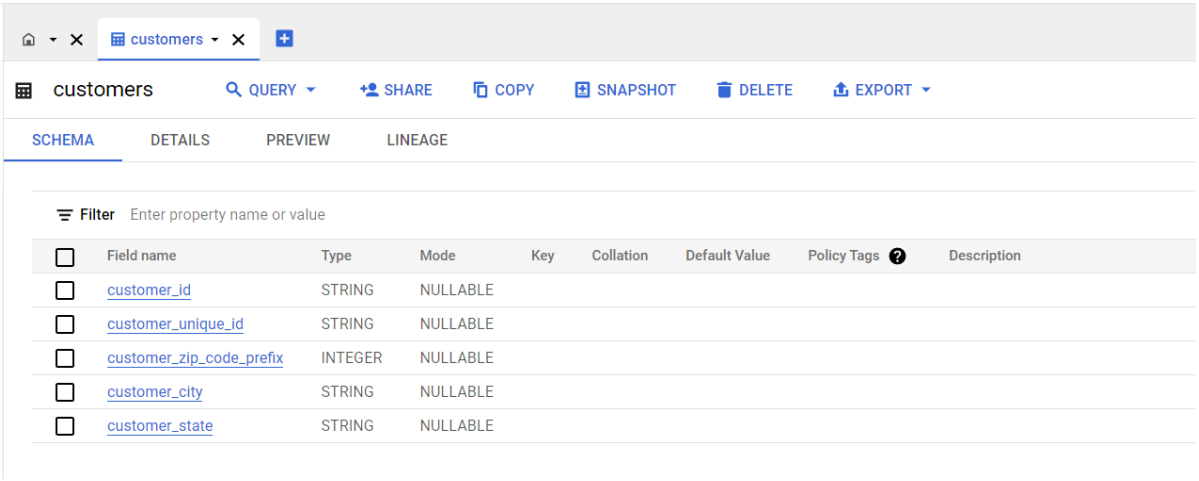
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

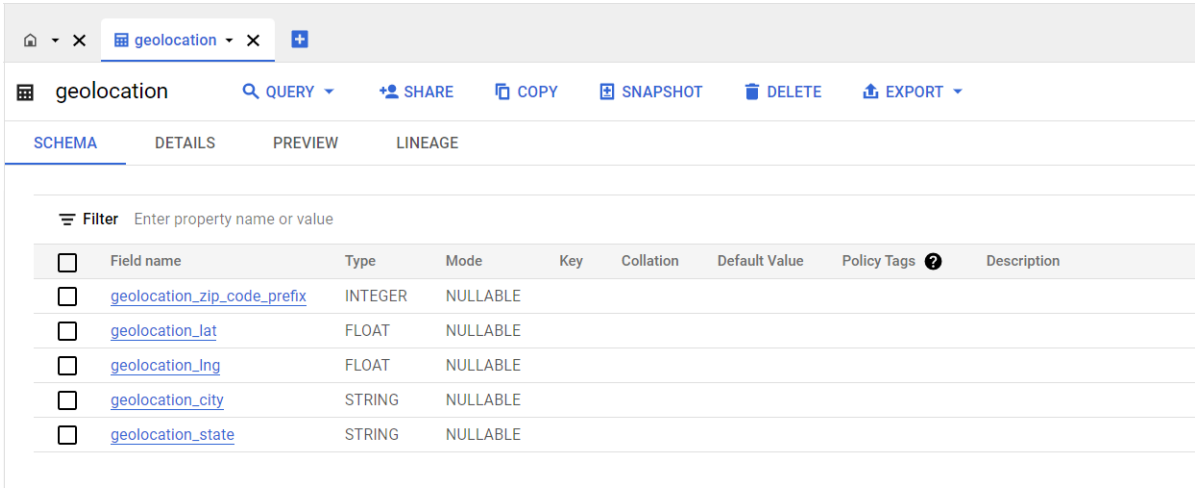
- A) Data type of columns in a table

  Ans:

