TARGET SQL QUERY

Problem Statement:

Analyze the target dataset to extract valuable insights and provide actionable recommendations.

- 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.

Query

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

Sample Output

Query results

JOB IN	IFORMATION	RESULTS	JSON	EXECUTION DET	TAILS	EXECUTION GRAPH
Row	column_name		data_type ▼	11		
1	customer_id		STRING			
2	customer_unique	e_id	STRING			
3	customer_zip_co	de_prefix	INT64			
4	customer_city		STRING			
5	customer_state		STRING			

Insights:

Customer_id and customer_unique_id are in string. So can be converted into integer for further manipulation if required.

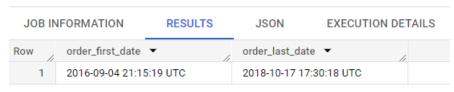
2. Get the time range between which the orders were placed.

Query

```
SELECT
  MIN(order_purchase_timestamp) AS order_first_date,
  MAX(order_purchase_timestamp) AS order_last_date
FROM
`target.orders`
```

Sample Output

Query results



Insights:

The order range is around two years.

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

Query

SELECT
 COUNT(DISTINCT customer_city) AS num_of_cities,
 COUNT(DISTINCT customer_state) AS num_of_states,
FROM
`target.customers`

Query results JOB INFORMATION RESULTS JSON Row num_of_cities num_of_states 1 4119 27

Insights:

There are around 4119 cities and 27 states in the dataset

2. In-depth Exploration:

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

```
SELECT
EXTRACT(YEAR FROM order_purchase_timestamp) AS year_of_purchase,
EXTRACT(MONTH FROM order_purchase_timestamp) AS month_of_purchase,
COUNT(*) AS order_count
FROM
`target.orders`
GROUP BY year_of_purchase,month_of_purchase
ORDER BY year_of_purchase,month_of_purchase
```

Quer	y results		
JOB IN	IFORMATION	RESULTS JS0	N EXECUTION
Row	year_of_purchase 🔻	month_of_purchase	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
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
1.4	2017	11	7511

Insights

September, October, November, December of 2016 and September, October of 2018 has less number of orders

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 month_of_purchase,
COUNT(*) AS order_count
FROM
`target.orders`
GROUP BY month_of_purchase
ORDER BY order_count DESC
```

Query results

JOB IN	IFORMATION	RESULTS	JSON
Row	month_of_purchase	order_count	· //
1	8		10843
2	5		10573
3	7		10318
4	3		9893
5	6		9412
6	4		9343
7	2		8508
8	1		8069
9	11		7544
10	12		5674
11	10		4959
12	9		4305

Insights

May, July, August month has highest number of orders.

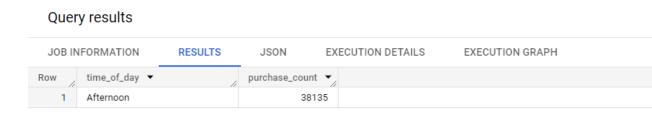
3. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

0-6 hrs: Dawn
7-12 hrs: Mornings
13-18 hrs: Afternoon
19-23 hrs: Night

Query

```
SELECT x.time_of_day, count(x.time_of_day) AS purchase_count
FROM(
SELECT
EXTRACT(HOUR FROM order_purchase_timestamp) AS hour_of_day,
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 'Morning'
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 time_of_day
FROM
'target.orders') AS x
GROUP BY x.time_of_day
ORDER BY purchase_count DESC
LIMIT 1
```

Sample Output



Insights

Most of the Brazilian customer do their purchase in the afternoon

- 3. Evolution of E-commerce orders in the Brazil region:
 - 1. Get the month on month no. of orders placed in each state.

Query

```
SELECT c.customer_state,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year_of_purchase,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month_of_purchase,
COUNT(*) AS order_count
FROM
`target.customers` AS c
JOIN `target.orders` AS o
ON c.customer_id = o.customer_id
GROUP BY c.customer_state, year_of_purchase, month_of_purchase
ORDER BY c.customer_state, year_of_purchase, month_of_purchase
```

Sample Output

Query results

JOB IN	IFORMATION	RESULTS	JSON EXI	ECUTION DETAILS	EXECUTION GRAP
Row	customer_state	• //	year_of_purchase 🍷	month_of_purchase	order_count ▼
1	AC		2017	1	2
2	AC		2017	2	3
3	AC		2017	3	2
4	AC		2017	4	5
5	AC		2017	5	8
6	AC		2017	6	4
7	AC		2017	7	5
8	AC		2017	8	4
9	AC		2017	9	5
10	AC		2017	10	6
11	AC		2017	11	5
12	AC		2017	12	5

INSIGHTS:

State BA and RJ has highest number of orders

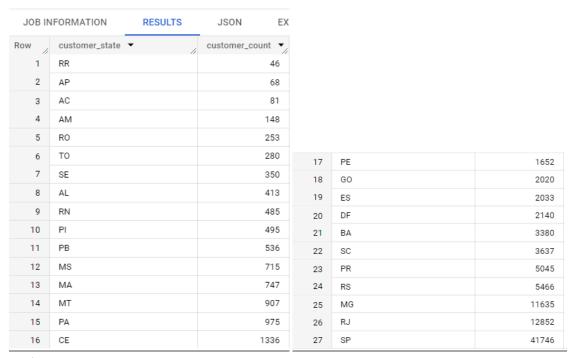
2. How are the customers distributed across all the states?

Query

