Business Case: Target SQL

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

1.1 Data type of all columns in the "customers" table

```
SELECT
column_name, data_type
FROM
`Target_project.INFORMATION_SCHEMA.COLUMNS`
WHERE
table_name = "customers"
```

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

- **INSIGHT -** This gives me the insight of the type of data present in the customers table . There are five columns present in the table

1.2 Get the time range between which the orders were placed

```
SELECT
MIN (order_purchase_timestamp) AS First_order,
MAX(order_purchase_timestamp) AS last_order
FROM
`Target_project.orders`
```

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

- **INSIGHT –** It is stating that the first order in our dataset was placed on 04th sep 2016 And the last order was placed on 17th oct 2018

1.3 Count the Cities & States of customers who ordered during the given period ?

```
SELECT
  COUNT(DISTINCT customer_city) AS city_count,
  COUNT(DISTINCT customer_state) AS state_count
FROM
  `Target_project.customers` AS c
JOIN
  `Target_project.orders` AS o
ON
  c.customer_id = o.customer_id
```

Row	city_count	state_count
1	4119	27

- **INSIGHT -** There are total 4119 city and 27 state from where the customer ordered the product.

2-In-depth Exploration:

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

```
SELECT
EXTRACT(year FROM order_purchase_timestamp) AS year,
COUNT(order_id) AS no_of_order
FROM
`Target_project.orders`
GROUP BY
1
ORDER BY
1
```

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

- **INSIGHT-** It can be said that 2016 was starting year so the order was less but order has increased over the year.

- **RECOMMENDATION -** No recommendation required as company is growing good.

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

```
SELECT
EXTRACT(month FROM order_purchase_timestamp) AS
peak_month,
COUNT(*) AS order_count
FROM
`Target_project.orders`
GROUP BY
1
ORDER BY
1
```

Row	peak_month	order_count
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-** This states that august month is the peak when the most number of order has been received

- **RECOMMENDATION-** We can give some offer to increase the order count in the month from where the order received is least

2.3 - 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
(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"
WHEN EXTRACT(hour FROM order_purchase_timestamp)
BETWEEN 19 AND 23 THEN "Night"
END
) AS Times_of_day,
COUNT(order_id) AS order_number
FROM
`Target_project.orders`
GROUP BY
1
ORDER BY
1
```

Row	Times_of_day	order_number
1	Afternoon	38135
2	Dawn	5242
3	Mornings	27733
4	Night	28331

- **INSIGHT-** This gives the information about the time during which customer place the order

- **RECOMMENDATION –** We can give discount at the Dawn time of the day to increase order or focus more on those time at which order

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

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

```
SELECT
c.customer_state,
EXTRACT(year FROM order_purchase_timestamp) AS
Year,
EXTRACT(month FROM order_purchase_timestamp) AS
month,
COUNT(order_id) as order_count
FROM
```

```
`Target_project.orders` AS o
JOIN
`Target_project.customers` AS c
ON
o.customer_id = c.customer_id
GROUP BY
1,2,3
ORDER BY
2
```

Row	customer_state	Year	month	order_count
1	RR	2016	9	1
2	RS	2016	9	1
3	SP	2016	9	2
4	SP	2016	10	113
5	RS	2016	10	24
6	RJ	2016	10	56
7	MT	2016	10	3
8	GO	2016	10	9
9	MG	2016	10	40
10	CE	2016	10	8

- **INSIGHT-** This gives the information about the order count across each state over the year 2016 – 2018

- **RECOMMENDATION-** Should focus on the state from where the order count is least.

3.2- How are the customers distributed across all the states?

