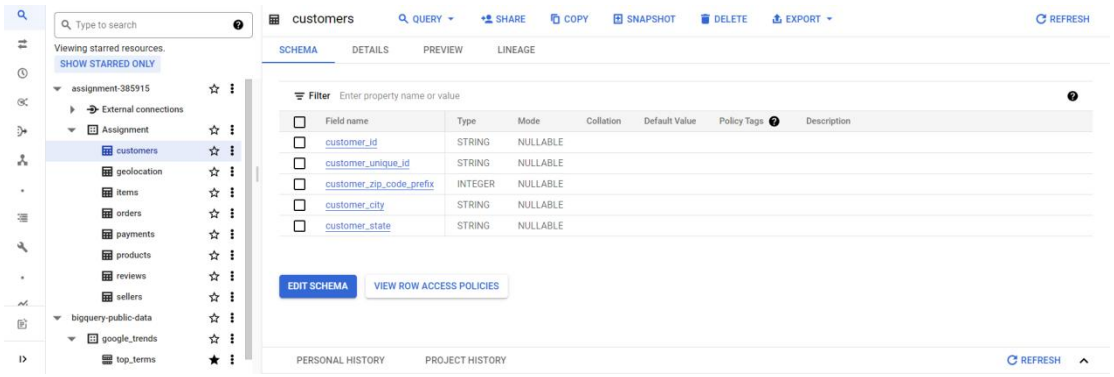
SQL Target BUSINESS CASE

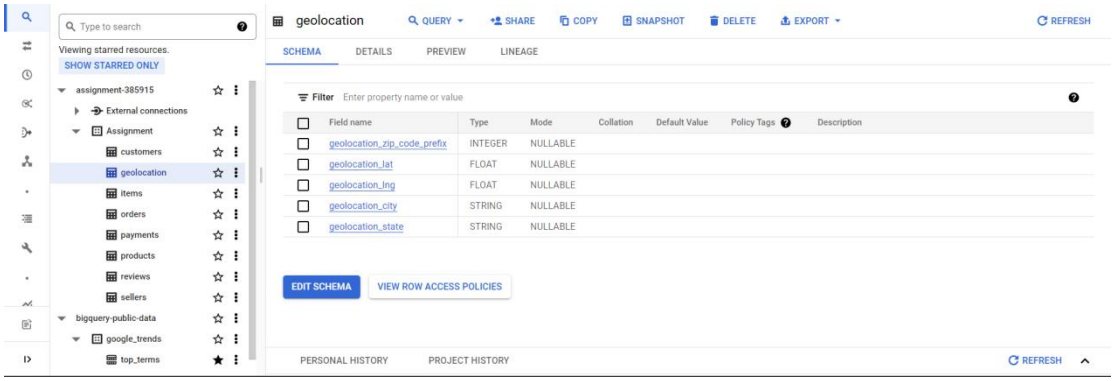
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset.

    1.1) Data type of columns in a table.

