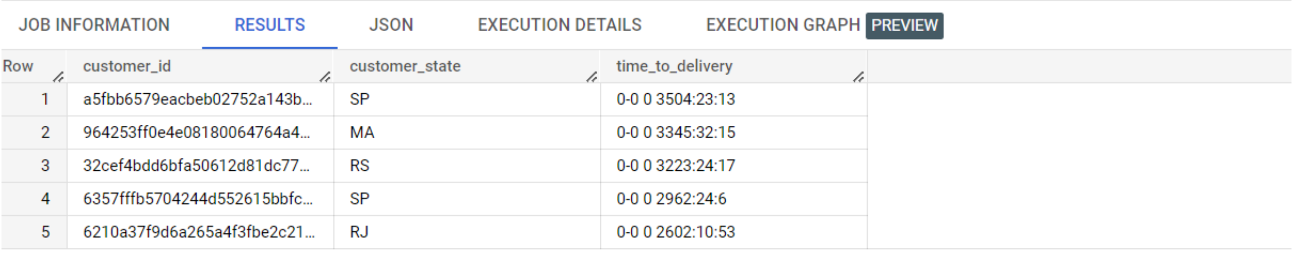
```sql
select c.customer_state, round(AVG(p.freight_value)) AS freight_avg
from `New_Project.customers` c
JOIN `New_Project.orders` o ON c.customer_id = o.customer_id
JOIN `New_Project.order_items` p ON p.order_id = o.order_id
group by c.customer_state
order by freight_avg ASC
limit 5;
```

Lowest



Row	customer_state	freight_avg
1	SP	15.0
2	PR	21.0
3	RJ	21.0
4	DF	21.0
5	MG	21.0

```sql
--6 Top 5 states with highest/lowest average time to delivery
--Highest
select o.customer_id, c.customer_state,
(order_estimated_delivery_date-order_delivered_customer_date) AS time_to_delivery
from `New_Project.orders` o
join `New_Project.customers` c ON o.customer_id = c.customer_id
order by time_to_delivery DESC
limit 5;
```



Row	customer_id	customer_state	time_to_delivery
1	a5fbb6579eacbeb02752a143b...	SP	0-0 0 3504:23:13
2	964253ff0e4e08180064764a4...	MA	0-0 0 3345:32:15
3	32cef4bdd6bfa50612d81dc77...	RS	0-0 0 3223:24:17
4	6357fffb5704244d552615bbfc...	SP	0-0 0 2962:24:6
5	6210a37f9d6a265a4f3fbe2c21...	RJ	0-0 0 2602:10:53

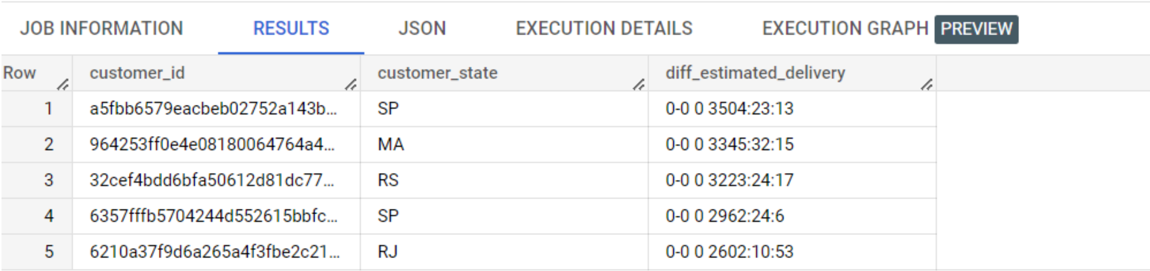
Lowest

```sql
select o.customer_id, c.customer_state,
(order_estimated_delivery_date-order_delivered_customer_date) AS time_to_delivery
from `New_Project.orders` o
join `New_Project.customers` c ON o.customer_id = c.customer_id
order by time_to_delivery ASC
limit 5;
```

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

Highest

```
select o.customer_id, c.customer_state,
(o.order_estimated_delivery_date-o.order_delivered_customer_date) AS
diff_estimated_delivery
from `New_Project.orders` o
join `New_Project.customers` c ON o.customer_id = c.customer_id
order by diff_estimated_delivery DESC
limit 5;
```



Row	customer_id	customer_state	diff_estimated_delivery
1	a5fbb6579eacbeb02752a143b...	SP	0-0 0 3504:23:13
2	964253ff0e4e08180064764a4...	MA	0-0 0 3345:32:15
3	32cef4bdd6bfa50612d81dc77...	RS	0-0 0 3223:24:17
4	6357fffb5704244d552615bbfc...	SP	0-0 0 2962:24:6
5	6210a37f9d6a265a4f3fbe2c21...	RJ	0-0 0 2602:10:53