```
SELECT customer_state, COUNT(*) AS customer_count
FROM `target.customers`
GROUP BY customer_state
ORDER BY customer_count
```

Sample Output

Query results



Insights

Largest number of customer from SP and lowest number of customer from RR

- 4. Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.
 - 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 y2017_summary AS(
    SELECT ROUND(SUM(p.payment_value),2) AS cost_of_order_2017
    FROM
    `target.payments` AS p
    JOIN `target.orders`AS o
    ON p.order_id = o.order_id
    WHERE (EXTRACT(YEAR FROM order_purchase_timestamp)= 2017) AND (EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8)),
    y2018_summary AS(
    SELECT ROUND(SUM(p.payment_value),2) AS cost_of_order_2018
```

```
FROM
`target.payments` AS p
JOIN `target.orders`AS o
ON p.order_id = o.order_id
WHERE (EXTRACT(YEAR FROM order_purchase_timestamp) = 2018) AND (EXTRACT(MONTH FROM order_purchase_timestamp) BETWEEN 1 AND 8))

SELECT
ROUND(((y18.cost_of_order_2018 - y17.cost_of_order_2017)/
y17.cost_of_order_2017)*100,2) AS per_cost_increase
FROM y2017_summary AS y17,
y2018_summary AS y18
```

Output for the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).



Output for the cost of order in 2017 from Jan to Aug



Output for the cost of order in 2018 from Jan to Aug

Query results

JOB IN	IFORMATION	RESULTS	JSON	EXECUTION DETAILS	EXECUTION GRAPH
Row	cost_of_order_201	8			
1	8694733.84				

Insights

In Brazilian country the there is 136.98% increase in cost happened from 2017 to 2018 in the month January to August

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

```
SELECT c.customer_state,
ROUND(SUM(oi.price),2) AS Total_order_price,
ROUND(AVG(oi.price),2) AS Average_order_price,
FROM `target.customers` AS c
JOIN `target.orders` AS o
ON c.customer_id = o.customer_id
JOIN `target.order_items` AS oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
```

Query results

JOB IN	IFORMATION RESULTS	JSON EX	ECUTION DETAILS	EXECUTION GRAPH
Row	customer_state ▼	Total_order_price 🔻	Average_order_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	ВА	511349.99	134.6	
8	DF	302603.94	125.77	
9	GO	294591.95	126.27	
10	ES	275037.31	121.91	

Insights

- Sao Paulo (SP) in brazil has highest total order price with the value of 5202955 and Roraima(RR) has lowest total cost price with the value of 7829
- Paraiba(PB) has highest average order price with the value of 191 and SP has lowest average order price with the value of 109
- Even though SP has more total cost than other state, since the average value is low, it indicates more number of customers made each order with low value of cost

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

```
SELECT c.customer_state,
ROUND(SUM(oi.freight_value),2) AS Total_order_freight,
ROUND(AVG(oi.freight_value),2) AS Average_order_freight,
FROM `target.customers` AS c
JOIN `target.orders` AS o
ON c.customer_id = o.customer_id
JOIN `target.order_items` AS oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY Total_order_freight DESC
```

Query results

JOB IN	IFORMATION RESULTS		JSON EX	ECUTION DETAILS	EXECUT
Row	customer_state ▼	//	Total_order_freight	Average_order_freight ▼	/
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	

Insights

- Sao Paulo (SP) in brazil has highest total order price with the value of 718723 and Roraima(RR) has lowest total cost price with the value of 2235
- Roraima(RR) has highest average order price with the value of 42.98 and SP has lowest average order price with the value of 15.15

5. 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.

