

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

1. Data type of all columns in the "customers" table.

The screenshot shows the Google Cloud BigQuery console interface. On the left, the 'Explorer' pane lists workspace resources, including a folder 'business_case_study_1' containing the 'customers' table. The main pane displays the 'customers' table schema. The schema table has columns: Field name, Type, Mode, Key, Collation, Default Value, Policy Tags, and Description. The data is as follows:

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

Insight :- All columns are of **string** data type excluding customer_zip_code_prefix columns and Mode of all columns is **Null able** which shows that data can be null.

2. Get the time range between which the orders were placed

```
SELECT
MIN(order_purchase_timestamp) order_purchase_start_date,
MAX(order_purchase_timestamp) order_purchase_end_date
FROM
`business_case_study_1.orders`
```

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

Insight: - The range of the time period is from sep-2016 to oct-2018 and the data of only year 2017 is available as a full year data where for the year 2016 and 2018 is partial data for that year.

3. Count the number of Cities and States in our dataset.

```
SELECT
COUNT(DISTINCT geolocation_city) Total_City,
COUNT(DISTINCT geolocation_state) Total_State
```

```
FROM
`business_case_study_1.geolocations`
```

Row	Total_City	Total_State
1	8011	27

Insight:- As per the analysis there are total 8011 unique Cities and 27 states are present in our dataset

In-depth Exploration:

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

```
SELECT
year,
COUNT(x.order_purchase_timestamp) Total_Orders
FROM (
SELECT
*,
EXTRACT(year
FROM
order_purchase_timestamp) year
FROM
`business_case_study_1.orders`)x
GROUP BY
x.year
ORDER BY
x.year
```

Row	year	Total_Orders
1	2016	329
2	2017	45101
3	2018	54011

Insight:- As there are partial data is available for the year 2016 and 2018, So we can't exactly identify the trend over a year.

But still as per the analysis we can say that there is growing trend over a period of time by observing the total_orders placed in year 2017 and 2018 are increased, even if there is only upto Oct month of data is available.

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

```
SELECT
x.month Month,
COUNT(x.order_purchase_timestamp) Order_Placed
```

```

FROM (
  SELECT
    *,
    EXTRACT(month
  FROM
    order_purchase_timestamp) month
  FROM
    `business_case_study_1.orders`)x
GROUP BY
  x.month
ORDER BY
  x.month

```

Row	Month	Order_Placed
1	1	8069
2	2	8508
3	3	9893
4	4	9343
5	5	10573
6	6	9412
7	7	10318
8	8	10843
9	9	4305
10	10	4959

InSight:- As per the analysis the months from jan to May are growing trend but from August it is suddenly in the decreasing in count of order placed.

3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)
 - 0-6 hrs : Dawn
 - 7-12 hrs : Mornings
 - 13-18 hrs : Afternoon
 - 19-23 hrs : Night

```

SELECT
  COUNT(CASE
    WHEN x.hour_of_purchase BETWEEN 0 AND 6 THEN "Dawn"
  END
  ) AS Dawn,
  COUNT(CASE
    WHEN x.hour_of_purchase BETWEEN 7 AND 12 THEN "Morning"

```

```

END
) Morning,
COUNT(CASE
  WHEN x.hour_of_purchase BETWEEN 13 AND 18 THEN "Afternoon"
END
) Afternoon,
COUNT(CASE
  WHEN x.hour_of_purchase BETWEEN 19 AND 23 THEN "Night"
END
) Night
FROM (
SELECT
  order_purchase_timestamp,
  EXTRACT(hour
FROM
  order_purchase_timestamp) hour_of_purchase
FROM
  `business_case_study_1.orders`)x

```

Row	Dawn	Morning	Afternoon	Night
1	5242	27733	38135	28331

Insight:- As per the analysis Brazilian customer are mostly like to purchase the order in Afternoon and not much interested in Dawn.

Evolution of E-commerce orders in the Brazil region:

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

```

SELECT
  c.customer_state,EXTRACT(month
FROM
  o.order_purchase_timestamp) month_of_purchase,
COUNT(o.order_id) number_of_order_placed

FROM
  `business_case_study_1.customers` c
INNER JOIN
  `business_case_study_1.orders` o
ON
  o.customer_id = c.customer_id
GROUP BY
  c.customer_state,
  month_of_purchase
ORDER BY
  customer_state,
  month_of_purchase

```

Row	customer_state	month_of_purchase	number_of_order_pla
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

2. How are the customers distributed across all the states?

4. `SELECT`
5. `customer_state State,`
6. `COUNT(customer_id) Count_Of_Customers`
7. `FROM`
8. ``business_case_study_1.customers``
9. `GROUP BY`
10. `customer_state`
11. `ORDER BY`
12. `COUNT(customer_id) DESC`

Row	State	Count_Of_Customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

Insight: - Highest number of customers are from "SP" state ie.41746 and lowest number of customers i.e. Only 46

