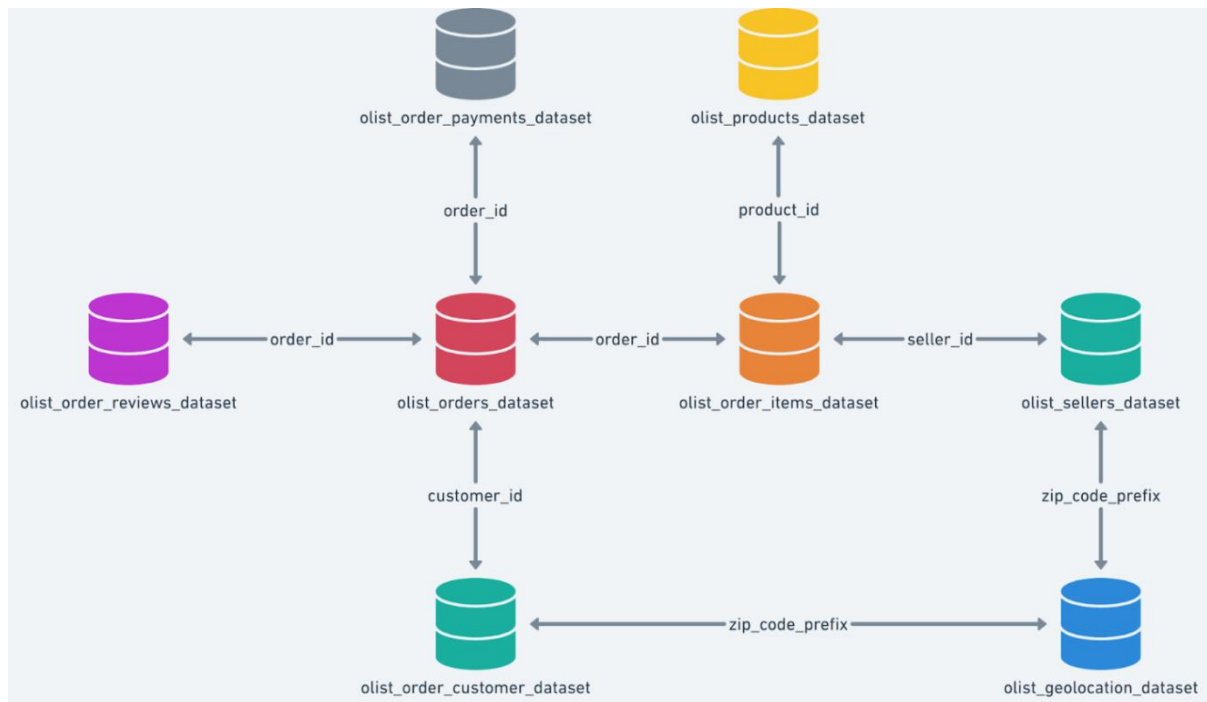


Business Case: Target SQL



Q1. 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 - Below is data related to orders table

Orders Table:

Field name	Type
order_id	STRING
customer_id	STRING
order_status	STRING
order_purchase_timestamp	TIMESTAMP
order_approved_at	TIMESTAMP
order_delivered_carrier_date	TIMESTAMP
order_delivered_customer_date	TIMESTAMP
order_estimated_delivery_date	TIMESTAMP

2. Time period for which the data is given

Time period is between 04th Sep-2016 and 17th Oct-2018

```
select min(order_purchase_timestamp) as min_time_stamp , max(order_purchase_timestamp) as max_time_stamp from `targetsql-sql.Target_SQL.orders`;
```

min_time_stamp	max_time_stamp
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

```
select distinct customer_city, customer_state from `targetsql-  
sql.Target_SQL.orders` o inner join `targetsql-  
sql.Target_SQL.customers` c on o.customer_id=c.customer_id;
```

Output of Top 10 rows below.

customer_city	customer_state
rio de janeiro	RJ
sao leopoldo	RS
general salgado	SP
brasilia	DF
paranavai	PR
cuiaba	MT
sao luis	MA
maceio	AL
hortolandia	SP
varzea grande	MT

Q2) 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?

```
select distinct count(order_id) over(partition by year, month order by year, month) as  
order_count, month, year  
  
from  
(  
    select order_id, extract(month from order_purchase_timestamp) as month, extract(year  
from order_purchase_timestamp) as year  
from `targetsql-sql.Target_SQL.orders` o inner join `targetsql-  
sql.Target_SQL.customers` c  
on o.customer_id = c.customer_id  
) a
```

order_count	month	year
4	9	2016
324	10	2016
1	12	2016
800	1	2017
1780	2	2017
2682	3	2017
2404	4	2017
3700	5	2017
3245	6	2017
4026	7	2017

order_count	year
329	2016
45101	2017
54011	2018

There is growing trend on e-commerce in terms of orders from 2016 to 2018. There is decrease in orders of Month December both for 2016 and 2017 years & drastic fall of orders in the month of Sep 2018

And there is a seasonal peak in the month of January of every year in given time period.

