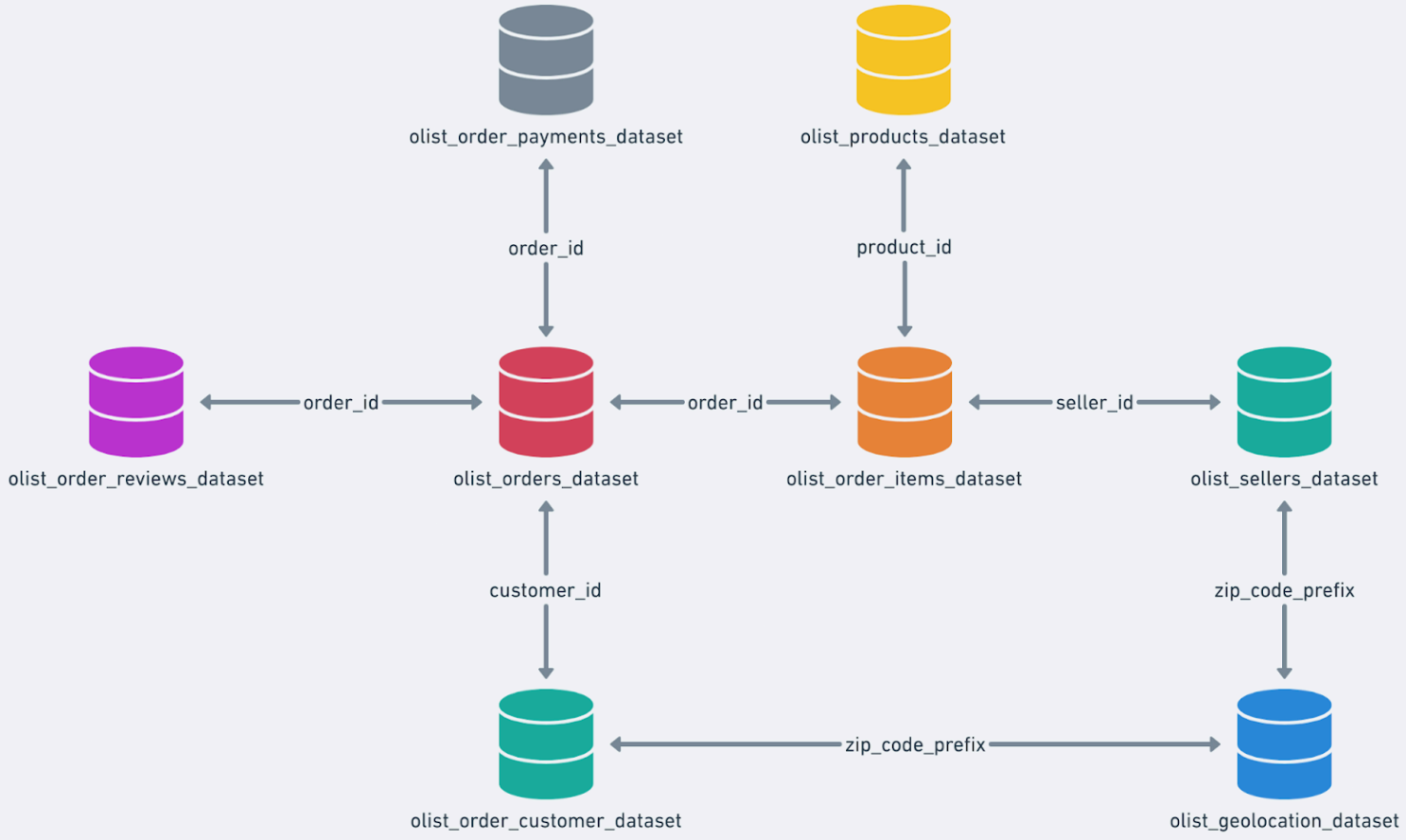
BUSINESS CASE STUDY - Brazilian E-commerce

Problem Statement :-

Assuming you are a data analyst/ scientist at Brazilian E-commerce, you have been assigned the task of analyzing the given dataset to extract valuable insights and provide actionable recommendations.