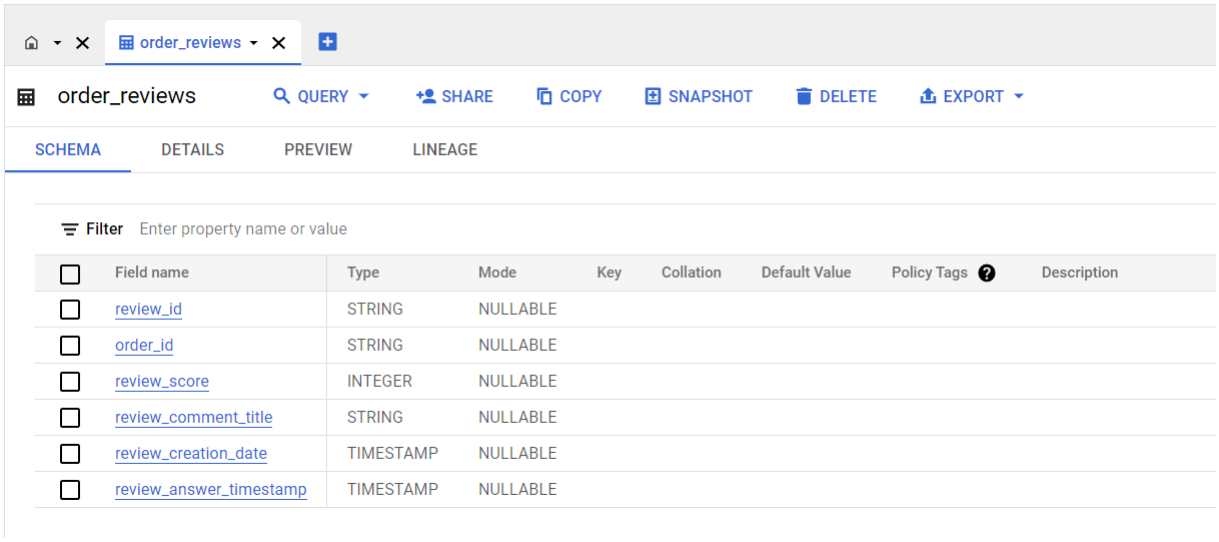
Customers :



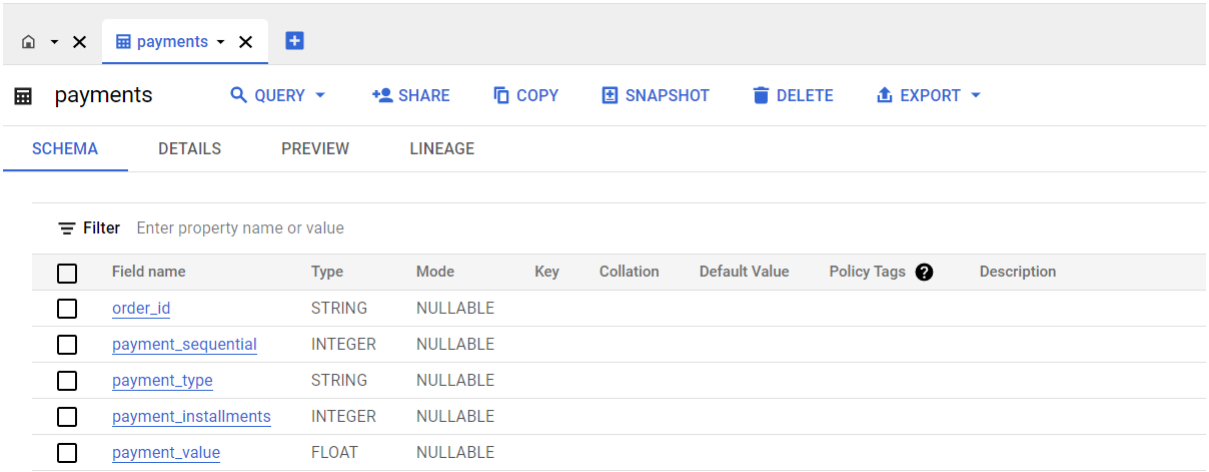
Geolocation :