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

1. 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.

```
SELECT
  z.Year,
  z.current_year_purchase,
  z.last_year_purchase,
  ROUND(((z.current_year_purchase-z.last_year_purchase)/z.last_year_purchase)*100,2) AS
percentage_increase
FROM (
  SELECT
    y.Year,
    y.Total_cost_of_orders current_year_purchase,
    LEAD(y.Total_cost_of_orders) OVER (ORDER BY y.Year DESC) last_year_purchase
  FROM (
    SELECT
      EXTRACT(year
    FROM
      x.order_purchase_timestamp) Year,
      SUM(x.payment_value) Total_cost_of_orders
    FROM (
      SELECT
        o.order_id,
        o.customer_id,
        o.order_purchase_timestamp,
        p.payment_value
      FROM
        `business_case_study_1.orders` o
      INNER JOIN
        `business_case_study_1.payments` p
      ON
        p.order_id = o.order_id
    WHERE
      (EXTRACT(year
      FROM
        o.order_purchase_timestamp) BETWEEN 2017
      AND 2018)
      AND (EXTRACT(month
      FROM
        o.order_purchase_timestamp) BETWEEN 01
      AND 08)
      AND (LOWER(order_status) NOT IN ("canceled",
      "unavailable"))))x
    GROUP BY
      Year)y)z
```

Year ▼	current_year_purchase	last_year_purchase	percentage_increase
2018	8594665.519999...	3575957.459999...	140.35
2017	3575957.459999...	null	null

Insight:- So there is almost 140 % increase in purchasing the orders from year 2017 to 2018 in including only Jan-Aug

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

```

SELECT
  DISTINCT c.customer_state,
  ROUND(SUM(x.price),2) Total_Price,
  ROUND(AVG(x.price),2) Avg_Price
FROM
  `business_case_study_1.customers` c
INNER JOIN (
  SELECT
    o.order_id,
    o.customer_id,
    ot.price,
    ot.freight_value
  FROM
    `business_case_study_1.order_items` ot
  INNER JOIN
    `business_case_study_1.orders` o
  ON
    o.order_id = ot.order_id ) x
ON
  c.customer_id = x.customer_id
GROUP BY
  c.customer_state
ORDER BY
  Total_price DESC

```

Row	customer_state ▼	Total_Price ▼	Avg_Price ▼
1	SP	5202955.05	109.65
2	RJ	1824092.67	125.12
3	MG	1585308.03	120.75
4	RS	750304.02	120.34
5	PR	683083.76	119.0
6	SC	520553.34	124.65
7	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27
10	ES	275037.31	121.91

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

```
SELECT
  DISTINCT c.customer_state,
  ROUND(SUM(x.freight_value),2) Total_freight_value,
  ROUND(AVG(x.freight_value),2) Avg_freight_value
FROM
  `business_case_study_1.customers` c
INNER JOIN (
  SELECT
    o.order_id,
    o.customer_id,
    ot.price,
    ot.freight_value
  FROM
    `business_case_study_1.order_items` ot
  INNER JOIN
    `business_case_study_1.orders` o
  ON
    o.order_id = ot.order_id ) x
ON
  c.customer_id = x.customer_id
GROUP BY
  c.customer_state
ORDER BY
  Total_freight_value DESC
```

Row	customer_state	Total_freight_value	Avg_freight_value
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92
9	GO	53114.98	22.77
10	DF	50625.5	21.04

InSight:- State **SP** has the highest fright value and **RR** has the minimum value

As per the analysis we can say that both States SP and RR are maximum and minimum in price and freight value simultaneously

Analysis based on sales, freight and delivery time.

1. 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.
Do this in a single query.

```
SELECT
  customer_id,order_id,
  DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp,day)
time_to_deliver_in_days,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day)
estimate_vs_actual_delivery
FROM
  `business_case_study_1.orders`
WHERE
  order_status = "delivered"
```

Row	customer_id	order_id	time_to_deliver_in_days	estimate_vs_actual_delivery
1	7a34a8e890765ad6f90db76d0...	635c894d068ac37e6e03dc54e...	30	1
2	065d53860347d845788e041c...	3b97562c3aee8bdedcb5c2e45...	32	0
3	0378e1381c730d4504ebc07d2...	68f47f50f04c4cb6774570cfde...	29	1
4	d33e520a99eb4cfc0d3ef2b6ff...	276e9ec344d3bf029ff83a161c...	43	-4
5	a0bc11375dd3d8bdd0e0bfcabc...	54e1a3c2b97fb0809da548a59...	40	-4
6	8fe0db7abbccaf2d788689e91...	fd04fa4105ee8045f6a0139ca5...	37	-1
7	22c0028cdec95ad1808c1fd50...	302bb8109d097a9fc6e9cefc5...	33	-5
8	dca924c5e55e17bdba2ad42ae...	66057d37308e787052a32828...	38	-6

Insight :-

1. Total order deliver is 96478 out of 99441 means 2963 orders are may be cancelled or not delivered.
2. Almost 50% of the order are delivered between 10 and 30 days of order purchase date i.e count of orders are 45873
3. There are 6534 orders which placed after estimated delivery date

