Business Case: Target SQL

1. Exploratory Analysis

1.Data type of columns in a table:

i) Customers:

Field name	Type	Mode
customer_id	STRING	NULLABLE
customer_unique_id	STRING	NULLABLE
customer_zip_code_prefix	INTEGER	NULLABLE
customer_city	STRING	NULLABLE
customer_state	STRING	NULLABLE

ii) Geolocation:

Field name	Type	Mode
geolocation_zip_code_prefix	INTEGER	NULLABLE
geolocation_lat	FLOAT	NULLABLE
geolocation_lng	FLOAT	NULLABLE
geolocation_city	STRING	NULLABLE
geolocation_state	STRING	NULLABLE

iii) Order_items:

Field name	Type	Mode
order_id	STRING	NULLABLE
order_item_id	INTEGER	NULLABLE
product_id	STRING	NULLABLE
seller_id	STRING	NULLABLE
shipping_limit_date	TIMESTAMP	NULLABLE
price	FLOAT	NULLABLE
freight_value	FLOAT	NULLABLE

iv) Order_reviews

Field name	Type	Mode
review_id	STRING	NULLABLE
order_id	STRING	NULLABLE
review_score	INTEGER	NULLABLE
review_comment_title	STRING	NULLABLE
review_creation_date	TIMESTAMP	NULLABLE
review_answer_timestamp	TIMESTAMP	NULLABLE

v) Orders:

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

vi) Payments:

Field name	Type	Mode
order_id	STRING	NULLABLE
payment_sequential	INTEGER	NULLABLE
payment_type	STRING	NULLABLE
payment_installments	INTEGER	NULLABLE
payment_value	FLOAT	NULLABLE

vii) Products:

product_id	STRING	NULLABLE
product_category	STRING	NULLABLE
product_name_length	INTEGER	NULLABLE
product_description_length	INTEGER	NULLABLE
product_photos_qty	INTEGER	NULLABLE
product_weight_g	INTEGER	NULLABLE
product_length_cm	INTEGER	NULLABLE
product_height_cm	INTEGER	NULLABLE
product_width_cm	INTEGER	NULLABLE

viii) Sellers:

Field name	Type	Mode
seller_id	STRING	NULLABLE
seller_zip_code_prefix	INTEGER	NULLABLE
seller_city	STRING	NULLABLE
seller_state	STRING	NULLABLE

2.Time period for which the data is given:

It is mentioned in the business case that it has information of 100k orders from 2016 to 2018 made at Target in Brazil. We can also check through order_purchase_timestamp from orders table.

```
select min(order_purchase_timestamp), max(order_purchase_timestamp)
from `Project_01.orders`
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	f0_	f1_
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:

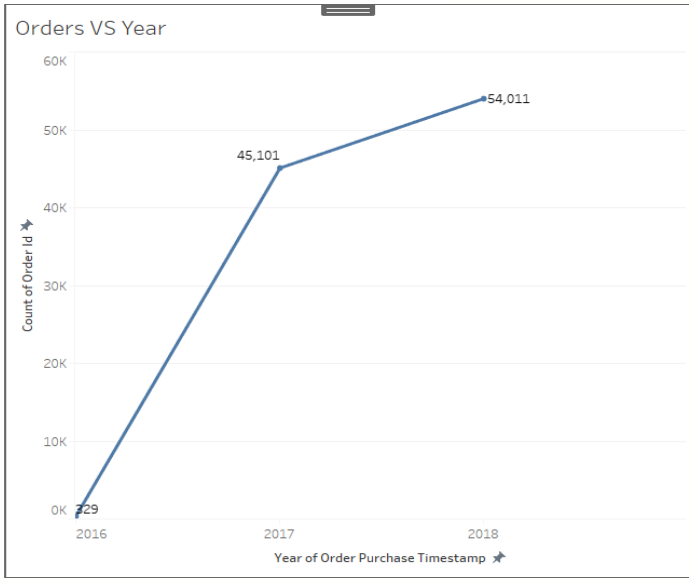
```
select distinct customer_city, customer_state
from `Project_01.customers`
```

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

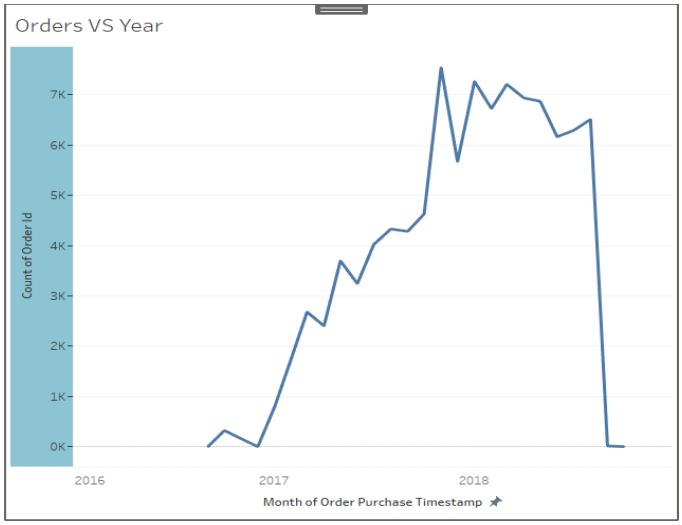
```
1.
select
  count(order_id) as Total_count_of_orders, extract(year from order_purchase_timestamp)
as year
from `Project_01.orders`
group by year
```

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



On observing the above count of orders per year, we can clearly say that there is a growing trend on e-commerce in Brazil.