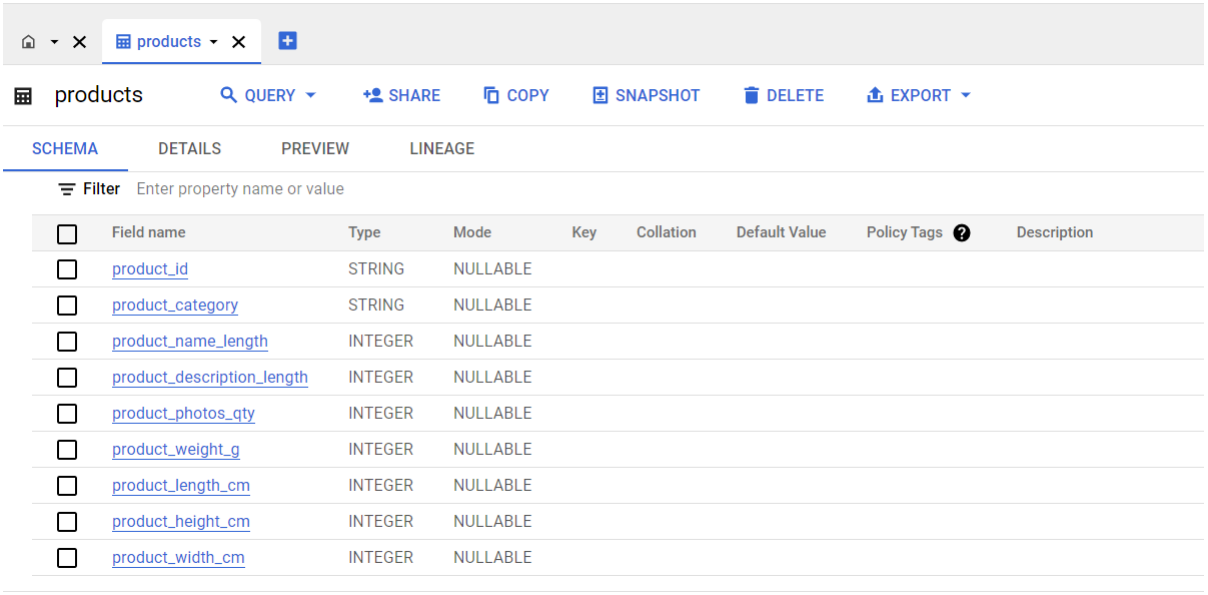
Order_reviews :



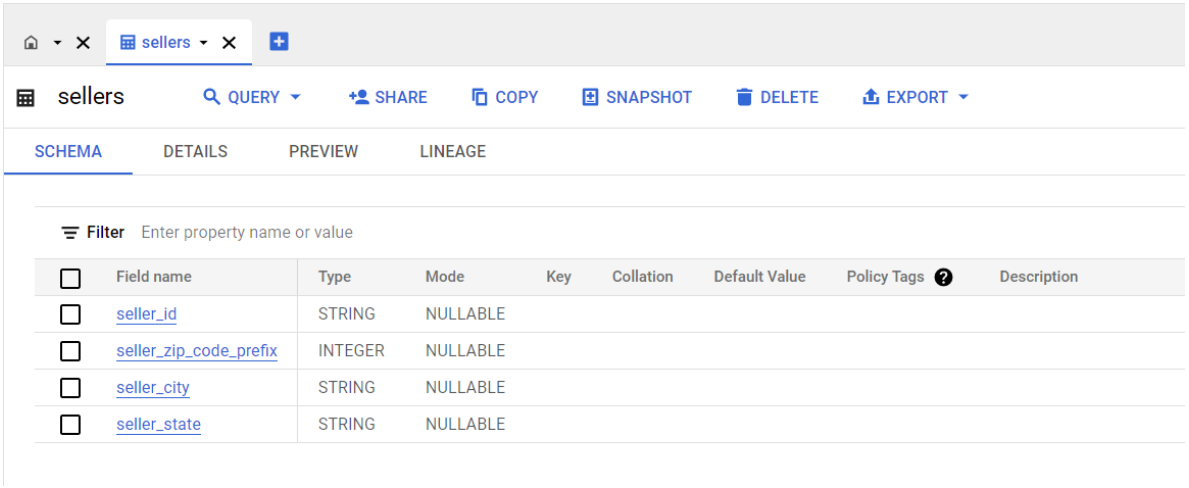
Payments :



