

## Target SQL (Project -1)

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset
  1. Data type of columns in a table

**QUERY:** `select column_name, data_type from  
`target-sql-380716.TargetDataset.INFORMATION_SCHEMA.COLUMNS`  
where table_name = 'customers' LIMIT 10;`

**Output:**

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS
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		

2. Time period for which the data is given

**Query:** `SELECT min(order_purchase_timestamp) as StartDate,  
max(order_purchase_timestamp) as EndDate  
FROM `target-sql-380716.TargetDataset.orders` LIMIT 10;`

**Output:**

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	EXECUTION TIME
Row	StartDate	EndDate			
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC			

3. Cities and States of customers ordered during the given period

**Query:** `With a as(select min(order_purchase_timestamp) as min_ts,  
max(order_purchase_timestamp) as max_ts  
from `target-sql-380716.TargetDataset.orders` )  
SELECT distinct c.customer_city,c.customer_state  
FROM `target-sql-380716.TargetDataset.customers` as c  
inner join `target-sql-380716.TargetDataset.orders` as o on o.customer_id = c.customer_id,a  
where o.order_purchase_timestamp >= a.min_ts  
and o.order_purchase_timestamp <= a.max_ts  
LIMIT 10`

**Output:**

Row	customer_city	customer_state
1	acu	RN
2	ico	CE
3	ipe	RS
4	ipu	CE
5	ita	SC
6	itu	SP
7	jau	SP
8	luz	MG
9	poa	SP
10	uba	MG

## 2. In-depth Exploration:

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

**Query:** `SELECT count(distinct o.order_id) as count_of_orders,  
EXTRACT(month FROM o.order_purchase_timestamp) as month FROM `target-sql-380716.TargetDataset.orders` as o  
inner join `target-sql-380716.TargetDataset.customers` as c on c.customer_id = o.customer_id  
group by month order by month LIMIT 10`

**Output:**

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

2. What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

**Query:** `SELECT part_of_day,  
COUNT(order_id) AS Number_of_order FROM (SELECT *,  
CASE  
WHEN time_ BETWEEN time "00:00:00" AND "06:00:00" THEN "Dawn"  
WHEN time_ BETWEEN time "06:00:01" AND "12:00:00" THEN "Morning"  
WHEN time_ BETWEEN time "12:00:01" AND "18:00:00" THEN "Afternoon"  
WHEN time_ BETWEEN time "18:00:01" AND "23:59:59" THEN "Night"  
END AS part_of_day  
FROM (SELECT order_id, order_purchase_timestamp,  
EXTRACT(time FROM order_purchase_timestamp) AS time_ FROM  
(SELECT DISTINCT * FROM `target-sql-380716.TargetDataset.orders`)  
ORDER BY order_purchase_timestamp) AS y ) as a  
GROUP BY part_of_day  
ORDER BY COUNT(order_id)  
LIMIT 10;`

**Output:**

JOB INFORMATION		RESULTS	JSON
Row	part_of_day	Number_of_order	
1	Dawn	4740	
2	Morning	22240	
3	Night	34096	
4	Afternoon	38365	

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

#### 1. Get month on month orders by states

**Query:** `select c.customer_state,extract(year from o.order_purchase_timestamp) as Year,  
extract(month from o.order_purchase_timestamp) as Month,count(o.order_id) as count_of_orders  
from `target-sql-380716.TargetDataset.customers` c  
left join `target-sql-380716.TargetDataset.orders` o  
on o.customer_id = c.customer_id  
group by c.customer_state, Year,Month  
order by c.customer_state, Year,Month  
LIMIT 10;`

**Output:**

Row	customer_state	Year	Month	count_of_orders
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

#### 2. Distribution of customers across the states in Brazil

**Query:** `SELECT customer_state,count(distinct customer_id ) as customerid FROM `target-sql-380716.TargetDataset.customers`  
group by customer_state  
Limit 10;`

**Output:**

Row	customer_state	customerid
1	RN	485
2	CE	1336
3	RS	5466
4	SC	3637
5	SP	41746
6	MG	11635
7	BA	3380
8	RJ	12852
9	GO	2020
10	MA	747