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

Dawn	00:00 AM to 06:00 AM
Morning	06:00 AM to 12:00 PM
Afternoon	12:00 PM to 18:00 PM
Night	18:00 PM to 24:00 PM

Brazil customer tend to buy more in Afternoon(12:00 to 18:00)

```
select count(order_id) as Order_count, Time_zone
from
(
select order_id,
order_purchase_timestamp,
Time_stamp,
CASE
WHEN Time_stamp > '00:00:00' and Time_stamp <= '06:00:00' then 'Dawn'
when Time_stamp > '06:00:00' and Time_stamp <= '12:00:00' then 'Morning'
when Time_stamp > '12:00:00' and Time_stamp <= '18:00:00' then 'Afternoon'
ELSE 'Night'
END as Time_zone
from
(
select order_id,order_purchase_timestamp,time(order_purchase_timestamp) as Time_s
tamp from `targetsql-sql.Target_SQL.orders`
) a
) b
group by Time_zone
ORDER BY CASE WHEN Time_zone = 'Dawn' THEN 1
when Time_zone = 'Morning' THEN 2
when Time_zone = 'Afternoon' THEN 3
when Time_zone = 'Night' THEN 4 END
```

Order_count	Time_zone
4739	Dawn
22240	Morning
38365	Afternoon
34097	Night

Q3) Evolution of E-commerce orders in the Brazil region:

1. Get month on month orders by states

Month on month orders (count) with respect to each state has been provided below:

```
select distinct customer_state ,
month,
year,
count(order_id) over(partition by customer_state,year,month order by customer_
state,year,month) as m_on_m_bystates
from
(
select distinct order_id,
extract(month from order_purchase_timestamp) as month,
extract(year from order_purchase_timestamp) as year,
customer_state
from `targetsql-sql.Target_SQL.orders` o inner join `targetsql-
sql.Target_SQL.customers` c
```

```

on o.customer_id=c.customer_id
order by customer_state,year,month
) a

```

customer_state	month	year	m_on_m_bystate
AC	1	2017	2
AC	2	2017	3
AC	3	2017	2
AC	4	2017	5
AC	5	2017	8
AC	6	2017	4
AC	7	2017	5
AC	8	2017	4
AC	9	2017	5
AC	10	2017	6

2. Distribution of customers across the states in Brazil

Count of customers from across each state of brazil as below:

```

select distinct customer_state,
count(customer_id) over(partition by customer_state order by customer_state)as
Customer_distribution_across_states
from `targetsql-sql.Target_SQL.customers`;

```

customer_state	Customer_distribution_across_states
AC	81
AL	413
AM	148
AP	68
BA	3380
CE	1336
DF	2140
ES	2033
GO	2020
MA	747

Q4) 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

```

With year_17 as (
select
round(sum(payment_value),2) as Total_cost_2017
from `targetsql-sql.Target_SQL.payments` p
join `targetsql-sql.Target_SQL.orders` o ON p.order_id = o.order_id
where o.order_status = 'delivered' and (extract (year from o.order_purchase_timestamp) = 2017
)
and extract (month from o.order_purchase_timestamp) between 1 and 8),
year_18 AS (

```

```

select
    round(sum(payment value),2) as Total cost 2018
from `targetsql-sql.Target_SQL.payments` p
join `targetsql-sql.Target_SQL.orders` o on p.order_id = o.order_id
where o.order_status = 'delivered'AND (extract (year from o.order_purchase_timestamp) = 2018)
AND extract (month from o.order_purchase_timestamp) between 1 and 8)

Select
    Total_cost_2018,
    Total_cost_2017,
    round(((Total_cost_2018 -
    Total_cost_2017)/Total_cost_2017)*100,2) AS percentage_of_cost_increase
from year_17,year_18

```

Total_cost_2018 //	Total_cost_2017 //	percentage_of_cost_increase //
8452975.2	3473862.76	143.33

