



SQL PROJECT

Q1.1→ Import the dataset and do usual exploratory analysis steps like checking the structure and characteristics of the dataset.


Ans:-- **For customer's Table**

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	customer_id	STRING	NULLABLE				
<input type="checkbox"/>	customer_unique_id	STRING	NULLABLE				
<input type="checkbox"/>	customer_zip_code_prefix	INTEGER	NULLABLE				
<input type="checkbox"/>	customer_city	STRING	NULLABLE				
<input type="checkbox"/>	customer_state	STRING	NULLABLE				


For Geolocation Table

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	geolocation_zip_code_prefix	INTEGER	NULLABLE				
<input type="checkbox"/>	geolocation_lat	FLOAT	NULLABLE				
<input type="checkbox"/>	geolocation_lng	FLOAT	NULLABLE				
<input type="checkbox"/>	geolocation_city	STRING	NULLABLE				
<input type="checkbox"/>	geolocation_state	STRING	NULLABLE				


For order items Table

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	order_item_id	INTEGER	NULLABLE				
<input type="checkbox"/>	product_id	STRING	NULLABLE				
<input type="checkbox"/>	seller_id	STRING	NULLABLE				
<input type="checkbox"/>	shipping_limit_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	price	FLOAT	NULLABLE				
<input type="checkbox"/>	freight_value	FLOAT	NULLABLE				


For orders_review Table

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	review_id	STRING	NULLABLE				
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	review_score	INTEGER	NULLABLE				
<input type="checkbox"/>	review_comment_title	STRING	NULLABLE				
<input type="checkbox"/>	review_creation_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	review_answer_timestamp	TIMESTAMP	NULLABLE				

For orders Table

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	customer_id	STRING	NULLABLE				
<input type="checkbox"/>	order_status	STRING	NULLABLE				
<input type="checkbox"/>	order_purchase_timestamp	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_approved_at	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_delivered_carrier_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_delivered_customer_date	TIMESTAMP	NULLABLE				
<input type="checkbox"/>	order_estimated_delivery_date	TIMESTAMP	NULLABLE				

For Payments Table

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags 	Description
<input type="checkbox"/>	order_id	STRING	NULLABLE				
<input type="checkbox"/>	payment_sequential	INTEGER	NULLABLE				
<input type="checkbox"/>	payment_type	STRING	NULLABLE				
<input type="checkbox"/>	payment_installments	INTEGER	NULLABLE				
<input type="checkbox"/>	payment_value	FLOAT	NULLABLE				

For Products Table

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags ?	Description
<input type="checkbox"/>	product_id	STRING	NULLABLE				
<input type="checkbox"/>	product_category	STRING	NULLABLE				
<input type="checkbox"/>	product_name_length	INTEGER	NULLABLE				
<input type="checkbox"/>	product_description_length	INTEGER	NULLABLE				
<input type="checkbox"/>	product_photos_qty	INTEGER	NULLABLE				
<input type="checkbox"/>	product_weight_g	INTEGER	NULLABLE				
<input type="checkbox"/>	product_length_cm	INTEGER	NULLABLE				
<input type="checkbox"/>	product_height_cm	INTEGER	NULLABLE				
<input type="checkbox"/>	product_width_cm	INTEGER	NULLABLE				

For Sellers Table

<input type="checkbox"/>	Field name	Type	Mode	Collation	Default Value	Policy Tags ?	Description
<input type="checkbox"/>	seller_id	STRING	NULLABLE				
<input type="checkbox"/>	seller_zip_code_prefix	INTEGER	NULLABLE				
<input type="checkbox"/>	seller_city	STRING	NULLABLE				
<input type="checkbox"/>	seller_state	STRING	NULLABLE				

Q1.2→ Time period for which the data is given.

Ans:-- **SELECT MIN(order_purchase_timestamp),
MAX(order_purchase_timestamp)
FROM 'target-e-commerce-372814.Target_SQL.orders'**

Row	f0_	f1_
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Explanation:

MIN syntax will get oldest order date from table and that will be the first order.

MAX syntax will get most recent order date from table and that will be last order.