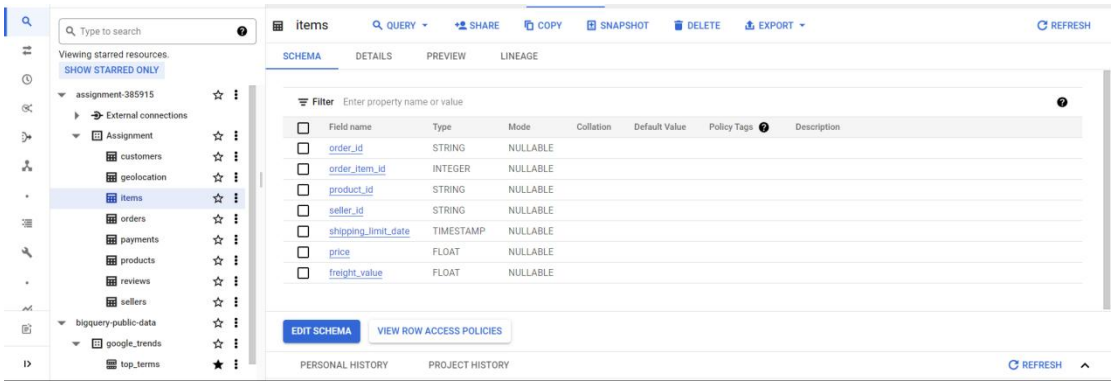
CUSTOMERS TABLE



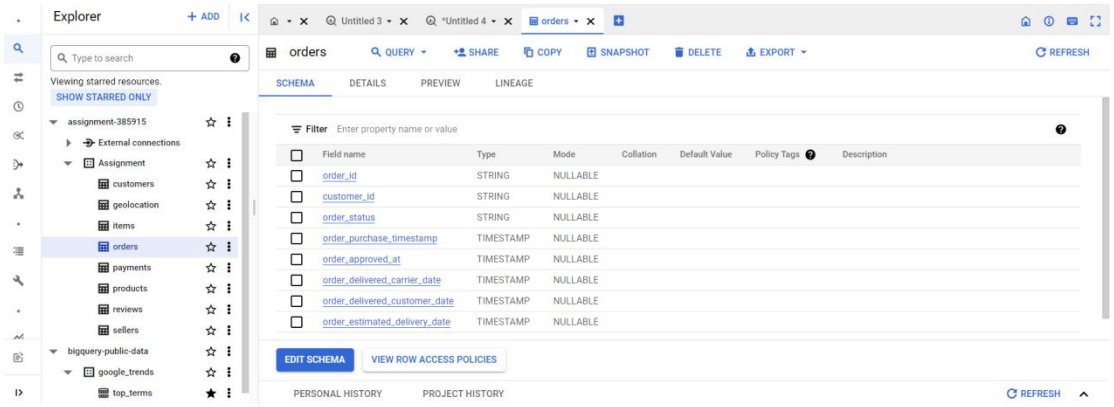
GEOLOCATION TABLE



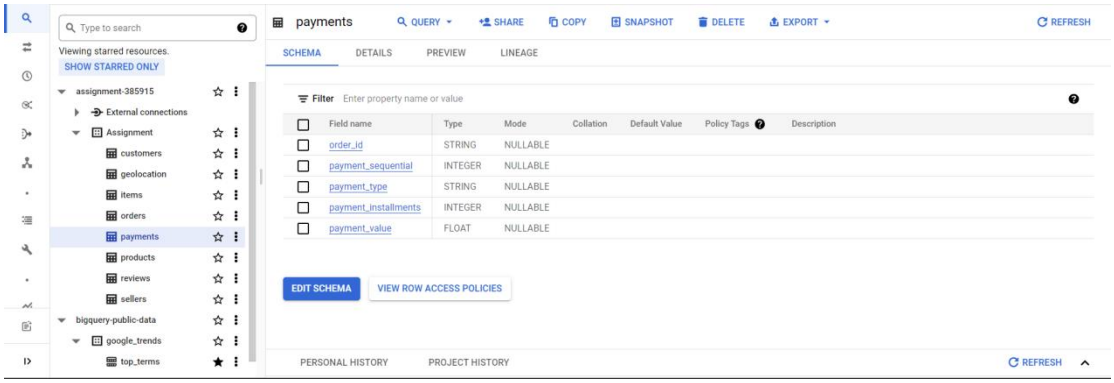
ITEMS TABLE



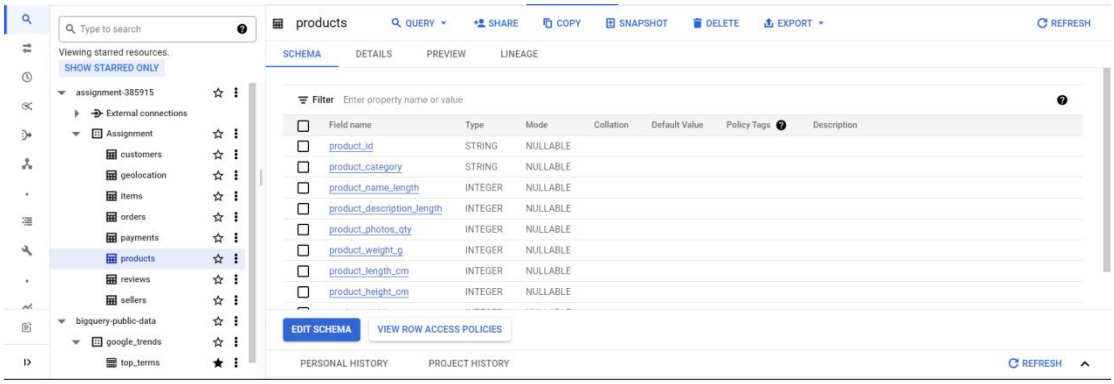
ORDERS TABLE



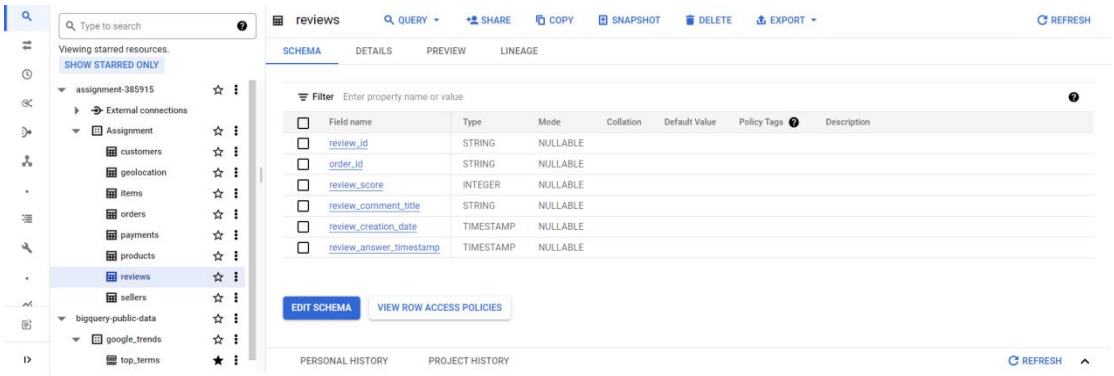
PAYMENTS TABLE



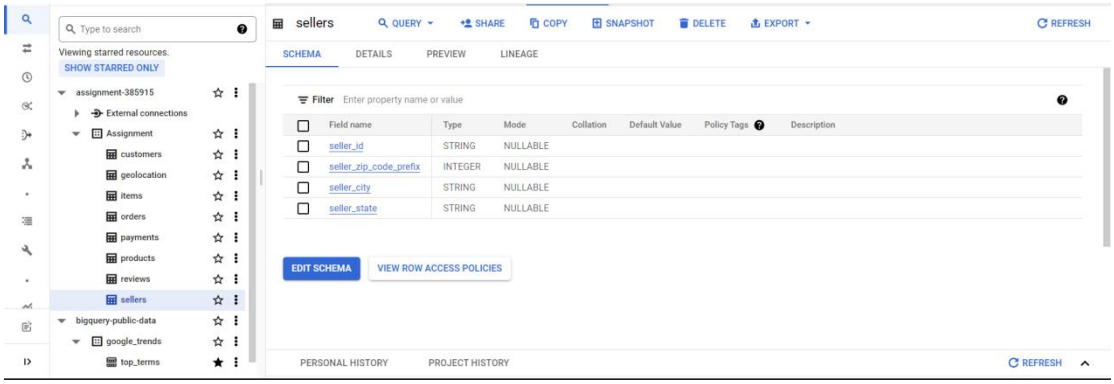
PRODUCTS TABLE



REVIEWS TABLE



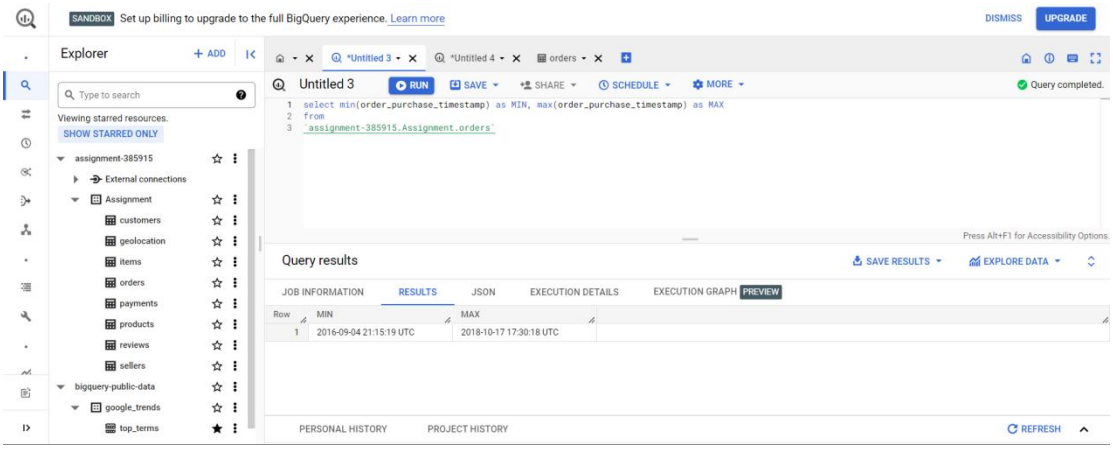
SELLERS TABLE



Actionable Insights - Here we can see that every column in the database have a name and a data type. The data type of a column defines what kind of data can be stored in that column.The data type is a guideline for SQL to understand what type of data is expected in each column.It also identifies how SQL will interact with the stored data.

1.2) Time period for which the data is given.

Ans - select min(order_purchase_timestamp) MIN , max(order_purchase_timestamp) MAX
from
`assignment-385915.Assignment.orders`