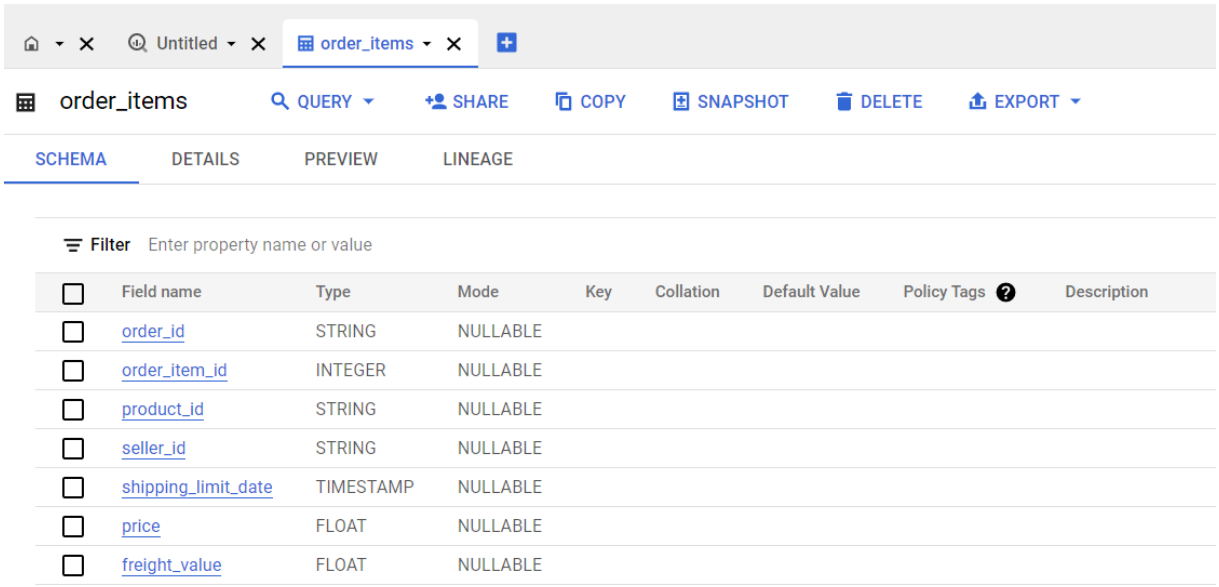
Prooducts :



Sellers :

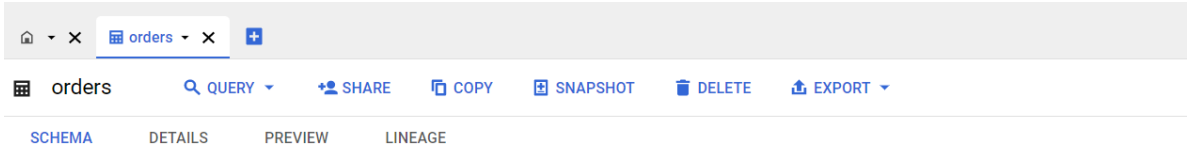


Order_items :