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_estimated_delivery_date order_delivered_customer_date

Query 1

SELECT

order_id,customer_id,order_purchase_timestamp,order_estimated_delivery_date,order_delivered_customer_date,

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.orders`
WHERE order_delivered_customer_date IS NOT NULL
ORDER BY time_to_deliver DESC
```

Quer	y results				≛ SA	VE RESULTS ▼	EXPLORE DATA
JOB IN	FORMATION RESULTS	JSON EXECUTION DE	TAILS EXECUTION GRAPH				
Row	order_id ▼	customer_id ▼	order_purchase_timestamp ▼	order_estimated_delivery_date 🔻	order_delivered_customer_date 🔻	time_to_deliver ▼	diff_estimated_deliv
1	ca07593549f1816d26a572e06	75683a92331068e2d281b11a	2017-02-21 23:31:27 UTC	2017-03-22 00:00:00 UTC	2017-09-19 14:36:39 UTC	209	-181
2	1b3190b2dfa9d789e1f14c05b	d306426abe5fca15e54b645e4	2018-02-23 14:57:35 UTC	2018-03-15 00:00:00 UTC	2018-09-19 23:24:07 UTC	208	-188
3	440d0d17af552815d15a9e41a	7815125148cfa1e8c7fee1ff79	2017-03-07 23:59:51 UTC	2017-04-07 00:00:00 UTC	2017-09-19 15:12:50 UTC	195	-165
4	0f4519c5f1c541ddec9f21b3bd	1a8a4a30dc296976717f44e78	2017-03-09 13:26:57 UTC	2017-04-11 00:00:00 UTC	2017-09-19 14:38:21 UTC	194	-161
5	285ab9426d6982034523a855f	9cf2c3fa2632cee748e1a59ca9	2017-03-08 22:47:40 UTC	2017-04-06 00:00:00 UTC	2017-09-19 14:00:04 UTC	194	-166
6	2fb597c2f772eca01b1f5c561b	217906bc11a32c1e470eb7e08	2017-03-08 18:09:02 UTC	2017-04-17 00:00:00 UTC	2017-09-19 14:33:17 UTC	194	-155
7	47b40429ed8cce3aee9199792	cb2caaaead400c97350c37a3f	2018-01-03 09:44:01 UTC	2018-01-19 00:00:00 UTC	2018-07-13 20:51:31 UTC	191	-175
8	2fe324febf907e3ea3f2aa9650	65b14237885b3972ebec28c0f	2017-03-13 20:17:10 UTC	2017-04-05 00:00:00 UTC	2017-09-19 17:00:07 UTC	189	-167
9	2d7561026d542c8dbd8f0daea	8199345f57c6d1cbe9701f924	2017-03-15 11:24:27 UTC	2017-04-13 00:00:00 UTC	2017-09-19 14:38:18 UTC	188	-159
10	437222e3fd1b07396f1d9ba8c	9b39de85d94d55a21991e70b	2017-03-16 11:36:00 UTC	2017-04-28 00:00:00 UTC	2017-09-19 16:28:58 UTC	187	-144

Insights

Maximum number of days it has taken to deliver is 209 and it has taken 188 more days from the estimated date.

Query 2

SELECT

order_id,customer_id,order_purchase_timestamp,order_estimated_delivery_date,order_delivered_customer_date,

DATE_DIFF(order_delivered_customer_date_order_purchase_timestamp_DAY) AS

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

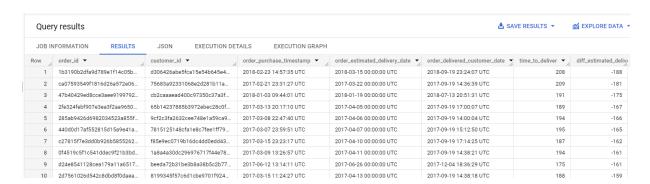
FROM

`target.orders`

WHERE order_delivered_customer_date IS NOT NULL AND

DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY) < 0
ORDER BY diff_estimated_delivery

SAMPLE OUTPUT



INSIGHTS

There are 6535 out of 96476(6%) order are delivered after estimated date

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

Query 1:Top 5 states with highest average freight value

```
SELECT c.customer_state,
ROUND(AVG(oi.freight_value),2) AS Average_order_freight,
FROM `target.customers` AS c
JOIN `target.orders` AS o
ON c.customer_id = o.customer_id
JOIN `target.order_items` AS oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY Average_order_freight DESC
LIMIT 5
```

Sample Output

Quer	y results			
JOB IN	FORMATION	RESULTS	JSON	EXE
Row	customer_state	•	Average_orde	r_freigh
1	RR		4	42.98
2	PB		4	42.72
3	RO		4	41.07
4	AC		4	40.07
5	PI			39.15

<u>Insights</u>

RR, PB, RO, AC, PI has highest average freight value

Query 2: Top 5 states with lowest average freight value