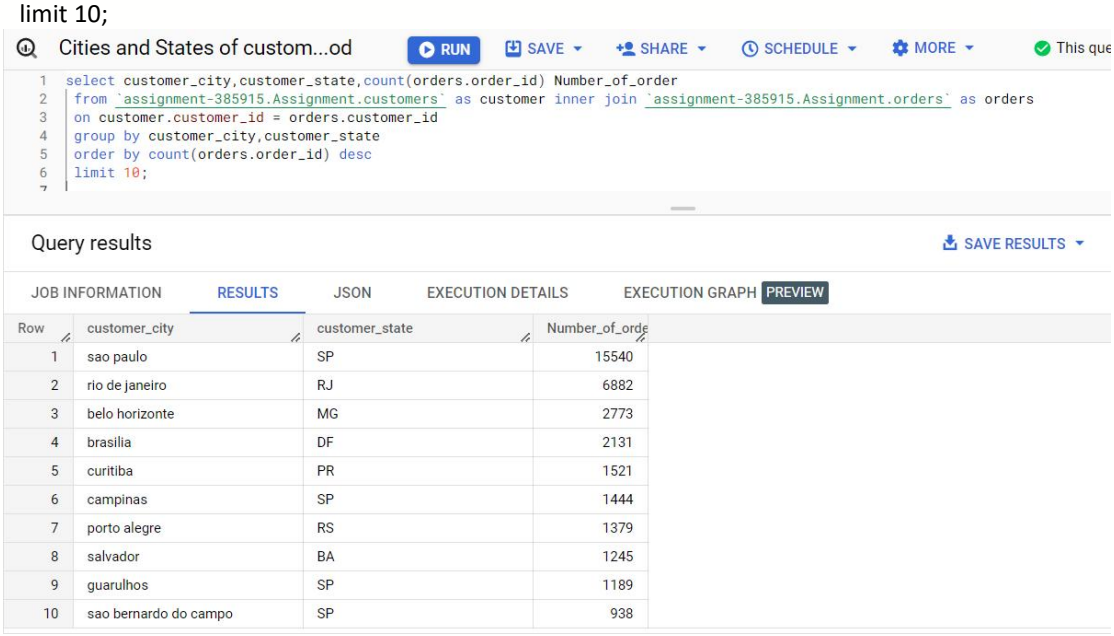


Actionable Insights - The SQL query above retrieves the earliest and latest order purchase timestamps from the orders table in the Assignment dataset.

This information can be used to determine the time frame of the available order data and to identify any potential data gaps or inconsistencies. It can also be used to track the growth or decline of orders over time and to analyze trends in customer behavior.

1.3) Cities and States of customers ordered during the given period.

Ans - select customer_city,customer_state,count(orders.order_id) Number_of_order
from `assignment-385915.Assignment.customers` as customer
inner join `assignment-385915.Assignment.orders` as orders
On customer.customer_id = orders.customer_id
group by customer_city,customer_state
Order by count(orders.order_id) desc
limit 10;



Actionable Insights - The SQL query above retrieves the top 10 cities and states with the highest number of orders made by customers.

It does this by joining the customers and orders tables on the customer_id column and grouping the results by customer_city and customer_state.

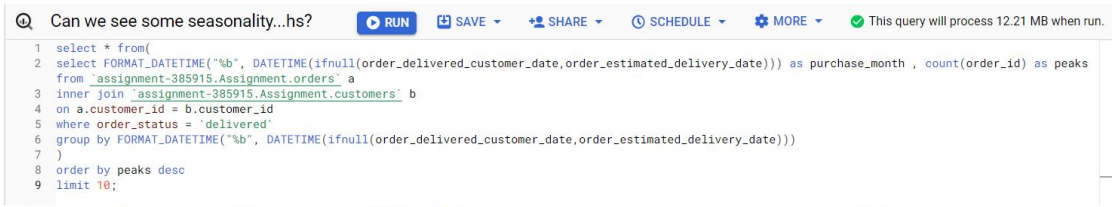
The count() function is then used to count the number of orders made by each customer in each city and state. The results are sorted in descending order based on the number of orders and limited to the top 10.

This information can be used to identify the most popular locations for the business and to tailor marketing strategies to target those areas. It can also be used to optimize inventory and supply chain management to ensure that the business is meeting the demand in those locations.

2) In-depth Exploration:

2.1) 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 * from(
select FORMAT_DATETIME("%b", DATETIME(ifnull(order_delivered_customer_date,order_estimated_delivery_ date))) as purchase_month , count(order_id) as peaks from `assignment-385915.Assignment.orders` a
inner join `assignment-385915.Assignment.customers` b
on a.customer_id = b.customer_id
where order_status = 'delivered'
group by FORMAT_DATETIME("%b", DATETIME(ifnull(order_delivered_customer_date,order_estimated_delivery_ date)))
)
order by peaks desc
limit 10;



JOB INFORMATION | RESULTS | JSON | EXECUTION DI

Row	purchase_month	peaks
1	Aug	12616
2	May	10862
3	Jun	10054
4	Apr	9699
5	Jul	9299
6	Mar	9206
7	Dec	7210
8	Feb	7201
9	Jan	6880
10	Nov	4728

Actionable Insights - The SQL query above retrieves the top 10 months with the highest number of delivered orders.It does this by joining the 'orders' and 'customers' tables and grouping the results by the month of the order delivery or estimated delivery date. The 'ifnull' function is used to handle cases where the actual delivery date is not available and the estimated delivery date is used instead.

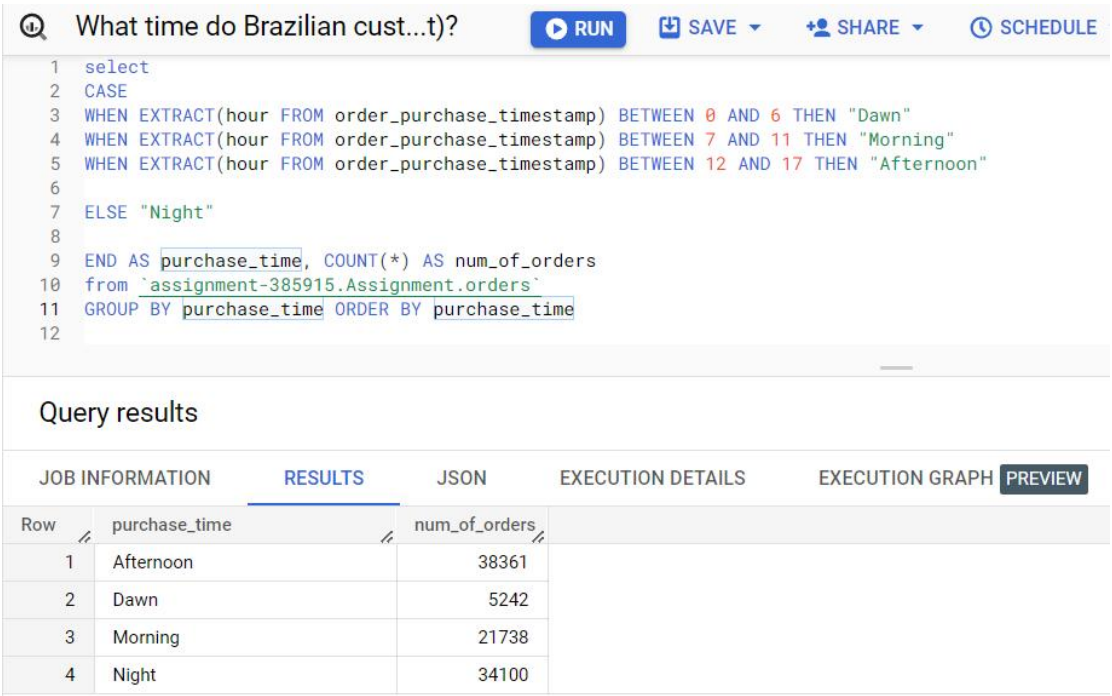
The results are then sorted in descending order by the number of delivered orders and limited to the top 10.

This information can be useful for identifying peak periods of order deliveries and planning for staffing, inventory, and logistics accordingly. It can also help to identify trends in customer behavior and preferences. Also we can see at the month of May,June and August are the peak periods where target can focus to increase their sales.

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

Ans -

```
select
CASE
WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 0 AND 6 THEN "Dawn"
  WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 7 AND 11 THEN
  "Morning"
  WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 12 AND 17 THEN
"Afternoon"
ELSE "Night"
END AS purchase_time, COUNT(*) AS num_of_orders
from `assignment-385915.Assignment.orders`
GROUP BY purchase_time ORDER BY purchase_time
```



Actionable Insights - The SQL query above groups the orders from a table named "orders" based on the time of day they were purchased.