```
SELECT
DISTINCT customer_state,
COUNT(customer_id) AS cust_cnt_in_each_state
FROM
`Target_project.customers`
GROUP BY
1
ORDER BY
1
```

Row	customer_state	cust_cnt_in_each_state
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747

- **INSIGHT-** State “SP” has the highest customer count. That means most order comes from that particular state.

- **RECOMMENDATION–** Should focus on those state from where the customer count is less

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

4.1- 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

```
WITH
  cte1 AS (
  SELECT
    *
  FROM
    `Target_project.orders` AS o
  JOIN
    `Target_project.payments` AS 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 1 AND 8 ),
  cte2 AS (
  SELECT
    EXTRACT(year
    FROM order_purchase_timestamp) AS year,
    ROUND(SUM(payment_value),2) AS cost
  FROM
    cte1
  GROUP BY
    year
```

```
    )
SELECT
  *,
  ROUND(LAG(cost, 1) OVER(ORDER BY year),2) AS
prev_cost,
  ROUND((cost - LAG(cost, 1) OVER(ORDER BY year))*100
/ (LAG(cost, 1) OVER(ORDER BY year)),2) AS perc_inc
FROM
  cte2
ORDER BY
  year
ASC
```

Row	year	cost	prev_cost	perc_inc
1	2017	3669022.12	null	null
2	2018	8694733.84	3669022.12	136.98

➢ **INSIGHT-** This states that percentage increase in the cost of order over the year 2017-2018

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

```
SELECT
  c.customer_state,
  ROUND(SUM(oi.price),2) AS total_price,
  ROUND(SUM(oi.price)/ COUNT(o.order_id),2) AS
avg_price_per_each_state
```

```
FROM
  `Target_project.order_items` AS oi
JOIN
  `Target_project.orders` AS o
ON
  oi.order_id = o.order_id
JOIN
  `Target_project.customers` AS c
ON
  c.customer_id = o.customer_id
GROUP BY
  1
ORDER BY
  1
```

Row	customer_state	total_price	avg_price_per_each_state
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

- **INSIGHT -** It states the total and average price of product ordered from each state.

- **RECOMMENDATION-** Should focus on those state where revenue is less

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

```
SELECT
  c.customer_state,
  ROUND(SUM(oi.freight_value),2) AS total_freight,
  ROUND(SUM(oi.freight_value) / COUNT(o.order_id),2)
AS average_freight_value
FROM
  `Target_project.order_items` AS oi
JOIN
  `Target_project.orders` AS o
ON
  oi.order_id = o.order_id
JOIN
  `Target_project.customers` AS c
ON
  c.customer_id = o.customer_id
GROUP BY
  1
ORDER BY
  1
```

Row	customer_state	total_freight	average_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

- **INSIGHT –** It states the total and average price rate at which the product is delivered from one point to another.

5- Analysis based on sales, freight and delivery time

5.1- 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. You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- **time_to_deliver = order_delivered_customer_date - order_purchase_timestamp**
- **diff_estimated_delivery = order_delivered_customer_date - order_estimated_delivery_date**

```
 SELECT
    DISTINCT 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
    `Target_project.orders`
  WHERE
    order_status = "delivered"
    AND order_delivered_customer_date IS NOT NULL
  ORDER BY
    2 DESC
```

Row	order_id	time_to_deliver	Diff_estimated_delivery
1	ca07593549f1816d26a572e06...	209	-181
2	1b3190b2dfa9d789e1f14c05b...	208	-188
3	440d0d17af552815d15a9e41a...	195	-165
4	0f4519c5f1c541ddec9f21b3bd...	194	-161
5	285ab9426d6982034523a855f...	194	-166
6	2fb597c2f772eca01b1f5c561b...	194	-155
7	47b40429ed8cce3aee9199792...	191	-175
8	2fe324febf907e3ea3f2aa9650...	189	-167
9	2d7561026d542c8dbd8f0daea...	188	-159
10	437222e3fd1b07396f1d9ba8c...	187	-144

- **INSIGHT-** We can state that the maximum delay that occurred in reaching the order to customer is 209 days

- **RECOMMENDATION-** We can partner with some company to reduce the delay

5.2-Find out the top 5 states with the highest & lowest average freight value