```
SELECT c.customer_state,
ROUND(AVG(oi.freight_value),2) AS Average_order_freight,
FROM `target.customers` AS c
JOIN `target.orders` AS o
ON c.customer_id = o.customer_id
JOIN `target.order_items` AS oi
ON o.order_id = oi.order_id
GROUP BY c.customer_state
ORDER BY Average_order_freight
LIMIT 5
```

Sample Output Ouerv results

JOB IN	FORMATION	RESULTS	JSON	EXE
Row	customer_state	~	Average_ord	er_freigl
1	SP			15.15
2	PR			20.53
3	MG			20.63
4	RJ			20.96
5	DF			21.04

Insights

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

Query 1: Top 5 states with the highest average delivery time.

Sample Output

Quer	y results				
JOB IN	IFORMATION	RESULTS		JSON	EXE
Row	customer_state	~	/	average_deliv	ery_tim
1	RR			2	28.98
2	AP			2	26.73
3	AM			2	25.99
4	AL			2	24.04
5	PA			2	23.32

Insights:

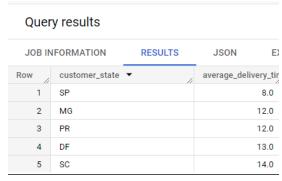
Top 5 states with the highest average delivery time are RR,AP,AM,AL,PA

Query 2: Top 5 states with the lowest average delivery time.

```
WITH cte AS(
SELECT order_id, customer_id, order_purchase_timestamp, order_delivered_customer_date,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, DAY) AS
time_to_deliver,
FROM
   `target.orders`
WHERE order_delivered_customer_date IS NOT NULL
)

SELECT c.customer_state,
ROUND(AVG(ct.time_to_deliver)) AS average_delivery_time
FROM cte AS ct
```

```
JOIN `target.customers` AS c
ON ct.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY average_delivery_time
LIMIT 5
```



<u>Insights</u>

Top 5 states with the lowest average delivery time are SP, PR, MG, DF, SC

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.

```
WITH cte AS(
SELECT
order_id,customer_id,order_purchase_timestamp,order_estimated_delivery_date,order_deli
vered_customer_date,
DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY) AS
diff_estimated_delivery
FROM
`target.orders`
WHERE order_delivered_customer_date IS NOT NULL
)
SELECT c.customer_state,
ROUND(AVG(ct.diff_estimated_delivery)) AS delivery_time
FROM cte AS ct
JOIN `target.customers` AS c
ON ct.customer_id = c.customer_id
GROUP BY c.customer_state
ORDER BY delivery_time
LIMIT 5
```

Query results

Row customer_state ▼ delivery_time ▼
1 AL 8.0
2 MA 9.0
3 SE 9.0
4 ES 10.0
5 SP 10.0

Insights

Top 5 states in Brazil where average of delivery date is approximately equal to delivery estimated date are AL,MA,SE,ES,SP

- 6. Analysis based on the payments:
 - 1. Find the month on month no. of orders placed using different payment types.

Query

```
SELECT p.payment_type,
EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year_of_purchase,
EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month_of_purchase,
COUNT(*) AS order_count
FROM
   `target.payments` AS p
JOIN `target.orders` AS o
ON p.order_id = o.order_id
GROUP BY p.payment_type, year_of_purchase, month_of_purchase
ORDER BY p.payment_type, year_of_purchase, month_of_purchase
Sample Output
```

Query results

JOB INFORMATION RESULTS		JSON EX	JSON EXECUTION DETAILS	
Row	payment_type ▼	year_of_purchase	month_of_purchase	order_count ▼
1	UPI	2016	10	63
2	UPI	2017	1	197
3	UPI	2017	2	398
4	UPI	2017	3	590
5	UPI	2017	4	496
6	UPI	2017	5	772
7	UPI	2017	6	707
8	UPI	2017	7	845
9	UPI	2017	8	938
10	UPI	2017	9	903
11	UPI	2017	10	993
12	UPI	2017	11	1509
13	UPI	2017	12	1160
14	HPI	2018	1	1518

Insights

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

Query

```
SELECT p.payment_installments,

COUNT(*) AS order_count

FROM

`target.payments` AS p

JOIN `target.orders` AS o

ON p.order_id = o.order_id

GROUP BY p.payment_installments

ORDER BY order_count DESC
```

Sample Output

Query results

JOB IN	FORMATION	RESULTS JS	ON
Row	payment_installment	order_count ▼	/
1	1	52546	
2	2	12413	
3	3	10461	
4	4	7098	
5	10	5328	
6	5	5239	
7	8	4268	
8	6	3920	
9	7	1626	
10	9	644	
11	12	133	

<u>Insights</u>

Most of the orders prefer 1,2 3 4 installments only

Recommendations:

- Since most of the Brazilian customers purchase products in the afternoon, we can predict that most of the orders are done by homemakers so household products can be increased to increase the sale.
- Largest number of customers are from the state São Paulo (SP), so a greater number of delivery units can be placed over there.
- The number of orders has increased from 2017 to 2018 by 136 %, we can say it will increase in 2018 also.
- 6% of order delivered after estimated time so need to take care of it.