The query uses the EXTRACT function to extract the hour from the "order_purchase_timestamp" column and categorizes the orders into four groups: Dawn, Morning, Afternoon, and Night.

It then counts the number of orders in each group and returns the result in ascending order based on the time of day.

This information can be used to analyze customer behavior and preferences for purchasing products at different times of the day. It can also be used to optimize marketing strategies and promotions to target customers during specific times of the day.

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

3.1) Get month on month orders by states

Ans - SELECT
customer.customer_state,extract(month from orders.order_purchase_timestamp) MONTHS,
COUNT(orders.order_id) AS num_of_orders
FROM `assignment-385915.Assignment.customers` AS customer
JOIN `assignment-385915.Assignment.orders` AS orders
ON customer.customer_id = orders.customer_id
GROUP BY customer.customer_state,MONTHS
ORDER BY customer.customer_state,MONTHS
limit 10;

Get month on month orders b...es. RUN SAVE SHARE

```
SELECT
customer.customer_state,extract(month from orders.order_purchase_timestamp) MONTHS,
COUNT(orders.order_id) AS num_of_orders
FROM `assignment-385915.Assignment.customers` AS customer
JOIN `assignment-385915.Assignment.orders` AS orders
ON customer.customer_id = orders.customer_id
GROUP BY customer.customer_state,MONTHS
ORDER BY customer.customer_state,MONTHS
limit 10;
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	MONTHS	num_of_orders
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6

Actionable Insights - The SQL query above retrieves the number of orders made by customers in each state of a particular country, grouped by month.

This information can be used to identify the states with the highest and lowest order volumes, as well as the months with the highest and lowest order volumes. This data can help businesses to optimize their inventory management, logistics, and marketing strategies.

For example, if a state has a high order volume in a particular month, a business can increase its inventory levels and marketing efforts in that state during that month to maximize sales. Conversely, if a state has a low order volume in a particular month, a business can reduce its inventory levels and marketing efforts in that state during that month to minimize costs.

3.2) Distribution of customers across the states in Brazil

Ans - SELECT
customer_state,
COUNT(customer_id) AS number_customers,
ROUND(COUNT(customer_id) * 100.0 / (SELECT COUNT(*) FROM `assignment-385915.Assignment.customers`), 2) AS percentage_customers
FROM `assignment-385915.Assignment.customers`
GROUP BY customer_state
ORDER BY 2 DESC
limit 10;

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	number_custom	percentage_cust
1	SP	41746	41.98
2	RJ	12852	12.92
3	MG	11635	11.7
4	RS	5466	5.5
5	PR	5045	5.07
6	SC	3637	3.66
7	BA	3380	3.4
8	DF	2140	2.15
9	ES	2033	2.04
10	GO	2020	2.03

Actionable insights - The SQL query above retrieves the number and percentage of customers from each state in a given dataset.

The query groups the customers by their state and counts the number of customers in each state. It then calculates the percentage of customers in each state by dividing the number of customers in that state by the total number of customers in the dataset and multiplying by 100.

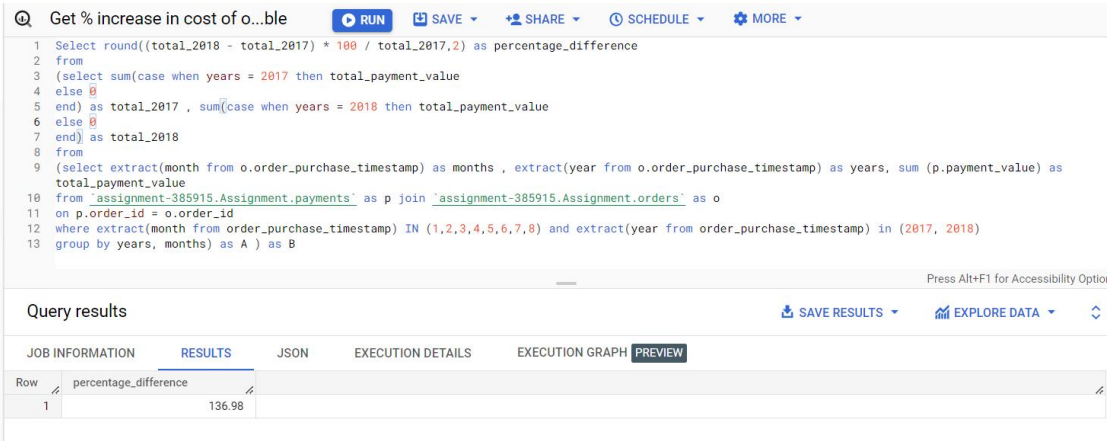
The results are ordered in descending order by the number of customers and limited to the top 10 states.

This information can be useful for businesses to understand their customer base and identify areas where they may want to focus their marketing efforts or improve their services. It can also help them to identify potential opportunities for growth in certain regions.

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

4.1) 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 -Select round((total_2018 - total_2017) * 100 / total_2017,2) as percentage_difference
from
(select sum(case when years = 2017 then total_payment_value
else 0
end) as total_2017 , sum(case when years = 2018 then total_payment_value
else 0
end) as total_2018
from
(select extract(month from o.order_purchase_timestamp) as months , extract(year from o.order_purchase_timestamp) as years, sum (p.payment_value) as total_payment_value
from `assignment-385915.Assignment.payments` as p join `assignment-385915.Assignment.orders` as o
on p.order_id = o.order_id
where extract(month from order_purchase_timestamp) IN (1,2,3,4,5,6,7,8) and extract(year from order_purchase_timestamp) in (2017, 2018)
group by years, months) as A) as B



Actionable Insights - The SQL query above calculates the percentage difference between the total payment value for the first eight months of 2018 and the same period in 2017.

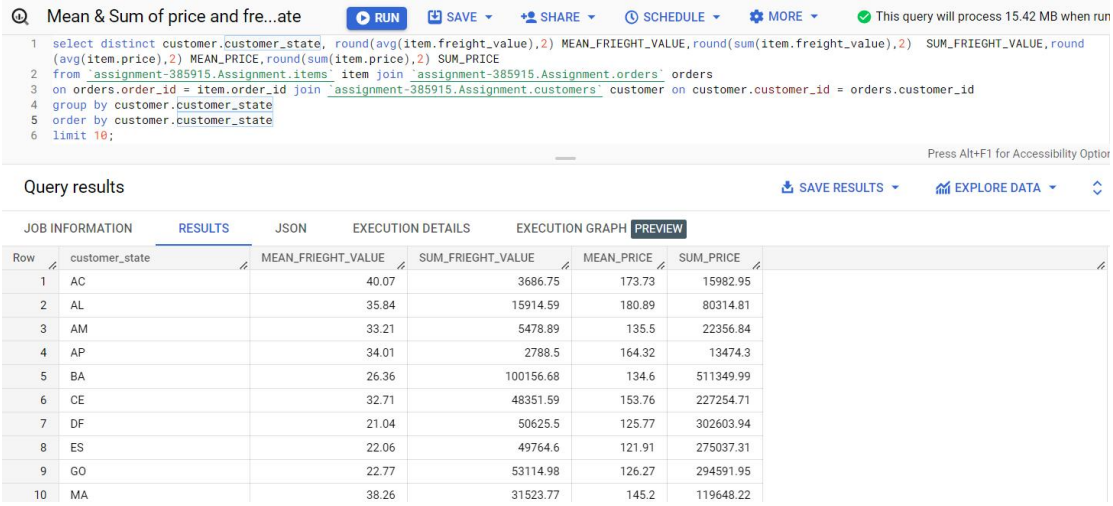
It does this by first calculating the total payment value for each year (2017 and 2018) and then subtracting the total payment value for 2017 from the total payment value for 2018. The result is then divided by the total payment value for 2017 and multiplied by 100 to get the percentage difference.

This information can be useful for businesses to track their revenue growth and make informed decisions about their future strategies. It can also help identify trends and patterns in customer behavior and preferences.

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

Ans -