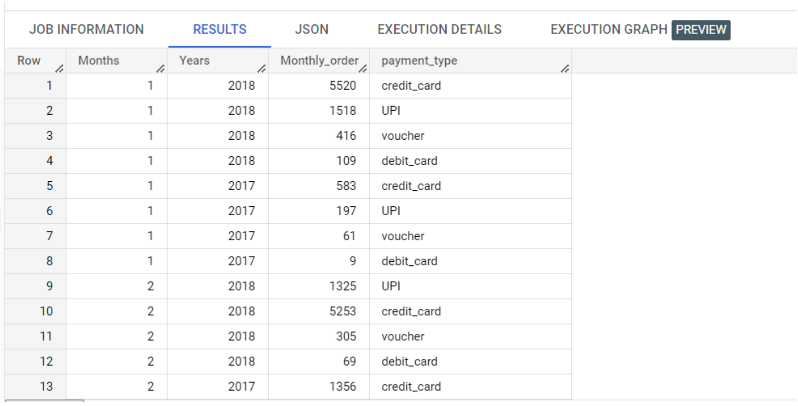
Lowest

```
select o.customer_id, c.customer_state,
(o.order_estimated_delivery_date-o.order_delivered_customer_date) AS
diff_estimated_delivery
from `New_Project.orders` o
join `New_Project.customers` c ON o.customer_id = c.customer_id
order by diff_estimated_delivery ASC
limit 5;
```

Q6 Payment type analysis:

1 Month over Month count of orders for different payment types

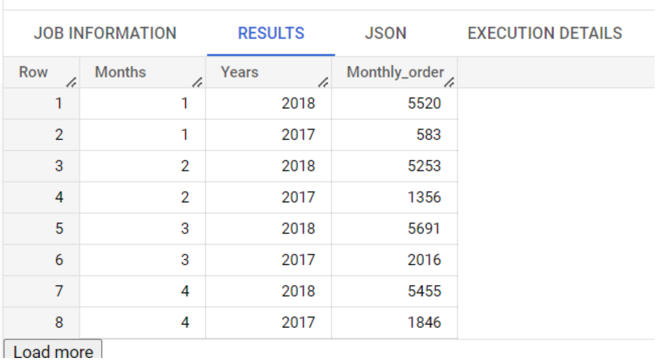
```
select extract(month from o.order_purchase_timestamp) AS Months,
extract(Year from o.order_purchase_timestamp) AS Years,
count(o.order_id) as Monthly_order, p.payment_type
from `New_Project.orders` o
JOIN `New_Project.payments`p ON p.order_id = o.order_id
group by Months,Years,p.payment_type
order by Months ASC, Years DESC;
```



Row	Months	Years	Monthly_order	payment_type
1	1	2018	5520	credit_card
2	1	2018	1518	UPI
3	1	2018	416	voucher
4	1	2018	109	debit_card
5	1	2017	583	credit_card
6	1	2017	197	UPI
7	1	2017	61	voucher
8	1	2017	9	debit_card
9	2	2018	1325	UPI
10	2	2018	5253	credit_card
11	2	2018	305	voucher
12	2	2018	69	debit_card
13	2	2017	1356	credit_card

credit_card

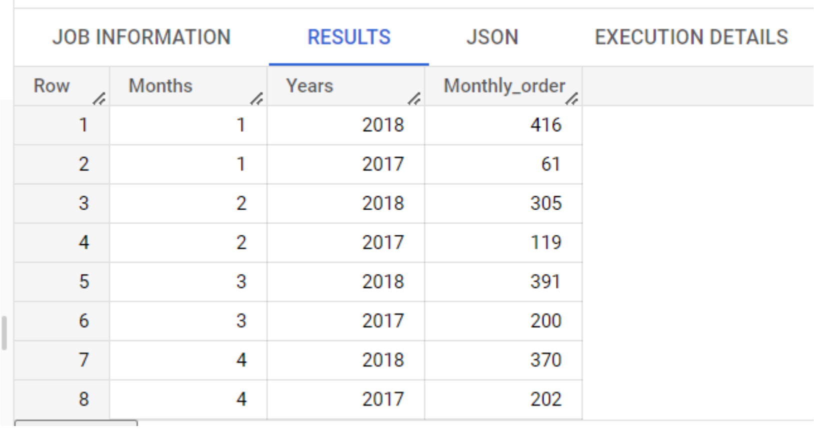
```
select extract(month from o.order_purchase_timestamp) AS Months,
extract(Year from o.order_purchase_timestamp) AS Years,
count(o.order_id) as Monthly_order
from `New_Project.orders` o
JOIN `New_Project.payments`p ON p.order_id = o.order_id
WHERE p.payment_type = 'credit_card'
group by Months,Years
order by Months ASC, Years DESC;
```