Q1.3→ Cities and State of customers ordered during the given period.

```
SELECT S.CUSTOMER_ID,  
S.CUSTOMER_CITY,  
S.CUSTOMER_STATE  
FROM TARGET-E-COMMERCE-372814.TARGET_SQL.CUSTOMERS AS S  
LEFT JOIN TARGET-E-COMMERCE-372814.TARGET_SQL.ORDERS AS O  
ON S.CUSTOMER_ID=O.CUSTOMER_ID  
WHERE ORDER_ID BETWEEN "2016-09-04" AND "2018-10-17"
```

Ans:--

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	customer_id	customer_city	customer_state			
1	cc1728aef35e20436d830cab...	santos	SP			
2	41f7392232f044ac07529df633...	niteroi	RJ			
3	b64106d2d4647c03dc631238...	queimadas	BA			
4	1cf7dea919ba9ca8bb252c4da...	sao paulo	SP			
5	597f5c79639cb341754f01842...	sao paulo	SP			
6	a9cb426b0db0ee6d1ba8163ac...	rio de janeiro	RJ			
7	bbc8c12f7afeba03337723117f...	santana de parnaiba	SP			
8	d86a084c7fa7102fa0a6ca64c...	sao joao dos pinhal	SP			

Q2.1→ Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario?
Can we see some seasonality with peak at specific month?

Ans:--

```
SELECT
EXTRACT(YEAR FROM ORDER_PURCHASE_TIMESTAMP) AS YEAR,
EXTRACT(MONTH FROM ORDER_PURCHASE_TIMESTAMP) AS MONTH,
COUNT(ORDER_ID) AS ORDER_COUNT
FROM TARGET_SQL.ORDER
GROUP BY YEAR, MONTH
ORDER BY YEAR, MONTH
```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	year	month	order_count			
1	2016	9	4			
2	2016	10	324			
3	2016	12	1			
4	2017	1	800			
5	2017	2	1780			
6	2017	3	2682			
7	2017	4	2404			

Q2.2→ What time do Brazilian customers tends to buy (Drawn,Morning,Evening,Night)

Ans:--

```

select
CASE when EXTRACT(hour from orders.order_purchase_timestamp) between 0 and 6 then 'Before Dawn'
when EXTRACT(hour from orders.order_purchase_timestamp) between 6 and 12 then 'Morning'
when EXTRACT(hour from orders.order_purchase_timestamp) between 12 and 18 then 'Afternoon'
else 'Night'
END as time_of_day,
count(orders.order_id) as no_of_orders
from `target-e-commerce-372814.Target_SQL.orders` as orders
group by time_of_day
order by no_of_orders;

```

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH	PREVIEW
Row	time_of_day	no_of_orders				
1	Before Dawn	5242				
2	Morning	27733				
3	Night	28331				
4	Afternoon	38135				

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

Q3.1→ Get month on month orders by state

Ans:--

```

WITH orders_state AS (SELECT DISTINCT order_id, EXTRACT(MONTH FROM order_purchase_timestamp) AS Month, EXTRACT(YEAR FROM order_purchase_timestamp) AS Year, customer_state AS State
FROM `Target.orders` as o
INNER JOIN `Target.customers` as c ON o.customer_id = c.customer_id)

SELECT COUNT(order_id) AS orders_count, Month, Year, State
FROM orders_state
GROUP BY Month, Year, State
ORDER BY Year, Month, State

```

Row	orders_count	Month	Year	State
1	1	9	2016	RR
2	1	9	2016	RS
3	2	9	2016	SP
4	2	10	2016	AL
5	4	10	2016	BA
6	8	10	2016	CE
7	6	10	2016	DF
8	4	10	2016	ES
9	9	10	2016	GO
10	4	10	2016	MA

Q3.2→ Distribution of customers across the states of Brazil

Ans:--

```
SELECT COUNT(customer_unique_id) AS customers_count, customer_state AS State
FROM `Target.customers`
GROUP BY customer_state
ORDER BY State
```

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

4→ Impact of economy: Analyze the money movement by e-commerce by looking at order prices, freight and others

Q4.1→ Get %increase in cost of order from 2017 to 2018 (include month between JAN to AUG only)-you can use "payment_value" column in the payment table