```
select distinct customer.customer_state, round(avg(item.freight_value),2) MEAN_FRIEGHT_VALUE,ro
und(sum(item.freight_value),2)  SUM_FRIEGHT_VALUE,round(avg(item.price),2) MEAN_PRICE,round(
sum(item.price),2) SUM_PRICE
from `assignment-385915.Assignment.items` item join `assignment-
385915.Assignment.orders` orders
on orders.order_id = item.order_id join `assignment-
385915.Assignment.customers` customer on customer.customer_id = orders.customer_id
group by customer.customer_state
order by customer.customer_state
limit 10;
```



Row	customer_state	MEAN_FRIEGHT_VALUE	SUM_FRIEGHT_VALUE	MEAN_PRICE	SUM_PRICE
1	AC	40.07	3686.75	173.73	15982.95
2	AL	35.84	15914.59	180.89	80314.81
3	AM	33.21	5478.89	135.5	22356.84
4	AP	34.01	2788.5	164.32	13474.3
5	BA	26.36	100156.68	134.6	511349.99
6	CE	32.71	48351.59	153.76	227254.71
7	DF	21.04	50625.5	125.77	302603.94
8	ES	22.06	49764.6	121.91	275037.31
9	GO	22.77	53114.98	126.27	294591.95
10	MA	38.26	31523.77	145.2	119648.22

Actionable Insights - The SQL query above calculates the average and total freight value and price of items purchased by customers from each state.

This information can be used to analyze the purchasing behavior of customers from different states and identify patterns or trends in terms of the amount spent on freight and items.

For example, if customers from a particular state tend to spend more on freight, it may be beneficial to offer free shipping or discounted rates to incentivize more purchases from that state. Similarly, if customers from a particular state tend to purchase more expensive items, it may be beneficial to offer more high-end products to cater to that market.

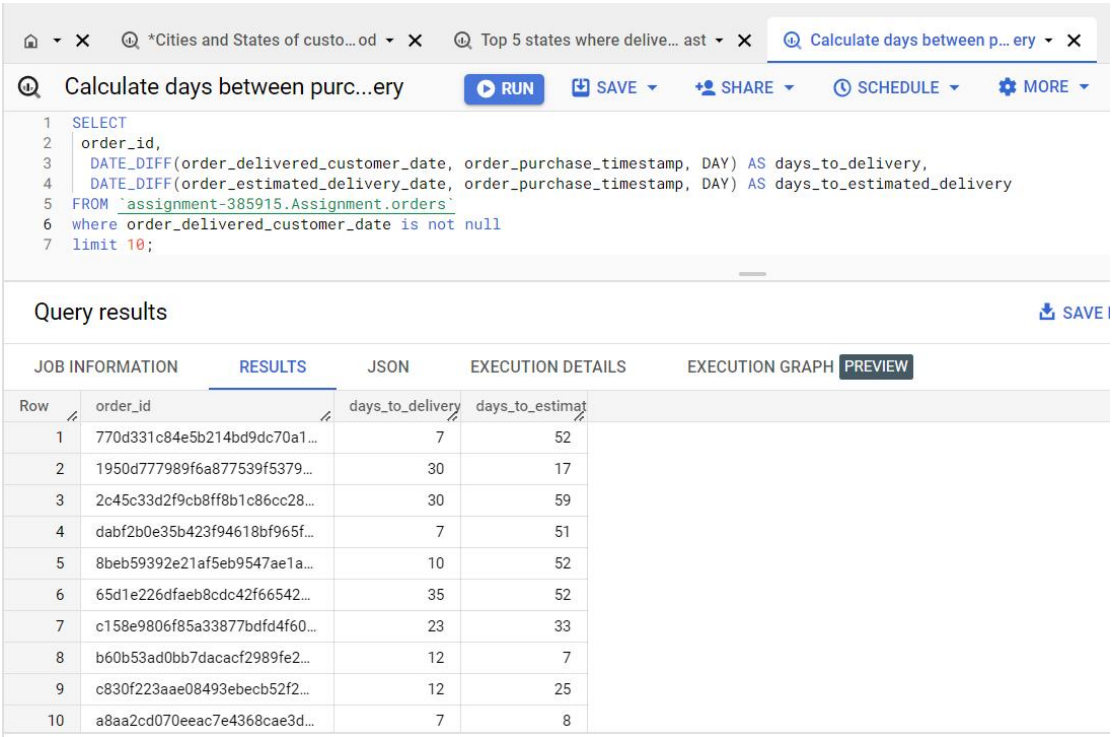
Overall, this data can be used to make data-driven decisions to optimize revenue streams and improve customer satisfaction.

5) Analysis on sales, freight and delivery time

5.1) Calculate days between purchasing, delivering and estimated delivery

Ans -

```
SELECT
 order_id,
 DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS days_to_delivery,
 DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY) AS days_to_estimate
d_delivery
FROM `assignment-385915.Assignment.orders`
where order_delivered_customer_date is not null
limit 10;
```



Row	order_id	days_to_delivery	days_to_estimat
1	770d331c84e5b214bd9dc70a1...	7	52
2	1950d777989f6a877539f5379...	30	17
3	2c45c33d2f9cb8ff8b1c86cc28...	30	59
4	dabf2b0e35b423f94618bf965f...	7	51
5	8beb59392e21af5eb9547ae1a...	10	52
6	65d1e226dfaeb8cdc42f66542...	35	52
7	c158e9806f85a33877bdfd4f60...	23	33
8	b60b53ad0bb7dacacf2989fe2...	12	7
9	c830f223aae08493ebecb52f2...	12	25
10	a8aa2cd070eeac7e4368cae3d...	7	8

Actionable Insights - The SQL query above retrieves the number of days it took for an order to be delivered to the customer and the number of days it was estimated to take for the order to be delivered.

The query filters out orders that have not been delivered yet by checking if the order_delivered_customer_date is not null.

This data can be used to analyze the efficiency of the delivery process and identify any delays or issues in the supply chain. It can also be used to improve customer satisfaction by providing accurate delivery estimates and reducing the time it takes for orders to be delivered.

5.2) 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
order_id,
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 `assignment-385915.Assignment.orders`
where order_delivered_customer_date is not null
limit 10;

Find time_to_delivery & dif...ery RUN SAVE SHARE SCHEDULE MORE