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

Please find the below query and top 10 rows o/p of sum & mean of price and freight value w.r.t customer_state

```

select customer_state,
    round(sum(price),2) as price_sum,
    round(sum(freight_value),2) as freight_sum,
    round(avg(price),2) as mean_price,
    round(avg(freight_value),2) as mean_freight
from `targetsql-sql.Target_SQL.customers` c
inner join `targetsql-sql.Target_SQL.orders` o on c.customer_id=o.customer_id
inner join `targetsql-sql.Target_SQL.order_items` oi on oi.order_id=o.order_id
group by customer_state
order by customer_state

```

customer_state //	price_sum //	freight_sum //	mean_price //	mean_freight //
AC	15982.95	3686.75	173.73	40.07
AL	80314.81	15914.59	180.89	35.84
AM	22356.84	5478.89	135.5	33.21
AP	13474.3	2788.5	164.32	34.01
BA	511349.99	100156.68	134.6	26.36
CE	227254.71	48351.59	153.76	32.71
DF	302603.94	50625.5	125.77	21.04
ES	275037.31	49764.6	121.91	22.06
GO	294591.95	53114.98	126.27	22.77
MA	119648.22	31523.77	145.2	38.26

Q5) Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

```

with delivery as
(
select order_id,
    DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) as Actual
    _days_of_delivery,
    DATE_DIFF(order_estimated_delivery_date, order_purchase_timestamp, day) as Expect
    ed_days_of_delivery,

```

```

DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_bt看w_expected_n_actual_delivery,
from `targetsql-sql.Target_SQL.orders`
where order_status='delivered'
)

```

```

select order_id, Actual_days_of_delivery, Expected_days_of_delivery, diff_bt看w_expected_n_actual_delivery,
CASE
  when diff_bt看w_expected_n_actual_delivery < 0 then 'Delayed'
  when diff_bt看w_expected_n_actual_delivery = 0 then 'On_time'
  else 'Early_Delivery'
END as Delivery_status
from delivery

```

order_id	Actual_days_of_delivery	Expected_days_of_delivery	diff_bt看w_expected_n_actual_delivery	Delivery_status
b60b53ad0bb7dacacf2989fe2...	12	7	-5	Delayed
276e9ec344d3bf029ff83a161c...	43	39	-4	Delayed
1a0b31f08d0d7e87935b819ed...	6	36	29	Early_Delivery
cec8f5f7a13e5ab934a486ec9e...	20	61	40	Early_Delivery
54e1a3c2b97fb0809da548a59...	40	36	-4	Delayed
58527ee4726911bee84a0f42c...	10	58	48	Early_Delivery
302bb8109d097a9fc6e9cefc5...	33	28	-5	Delayed
10ed5499d1623638ee810eff1...	28	57	29	Early_Delivery
cb837ba275cf8ffa9ded7e18f7...	12	7	-4	Delayed
66057d37308e787052a32828...	38	32	-6	Delayed

order_count	Delivery_status
87187	Early_Delivery
6535	Delayed
2754	On_time

→ order count on delivery status

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

```

select order_id,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, day) as time_to_delivery,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day) as diff_estimated_delivery
from `targetsql-sql.Target_SQL.orders`;

```

order_id	time_to_delivery	diff_estimated_delivery
1950d777989f6a877539f5379...	30	-12
2c45c33d2f9cb8ff8b1c86cc28...	30	28
65d1e226dfaeb8cdc42f66542...	35	16
635c894d068ac37e6e03dc54e...	30	1
3b97562c3aee8bdedcb5c2e45...	32	0
68f47f50f04c4cb6774570cfde...	29	1
276e9ec344d3bf029ff83a161c...	43	-4
54e1a3c2b97fb0809da548a59...	40	-4
fd04fa4105ee8045f6a0139ca5...	37	-1
302bb8109d097a9fc6e9cefc5...	33	-5

3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
select customer_state,
       round(AVG(time_to_delivery),2) as mean_time_to_delivery,
       round(AVG(diff_estimated_delivery),2) as mean_diff_estimated_delivery,
       round(AVG(freight_value),2) as mean_freight_value
from
(
select o.order_id,
       DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp, day) as time_to_delivery,
       DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_estimate
d_delivery,
       customer_state,
       freight_value
from `targetsql-sql.Target_SQL.orders` o
inner join `targetsql-sql.Target_SQL.customers` c on o.customer_id=c.customer_id
inner join `targetsql-sql.Target_SQL.order_items` oi on o.order_id=oi.order_id
) a
group by customer_state
order by customer_state
```

customer_state	mean_time_to_delivery	mean_diff_estimated_delivery	mean_freight_value
AC	20.33	20.01	40.07
AL	23.99	7.98	35.84
AM	25.96	18.98	33.21
AP	27.75	17.44	34.01
BA	18.77	10.12	26.36
CE	20.54	10.26	32.71
DF	12.5	11.27	21.04
ES	15.19	9.77	22.06
GO	14.95	11.37	22.77
MA	21.2	9.11	38.26

4. Sort the data to get the following:

5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Please find the below query for Top 5 highest average freight value w.r.t states