Data Schema :-



The dataset have 8 tables :-

- customers Table
- geolocation Table
- order_items Table
- orders Table
- payments Table
- products Table
- reviews Table
- sellers Table

Questions :-

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

A) Data type of all columns in the "customers" table.

i) Query -

```
SELECT column_name,
       data_type
FROM   ecommerce.INFORMATION_SCHEMA.COLUMNS
WHERE  table_name = 'customers' ;
```

ii) Result -

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

iii) Insight - This query provides the information of data type present in the table (customers) , which is helpful in understanding the table structure. All columns use string type except the customer_zip_code_prefix which is in INT64. Also, there are two customer id i.e. customer_id (is for per order placed) and customer_unique_id(is for per customer).

iv) Recommendation - There may be some problems with how zip codes are stored as INT64 drops leading zeros like if the zip code of a city is ‘012345’ then it will show ‘12345’. Hence, it is much preferred to store customer_zip_code_prefix in strings.

B) Get the time range between which the orders were placed.

i) Query -

```
SELECT min(order_purchase_timestamp) as first_order,
       max(order_purchase_timestamp) as last_order
FROM   ecommerce.orders ;
```

ii) Result -

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

iii) Insight - This query provides the information of the date and time of the first order placed (on Sept 9, 2016) and last order placed (on Oct 17, 2018). This tells us of approximately 2 years and 1 month of time range, which serves as a crucial pattern for identifying order patterns and analytics.

C) Count the Cities & States of customers who ordered during the given period.

i) Query -

```
SELECT COUNT(distinct customer_city) AS number_of_city,
       COUNT(distinct customer_state ) AS number_of_state
FROM ecommerce.orders AS o
JOIN   ecommerce.customers AS c USING(customer_id)
```

ii) Result -

Row	number_of_city	number_of_state
1	4119	27

iii) Insight - This dataset provides the information that the customers are placing orders from 27 states and 4119 cities, which shows that Brazilian E-commerce has established a significant presence across various regions in Brazil i.e. Brazilian E-commerce has been established in all the states and 4119 cities which are operating in more 74% cities. It offers valuable information for conduction geographic analysis and scope for expansions.

Q2. In-depth Exploration:

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

i) Query -

```
WITH order_yearly AS (
    select extract(year from order_purchase_timestamp) as year,
           extract(month from order_purchase_timestamp) as month,
           count( order_id) as total_order_placed
     from ecommerce.orders
     group by extract(year from order_purchase_timestamp),
              extract(month from order_purchase_timestamp)
      )
select concat(year, '-', month) as date_range,
       total_order_placed,
       lag(total_order_placed,1) over(order by year, month) as last_month_order,
       Round(((total_order_placed - lag(total_order_placed,1) over(order by year, month)) /
       lag(total_order_placed,1) over(order by year, month)) * 100,2) as MoM_growth_percentage
from order_yearly
order by year, month;
```

ii) Result -

Row	date_range	total_order_placed	last_month_order	MoM_growth_percentage
1	2016-9	4	*null*	*null*
2	2016-10	324	4	8000.0
3	2016-12	1	324	-99.69
4	2017-1	800	1	79900.0
5	2017-2	1780	800	122.5
6	2017-3	2682	1780	50.67
7	2017-4	2404	2682	-10.37
8	2017-5	3700	2404	53.91
9	2017-6	3245	3700	-12.3
10	2017-7	4026	3245	24.07

iii) Insight - From this dataset we get to see volatile growth patterns from Jan 2017 with 800 orders to August 2018 with 6512 orders, but failed to stabilize the operations which resulted in the sudden drop down of orders in the last of the year 2018. Early months show the rapid growth i.e. January 2017 with 79900% growth in the sales, which followed by further increase of 122.5% in February 2017 and sudden drop out of sales in September 2018 i.e. -99.75%. Overall, growth was inconsistent, with peaks in mid-2017 (8% to 122.5% MoM) but frequent declines, indicating potential seasonal or operational challenges.

iv) Recommendation -
-> Should maintain some stock in reserve for the peak season to prevent the storages.
-> The company should plan a strategic promotion and marketing scheme so that sales would not drop down by giving discounts to customers or adding combo stocks.