Ans:--

```
WITH cost AS (SELECT payment_value, EXTRACT(MONTH FROM order_purchase_timestamp)
  AS Month, EXTRACT(YEAR FROM order_purchase_timestamp) AS Year
FROM `Target.orders` AS o
INNER JOIN `Target.payments` AS p ON o.order_id = p.order_id),
```

```
cost_year AS (SELECT ROUND(SUM(payment_value),2) AS payment_sum, Year
FROM cost
WHERE Month BETWEEN 1 AND 8 AND Year BETWEEN 2017 AND 2018
GROUP BY Year
ORDER BY Year)
```

```
SELECT percent_change
FROM (
SELECT ROUND(((payment_sum/LAG(payment_sum,1) OVER(ORDER BY Year))*100),2) AS pe
rcent_change, Year
FROM cost_year)
WHERE percent_change IS NOT NULL
```

Row	percent_change
1	236.98

Q4.2→ Mean & Sum of price and freight value by customer state

Ans→

```
SELECT
  ROUND(AVG(i.price),2) AS mean_price,
  ROUND(SUM(i.price),2) AS sum_price,
  ROUND(AVG(i.freight_value),2) AS mean_freight_value,
  ROUND(SUM(i.freight_value),2) AS sum_freight_value,
  c.customer_state AS State
FROM `Target.order_items` AS i
INNER JOIN `Target.orders` AS o ON i.order_id = o.order_id
INNER JOIN `Target.customers` AS c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY c.customer_state
```


Row	mean_price	sum_price	mean_freight_value	sum_freight_value	State
1	173.73	15982.95	40.07	3685.75	AC
2	180.89	80314.81	35.84	15914.59	AL
3	135.5	22356.84	33.21	5478.89	AM
4	164.32	13474.3	34.01	2788.5	AP
5	134.6	511349.09	26.36	100156.68	BA
6	153.76	227254.71	32.71	48351.59	CE
7	125.77	302603.94	21.04	50625.5	DF
8	121.91	275037.31	22.06	49754.6	ES
9	126.27	294591.95	22.77	53114.98	GO
10	145.2	119648.22	38.26	31523.77	MA

Q5→ Analysis on sales, freight and delivery time

Q5.1→ Calculate days between purchasing, delivering and estimated delivery

Ans:--

```
SELECT
    order_id,
    DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
days_delivery_purchase,
    DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp, DAY) AS
days_estimated_purchase,
    DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY
) AS days_estimated_delivery,
FROM `Target.orders`
ORDER BY days_estimated_delivery DESC
```

Row	order_id	days_delivery_pi	days_estimated	days_estimated
1	0607f0efea4b566f1eb8f7d3c2...	3	149	146
2	c72727d29cde4cf870d569bf6...	6	146	139
3	eec7f369423b033e549c02f3c...	20	155	134
4	c2bb89b5c1dd978d507284be...	16	140	123
5	40dc2ba6f322a17626aac6244...	7	116	108
6	1a695d543b7302aa9446c8d5f...	12	95	83
7	39e0115911bf404857e14baa7...	11	94	82
8	38930f76efb00b138f4d632e4d...	11	88	77
9	c5132855100a12d63ed4e8ae0...	12	89	77
10	559eea5a72341a4c82dbce988...	13	90	77

Q5.2→ Find time_to_delivery and diff_estimated_delivery. Formula for the same given below:

a) Time_to_delivery=order_purchases_timestamp – order_delivered_customer_date

Ans:--

```
SELECT
    order_id,DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp
,DAY) AS time_to_delivery
FROM `Target.orders`
ORDER BY time_to_delivery DESC
```

Row	order_id	time_to_delivery
1	ca07593549f1816d26a572e06...	209
2	1b3190b2dfa9d789e1f14c05b...	208
3	440d0d17af552815d15a9e41a...	195
4	0f4519c5f1c541ddec9f21b3bd...	194
5	285ab9426d6982034523a855f...	194
6	2fb597c2f772eca01b1f5c561b...	194
7	47b40429ed8cce3aee9199792...	191
8	2fe324feb907e3ea3f2aa9650...	189
9	2d7561026d542c8dbd8f0daea...	188
10	437222e3fd1b07396f1d9ba8c...	187

b) $\text{Diff_estimated_delivery} = \text{order_estimated_delivery_date} - \text{order_delivered_customer_date}$

Ans:--

```
SELECT
    order_id,
    DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY
) AS diff_estimated_delivery
FROM `Target.orders`
ORDER BY diff_estimated_delivery DESC
```

Row	order_id	diff_estimated_delivery
1	0607f0efea4b566f1eb8f7d3c2...	146
2	c72727d29cde4cf870d569bf6...	139
3	eec7f369423b033e549c02f3c...	134
4	c2bb89b5c1dd978d507284be...	123
5	40dc2ba6f322a17626aac6244...	108
6	1a695d543b7302aa9446c8d5f...	83
7	39e0115911bf404857e14baa7...	82
8	38930f76efb00b138f4d632e4d...	77
9	c5132855100a12d63ed4e8ae0...	77
10	559eea5a72341a4c82dbce988...	77

Q5.3→ Group data by state, taken mean of freight_value, time_to_delivery, diff_estimated_delivery.