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					

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					

- B) Time period for which the data is given

Ans: "
```
select min(order_purchase_timestamp) as
first_order_date,max(order_purchase_timestamp) as last_order_date
from `scaler_project.orders`"
```

Query results

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

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

- C) Cities and States of customers ordered during the given period

Ans: "
```
select  distinct  customer_state,customer_city
from `scaler_project.orders` o inner join `scaler_project.customers` c
on o.customer_id=c.customer_id"
```

Row	customer_state	customer_city
1	RJ	rio de janeiro
2	RS	sao leopoldo
3	SP	general salgado
4	DF	brasilia
5	PR	paranavai
6	MT	cuiaba
7	MA	sao luis
8	AL	maceio
9	SP	hortolandia
10	MT	varzea grande

2. In-depth Exploration:

- A) Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

Ans: "
```
select extract(year from order_purchase_timestamp) as year,extract(month from order_purchase_timestamp) as month,count(order_id) as no_of_orders
from `scaler_project.orders` group by 1,2 order by 1,2 "
```

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

Insight : There is a gradual increase in the sales from the month of January 2017 to august 2018 with a peak in November 2017.

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

Ans: "
```
with time_data as (select *,case when extract(hour from order_purchase_timestamp) between 0 and 6 then "Dwan" when extract(hour from order_purchase_timestamp) between 7 and 12 then "Morning" when extract(hour from order_purchase_timestamp) between 13 and 18 then "Afternoon" else "Night" end as time_division from `scaler_project.orders` )

select time_division,count(order_id) from time_data group by 1"
```

Row	time_division	f0_
1	Morning	27733
2	Dwan	5242
3	Afternoon	38135
4	Night	28331

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

- A) Get month on month orders by states

Ans: "select customer_state, extract(year from order_purchase_timestamp) as year,extract(month from order_purchase_timestamp) as month,count(order_id) as no_of_orders

from `scaler_project.customers` b left join `scaler_project.orders` c on b.customer_id=c.customer_id group by 1,2,3 order by 1,2,3"

Row	customer_state	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

- B) Distribution of customers across the states in Brazil

Ans: "select customer_state,count(customer_unique_id) as no_of_customers from `scaler_project.customers` group by 1"

Row	customer_state	no_of_customer
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747

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

- A) Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment_value" column in payments table

Ans: "
```
with cost_each_order as (select order_id,sum(payment_value) as cost_of_order from `scaler_project.payments` group by 1),

total_cost_year as (select extract(year from order_purchase_timestamp) as year,sum(cost_of_order) as total_cost
from `scaler_project.orders` a inner join cost_each_order b
 on a.order_id=b.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 1)
 select * ,round(((total_cost-(lag(total_cost) over(order by year)))/lag(total_cost) over(order by year))*100,2)
 from total_cost_year order by year"
```

Row	year	total_cost	f0_
1	2017	3669022.11...	*null*
2	2018	8694733.83...	136.98

- B) Mean & Sum of price and freight value by customer state

Ans: "
```
select customer_state,sum(price) as price_sum,sum(freight_value) as freight_sum,avg(price) as price_avg,avg(freight_value)as freight_avg

 from `scaler_project.order_items` a inner join `scaler_project.orders` b on a.order_id=b.order_id
 inner join `scaler_project.customers` c on b.customer_id=c.customer_id
group by 1"
```

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

Results per page: 50 1 – 27 of 27

5. Analysis on sales, freight and delivery time

- A) Calculate days between purchasing, delivering and estimated delivery

Ans: "
```
select
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
diff_purchase_delivered,

 date_diff(order_estimated_delivery_date,order_delivered_customer_date,day
) as diff_delivered_estimate,
date_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as
diff_purchase_estimate
 from `scaler_project.orders` order by 3 desc"
```