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

i) Query -

```
WITH date_range AS (
 SELECT order_id,
        order_purchase_timestamp,
        FORMAT_TIMESTAMP('%Y-%m',order_purchase_timestamp) AS year_month,
        EXTRACT(month FROM order_purchase_timestamp) AS month
 from ecommerce.orders
),
cnt_order AS (
 SELECT year_month,
        COUNT(DISTINCT order_id) AS no_of_orders
 FROM date_range
 GROUP BY year_month
 ORDER BY year_month ASC
 ),
percent_change AS (
 SELECT year_month,
        co.no_of_orders,
```

```
            ROUND((no_of_orders - lag(no_of_orders) over(ORDER BY year_month asc))/lag(no_of_orders)
over (ORDER BY year_month) * 100, 2) AS percent_change
 FROM cnt_order AS co
),
final AS (
 SELECT  year_month,
            no_of_orders,
            ROUND((no_of_orders - lag(no_of_orders) over(order by year_month asc))/lag(no_of_orders)
over (order by year_month) * 100, 2) AS percent_change,
            ROUND(avg(no_of_orders) over()) AS average_no_of_orders,
            ROUND(no_of_orders-avg(no_of_orders) over()) AS dev_from_avg
 FROM percent_change
 GROUP BY year_month,
              no_of_orders
)
SELECT *
FROM final
ORDER BY year_month ASC;
```

ii) Result -

Row	year_month	no_of_orders	percent_change	average_no_of_orders	dev_from_avg
1	2016-09	4	*null*	3978.0	-3974.0
2	2016-10	324	8000.0	3978.0	-3654.0
3	2016-12	1	-99.69	3978.0	-3977.0
4	2017-01	800	79900.0	3978.0	-3178.0
5	2017-02	1780	122.5	3978.0	-2198.0
6	2017-03	2682	50.67	3978.0	-1296.0
7	2017-04	2404	-10.37	3978.0	-1574.0
8	2017-05	3700	53.91	3978.0	-278.0
9	2017-06	3245	-12.3	3978.0	-733.0
10	2017-07	4026	24.07	3978.0	48.0

iii) Insight - The dataset highlights important trends and patterns among the order patterns. During initial start, Brazilian E-commerce showed extreme downfall and volatility, Order's drastically shifted from 1 in December 2016 to 800 in January 2017, suggesting early stage instability. However, from February 2017, Brazilian E-commerce took a drastic shift in the upward trend where maximum orders were in November 2017. Also, there is an unexplained drop and sudden drop to 16 in September 2018 and 4 in October 2018.

iv) Recommendation -

-> To investigate the abnormalities in the last of operation i.e. September and October in 2018 and fix the problem.

-> Should maintain the warehouse inventory to handle the fluctuations in the demand and supply, or to improve communication with other warehouses of cities/states.

C) During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

- ***i) 0-6 hrs : Dawn***
- ***ii) 7-12 hrs : Mornings***
- ***iii) 13-18 hrs : Afternoon***
- ***iv) 19-23 hrs : Night***

i) Query -

```
WITH duration_of_day AS (
   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 12 THEN 'Mornings'
WHEN EXTRACT(hour FROM order_purchase_timestamp) BETWEEN 13 AND 18 THEN 'Afternoon'
        ELSE 'Night' END) AS Duration
    FROM ecommerce.orders
 )
SELECT Duration,
          COUNT(order_id) AS order_place_duration
FROM duration_of_day
GROUP BY Duration
ORDER BY order_place_duration DESC;
```

ii) Result -

Row	Duration	order_place_duration
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

iii) Insight - The data highlights that afternoon is the peak time for the orders (38135 orders), followed by night and morning where dawn has lesser orders (5242 orders). This data suggests that customers prefer to shop during the afternoon or the late evening possibly due to their break time after working hours.

iv) Recommendation -

-> to maximize sales, Brazilian E-commerce should focus its marketing strategy on peak afternoon hours — such as launching flesh sales or running targeted digital ads, when customer activity is highest.

-> during the low-activity dawn period, Brazilian E-commerce could implement campaigns like limited time offers or a rewards system to increase the sales.

Q3. Evolution of E-commerce orders in the Brazil region:

A) Get the month on month no. of orders placed in each state.

i) Query -

```
SELECT c.customer_state AS state,
       extract(year FROM o.order_purchase_timestamp) as Year,
       Format_date('%B',o.order_purchase_timestamp) as month,
       count(o.order_id) as total_order
FROM ecommerce.orders as o
JOIN ecommerce.customers as c
on o.customer_id = c.customer_id
GROUP BY c.customer_state,
       extract(year FROM o.order_purchase_timestamp),
       Format_date('%B',o.order_purchase_timestamp)
ORDER BY state, Year,month;
```

iii) Result -

Row	state	Year	month	total_order
1	AC	2017	April	5
2	AC	2017	August	4
3	AC	2017	December	5
4	AC	2017	February	3
5	AC	2017	January	2
6	AC	2017	July	5
7	AC	2017	June	4
8	AC	2017	March	2
9	AC	2017	May	8
10	AC	2017	November	5

iii) Insight - By analyzing the query results, we gain valuable insights into the monthly over order counts for each state. This allows us to identify trends, patterns, and seasonality in order volumes across the different states. The state SP dominates orders, maximum orders in August 2018 i.e. 3253, which consistently follow by the state RJ and MG but in lower volumes, while the states AC and AP have least orders (less than 10 orders per month). Regional disparities show the stronger e-commerce adoption in the SP, RJ and MG can be due to economically developed states then AC and AP.

iv) Recommendation -

-> The Brazilian E-commerce should focus more inventory allocation on the states SP, RJ and MG. And also focus more marketing strategy on the states RJ and MG to increase the sales.

