1st Question

1. Data type of columns in a table

Ans. **CUSTOMERS TABLE**

Customer_id → STRING

Customer_unique_id → STRING

Customer_zip_code_prefix → INTEGER

Customer_city → STRING

Customer_state → STRING

ORDER_ITEMS TABLE

order_id → STRING

order_item_id → INTEGER

product_id → STRING

seller_id → STRING

shipping_limit_date → TIMESTAMP

price → FLOAT

freight_value → FLOAT

ORDER_REVIEWS TABLE

review_id → STRING

order_id → STRING

review_score → INTEGER

review_comment_title → STRING

review_creation_date → TIMESTAMP

review_answer_timestamp → TIMESTAMP

ORDERS TABLE

order_id → STRING

customer_id → STRING

order_status → STRING

order_purchase_timestamp → TIMESTAMP

order_approved_at → TIMESTAMP

order_delivered_carrier_date → TIMESTAMP

order_delivered_customer_date → TIMESTAMP

```
order_estimated_delivery_date → TIMESTAMP
```

PAYMENTS

order_id \rightarrow STRING

payment_sequential → INTEGER

payment_type → STRING

payment_installments → INTEGER

payment_value → FLOAT

PRODUCTS

product_id → STRING

product_category →STRING

product_name_length → INTEGER

product_description_length → INTEGER

product_photos_qty → INTEGER

product_weight_g → INTEGER

product_length_cm → INTEGER

product_height_cm → INTEGER

product_width_cm → INTEGER

SELLERS

seller_id → STRING

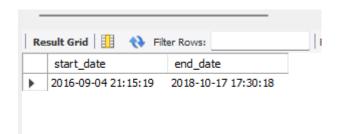
seller_zip_code_prefix → INTEGER

seller_city → STRING

seller_state → STRING

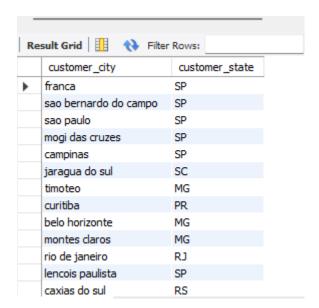
2. Time period for which the data is given

Ans. SELECT min(order_purchase_timestamp) as 'start_date', max(order_purchase_timestamp) as 'end_date' FROM orders;



3. Cities and States covered in the dataset

Ans. SELECT DISTINCT customer city, customer state FROM customers;

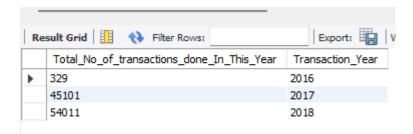


2nd Question

1. <u>Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario?</u> <u>Can we see some seasonality with peaks at specific months?</u>

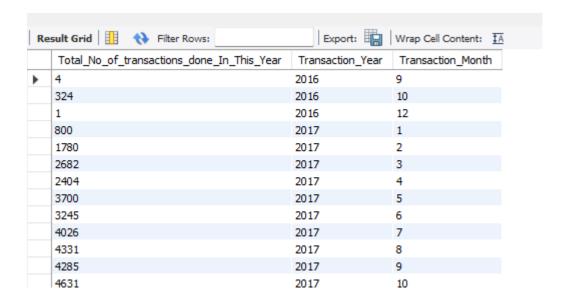
Ans. SELECT COUNT(1) as

'Total_No_of_transactions_done_In_This_Year',YEAR(order_purchase_timestamp) as 'Transaction_Year' FROM ORDERS GROUP BY YEAR(order_purchase_timestamp) ORDER BY YEAR(order_purchase_timestamp);



SELECT COUNT(1) as

'Total_No_of_transactions_done_In_This_Year',YEAR(order_purchase_timestamp) as 'Transaction_Year',MONTH(order_purchase_timestamp) as 'Transaction_Month' FROM ORDERS GROUP BY YEAR(order_purchase_timestamp),MONTH(order_purchase_timestamp) ORDER BY YEAR(order_purchase_timestamp),MONTH(order_purchase_timestamp);



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

Ans. WITH cte AS (SELECT

CASE

WHEN TIME(order_purchase_timestamp) BETWEEN '00:00:00.000000' AND '05:00:00.000000' THEN 'Dawn'

WHEN TIME(order_purchase_timestamp) BETWEEN '05:00:00.000001' AND '12:00:00.000000' THEN 'Morning'