Row	diff_purchase_d	diff_delivered_e	diff_purchase_e
1	20	134	155
2	3	146	149
3	6	139	146
4	null	null	144
5	null	null	144
6	null	null	142
7	16	123	140
8	7	108	116
9	54	55	109
10	null	null	106

Results per page: 50 1 – 50 of 99441

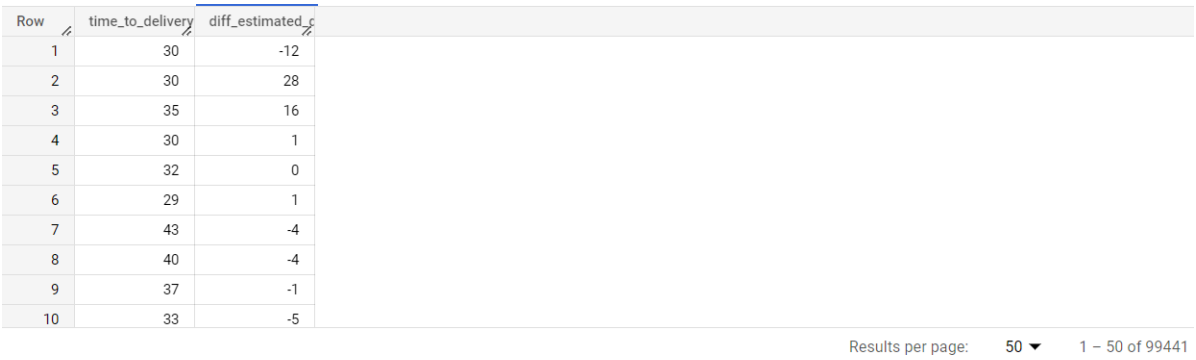
- B) Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

  time_to_delivery = order_delivered_customer_date-order_purchase_timestamp

  diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

Ans: "select
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,

date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
from `scaler_project.orders`"



Row	time_to_delivery	diff_estimated_c
1	30	-12
2	30	28
3	35	16
4	30	1
5	32	0
6	29	1
7	43	-4
8	40	-4
9	37	-1
10	33	-5

Results per page: 50 1 – 50 of 99441

- C) Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

Ans: "with orders_days_calc as (select *,date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,

date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimated_delivery
from `scaler_project.orders`)

select customer_state,avg(time_to_delivery) as avg_time_to_delivery, avg(diff_estimated_delivery) as avg_diff_estimated_delivery,avg(freight_value) as avg_freight_value
from `scaler_project.customers` a left join orders_days_calc b
on a.customer_id= b.customer_id left join `scaler_project.order_items` c
on c.order_id=b.order_id group by 1"

Row	customer_state	avg_time_to_deli	avg_diff_estimat	avg_freight_valu
1	RN	18.8733205...	13.0556621...	35.6523629...
2	CE	20.5371669...	10.2566619...	32.7142016...
3	RS	14.7082993...	13.2030001...	21.7358043...
4	SC	14.5209858...	10.6688628...	21.4703687...
5	SP	8.25960855...	10.2655943...	15.1472753...
6	MG	11.5155221...	12.3971510...	20.6301668...
7	BA	18.7746402...	10.1194678...	26.3639589...
8	RJ	14.6893821...	11.1444931...	20.9609239...
9	GO	14.9481774...	11.3728590...	22.7668152...
10	MA	21.2037499...	9.10999999...	38.2570024...

Results per page: 50 1 – 27 of 27

- D) Sort the data to get the following:

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

Ans: "

```
with answer as (

with orders_days_calc as (select
*,date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
as diff_estimated_delivery
from `scaler_project.orders`)

select customer_state,avg(time_to_delivery) as avg_time_to_delivery,
avg(diff_estimated_delivery) as
avg_diff_estimated_delivery,avg(freight_value) as avg_freight_value
from `scaler_project.customers` a left join orders_days_calc b
on a.customer_id= b.customer_id left join `scaler_project.order_items` c
on c.order_id=b.order_id group by 1)

select customer_state,avg_freight_value
from answer order by 2 desc limit 5"
```

Row	customer_state	avg_freight_valu
1	RR	42.9844230...
2	PB	42.7238039...
3	RO	41.0697122...
4	AC	40.0733695...
5	PI	39.1479704...