-> For the lower order states like AC and AP, Brazilian E-commerce should start collaborating with local businesses for better sales and start localized marketing campaigns in those regions.

B) How are the customers distributed across all the states?

i) Query -

```
SELECT customer_state,
        COUNT( DISTINCT customer_unique_id) AS num_of_customer
FROM ecommerce.customers
GROUP BY customer_state
ORDER BY customer_state;
```

ii) Result -

Row	customer_state	num_of_customer
1	AC	77
2	AL	401
3	AM	143
4	AP	67
5	BA	3277
6	CE	1313
7	DF	2075

iii)Insight - The dataset shows that the customer base is highly concentrated, with SP alone accounting for over 40K customers (~50% of total). RJ and MG follow as key markets. The states RS, PR, SC,BA, PE and CE show moderate traction, states (AC, AP, RR, RO) have minimal representation (<500 customers). This suggests untapped potential in low-density regions, requiring targeted strategies to boost adoption.

iv) Recommendation -

-> For high order volumes states and customers focus on premium or high end sales services (e.g., loyalty programs, fast delivery, some discounts or extra services) to maximize revenue from these dominant markets.

-> To boost the mid - tier states like RS, PR, SC,BA, PE and CE we can invest in the Brazilian E-commerce promotion to increase order frequency.

-> for lower order volumes states, Brazilian E-commerce can start collaborating with local business or to start with free delivery or some other beneficial programmes.

Q4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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

i) Query - **(MONTH-WISE)**

```
 WITH Date_range AS (
   SELECT *,
      EXTRACT(year FROM order_purchase_timestamp) AS Year,
      EXTRACT(month FROM order_purchase_timestamp) AS month,
      FORMAT_DATE('%B', order_purchase_timestamp) AS month_name,
  FROM ecommerce.orders ),
Amount_2017 AS (
SELECT month_name,
     ROUND(SUM(payment_value),2) AS cost_2017
FROM ecommerce.payments AS p
JOIN Date_range AS d
ON p.order_id = d.order_id
WHERE d.month < 9 AND year = 2017
GROUP BY month_name ),
Amount_2018 AS (
SELECT month_name,
     ROUND(SUM(payment_value),2) AS cost_2018
FROM ecommerce.payments AS p
JOIN Date_range AS d
ON p.order_id = d.order_id
WHERE d.month < 9 AND year = 2018
GROUP BY month_name )
SELECT a1.month_name,
     a1.cost_2017,
     a2.cost_2018,
```

```
    ROUND(((a2.cost_2018 - a1.cost_2017)/ a1.cost_2017) * 100,2) AS increased_cost_percentage
FROM Amount_2017 as a1
JOIN Amount_2018 AS a2
ON a1.month_name = a2.month_name
ORDER BY increased_cost_percentage DESC;
```

ii) Result -

(MONTH-WISE)

Row	month_name	cost_2017	cost_2018	increased_cost_p...
1	January	138488.04	1115004.18	705.13
2	February	291908.01	992463.34	239.99
3	April	417788.03	1160785.48	177.84
4	March	449863.6	1159652.12	157.78
5	June	511276.38	1023880.5	100.26
6	May	592918.82	1153982.15	94.63
7	July	592382.92	1066540.75	80.04
8	August	674396.32	1022425.32	51.61

i) Query -

(YEAR-WISE)

```
WITH date_range AS (
 SELECT *,
       EXTRACT(year FROM order_purchase_timestamp) AS year,
       EXTRACT(month FROM order_purchase_timestamp) AS month
 FROM ecommerce.orders ),
amount AS (
 SELECT ROUND(SUM (CASE WHEN dr.year = 2017 THEN p.payment_value ELSE 0 END), 2) AS
Amt_2017,
            ROUND(SUM (CASE WHEN dr.year = 2018 THEN p.payment_value ELSE 0 END), 2) AS
Amt_2018
 FROM ecommerce.payments AS p
 JOIN date_range AS dr USING(order_id)
 WHERE dr.month < 9 and year <> 2016 )
```

```
SELECT Amt_2017,
       Amt_2018,
       ROUND(((Amt_2018 - Amt_2017)/Amt_2017) * 100, 2) AS percent_increase
FROM amount;
```

i) Result -

(YEAR-WISE)