WHEN TIME(order_purchase_timestamp) BETWEEN '12:00:00.000001' AND '19:00:00.000000' THEN 'Afternoon'

WHEN TIME(order_purchase_timestamp) BETWEEN '19:00:00.000001' AND '23:59:59.000000' THEN 'Night'

END AS 'Timing'

FROM orders)

SELECT count(*) as 'Total_transactions_done_in_that_time', Timing FROM cte GROUP BY Timing ORDER BY count(*) DESC;

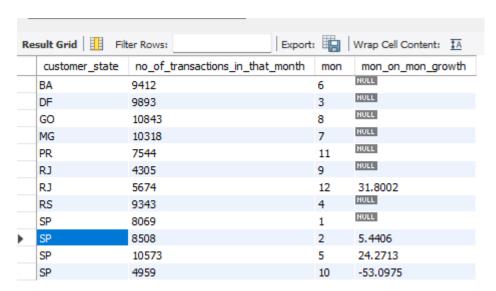


3rd Question

1. Get month on month orders by region, states.

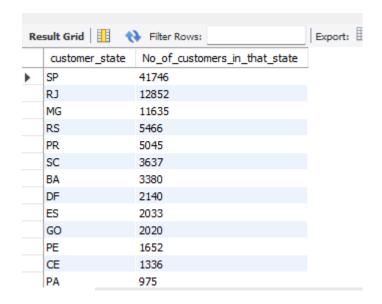
Ans. WITH m_o_m AS (SELECT customer_state,MONTH(order_purchase_timestamp) as mon FROM orders as o JOIN customers as c ON o.customer id=c.customer id)

SELECT customer_state,count(*) as 'no_of_transactions_in_that_month',mon,(count(*)-LAG(count(*),1,0) OVER(PARTITION BY customer_state ORDER BY mon))*100/LAG(count(*),1,0) OVER(PARTITION BY customer_state ORDER BY mon) as 'mon_on_mon_growth' FROM m_o_m GROUP BY mon;

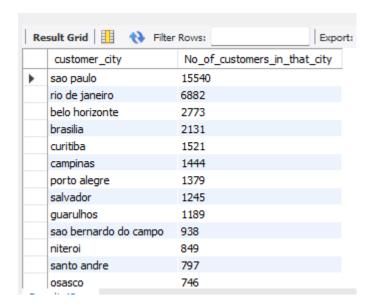


2. How are customers distributed in Brazil

Ans. SELECT customer_state,count(*) as 'No_of_customers_in_that_state' FROM customers GROUP BY customer_state ORDER BY count(*) DESC;



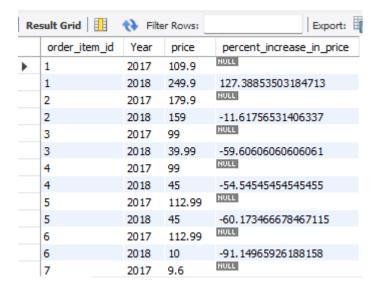
SELECT customer_city,count(*) as 'No_of_customers_in_that_city' FROM customers GROUP BY customer city ORDER BY count(*) DESC;



4th Question

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

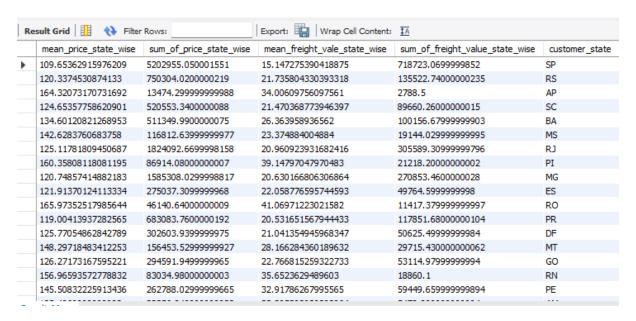
Ans. SELECT order_item_id,YEAR(order_purchase_timestamp) as 'Year',price,(price-LAG(price,1) OVER(PARTITION BY order_item_id ORDER BY YEAR(order_purchase_timestamp)))*100/LAG(price,1) OVER(PARTITION BY order_item_id ORDER BY YEAR(order_purchase_timestamp)) as 'percent_increase_in_price' FROM order_items as oi JOIN orders as o on oi.order_id=o.order_id WHERE MONTH(order_purchase_timestamp) BETWEEN 1 AND 8 GROUP BY YEAR(order_purchase_timestamp),order_item_id ORDER BY order_item_id,YEAR(order_purchase_timestamp);