Ans:--

```

SELECT
    ROUND(AVG(i.freight_value),2) AS mean_freight_value,
    ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestam
p, DAY)),2) AS time_to_delivery,
    ROUND(AVG(DATE_DIFF(order_estimated_delivery_date, order_delivered_custome
r_date, DAY)),2) AS diff_estimated_delivery,
    c.customer_state AS State
FROM `Target.order_items` AS i
INNER JOIN `Target.orders` AS o ON i.order_id = o.order_id
INNER JOIN `Target.customers` AS c ON o.customer_id = c.customer_id
GROUP By c.customer_state
ORDER BY c.customer_state

```

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

Q5.4→ Sort the data to get the following:

- Top 5 state with heights/lowest average freight value and sort in desc/asc limit 5

Ans:--

```

SELECT
    ROUND(AVG(i.freight_value),2) AS mean_freight_value,
    ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestam
p, DAY)),2) AS time_to_delivery,
    ROUND(AVG(DATE_DIFF(order_estimated_delivery_date, order_delivered_custome
r_date, DAY)),2) AS diff_estimated_delivery,
    c.customer_state AS State
FROM `Target.order_items` AS i
INNER JOIN `Target.orders` AS o ON i.order_id = o.order_id
INNER JOIN `Target.customers` AS c ON o.customer_id = c.customer_id
GROUP By c.customer_state
ORDER BY mean_freight_value ASC
LIMIT 5

```

Row	mean_freight_value	time_to_delivery	diff_estimated_delivery	State
1	15.15	8.26	10.27	SP
2	20.53	11.48	12.53	PR
3	20.63	11.52	12.4	MG
4	20.96	14.69	11.14	RJ
5	21.04	12.5	11.27	DF

b) Top 5 state with highest/lowest average time to delivery

Ans:--

```
SELECT
    ROUND(AVG(i.freight_value),2) AS mean_freight_value,
    ROUND(AVG(DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY)),2) AS time_to_delivery,
    ROUND(AVG(DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY)),2) AS diff_estimated_delivery,
    c.customer_state AS State
FROM `Target.order_items` AS i
INNER JOIN `Target.orders` AS o ON i.order_id = o.order_id
INNER JOIN `Target.customers` AS c ON o.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY time_to_delivery ASC
LIMIT 5
```

Row	mean_freight_value	time_to_delivery	diff_estimated_delivery	State
1	15.15	8.26	10.27	SP
2	20.53	11.48	12.53	PR
3	20.63	11.52	12.4	MG
4	21.04	12.5	11.27	DF
5	21.47	14.52	10.67	SC

Q6→ Payment type analysis:

Q6.1→Month over month count of orders for different payment types.

Ans:--

```

WITH cte AS (
SELECT
    p.order_id AS order_id, payment_type,
    EXTRACT(MONTH FROM order_purchase_timestamp) AS Month,
    EXTRACT(YEAR FROM order_purchase_timestamp) AS Year
FROM `Target.payments` AS p
INNER JOIN `Target.orders` as o ON p.order_id = o.order_id)

SELECT COUNT(order_id) AS order_count, payment_type, Month, Year
FROM cte
GROUP BY payment_type, Month, Year
ORDER BY Year, Month, payment_type

```

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

Q6.2→ Count of order based on the no. of payment instalments

Ans:--

```

SELECT COUNT(p.order_id) AS orders_count, payment_installments
FROM `Target.payments` AS p
INNER JOIN `Target.orders` as o ON p.order_id = o.order_id
GROUP BY payment_installments
ORDER BY payment_installments

```

Row	orders_count	payment_installments
1	2	0
2	52546	1
3	12413	2
4	10461	3
5	7098	4
6	5239	5
7	3920	6
8	1626	7
9	4268	8
10	644	9