Row	Months	Years	Monthly_order
1	1	2018	5520
2	1	2017	583
3	2	2018	5253
4	2	2017	1356
5	3	2018	5691
6	3	2017	2016
7	4	2018	5455
8	4	2017	1846

voucher

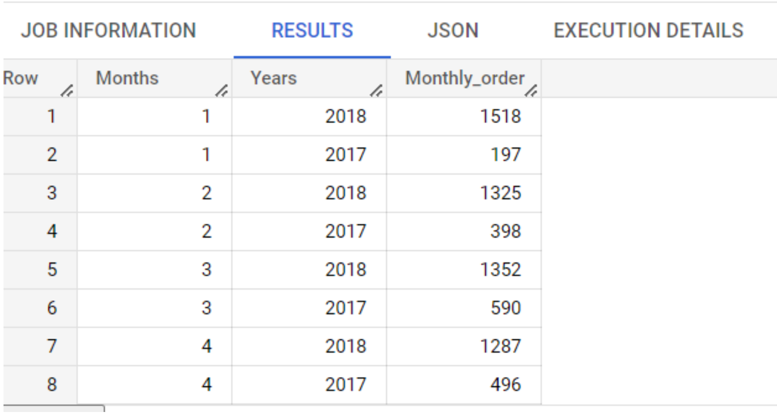
```
select extract(month from o.order_purchase_timestamp) AS Months,
extract(Year from o.order_purchase_timestamp) AS Years,
count(o.order_id) as Monthly_order
from `New_Project.orders` o
JOIN `New_Project.payments`p ON p.order_id = o.order_id
WHERE p.payment_type = 'voucher'
group by Months,Years
order by Months ASC, Years DESC;
```



Row	Months	Years	Monthly_order
1	1	2018	416
2	1	2017	61
3	2	2018	305
4	2	2017	119
5	3	2018	391
6	3	2017	200
7	4	2018	370
8	4	2017	202

UPI

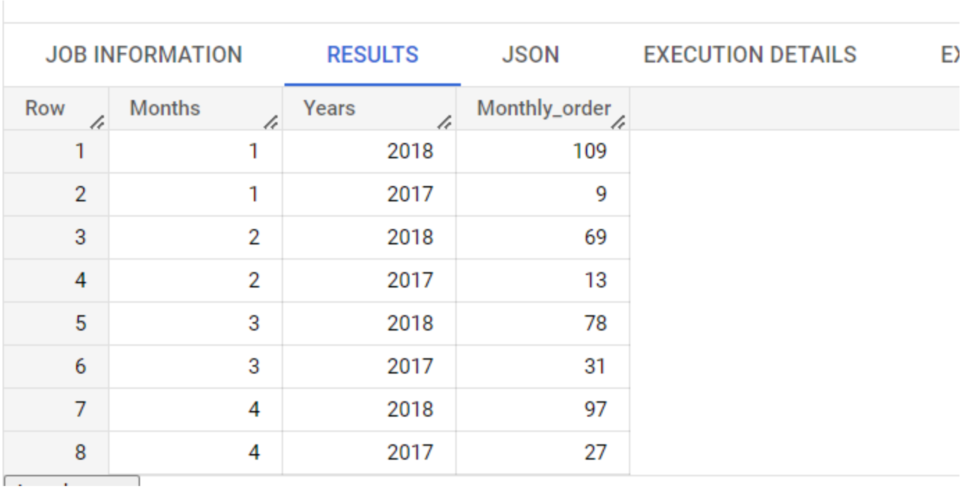
```
select extract(month from o.order_purchase_timestamp) AS Months,
extract(Year from o.order_purchase_timestamp) AS Years,
count(o.order_id) as Monthly_order
from `New_Project.orders` o
JOIN `New_Project.payments`p ON p.order_id = o.order_id
WHERE p.payment_type = 'UPI'
group by Months,Years
order by Months ASC, Years DESC;
```