Row	Amt_2017	Amt_2018	percent_increased
1	3669022.12	8694733.84	136.98 %

iii) Insight - The data highlights a significant surge in order costs from 2017 to 2018, with the highest increase in January (705%), likely due to post-holiday demand or pricing adjustments. Growth rates gradually declined but remained strong (51-239%) for other months, suggesting rising transaction values, inflation, or expanded product offerings. July-August saw slower growth, possibly indicating market stabilization or seasonal trends.

iv) Recommendation -

-> January 2018 shows a growth of 705%, which is significantly higher than other months. We need to analyze this trend to identify its causes, allowing us to replicate similar sales methods in other months.

-> Implementing budget planning and analyzing seasonal trends can help us achieve better cost control.

B) Calculate the Total & Average value of order price for each state.

i) Query -

```
SELECT c.customer_state,
    ROUND(SUM(oi.price),2) AS total_order_price,
    ROUND(AVG(oi.price),2) AS avg_order_price
FROM ecommerce.customers AS c
JOIN ecommerce.orders AS o USING(customer_id)
```

```
JOIN `ecommerce.order_items` AS oi USING(order_id)
GROUP BY c.customer_state
ORDER BY 1;
```

ii) Result -

Row	customer_state	total_order_price	avg_order_price
1	AC	15982.95	173.73
2	AL	80314.81	180.89
3	AM	22356.84	135.5
4	AP	13474.3	164.32
5	BA	511349.99	134.6
6	CE	227254.71	153.76
7	DF	302603.94	125.77
8	ES	275037.31	121.91
9	GO	294591.95	126.27
10	MA	119648.22	145.2

iii) Insight - The data reveals significant variations in purchasing power across states. While SP and MG drive the highest total sales, their average order values (SP - $109.65, MG - $120.75) are below the national trend. States like PB ($191.48) and RO ($165.97) show higher per-order spending despite lower volumes, suggesting opportunities for premium targeting. Focus on high- average order prices like PA, TO and RN regions to boost profitability, while scaling volume in high-demand states like RJ and PR.

iv) Recommendation - The data reveals significant regional variations in order values :-
-> states with high average order prices like PB (₹191 avg.) and RO (₹166 avg.) showing premium potential—ideal for upselling strategies.
-> High-volume states like SP and MG deliver scale but lower margins (₹109-₹120 avg.), suggesting opportunities for bulk discounts.

-> For low-performing states (RR, AP), targeted incentives like free shipping could boost adoption. Prioritize tailored pricing and promotions to maximize revenue across all regions while addressing local market dynamics.

C) Calculate the Total & Average value of order freight for each state.

i) Query -

```sql
SELECT c.customer_state,
       ROUND(SUM(oi.freight_value),2) AS total_freight_value,
       ROUND(AVG(oi.freight_value),2) AS avg_freight_value
FROM ecommerce.customers AS c
JOIN ecommerce.orders AS o USING(customer_id)
JOIN `ecommerce.order_items` AS oi USING(order_id)
GROUP BY c.customer_state
ORDER BY 1;
```

iii) Result -

Row	customer_state	total_freight_value	avg_freight_value
1	AC	3686.75	40.07
2	AL	15914.59	35.84
3	AM	5478.89	33.21
4	AP	2788.5	34.01
5	BA	100156.68	26.36
6	CE	48351.59	32.71
7	DF	50625.5	21.04
8	ES	49764.6	22.06
9	GO	53114.98	22.77
10	MA	31523.77	38.26

iii) Insight - This data reveals the information for the freight rates between different states. The States (AC, RR, PR and RO) show the highest average freight fees with lowest which is likely due to economically weaker or bad infrastructure for transportation. With high volume states like SP, MG and RJ the average freight value is cheaper due to well developed infrastructure and high scale of operations.

iv) Recommendation - For remote or under-developed regions, we should pair up regional courier service companies, we can use an established local warehouse.

Q5. Analysis based on sales, freight and delivery time.