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

#### 1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment\_value" column in payments table

**Query:** `With a as  
(SELECT sum(p.payment_value) as sumofPayments,  
EXTRACT(YEAR from o.order_approved_at) as year  
FROM `target-sql-380716.TargetDataset.orders` as o  
inner join `target-sql-380716.TargetDataset.payments` as p on p.order_id=o.order_id  
WHERE order_approved_at BETWEEN '2017-01-01' AND '2017-08-31'  
group by year),`

`b as(SELECT sum(p.payment_value) as sumofPayments,  
EXTRACT(YEAR from o.order_approved_at) as year  
FROM `target-sql-380716.TargetDataset.orders` as o`

```
inner join `target-sql-380716.TargetDataset.payments` as p on p.order_id=o.order_id
WHERE order_approved_at BETWEEN '2018-01-01' AND '2018-08-31'
group by year)
```

```
select sumofPayments,year from a As ak17
union all
select sumofPayments,year from b as bk18
LIMIT 10
```

**Output:**

Row	sumofPayments	year
1	8685333.07...	2018
2	3617410.23...	2017

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

**Query:** `SELECT AVG(ot.price) as price, sum(ot.freight_value) as sumoffreightvalue, c.customer_state FROM `target-sql-380716.TargetDataset.order_items` ot inner join `target-sql-380716.TargetDataset.orders` o on o.order_id = ot.order_id inner join `target-sql-380716.TargetDataset.customers` c on c.customer_id = o.customer_id group by c.customer_state LIMIT 10`

**Output:**

Row	price	sumoffreightval	customer_state
1	109.653629...	718723.069...	SP
2	125.117818...	305589.310...	RJ
3	119.004139...	117851.680...	PR
4	124.653577...	89660.2600...	SC
5	125.770548...	50625.4999...	DF
6	120.748574...	270853.460...	MG
7	165.692416...	38699.3000...	PA
8	134.601208...	100156.679...	BA
9	126.271731...	53114.9799...	GO
10	120.337453...	135522.740...	RS

## 5. Analysis on sales, freight and delivery time

### 1. Calculate days between purchasing, delivering and estimated delivery

**Query:** `SELECT order_id, DATE_DIFF(order_purchase_timestamp, order_delivered_customer_date, day) as delivery_difference, DATE_DIFF(order_purchase_timestamp, order_estimated_delivery_date, day) as estimated_delivery, FROM `target-sql-380716.TargetDataset.orders` limit 10`

**Output:**

Row	order_id	delivery_differen	estimated_deliv
1	7a4df5d8cff4090e541401a20a...	null	-16
2	35de4050331c6c644cddc86f4...	null	-33
3	b5359909123fa03c50bdb0cfe...	null	-36
4	dba5062fbd3af4fb6c33b1e04...	null	-25
5	90ab3e7d52544ec7bc3363c82...	null	-24
6	fa65dad1b0e818e3ccc5cb0e3...	null	-27
7	1df2775799eecd9dd8502425...	null	-31
8	6190a94657e1012983a274b8...	null	-33
9	58ce513a55c740a3a81e8c8b7...	null	-15
10	088683f795a3d30bfd61152c4f...	null	-31

2. Find time\_to\_delivery & diff\_estimated\_delivery. Formula for the same given below:
  - time\_to\_delivery = order\_purchase\_timestamp-order\_delivered\_customer\_date
  - diff\_estimated\_delivery = order\_estimated\_delivery\_date-order\_delivered\_customer\_date

**Query:** `SELECT order_id,  
DATE_DIFF(order_purchase_timestamp, order_delivered_customer_date, DAY) AS time_to_delivery,  
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS diff_estimated_delivery  
from `target-sql-380716.TargetDataset.orders`  
WHERE  
order_purchase_timestamp is not null  
and order_delivered_customer_date is not null  
and order_estimated_delivery_date is not null  
limit 10`