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

Ans. SELECT AVG(price) as 'mean_price_state_wise',sum(price) as 'sum_of_price_state_wise',AVG(freight_value) as 'mean_freight_vale_state_wise',sum(freight_value) as

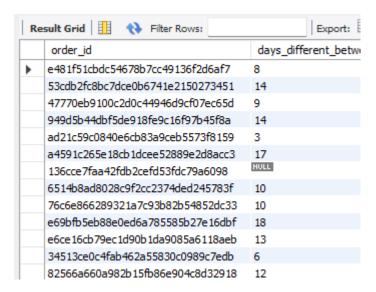
'sum_of_freight_value_state_wise',customer_state FROM order_items as oi JOIN orders as o ON oi.order_id=o.order_id JOIN customers as c ON o.customer_id=c.customer_id GROUP BY customer_state;



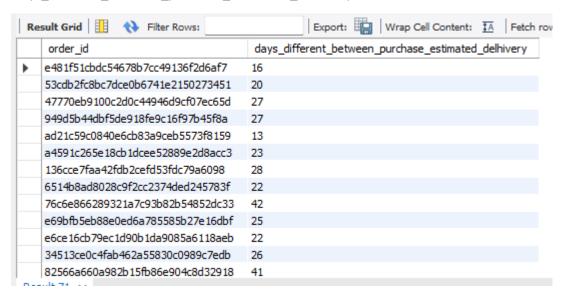
5th Question

1. Calculate days between purchasing, delivering and estimated delivery

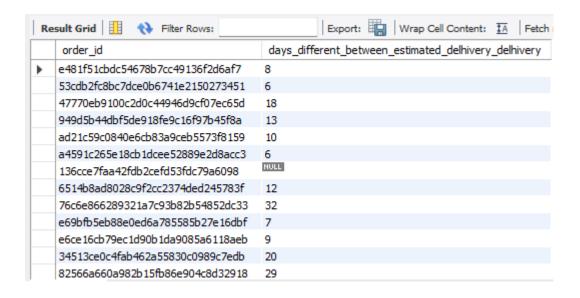
Ans. SELECT order_id,DATEDIFF(order_delivered_customer_date,order_purchase_timestamp) as 'days_different_between_purchase_delhivery' FROM orders;



SELECT order_id,DATEDIFF(order_estimated_delivery_date,order_purchase_timestamp) as 'days_different_between_purchase_estimated_delhivery' FROM orders;



SELECT order_id,DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date) as 'days_different_between_estimated_delhivery' FROM orders;



2. Create columns:

time_to_delivery = order_purchase_timestamp-order_delivered_customer_date

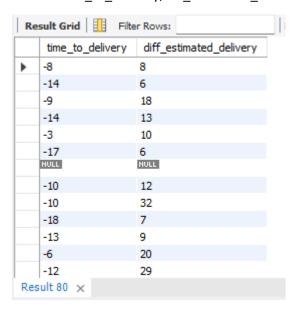
diff estimated delivery = order estimated delivery date-order delivered customer date

Ans. WITH create_columns AS (SELECT

*,DATEDIFF(order_purchase_timestamp,order_delivered_customer_date) as 'time_to_delivery',

DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date) as 'diff_estimated_delivery' FROM orders)

SELECT time_to_delivery,diff_estimated_delivery FROM create_columns;



3. Group data by state, take mean of freight value, time to delivery, diff estimated delivery

Ans. WITH create columns AS (SELECT

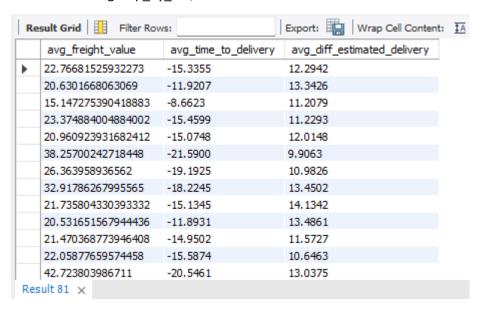
*,DATEDIFF(order_purchase_timestamp,order_delivered_customer_date) as 'time_to_delivery',

DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date) as 'diff_estimated_delivery' FROM orders),group_by_cte AS(