A) Find the no. of days taken to deliver each order from the order's purchase date as delivery time. Also, calculate the difference (in days) between the estimated & actual delivery date of an order.

i) Query -

```
SELECT order_id,
    DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,
    day) AS time_to_deliver,
    DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,
    day) AS diff_estimated_delivery
FROM ecommerce.orders
WHERE order_status = 'delivered' and
     order_delivered_customer_date is not null
and order_estimated_delivery_date is not null
ORDER BY order_id ASC;
```

ii) Result -

Row	order_id	time_to_deliver	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792cb16214	7	8
2	00018f77f2f0320c557190d7a144bdd3	16	2
3	000229ec398224ef6ca0657da4fc703e	7	13
4	00024acbcdf0a6daa1e931b038114c75	6	5
5	00042b26cf59d7ce69dfabb4e55b4fd9	25	15
6	00048cc3ae777c65dbb7d2a0634bc1ea	6	14
7	00054e8431b9d7675808bcb819fb4a32	8	16
8	000576fe39319847cbb9d288c5617fa6	5	15
9	0005a1a1728c9d785b8e2b08b904576c	9	0
10	0005f50442cb953dcd1d21e1fb923495	2	18

iii) Insight - In this query, 'time_to_deliver' column represents the number of days taken to deliver the product to the customers from the purchase date, while the column 'diff_estimated_delivery' column indicates the difference between estimated delivery date and the actual delivery date. Through this query we can identify the orders which take more delivery time than expected. The negative sign shows the delay in the delivery, while the positive sign shows the early delivery.

iv) Recommendation - Analyze delayed orders (negative values) to improve last-mile logistics and carrier partnerships. For early deliveries (positive values), refine estimation models. Prioritize reducing delivery times for high-value orders. Implement proactive customer notifications for delays and consider compensation policies for significant setbacks. Focus on regional bottlenecks and optimize routes to enhance overall efficiency, balancing speed with reliable delivery promises.

B) ***Find out the top 5 states with the highest & lowest average freight value.***

i) Query -

With Highest Avg Freight Value

```
SELECT c.customer_state AS state,
       ROUND(AVG(oi.freight_value),2) AS high_avg_freight_value
```

```
FROM ecommerce.customers AS c
JOIN ecommerce.orders AS o USING(customer_id)
JOIN ecommerce.order_items AS oi USING(order_id)
GROUP BY c.customer_state
ORDER BY 2 DESC LIMIT 5 ;
```

With Lowest Avg Freight Value

```
SELECT c.customer_state AS state,
       ROUND(AVG(oi.freight_value),2) AS low_avg_freight_value
FROM ecommerce.customers AS c
JOIN ecommerce.orders AS o USING(customer_id)
JOIN ecommerce.order_items AS oi USING(order_id)
GROUP BY c.customer_state
ORDER BY 2 LIMIT 5 ;
```

ii) Result -

With Highest Avg Freight Value

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

With Lowest Avg Freight Value

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

Another Approach

i) Query -

```
WITH high_ranking AS (
  SELECT c.customer_state AS state,
        ROUND(AVG(oi.freight_value),2) AS avg_freight_value,
        'High' AS ranking
  FROM ecommerce.customers AS c
  JOIN ecommerce.orders AS o USING(customer_id)
  JOIN ecommerce.order_items AS oi USING(order_id)
  GROUP BY c.customer_state
  ORDER BY 2 DESC LIMIT 5 ),
low_ranking AS (
  SELECT c.customer_state AS state,
        ROUND(AVG(oi.freight_value),2) AS avg_freight_value,
        'Low' AS ranking
  FROM ecommerce.customers AS c
  JOIN ecommerce.orders AS o USING(customer_id)
  JOIN ecommerce.order_items AS oi USING(order_id)
  GROUP BY c.customer_state
  ORDER BY 2 LIMIT 5 )
SELECT * FROM high_ranking
UNION ALL
```

```
SELECT * FROM low_ranking
ORDER BY avg_freight_value DESC;
```

ii) Result -

Row	state	avg_freight_value	ranking
1	RR	42.98	High
2	PB	42.72	High
3	RO	41.07	High
4	AC	40.07	High
5	PI	39.15	High
6	DF	21.04	Low
7	RJ	20.96	Low
8	MG	20.63	Low
9	PR	20.53	Low
10	SP	15.15	Low

iii) Insight - This data reveals the information for the freight rates between different states. Analysing the freight value highlights different patterns and opportunities for optimization in our logistics department. States like RR and PB have the maximum freight value which can be due to remote location or bad developed infrastructure, in opposite states like SP and PR have minimum freight value which can be due to well developed infrastructure and economically strong states or high scale operation.

iv) Recommendation -

-> For remote regions, Brazilian E-commerce could partner with the local business and courier services for the freight cost reduction.