**Output:**

Row	order_id	time_to_delivery	diff_estimated_delivery
1	770d331c84e5b214bd9dc70a1...	-7	45
2	1950d777989f6a877539f5379...	-30	-12
3	2c45c33d2f9cb8ff8b1c86cc28...	-30	28
4	dabf2b0e35b423f94618bf965f...	-7	44
5	8beb59392e21af5eb9547ae1a...	-10	41
6	65d1e226dfaeb8cdc42f66542...	-35	16
7	c158e9806f85a33877bdfd4f60...	-23	9
8	b60b53ad0bb7dacacf2989fe2...	-12	-5
9	c830f223aae08493ebecb52f2...	-12	12
10	a8aa2cd070eeac7e4368cae3d...	-7	1

3. Group data by state, take mean of freight\_value, time\_to\_delivery, diff\_estimated\_delivery

**Query:** `SELECT c.customer_state,  
ROUND(AVG(freight_value), 2) as avg_of_freight_value,  
ROUND(AVG(DATE_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY)), 2) AS mean_time_to_delivery,  
ROUND(AVG(DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)), 2) AS mean_diff_estimated_delivery  
FROM `target-sql-380716.TargetDataset.orders` o  
inner join `target-sql-380716.TargetDataset.order_items` ot on o.order_id = ot.order_id  
inner join `target-sql-380716.TargetDataset.customers` c on c.customer_id = o.customer_id  
group by c.customer_state  
LIMIT 10`

**Output:**

Row	customer_state	avg_of_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
1	MT	28.17	17.51	13.64
2	MA	38.26	21.2	9.11
3	AL	35.84	23.99	7.98
4	SP	15.15	8.26	10.27
5	MG	20.63	11.52	12.4
6	PE	32.92	17.79	12.55
7	RJ	20.96	14.69	11.14
8	DF	21.04	12.5	11.27
9	RS	21.74	14.71	13.2
10	SE	36.65	20.98	9.17

4. Sort the data to get the following:
5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

**Query:** WITH state\_avg\_freight\_value AS ( SELECT c.customer\_state,  
ROUND(AVG(oi.freight\_value), 2) AS avg\_freight\_value  
FROM `target-sql-380716.TargetDataset.orders` AS o  
JOIN `target-sql-380716.TargetDataset.order\_items` AS oi  
ON o.order\_id = oi.order\_id  
JOIN `target-sql-380716.TargetDataset.customers` AS c  
ON o.customer\_id = c.customer\_id  
GROUP BY c.customer\_state  
)  
(SELECT  
"Top 5 States with Highest Average Freight Value" AS title,  
customer\_state,  
avg\_freight\_value  
FROM state\_avg\_freight\_value  
ORDER BY avg\_freight\_value DESC  
LIMIT 5)  
UNION ALL  
(SELECT  
"Top 5 States with Lowest Average Freight Value" AS title,  
customer\_state,  
avg\_freight\_value  
FROM state\_avg\_freight\_value  
ORDER BY avg\_freight\_value ASC  
LIMIT 5)

**Output:**

Row	title	customer_state	avg_freight_value
1	Top 5 States with Lowest Average Freight Value	SP	15.15
2	Top 5 States with Lowest Average Freight Value	PR	20.53
3	Top 5 States with Lowest Average Freight Value	MG	20.63
4	Top 5 States with Lowest Average Freight Value	RJ	20.96
5	Top 5 States with Lowest Average Freight Value	DF	21.04
6	Top 5 States with Highest Average Freight Value	RR	42.98
7	Top 5 States with Highest Average Freight Value	PB	42.72
8	Top 5 States with Highest Average Freight Value	RO	41.07
9	Top 5 States with Highest Average Freight Value	AC	40.07
10	Top 5 States with Highest Average Freight Value	PI	39.15