SELECT AVG(freight_value) as 'avg_freight_value',AVG(time_to_delivery) as 'avg_time_to_delivery',AVG(diff_estimated_delivery) as 'avg_diff_estimated_delivery' FROM create_columns as cc JOIN order_items as oi ON cc.order_id=oi.order_id JOIN customers as c ON cc.customer id=c.customer id

GROUP BY c.customer state)

SELECT * FROM group by cte;



4. Sort the data to get the following:

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

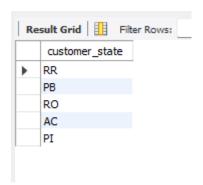
Ans. WITH create columns AS (SELECT

*,DATEDIFF(order_purchase_timestamp,order_delivered_customer_date) as 'time_to_delivery',

ON cc.customer_id=c.customer_id GROUP BY c.customer_state)

DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date) as 'diff_estimated_delivery' FROM orders),group_by_cte AS(SELECT c.customer_state,AVG(freight_value) as 'avg_freight_value',AVG(time_to_delivery) as 'avg_time_to_delivery',AVG(diff_estimated_delivery) as 'avg_diff_estimated_delivery' FROM create_columns as cc JOIN order_items as oi ON cc.order_id=oi.order_id JOIN customers as c

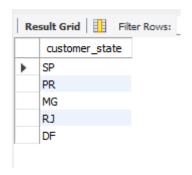
SELECT customer_state FROM group_by_cte ORDER BY avg_freight_value DESC LIMIT 5;



WITH create_columns AS (SELECT

*,DATEDIFF(order_purchase_timestamp,order_delivered_customer_date) as 'time_to_delivery',DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date) as 'diff_estimated_delivery' FROM orders),group_by_cte AS(SELECT c.customer_state,AVG(freight_value) as 'avg_freight_value',AVG(time_to_delivery) as 'avg_time_to_delivery',AVG(diff_estimated_delivery) as 'avg_diff_estimated_delivery' FROM create_columns as cc JOIN order_items as oi ON cc.order_id=oi.order_id JOIN customers as c ON cc.customer_id=c.customer_id GROUP BY c.customer_state)

SELECT customer_state FROM group_by_cte ORDER BY avg_freight_value LIMIT 5;



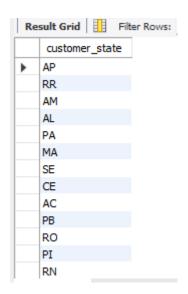
2. Top 5 states with highest/lowest average time to delivery

Ans. WITH create_columns AS (SELECT

*,DATEDIFF(order_purchase_timestamp,order_delivered_customer_date) as 'time_to_delivery',

DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date) as 'diff_estimated_delivery' FROM orders),group_by_cte AS(SELECT c.customer_state,AVG(freight_value) as 'avg_freight_value',AVG(time_to_delivery) as 'avg_time_to_delivery',AVG(diff_estimated_delivery) as 'avg_diff_estimated_delivery' FROM create_columns as cc JOIN order_items as oi ON cc.order_id=oi.order_id JOIN customers as c ON cc.customer_id=c.customer_id GROUP BY c.customer_state)

SELECT customer_state FROM group_by_cte ORDER BY avg_time_to_delivery;

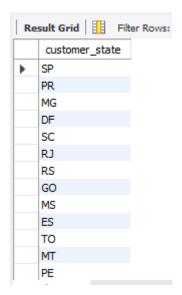


WITH create_columns AS (SELECT

*,DATEDIFF(order_purchase_timestamp,order_delivered_customer_date) as 'time_to_delivery',

DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date) as 'diff_estimated_delivery' FROM orders),group_by_cte AS(SELECT c.customer_state,AVG(freight_value) as 'avg_freight_value',AVG(time_to_delivery) as 'avg_time_to_delivery',AVG(diff_estimated_delivery) as 'avg_diff_estimated_delivery' FROM create_columns as cc JOIN order_items as oi ON cc.order_id=oi.order_id JOIN customers as c ON cc.customer_id=c.customer_id GROUP BY c.customer_state)

SELECT customer_state FROM group_by_cte ORDER BY avg_time_to_delivery DESC;



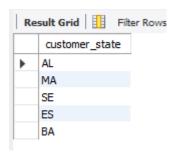
3. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Ans. WITH create_columns AS (SELECT

*,DATEDIFF(order_purchase_timestamp,order_delivered_customer_date) as 'time to delivery',

DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date) as 'diff_estimated_delivery' FROM orders),group_by_cte AS(SELECT c.customer_state,AVG(freight_value) as 'avg_freight_value',AVG(time_to_delivery) as 'avg_time_to_delivery',AVG(diff_estimated_delivery) as 'avg_diff_estimated_delivery' FROM create_columns as cc JOIN order_items as oi ON cc.order_id=oi.order_id JOIN customers as c ON cc.customer id=c.customer id GROUP BY c.customer state)

SELECT customer_state FROM group_by_cte ORDER BY avg_diff_estimated_delivery LIMIT 5;

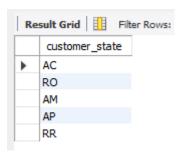


WITH create_columns AS (SELECT

*,DATEDIFF(order_purchase_timestamp,order_delivered_customer_date) as 'time_to_delivery',

DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date) as 'diff_estimated_delivery' FROM orders),group_by_cte AS(SELECT c.customer_state,AVG(freight_value) as 'avg_freight_value',AVG(time_to_delivery) as 'avg_time_to_delivery',AVG(diff_estimated_delivery) as 'avg_diff_estimated_delivery' FROM create_columns as cc JOIN order_items as oi ON cc.order_id=oi.order_id JOIN customers as c ON cc.customer_id=c.customer_id GROUP BY c.customer_state)

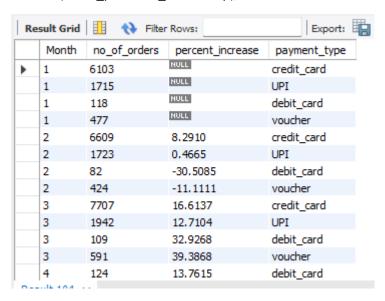
SELECT customer_state FROM group_by_cte ORDER BY avg_diff_estimated_delivery DESC LIMIT 5;



6th Question

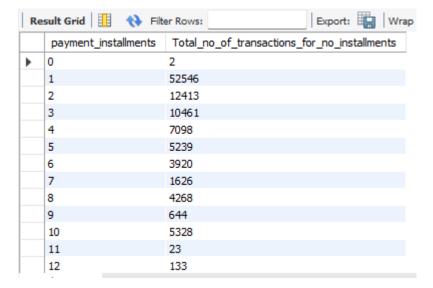
1. Month over Month count of orders for different payment types

Ans. SELECT MONTH(order_purchase_timestamp) as 'Month',count(o.order_id) as 'no_of_orders',(count(o.order_id)-LAG(count(o.order_id),1) OVER(PARTITION BY payment_type ORDER BY MONTH(order_purchase_timestamp)))*100/LAG(count(o.order_id),1) OVER(PARTITION BY payment_type ORDER BY MONTH(order_purchase_timestamp)) as 'percent_increase',payment_type FROM orders as o LEFT JOIN payments as p on o.order_id=p.order_id GROUP BY MONTH(order_purchase_timestamp),payment_type ORDER BY MONTH(order_purchase_timestamp);



2. Distribution of payment installments and count of orders

Ans. SELECT payment_installments,count(*) as 'Total_no_of_transactions_for_no_installments' FROM payments GROUP BY payment_installments ORDER BY payment_installments;



7Th Question

ACTIONABLE INSIGHTS

More number of customers are located in the city sao Paulo, rio de janerio, belo horizonte etc

More number of customers are located in the state SP,RJ,MG,RS,PR.

Most of the item prices have decreases in 2018 when compared to 2017.

More number of orders are from the states RR,PB,RO,AC,PI and in the similar way less number of orders are from the states SP,PR,MG,RJ,DF.

Average Time of delivery is least in the states AP,RR,AM,AI,PA and highest in the states SP,PR,MG,DF,SC.

The delivery time is less in the states AL,MA,SE,ES,BA and in the similar way the delivery time is high in the states AC,RO,AM,Ap,RR.

Most of the people are preferring to pay in single instalment followed by two and three.

8Th Question

RECOMMENDATION

As more number of customers are from the city sao Paulo, rio de janerio, belo horizonte etc we should try to attract more customers by giving discount to the people living in that particular state and city.

As most number of orders are from the states RR,PB,RO,AC,PI the ecommerce should find a way to deliver the goods faster and also more number of delivery persons should be present at any point of time.

As we cannot see any trend in which card the people are using the most, we have to incentivise the digital payment so that a greater number of people will be attracted towards the digital transactions.

As the delivery of some of the states is too high we should fine some alternative ways to deliver the good in quick manner.