Insights:
Above pic shows the top 5 states in avg_frieght_value that is these 5 states are have high freight values which takes more cost for delivery compared to others.

Top 5 states with highest/lowest average time to delivery

Ans: "with answer as (

```
with orders_days_calc as (select
*,date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
as diff_estimated_delivery
from `scaler_project.orders`)

select customer_state,round(avg(time_to_delivery),2) as
avg_time_to_delivery,
round(avg(diff_estimated_delivery),2) as
avg_diff_estimated_delivery,avg(freight_value) as avg_freight_value
from `scaler_project.customers` a left join orders_days_calc b
on a.customer_id= b.customer_id left join `scaler_project.order_items` c
on c.order_id=b.order_id group by 1)

select customer_state,avg_time_to_delivery
from answer order by 2 limit 5"
```

Row	customer_state	avg_time_to_deli
1	SP	8.26
2	PR	11.48
3	MG	11.52
4	DF	12.5
5	SC	14.52

Insights:
Above pic shows the lowest avg_time_delivery 5 states which means customers in this state getting orders delivered fast.

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

Ans: "with answer as (

```
with orders_days_calc as (select
*,date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_to_delivery,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)
as diff_estimated_delivery
from `scaler_project.orders`)
```

```
select customer_state,round(avg(time_to_delivery),2) as
avg_time_to_delivery,
round(avg(diff_estimated_delivery),2) as
avg_diff_estimated_delivery,avg(freight_value) as avg_freight_value
from `scaler_project.customers` a left join orders_days_calc b
on a.customer_id= b.customer_id left join `scaler_project.order_items` c
on c.order_id=b.order_id group by 1)

select customer_state,avg_diff_estimated_delivery
from answer order by 2 limit 5"
```

Row	customer_state	avg_diff_estimat
1	AL	7.98
2	MA	9.11
3	SE	9.17
4	ES	9.77
5	BA	10.12

Insights:
States in above pic are the top 5 states which are fast in delivery

6. Payment type analysis:

- A) Month over Month count of orders for different payment types

Ans: "select extract(year from order_purchase_timestamp) as year,

```
extract(month from order_purchase_timestamp) as month,
payment_type,count(a.order_id) as no_of_orders
from `scaler_project.payments` a inner join `scaler_project.orders` b
on a.order_id=b.order_id group by 1,2,3 order by 1,2,3"
```

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

Results per page: 50 1 – 50 of 90

- B) Count of orders based on the no. of payment installments

Ans: "`select payment_installments,count(order_id) as no_of_orders from `scaler_project.payments` group by 1`"

Row	payment_installr	no_of_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

Insights :

- 1)Customers who have chosen more than 1 instalment method used only credit card for payment.

Query : "`select distinct payment_type from `scaler_project.payments` where payment_installments>1`"

Row	payment_type
1	credit_card

- 2)Max number of products were sold from the sellers in "SP" state(71%).

Query : "
```
select seller_state,(products_count/sum(products_count)  over())*100 as pct_sold from

(select seller_state,count(a.seller_id) as no_of_sellers,sum(no_of_pro) as products_count from
(select seller_id,count(product_id) as no_of_pro,sum(price) as total_cost from `scaler_project.order_items`
group by 1) a inner join `scaler_project.sellers` b on a.seller_id=b.seller_id group by 1) order by 2 desc"
```

Row	seller_state	pct_sold
1	SP	71.32001775410...
2	MG	7.835774522858...
3	PR	7.697292498890...
4	RJ	4.276964047936...
5	SC	3.617399023524...
6	RS	1.952063914780...
7	DF	0.798047048379...
8	BA	0.570794496227...
9	GO	0.461606746560...
10	PE	0.397691966267...