```
SELECT
  order_id,
  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 `assignment-385915.Assignment.orders`
where order_delivered_customer_date is not null
limit 10;
```

Query results SAV

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

Row	order_id	time_to_delivery	diff_estimated_c
1	770d331c84e5b214bd9dc70a1...	7	45
2	1950d777989f6a877539f5379...	30	-12
3	2c45c33d2f9cb8ff8b1c86cc28...	30	28
4	dabf2b0e35b423f94618bf965f...	7	44
5	8beb59392e21af5eb9547ae1a...	10	41
6	65d1e226dfaeb8cdc42f66542...	35	16
7	c158e9806f85a33877bdfd4f60...	23	9
8	b60b53ad0bb7dacacf2989fe2...	12	-5
9	c830f223aae08493ebecb52f2...	12	12
10	a8aa2cd070eeac7e4368cae3d...	7	1

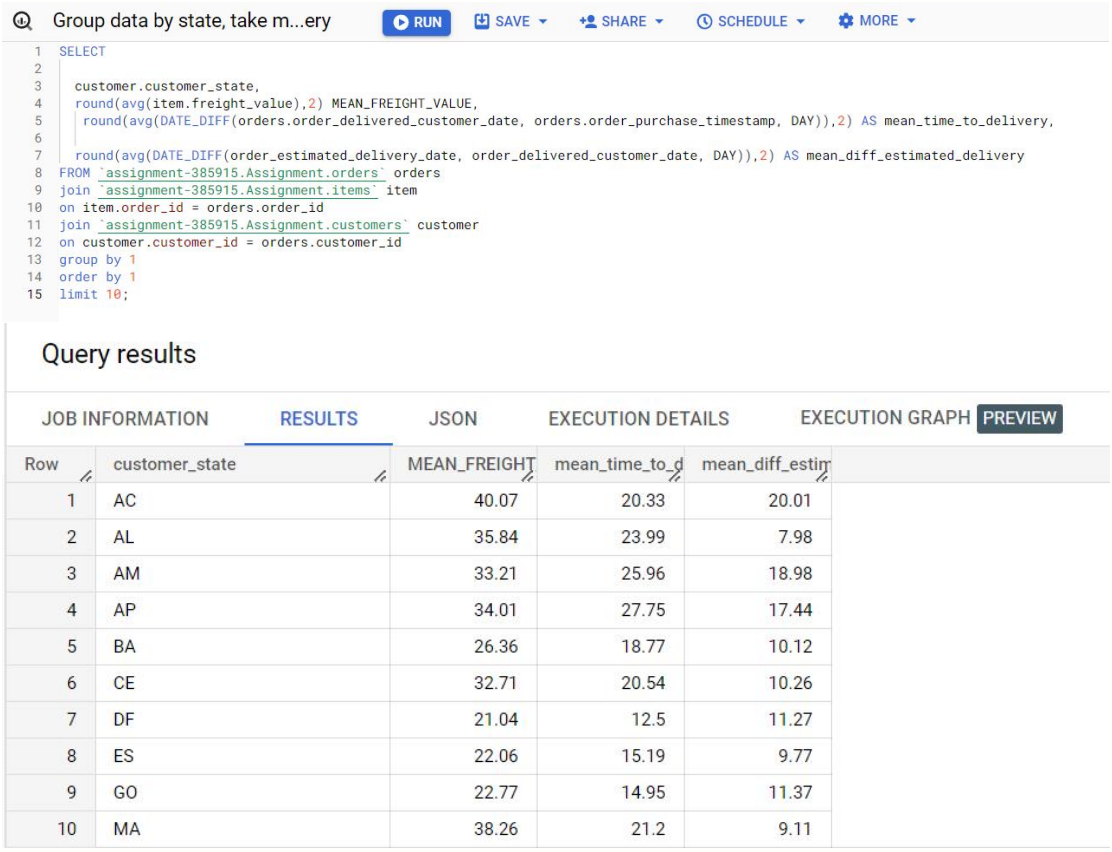
Actionable Insights - The SQL query above calculates the time it took for an order to be delivered to a customer and the difference between the estimated delivery date and the actual delivery date for the first 10 orders in the orders table.

This information can be used to track the performance of the delivery system and identify areas for improvement. It can also be used to improve customer satisfaction by providing accurate delivery estimates and reducing the time it takes for an order to be delivered. Additionally, this data can be used to identify any potential issues in the supply chain or delivery process that may be causing delays.

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

Ans - SELECT
customer.customer_state,
round(avg(item.freight_value),2) MEAN_FREIGHT_VALUE,
round(avg(DATE_DIFF(orders.order_delivered_customer_date, orders.order_purchase_timestamp, DAY)),2) AS mean_time_to_delivery,

round(avg(DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)),2) AS mean_diff_estimated_delivery
FROM `assignment-385915.Assignment.orders` orders
join `assignment-385915.Assignment.items` item
on item.order_id = orders.order_id
join `assignment-385915.Assignment.customers` customer
on customer.customer_id = orders.customer_id
group by 1
order by 1
limit 10;



Query results

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

Row	customer_state	MEAN_FREIGHT	mean_time_to_d	mean_diff_estim
1	AC	40.07	20.33	20.01
2	AL	35.84	23.99	7.98
3	AM	33.21	25.96	18.98
4	AP	34.01	27.75	17.44
5	BA	26.36	18.77	10.12
6	CE	32.71	20.54	10.26
7	DF	21.04	12.5	11.27
8	ES	22.06	15.19	9.77
9	GO	22.77	14.95	11.37
10	MA	38.26	21.2	9.11

Actionable Insights - The SQL query above calculates the average freight value, mean time to delivery, and mean difference between estimated delivery date and actual delivery date for each customer state.

This data can be used to identify which states have the highest and lowest average freight values, as well as which states have the longest and shortest mean time to delivery. This information can be used to optimize logistics and delivery processes, as well as to improve customer satisfaction by providing more accurate delivery estimates.

Additionally, this data can be used to identify potential issues with delivery in certain states and take corrective action to improve the overall customer experience.

5.4) Sort the data to get the following:

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

Ans - WITH state_avg_freight_value AS
(SELECT customer.customer_state, round(avg(item.freight_value),2) AS avg_freight_value,

FROM `assignment-385915.Assignment.orders` AS orders JOIN `assignment-385915.Assignment.items` AS item

ON orders.order_id = item.order_id

JOIN `assignment-385915.Assignment.customers` AS customer

ON orders.customer_id = customer.customer_id
GROUP BY customer.customer_state)
(SELECT "Top 5 states with Highest average freight value" AS title, customer_state, avg_freight_value

FROM state_avg_freight_value

ORDER BY avg_freight_value DESC LIMIT 5)

UNION ALL

(SELECT "Top 5 states with lowest average freight value" AS title, customer_state, avg_freight_value
FROM state_avg_freight_value ORDER BY avg_freight_value ASC LIMIT 5)

Query results

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

Row	title	customer_state	avg_freight_valu
1	Top 5 states with lowest average freight value	SP	15.15
2	Top 5 states with lowest average freight value	PR	20.53
3	Top 5 states with lowest average freight value	MG	20.63
4	Top 5 states with lowest average freight value	RJ	20.96
5	Top 5 states with lowest average freight value	DF	21.04
6	Top 5 states with Highest average freight value	RR	42.98
7	Top 5 states with Highest average freight value	PB	42.72
8	Top 5 states with Highest average freight value	RO	41.07
9	Top 5 states with Highest average freight value	AC	40.07
10	Top 5 states with Highest average freight value	PI	39.15

Actionable insights - The SQL query above calculates the average freight value for each state based on orders made by customers, and then retrieves the top 5 states with the highest average freight value and the top 5 states with the lowest average freight value.

This information can be used by businesses to identify which states have higher or lower shipping costs and adjust their pricing strategies accordingly. It can also help identify potential issues with shipping providers or delivery routes that may be causing higher freight costs in certain areas. Additionally, this data can be used to optimize logistics and supply chain operations by identifying areas where shipping costs can be reduced.

5.6) Top 5 states with highest/lowest average time to delivery

Ans - WITH state_avg_time_to_delivery AS
(SELECT customer.customer_state, ROUND(AVG(DATE_DIFF(orders.order_delivered_customer_date, orders.order_purchase_timestamp, DAY)), 2) AS avg_time_to_delivery,

FROM `assignment-385915.Assignment.orders` AS orders JOIN `assignment-385915.Assignment.items` AS item

ON orders.order_id = item.order_id

JOIN `assignment-385915.Assignment.customers` AS customer

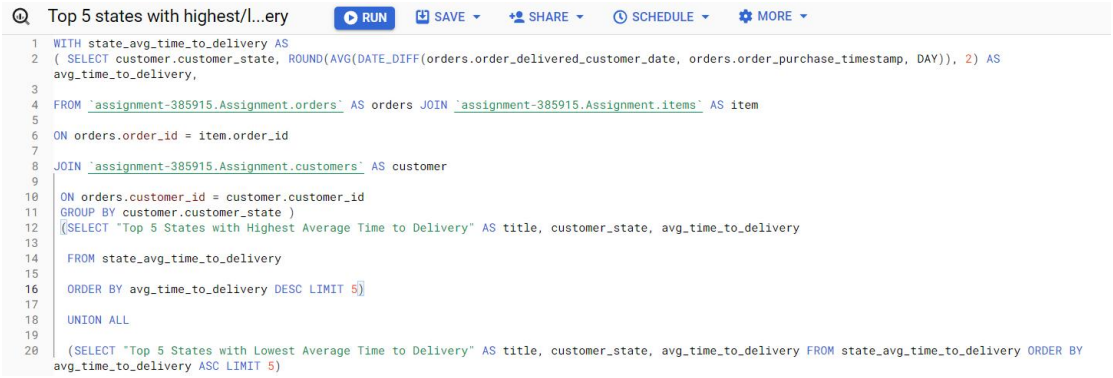
ON orders.customer_id = customer.customer_id
GROUP BY customer.customer_state)
(SELECT "Top 5 States with Highest Average Time to Delivery" AS title, customer_state, avg_time_to_delivery

FROM state_avg_time_to_delivery

ORDER BY avg_time_to_delivery DESC LIMIT 5)

UNION ALL

(SELECT "Top 5 States with Lowest Average Time to Delivery" AS title, customer_state, avg_time_to_delivery FROM state_avg_time_to_delivery ORDER BY avg_time_to_delivery ASC LIMIT 5)



Query results

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

Row	title	customer_state	avg_time_to_del
1	Top 5 States with Lowest Average Time to Delivery	SP	8.26
2	Top 5 States with Lowest Average Time to Delivery	PR	11.48
3	Top 5 States with Lowest Average Time to Delivery	MG	11.52
4	Top 5 States with Lowest Average Time to Delivery	DF	12.5
5	Top 5 States with Lowest Average Time to Delivery	SC	14.52
6	Top 5 States with Highest Average Time to Delivery	RR	27.83
7	Top 5 States with Highest Average Time to Delivery	AP	27.75
8	Top 5 States with Highest Average Time to Delivery	AM	25.96
9	Top 5 States with Highest Average Time to Delivery	AL	23.99
10	Top 5 States with Highest Average Time to Delivery	PA	23.3

Actionable Insights - This SQL query calculates the average time to delivery for orders in each state and then returns the top 5 states with the highest average time to delivery and the top 5 states with the lowest average time to delivery.

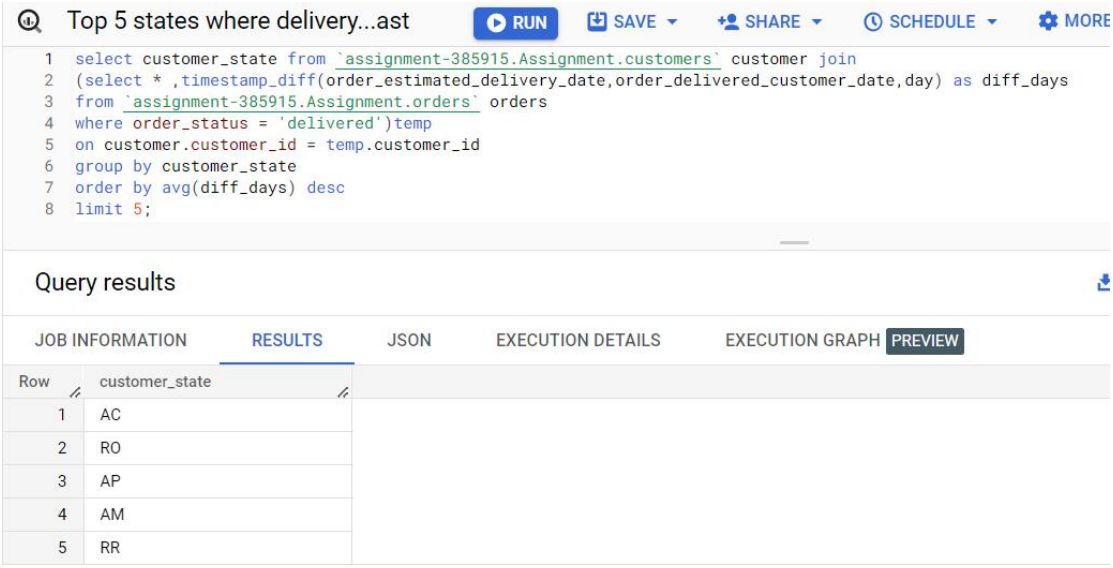
The data can be used to identify states where delivery times are longer than average and take corrective measures to improve delivery times. It can also be used to identify states where delivery times are shorter than average and use this information to promote the business in those states.

Overall, this query can help businesses optimize their delivery processes and improve customer satisfaction.

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

Ans - Top 5 states where delivery not so fast compared to estimated date