-> Find better freight to reduce the freight value for the customers which results in more demand or orders from them.

-> For developed regions, Brazilian E-commerce could start schemes like fast day or free delivery with specific amounts by maintaining the stock and improving the supply chain network.

C) Find out the top 5 states with the highest & lowest average delivery time.

i) Query -

State with Highest Avg Delivery Time

```
SELECT c.customer_state,
        ROUND(AVG(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,day)),
       2) AS high_avg_delivery_time
FROM ecommerce.customers AS c
JOIN ecommerce.orders AS o USING(customer_id)
WHERE order_status = 'delivered'
GROUP BY c.customer_state
ORDER BY 2 DESC LIMIT 5;
```

State with Lowest Avg Delivery Time

```
SELECT c.customer_state,
    ROUND(AVG(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,day)),
    2) AS low_avg_delivery_time
FROM ecommerce.customers AS c
JOIN ecommerce.orders AS o USING(customer_id)
WHERE order_status = 'delivered'
GROUP BY c.customer_state ORDER BY 2 LIMIT 5;
```

ii) Result -

State with Highest Avg Delivery Time

Row	customer_state	high_avg_delivery_time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

State with Lowest Avg Delivery Time

Row	customer_state	low_avg_delivery_time
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

Another Approach

i) Query -

```
WITH high_time AS (
 SELECT c.customer_state,
     ROUND(AVG(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,day)),
     2) AS avg_delivery_time,
     'HIGH' AS remark
 FROM ecommerce.customers AS c
 JOIN ecommerce.orders AS o USING(customer_id)
 WHERE order_status = 'delivered'
 GROUP BY c.customer_state
 ORDER BY 2 DESC LIMIT 5),
```

```
low_time AS (
 SELECT c.customer_state,
     ROUND(AVG(date_diff(o.order_delivered_customer_date, o.order_purchase_timestamp,day)),
     2) AS avg_delivery_time,
     'LOW' AS remark
 FROM ecommerce.customers AS c
 JOIN ecommerce.orders AS o USING(customer_id)
 WHERE order_status = 'delivered'
 GROUP BY c.customer_state
 ORDER BY 2 LIMIT 5
)
SELECT * FROM high_time
UNION ALL
SELECT * FROM low_time
ORDER BY avg_delivery_time;
```

ii) Result -

Row	customer_state	avg_delivery_time	remark
1	SP	8.3	LOW
2	PR	11.53	LOW
3	MG	11.54	LOW
4	DF	12.51	LOW
5	SC	14.48	LOW
6	PA	23.32	HIGH
7	AL	24.04	HIGH
8	AM	25.99	HIGH
9	AP	26.73	HIGH
10	RR	28.98	HIGH

iii) Insight - The dataset provides the information of the average time to deliver (between from when the customer has placed an order to when the customer has received the order). The states (SP,PR,MG,DF and SC) have the fastest delivery rate, while the states (PA,AL,AM,AP and RR) have the slowest delivery rate.

iv) Recommendation -

-> To raise the rate of delivery in states (PA,AL,AM,AP and RR), we can partner with the courier service company or we can expand our warehouse and logistic department to reduce the delays.

-> For the states (SP,PR,MG,DF and SC) we can use fast delivery as a scheme or service to increase the sales for the company.

D) ***Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery. You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.***

i) Query -

```
SELECT c.customer_state as state,
      ROUND(AVG(DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day)),2)
      AS avg_delivery_speed
FROM ecommerce.customers AS c
JOIN ecommerce.orders AS O USING(customer_id)
WHERE order_status = 'delivered'
GROUP BY c.customer_state
ORDER BY avg_delivery_speed DESC LIMIT 5 ;
```

ii) Result -

Row	state	avg_delivery_speed
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