```
select customer_state as State,
       round(avg(freight_value),2) as mean_freight
from `targetsql-sql.Target_SQL.customers` c
inner join `targetsql-sql.Target_SQL.orders` o on c.customer_id=o.customer_id
inner join `targetsql-sql.Target_SQL.order_items` oi on oi.order_id=o.order_id
group by customer_state
order by mean_freight desc
limit 5
```

State	mean_freight
RR	42.98
PB	42.72
RO	41.07
AC	40.07
PI	39.15

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

Please find the below query for lowest average time to delivery

```

select customer_state,
       round(AVG(time_to_delivery),2) as avg_time_to_delivery
from
(
select o.order_id,
       DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp, day) as time_to_delivery,
       customer_state
from `targetsql-sql.Target_SQL.orders` o
inner join `targetsql-sql.Target_SQL.customers` c on o.customer_id=c.customer_id
) a
group by customer_state
order by avg_time_to_delivery
limit 5

```

customer_state	avg_time_to_delivery
SP	8.3
PR	11.53
MG	11.54
DF	12.51
SC	14.48

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

```

select customer_state,
       round(AVG(days_btw_estimated_n_delivery),2) as delivery
from
(
select o.order_id,
       DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date, day) as days_btw_estimated_n_delivery,
       customer_state
from `targetsql-sql.Target_SQL.orders` o
inner join `targetsql-sql.Target_SQL.customers` c on o.customer_id=c.customer_id
) a
group by customer_state
order by delivery desc
limit 5

```

customer_state	delivery
AC	19.76
RO	19.13
AP	18.73
AM	18.61
RR	16.41

Q6) Payment type analysis:

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

Below is the query for month over month count of orders and have calculated percentage growth

```

with CTE1 as
(
select payment_type,month,year,count(order_id) as mm_order_count
from
(
select p.order_id,payment_type,
       extract(month from order_purchase_timestamp) as month,
       extract(year from order_purchase_timestamp) as year,
       from `targetsql-sql.Target_SQL.payments` p
inner join `targetsql-sql.Target_SQL.orders` o on p.order_id=o.order_id
) a

```



```

group by payment_type,year,month
),
CTE2 as
(
select payment_type,month,year,mm_order_count,
lag(mm_order_count) over(partition by payment_type ORDER BY payment_type,year,month) as count_lag
from CTE1
)

select payment_type,month,year,mm_order_count,count_lag,change,
round(((change/count_lag)*100),2) as Percent_Growth
from
(
SELECT payment_type,month,year,mm_order_count,count_lag,
(mm_order_count-count_lag)as change
FROM CTE2
)
order by payment_type,year,month

```

payment_type	month	year	mm_order_count	count_lag	change	Percent_Growth
UPI	10	2016	63	null	null	null
UPI	1	2017	197	63	134	212.7
UPI	2	2017	398	197	201	102.03
UPI	3	2017	590	398	192	48.24
UPI	4	2017	496	590	-94	-15.93
UPI	5	2017	772	496	276	55.65
UPI	6	2017	707	772	-65	-8.42
UPI	7	2017	845	707	138	19.52
UPI	8	2017	938	845	93	11.01
UPI	9	2017	903	938	-35	-3.73

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

```

select payment_installments,count(order_id) as Count_of_orders from `targetsql-
sql.Target_SQL.payments`
group by payment_installments

```

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

Q7. Actionable Insights:

- There is decrease in orders in the Month of December both in 2016 and 2017 years.
- January Month is the seasonal peak for the given time period.
- Brazil customer tend to buy more in Afternoon(12:00 to 18:00)

- There is at most 150 % increase in cost of orders from 2017 to 2018 (included months between Jan to Aug only)
- 90% of total orders are delivered faster than expected in given time period.

Q8. Recommendations:

- As there is a decrease of sales in the month of December and others, providing exclusive offers, discount coupons and lesser shipping charges could increase the sales/orders
- Customer feedback and recommendations in terms of service could also be helpful to improvise personalized experiences for customers, which intern increases sales.