```
select customer_state,round(avg(diff_days),2) as diff_days_count from `assignment-385915.Assignment.customers` customer join
(select * ,timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_days
from `assignment-385915.Assignment.orders` orders
where order_status = 'delivered')temp
on customer.customer_id = temp.customer_id
group by customer_state
order by avg(diff_days) desc
limit 5;
```

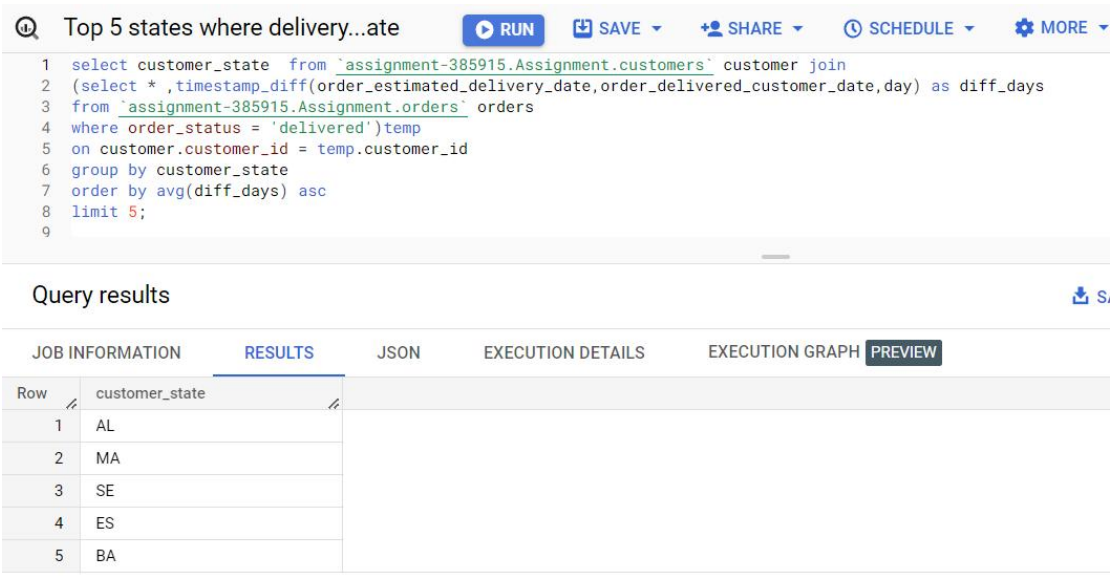


Actionable Insights - The SQL query above retrieves customer information and order details for orders that have been delivered. It then calculates the average number of days it takes for orders to be delivered to customers in each state. The results are then sorted in descending order by the average delivery time and the top 5 states with the longest delivery times are returned.

This information can be used to identify areas where delivery times are longer than expected and take corrective measures to improve delivery times. It can also help in identifying potential logistical issues and optimizing delivery routes to improve overall efficiency.

Top 5 states where delivery is really fast