Another Approach

i) Query -

```
WITH delivery_stats AS (
```

```
   SELECT c.customer_state,
            ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp,
day)),2)
            AS actual_delivery_time,
            ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date, o.order_purchase_timestamp,
day)),2)
            AS estimated_delivery_time
   FROM ecommerce.customers AS c
   JOIN ecommerce.orders AS o USING(customer_id)
   WHERE order_status = 'delivered'
   GROUP BY c.customer_state )
SELECT customer_state,
          actual_delivery_time,
          estimated_delivery_time,
          estimated_delivery_time - actual_delivery_time AS actual_vs_estimated
FROM delivery_stats
ORDER BY actual_vs_estimated DESC LIMIT 5 ;
```

ii) Result -

Row	customer_state	actual_delivery_time	estimated_deliver...	actual_vs_estimated
1	AC	20.64	40.72	20.08
2	RO	18.91	38.39	19.48
3	AP	26.73	45.87	19.139999999999997
4	AM	25.99	44.92	18.930000000000003
5	RR	28.98	45.63	16.650000000000002

iii) Insight - The data shows the AC has the fastest delivery time in comparison to other states which is followed by RO, AP, AM and RR. The actual delivery time is calculated by time when customers place the order and customer actually received the order, where as estimated delivery time is calculated by time when customers place the order and customers are expected to receive the orders.

iv) Recommendation - Based on the analysis, AC, RO, AP, AM, and RR are the top 5 states with the fastest deliveries, exceeding estimates by 16–20 days. To maintain customer satisfaction, consider:
-> Rewarding delivery partners in these states to sustain performance.
-> Adjusting estimated delivery time for accuracy, enhancing customer trust. Focus on replicating best practices in under - performing states.

Q6. Analysis based on the payments:

A) Find the month on month no. of orders placed using different payment types.

i) Query -

```
SELECT EXTRACT(year FROM o.order_purchase_timestamp) AS year,
       EXTRACT(month FROM o.order_purchase_timestamp) AS month,
       FORMAT_DATE('%B', o.order_purchase_timestamp) AS month_name,
       payment_type,
       COUNT(order_id) AS num_of_orders
FROM ecommerce.payments AS p
JOIN ecommerce.orders AS o USING(order_id)
GROUP BY EXTRACT(year FROM o.order_purchase_timestamp),
         EXTRACT(month FROM o.order_purchase_timestamp),
         FORMAT_DATE('%B', o.order_purchase_timestamp),
         payment_type
ORDER BY 1, 2, 5 DESC;
```

iii) Result -

Row	year	month	month_name	payment_type	num_of_orders
1	2016	9	September	credit_card	3
2	2016	10	October	credit_card	254
3	2016	10	October	UPI	63
4	2016	10	October	voucher	23
5	2016	10	October	debit_card	2
6	2016	12	December	credit_card	1
7	2017	1	January	credit_card	583
8	2017	1	January	UPI	197
9	2017	1	January	voucher	61
10	2017	1	January	debit_card	9

iii) Insight - The above data provides the information of the preference of payment method used by customers on a monthly basis. Through this information, we can expand the payment options and add some discounts to it. By watching this data, we can judge that credit cards are the most preferred payment options i.e. used for approx 75% while shopping which is followed by the UPI method, the second most popular payment method.

iv) Recommendation -

-> We can provide the vouchers to customers for the purchase of a particular amount to increase the sales.

-> As credit cards are the most popular source of payment, then we can provide some offers like cash back and rewards to increase the sales.

-> As for debit cards and UPI, we can add additional discounts on using these options so that sales can increase.

B) Find the no. of orders placed on the basis of the payment installments that have been paid.

i) Query -

```
SELECT payment_installments,
        count(distinct order_id) as num_of_order
FROM ecommerce.payments
WHERE payment_installments >= 1
GROUP BY payment_installments ORDER BY 2 DESC;
```

ii) Result -

Row	payment_installments	num_of_order
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	10	5315
6	5	5234
7	8	4253
8	6	3916
9	7	1623
10	9	644

iii) Insight - This dataset reveals the preference of the number of installment preferred by the customers. Here, we can see that for 49060 orders, people preferred to pay in single installment i.e. upfront payments. Also, people also prefer to pay the amount in 2 to 10

installments which is more than 50000 orders. However, only a few orders have 11+ installments.

iv) Recommendation - To optimize payment strategies and drive sales growth, the e-commerce company should focus on promoting mid-range installment plans (2-6 installments) by offering low or zero interest rates, as these are popular among customers and can attract budget-conscious shoppers. Simultaneously, the checkout process for single-payment buyers—who represent the majority (49060 orders)—should be streamlined with incentives like instant discounts to encourage faster conversions. Given the minimal demand for long-term payment plans (11+ installments), the company should consider phasing them out to reduce operational complexity and financial risk. Additionally, implementing dynamic messaging at checkout that highlights flexible payment options can further boost conversion rates. By balancing convenience and affordability, this strategy aims to enhance customer satisfaction while maximizing order volume and revenue.