```
(SELECT
  c.customer_state,
  ROUND(SUM(oi.freight_value) / COUNT(o.order_id),2) AS
avg_freight_value
FROM
  `Target_project.customers` AS c
JOIN
  `Target_project.orders` AS o
ON
  c.customer_id = o.customer_id
JOIN
  `Target_project.order_items` AS o
```

```
ON
  oi.order_id = o.order_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5)
UNION ALL (
  SELECT
    c.customer_state,
    ROUND(SUM(oi.freight_value) / COUNT(o.order_id),2) AS
avg_freight_value
  FROM
    `Target_project.customers` AS c
  JOIN
    `Target_project.orders` AS o
  ON
    c.customer_id = o.customer_id
  JOIN
    `Target_project.order_items` AS oi
  ON
    oi.order_id = o.order_id
  GROUP BY 1
  ORDER BY 2 ASC
  LIMIT 5)
ORDER BY avg_freight_value ASC
```

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

- **INSIGHT-** It says that “RR” state price rate is high as compared to other state for delivering the order.

- **RECOMMENDATION-** Should Focus on the states where order delivery rate is high.

5.3- Find out the top 5 states with the highest & lowest average delivery time

```
(SELECT
  c.customer_state,
  ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, Day)),2) AS avg_delivery_time
FROM
  `Target_project.orders` AS o
JOIN
  `Target_project.customers` AS c
ON
  o.customer_id = c.customer_id
GROUP BY
  1
ORDER BY
  avg_delivery_time DESC
LIMIT
  5)
UNION ALL (
  SELECT
    c.customer_state,
    ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date,
o.order_purchase_timestamp, Day)),2) AS avg_delivery_time
  FROM
    `Target_project.orders` AS o
  JOIN
    `Target_project.customers` AS c
  ON
    o.customer_id = c.customer_id
  GROUP BY
    1
```

```
  ORDER BY
    avg_delivery_time ASC
  LIMIT
    5)
ORDER BY
  avg_delivery_time ASC
```

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

- **INSIGHT-** It states that delivery time is faster in “SP” states comparing to other states

- **RECOMMENDATION-** Focus on those states where delivery rate is really slow.

5.4-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.

```sql
SELECT
  t.customer_state,
  t.fastest_delivery_state
FROM (
  SELECT
    c.customer_state,

ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,order_delive
red_customer_date, day)),2) AS fastest_delivery_state,
    ROW_NUMBER() OVER (ORDER BY
ROUND(AVG(DATE_DIFF(order_estimated_delivery_date,order_delive
red_customer_date, day)),2) desc) AS rw_num
  FROM
    `Target_project.orders` AS o
  JOIN
    `Target_project.customers` AS c
  ON
    o.customer_id = c.customer_id
  WHERE
    order_status = "delivered"
  GROUP BY
    1) AS t
WHERE
  t.rw_num < 6
```

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

- **INSIGHT-** We can state that order delivery is way more faster in these state.

6- Analysis based on the payments:

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

```
SELECT
  p.payment_type,
  EXTRACT(year FROM o.order_purchase_timestamp) year,
  EXTRACT(month FROM o.order_purchase_timestamp) month,
  COUNT(o.order_id) AS order_count
FROM
  `Target_project.orders` AS o
JOIN
  `Target_project.payments` AS p
ON
  o.order_id = p.order_id
GROUP BY
  1,
  2,
  3
ORDER BY
  2,
  3
```

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

- **INSIGHT-** Output shows the mode of payment which customer made to buy the order.

- **RECOMMENDATION-** Can give offers and discount to those payment type which are contributing less

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

```
SELECT
  p.payment_installments,
  COUNT(o.order_id) AS order_count
FROM
  `Target_project.payments` AS p
JOIN
  `Target_project.orders` AS o
ON
  p.order_id = o.order_id
WHERE
  p.payment_installments >= 1
GROUP BY 1
```

Row	payment_installments	order_count
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

- **INSIGHT-** This states that how many orders payment has been received in how many EMIs.