6. Top 5 states with highest/lowest average time to delivery

**Query:** WITH state\_avg\_time\_delivery AS ( SELECT c.customer\_state,  
ROUND(AVG(DATE\_DIFF(order\_purchase\_timestamp, order\_delivered\_customer\_date, DAY) ), 2)  
AS time\_to\_delivery  
FROM `target-sql-380716.TargetDataset.orders` AS o  
JOIN `target-sql-380716.TargetDataset.order\_items` AS oi  
ON o.order\_id = oi.order\_id  
JOIN `target-sql-380716.TargetDataset.customers` AS c  
ON o.customer\_id = c.customer\_id  
GROUP BY c.customer\_state  
)  
(SELECT  
"Top 5 States with Highest Average Time to delivery" AS title,  
customer\_state,

```

time_to_delivery
FROM state_avg_time_delivery
ORDER BY time_to_delivery DESC
LIMIT 5)
UNION ALL
(SELECT
"Top 5 States with Lowest Average Time to delivery" AS title,
customer_state,
time_to_delivery
FROM state_avg_time_delivery
ORDER BY time_to_delivery ASC
LIMIT 5)

```

Output:

Row	title	customer_state	time_to_delivery
1	Top 5 States with Lowest Average Time to delivery	RR	-27.83
2	Top 5 States with Lowest Average Time to delivery	AP	-27.75
3	Top 5 States with Lowest Average Time to delivery	AM	-25.96
4	Top 5 States with Lowest Average Time to delivery	AL	-23.99
5	Top 5 States with Lowest Average Time to delivery	PA	-23.3
6	Top 5 States with Highest Average Time to delivery	SP	-8.26
7	Top 5 States with Highest Average Time to delivery	PR	-11.48
8	Top 5 States with Highest Average Time to delivery	MG	-11.52
9	Top 5 States with Highest Average Time to delivery	DF	-12.5
10	Top 5 States with Highest Average Time to delivery	SC	-14.52

7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

**Query:** WITH state\_avg\_time\_delivery AS (  
SELECT c.customer\_state,  
ROUND(AVG(DATE\_DIFF(order\_purchase\_timestamp, order\_estimated\_delivery\_date, day)), 2) AS estimated\_delivery  
FROM `target-sql-380716.TargetDataset.orders` AS o  
JOIN `target-sql-380716.TargetDataset.order\_items` AS oi ON o.order\_id = oi.order\_id  
JOIN `target-sql-380716.TargetDataset.customers` AS c ON o.customer\_id = c.customer\_id  
GROUP BY c.customer\_state)  
(SELECT "Top 5 States with Fastest delivery" AS title, customer\_state, estimated\_delivery  
FROM state\_avg\_time\_delivery ORDER BY estimated\_delivery DESC LIMIT 5)  
UNION ALL  
(SELECT "Top 5 States with Not So Fast delivery" AS title, customer\_state, estimated\_delivery  
FROM state\_avg\_time\_delivery ORDER BY estimated\_delivery ASC LIMIT 5)

Output:

Row	title	customer_state	estimated_delivery
1	Top 5 States with Not So Fast delivery	RR	-45.98
2	Top 5 States with Not So Fast delivery	AP	-45.49
3	Top 5 States with Not So Fast delivery	AM	-45.21
4	Top 5 States with Not So Fast delivery	AC	-40.7
5	Top 5 States with Not So Fast delivery	RO	-38.65
6	Top 5 States with Fastest delivery	SP	-18.9
7	Top 5 States with Fastest delivery	DF	-24.19
8	Top 5 States with Fastest delivery	MG	-24.31
9	Top 5 States with Fastest delivery	PR	-24.38
10	Top 5 States with Fastest delivery	ES	-25.26

## 6. Payment type analysis:

### 1. Month over Month count of orders for different payment types

**Query:** SELECT  
(EXTRACT(year FROM o.order\_purchase\_timestamp)) AS year,  
(EXTRACT(month FROM o.order\_purchase\_timestamp)) AS month,  
p.payment\_type,  
COUNT( o.order\_id) AS order\_count  
FROM  
`target-sql-380716.TargetDataset.orders` o  
inner JOIN `target-sql-380716.TargetDataset.payments` p on o.order\_id = p.order\_id  
GROUP BY  
year,month,  
payment\_type  
ORDER BY  
year,month LIMIT 10

#### Output:

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

### 2. Count of orders based on the no. of payment installments

**Query:** SELECT payment\_installments,count(order\_id) as count\_of\_orders FROM `target-sql-380716.TargetDataset.payments`  
group by payment\_installments  
order by payment\_installments LIMIT 10

#### Output:

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