Row	Months	Years	Monthly_order
1	1	2018	1518
2	1	2017	197
3	2	2018	1325
4	2	2017	398
5	3	2018	1352
6	3	2017	590
7	4	2018	1287
8	4	2017	496

debit_card

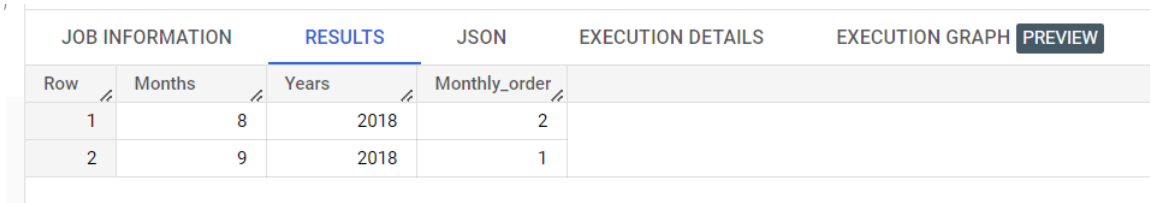
```
select extract(month from o.order_purchase_timestamp) AS Months,
extract(Year from o.order_purchase_timestamp) AS Years,
count(o.order_id) as Monthly_order
from `New_Project.orders` o
JOIN `New_Project.payments`p ON p.order_id = o.order_id
WHERE p.payment_type = 'debit_card'
group by Months,Years
order by Months ASC, Years DESC;
```



Row	Months	Years	Monthly_order
1	1	2018	109
2	1	2017	9
3	2	2018	69
4	2	2017	13
5	3	2018	78
6	3	2017	31
7	4	2018	97
8	4	2017	27

not_defined

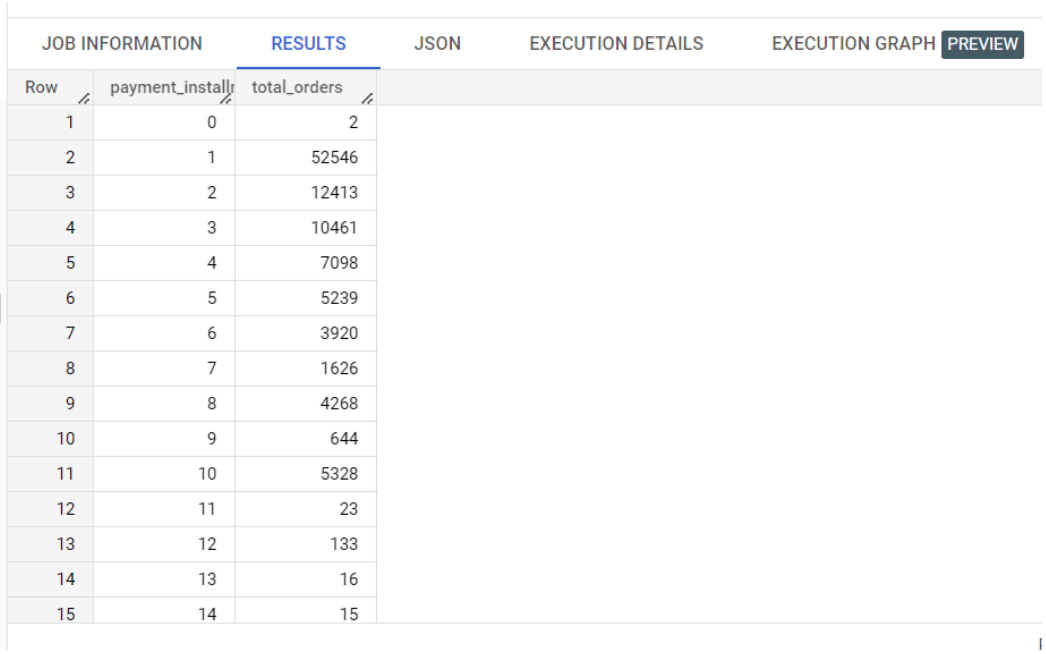
```
select extract(month from o.order_purchase_timestamp) AS Months,
extract(Year from o.order_purchase_timestamp) AS Years,
count(o.order_id) as Monthly_order
from `New_Project.orders` o
JOIN `New_Project.payments`p ON p.order_id = o.order_id
WHERE p.payment_type = 'not_defined'
group by Months,Years
order by Months ASC, Years DESC;
```



Row	Months	Years	Monthly_order
1	8	2018	2
2	9	2018	1

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

```
select payment_installments, count(order_id) AS total_orders
from `New_Project.payments`
group by payment_installments;
```



Row	payment_install	total_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644
11	10	5328
12	11	23
13	12	133
14	13	16
15	14	15

7. **Actionable Insights**

E-commerce on Brazil really has a growing trend along the time. It can be clearly seen in general that customers are more prone to buy things online than before. As per this observation, smooth and fast delivery service would increase the sales more

8. **Recommendations**

Monday are the preferred day for Brazilian's customers and they tend to buy more at afternoons. So that, more active and smooth customer services at this time should be , would really help to increase sales