2. Find out the top 5 states with the highest & lowest average freight value.

```
SELECT
  c.customer_state,
  AVG(ot.freight_value) Avg_freight,
  DENSE_RANK() OVER(ORDER BY AVG(freight_value)) Lowest_freight,
  DENSE_RANK() OVER(ORDER BY AVG(freight_value) DESC) Highest_freight
FROM
  `business_case_study_1.customers` c
INNER JOIN
```

```

`business_case_study_1.orders` o
ON
c.customer_id = o.customer_id
INNER JOIN
`business_case_study_1.order_items` ot
ON
o.order_id = ot.order_id
GROUP BY
c.customer_state
ORDER BY
AVG(freight_value)
LIMIT
5

```

Row	customer_state	Avg_freight	Lowest_freight	Highest_freight
1	SP	15.14727539041...	1	27
2	PR	20.53165156794...	2	26
3	MG	20.63016680630...	3	25
4	RJ	20.96092393168...	4	24
5	DF	21.04135494596...	5	23
6	SC	21.47036877394...	6	22
7	RS	21.73580433039...	7	21
8	ES	22.05877659574...	8	20
9	GO	22.76681525932...	9	19
10	MS	23.37488400488...	10	18

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

```

SELECT
customer_state,
AVG(x.diff_in_second) Avg_diff,
DENSE_RANK() OVER(ORDER BY AVG(diff_in_second)) lowest_average_delivery_time,
DENSE_RANK() OVER(ORDER BY AVG(x.diff_in_second) DESC)
highest_average_delivery_time
FROM (
SELECT
order_purchase_timestamp,
order_delivered_customer_date,
customer_state,
DATETIME_DIFF(order_delivered_customer_date,order_purchase_timestamp,second)diff_i
n_second
FROM
`business_case_study_1.orders` o
JOIN
`business_case_study_1.customers` c

```

```

ON
  c.customer_id = o.customer_id
WHERE
  order_status = "delivered")x
GROUP BY
  customer_state
ORDER BY
  lowest_average_delivery_time
LIMIT 5

```

Row	customer_state	Avg_diff	lowest_average_delivery_time	highest_average_delivery_time
1	SP	756983.7500617...	1	27
2	PR	1036072.704448...	2	26
3	MG	1037548.711819...	3	25
4	DF	1120397.884615...	4	24
5	SC	1292093.240552...	5	23

Insight:- This are the top 5 states with highest and lowest avg delivery time in second.

Where I calculate the delivery time difference in seconds

- Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

```

SELECT
  x.customer_state,
  AVG(x.day_diff) avg_day_taken_to_deliver
FROM (
  SELECT
    c.customer_id,
    c.customer_state,
    o.order_status,
    o.customer_id,
    o.order_delivered_customer_date,
    o.order_estimated_delivery_date,
    DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, day) day_diff
  FROM
    `business_case_study_1.customers` c
  INNER JOIN
    `business_case_study_1.orders` o
  ON
    o.customer_id = c.customer_id
  WHERE
    o.order_status = "delivered" )x
GROUP BY
  x.customer_state

```

```
ORDER BY
AVG(x.day_diff)
LIMIT
5
```

Row	customer_state	avg_day_taken_to_de
1	AL	7.9471032745592
2	MA	8.768479776847...
3	SE	9.173134328358...
4	ES	9.618546365914...
5	BA	9.934889434889...

This are the top 5 states which has less amount of time taken to deliver the product

Analysis based on the payments:

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

```
SELECT
*
FROM (
SELECT
payment_type,
EXTRACT(month
FROM
order_purchase_timestamp) Month,
COUNT(o.order_id) Total_Order_placed,
FROM
`business_case_study_1.orders` o
INNER JOIN
`business_case_study_1.payments` p
ON
p.order_id = o.order_id
GROUP BY
payment_type,
EXTRACT(month
FROM
order_purchase_timestamp) )x
ORDER BY
x.payment_type,
x.month
```

Row	payment_type	Month	Total_Order_placed
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077

As per the observation in 9th Month i.e in September there is very less number of orders placed by using every payment type as compare to other months orders placed

- Find the no. of orders placed on the basis of the payment instalments that have been paid.

```

SELECT
  payment_installments,
  COUNT(order_id) Orders_Placed
FROM
  `business_case_study_1.payments`
WHERE
  payment_installments <> 0
GROUP BY
  payment_installments
ORDER BY
  payment_installments

```

Row	payment_installment	Orders_Placed
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920
7	7	1626
8	8	4268
9	9	644
10	10	5328

Actionable Insights & Recommendations:-

1. Have to focus on mid of the year i.e from march-to august as there are more of the orders are placed
2. Provide offer or some coupon based scheme in the night to attract more customer to make order
3. Focus on the City, state where less number of ordered are placed
4. More focus on the need of customers from which more customers are placing an order
5. The cost of orders are increase year by year as we checked from 2017 to 2018, so we have to do follow some steps as we are already following
6. Have to do more focus on delivery time reduction