- 3)Product category with highest number of sales was "Bed table bath"(10%) .

Query : "
```
select  *,(no_of_products/sum(no_of_products)  over())*100  as pct_sales,(cost/sum(cost) over())*100 as pct_income from (

select product_category,count(c.product_id) as no_of_products,sum(price) as cost
from `scaler_project.order_items` c inner join `scaler_project.products` d on c.product_id=d.product_id where product_category is not null group by 1) order by 4 desc"
```

Row	product_category	no_of_products	cost	pct_sales	pct_income
1	bed table bath	11115	1036988.680000...	10.00927535187...	7.731734992939...
2	HEALTH BEAUTY	9670	1258681.339999...	8.7080245301539	9.384664219705...
3	sport leisure	8641	988048.9700000...	7.781389861950...	7.366842997829...
4	Furniture Decoration	8334	729762.4900000...	7.504930344808...	5.441072105500...
5	computer accessories	7827	911954.3200000...	7.048366907705...	6.799485147615...
6	housewares	6964	632248.6600000...	6.271218493070...	4.714013935774...
7	Watches present	5991	1205005.679999...	5.395012922456...	8.984461221645...
8	telephony	4545	323667.5299999...	4.092861581132...	2.413248684429...
9	Garden tools	4347	485256.4600000...	3.914558700370...	3.618047549305...
10	automotive	4235	592720.1100000...	3.813700505191...	4.419291072208...

Results per page: 50 1 – 50 of 73

- 4)"Health beauty"(9.4%) and "Watches present"(9%) product categories were the major contributors in income generation.

Query : "select *,(no_of_products/sum(no_of_products) over())*100 as pct_sales,(cost/sum(cost) over())*100 as pct_income from (

```
select product_category,count(c.product_id) as no_of_products,sum(price)
as cost
from `scaler_project.order_items` c inner join `scaler_project.products` d
on c.product_id=d.product_id where product_category is not null group by
1) order by 5 desc"
```

Row	product_category	no_of_products	cost	pct_sales	pct_income
1	HEALTH BEAUTY	9670	1258681.339999...	8.7080245301539	9.384664219705...
2	Watches present	5991	1205005.679999...	5.395012922456...	8.984461221645...
3	bed table bath	11115	1036988.680000...	10.00927535187...	7.731734992939...
4	sport leisure	8641	988048.9700000...	7.781389861950...	7.366842997829...
5	computer accessories	7827	911954.3200000...	7.048366907705...	6.799485147615...
6	Furniture Decoration	8334	729762.4900000...	7.504930344808...	5.441072105500...
7	Cool Stuff	3796	635290.8500000...	3.418372400875...	4.736696350088...
8	housewares	6964	632248.6600000...	6.271218493070...	4.714013935774...
9	automotive	4235	592720.1100000...	3.813700505191...	4.419291072208...
10	Garden tools	4347	485256.4600000...	3.914558700370...	3.618047549305...

Results per page: 50 1 – 50 of 73

- 5)More than three quarters of the orders got review score of 4(19.3%) and 5(57.8%)

Query : "select review_score,(no_of_orders/(sum(no_of_orders) over()))*100 from (

```
select review_score,count(order_id) as no_of_orders from
`scaler_project.order_reviews` group by 1) order by 1 desc"
```

Row	review_score	f0_
1	5	57.77634443279...
2	4	19.29170362009...
3	3	8.242965411593...
4	2	3.175642989599...
5	1	11.51334354591...

Recommendations :

- Based on insight-1 credit card was the only method used for payments with more than 1 instalments because this could be the only option allowed by the store. Payments through other modes like debit card can be allowed on the basis of certain eligibility criteria so that some more customers may add up.
- Based on insights-3,4 "watches present" product category was contributing to 9% of income generation with 5% of sales. Focusing on slight increase in "watches present" sales will generate relatively more income compared to other product categories.