```
select customer_state,round(avg(diff_days),2) as diff_days_count from `assignment-
385915.Assignment.customers` customer join
(select * ,timestamp_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff
_days
from `assignment-385915.Assignment.orders` orders
where order_status = 'delivered')temp
on customer.customer_id = temp.customer_id
group by customer_state
order by avg(diff_days) asc
limit 5;
```



Actionable Insights - The SQL query above retrieves the top 5 states with the shortest average delivery time for orders that have been marked as "delivered".

The query joins the "customers" table with a subquery that calculates the difference in days between the estimated delivery date and the actual delivery date for each order that has been delivered. The results are then grouped by state and ordered by the average delivery time in ascending order.

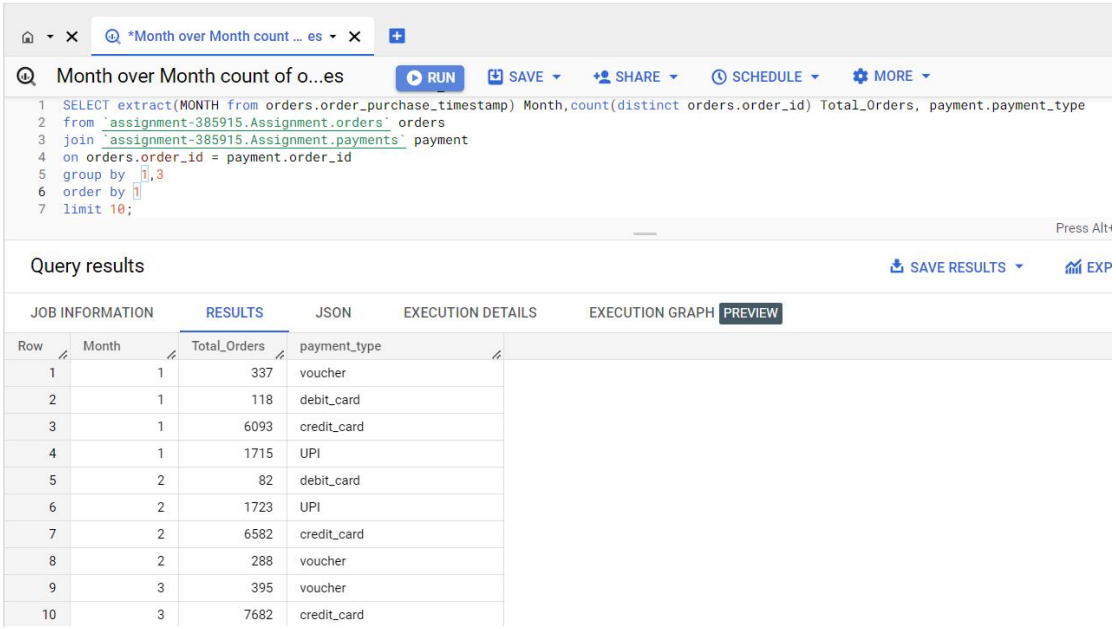
This information can be useful for businesses to identify areas where they are performing well in terms of delivery times and to potentially replicate those practices in other areas. It can also help identify areas where improvements can be made to reduce delivery times and improve customer satisfaction.

6) Payment type analysis:

6.1) Month over Month count of orders for different payment types

Ans -

```
SELECT extract(MONTH from orders.order_purchase_timestamp) Month,count(distinct orders.order_id) Total_Orders, payment.payment_type
from `assignment-385915.Assignment.orders` orders
join `assignment-385915.Assignment.payments` payment
on orders.order_id = payment.order_id
group by  1,3
order by 1
limit 10;
```



Row	Month	Total_Orders	payment_type
1	1	337	voucher
2	1	118	debit_card
3	1	6093	credit_card
4	1	1715	UPI
5	2	82	debit_card
6	2	1723	UPI
7	2	6582	credit_card
8	2	288	voucher
9	3	395	voucher
10	3	7682	credit_card

Actionable Insights - The SQL query above retrieves monthly order information for the top 10 payment types used by customers.

It extracts the month from the order purchase timestamp and counts the total number of distinct orders for each payment type. The results are grouped by month and payment type and ordered by month in ascending order.

This data can be used to analyze customer payment behavior and preferences over time. It can also be used to identify trends and patterns in payment types used by customers and make data-driven decisions to optimize payment processing and improve customer experience.

6.2) Count of orders based on the no. of payment installments

Ans -

```
SELECT count(distinct order_id) Total_Orders,payment_installments
from `assignment-385915.Assignment.payments`
group by 2
order by 2
limit 10;
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	Total_Orders	payment_installr
1	2	0
2	49060	1
3	12389	2
4	10443	3
5	7088	4
6	5234	5
7	3916	6
8	1623	7
9	4253	8
10	644	9

Actionable Insights - The SQL query above counts the total number of distinct order IDs for each payment installment value in the payments table. The results are then sorted in ascending order by the payment installment value and limited to the top 10 results.

This query can be useful in understanding the distribution of payment installments for orders in the payments table. It can help identify patterns in payment behavior and inform business decisions related to payment processing and customer payment options.

- Here are some recommendations for Target based on the business case:

1. Improve order status tracking:- The business case shows that a significant number of orders are not delivered on time. Target can improve order status tracking by using a more robust system that provides real-time updates to customers. Also where the delivery time is on time they can refer and come up with the solution in nearing areas.

2. Offer more payment options:- The business case shows that a significant number of customers are using credit card and and UPI to pay for their orders. Target can increase customer convenience by making it more reliable and safe payments options.

3. Improve freight performance:- The business case shows that a significant number of orders are delivered late due to damage or some other reasons. Target can improve freight performance by working with its shipping partners to ensure that orders are properly packaged and shipped.

4. Target marketing to specific customer segments:- The business case shows that different customer segments have different preferences. Target can improve its marketing efforts by targeting specific customer segments or during festivals can store up more inventory with relevant offers and promotions to make more sales.

5. Expand its product selection: The business case shows that Target's product selection is limited in some areas. Target can expand its product selection by adding new products and brands to its stores. Whenever there is no need of that item they can reduce that inventory.

By implementing these recommendations, Target can improve its customer experience and increase its sales.