```
select
  count(order_id) as Total_count_of_orders,extract(year from order_purchase_timestamp)
as year,extract(month from order_purchase_timestamp) as month
from `Project_01.orders`
group by year,month
order by year,month
```



Above is the graph showing the seasonality with peaks for every month from 2016 to 2018.

Row	Total_count_of_orders	year	month
1	4	2016	9
2	324	2016	10
3	1	2016	12
4	800	2017	1
5	1780	2017	2
6	2682	2017	3
7	2404	2017	4
8	3700	2017	5
9	3245	2017	6
10	4026	2017	7

2.Time at which Brazilian customers tend to buy the most:

```
select

case

  when extract(time from order_purchase_timestamp) between '05:00:00' and '07:00:00'
  then 'Dawn'
  when extract(time from order_purchase_timestamp) between '07:00:00' and '12:00:00'
  then 'Morning'
  when extract(time from order_purchase_timestamp) between '12:00:00' and '16:00:00'
  then 'Afternoon'
  when extract(time from order_purchase_timestamp) between '16:00:00' and '23:59:59'
  then 'Night'
  else 'Twilight'
end as Time,
count(c.customer_id) as No_of_customers
from `Project_01.customers` as c
join `Project_01.orders` as o
on c.customer_id = o.customer_id
group by Time
order by No_of_customers desc
```

Row	Time	No_of_customer
1	Night	46924
2	Afternoon	25537
3	Morning	21738
4	Twilight	4552
5	Dawn	690

From the above observation, we could say that most of the Brazilian customers tend to buy at night.

3.Evolution of e-commerce orders in Brazil region

1.Month on month orders by states:

```
select extract(month from order_purchase_timestamp) as month,count(order_id) as Total_c
ount_of_orders,customer_state

from `Project_01.orders` as o
join `Project_01.customers` as c
on o.customer_id = c.customer_id
group by customer_state,month
order by customer_state,month
```

Row	month	Total_count_of_orders	customer_state
1	1	8	AC
2	2	6	AC
3	3	4	AC
4	4	9	AC
5	5	10	AC
6	6	7	AC
7	7	9	AC
8	8	7	AC
9	9	5	AC
10	10	6	AC

2.Distribution of customers across states in Brazil:

```
select customer_state, count(customer_unique_id) as Distribution_of_customers

from `Project_01.customers`

group by customer_state

order by customer_state
```

Row	customer_state	Distribution_of_customers
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

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

1. % increase in cost of orders from 2017 to 2018:

```
with cost_of_orders as
(select extract(year from o.order_purchase_timestamp) as year,sum(payment_value) as tota
l_cost
from `Project_01.orders` as o
join `Project_01.payments` as p
on o.order_id = p.order_id
where extract(year from o.order_purchase_timestamp) between 2017 and 2018
  and extract(month from o.order_purchase_timestamp) between 01 and 08
group by year
order by year),
cost_2017 as(select c.total_cost as tc_2017 from cost_of_orders as c where c.year = 2017
 )

select year,total_cost,
case
  when year = 2017
  then 0
  else round(100*((total_cost - (select tc_2017 from cost_2017))/(select tc_2017 from co
st_2017)),2)
end as percentage_increase
from cost_of_orders as c
order by year
```

Row	year	total_cost	percentage_increase
1	2017	3669022.11...	0.0
2	2018	8694733.83...	136.98

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

```
select c.customer_state,

    sum(a.price+a.freight_value) as Sum_of_price_and_freight_value,

    sum(a.price+a.freight_value)/count(price) as Mean_of_price_and_freight_value
 from `Project_01.order_items` as a
 join `Project_01.orders` as b on a.order_id = b.order_id
 join `Project_01.customers` as c on b.customer_id = c.customer_id
 group by customer_state
```

Row	customer_state	Sum_of_price_and_freight_value	Mean_of_price_and_freight_value
1	MT	186168.96000000034	176.46346919431312
2	MA	151171.99000000019	183.4611529126216
3	AL	96229.399999999965	216.73288288288279
4	SP	5921678.1199998269	124.80090455014494
5	MG	1856161.4900000079	141.37874095513808
6	PE	322237.69000000006	178.42618493909194
7	RJ	2129681.98	146.07874202620206
8	DF	353229.44000000047	146.81190357439755
9	RS	885826.759999998	142.07325741780241
10	SE	73032.319999999992	189.69433766233763

5.Analysis on Sales, freight and delivery time

1. Calculation of days between purchasing, delivering and estimated delivery; time_to_delivery & diff_estimated_delivery:

```
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_carrier_date,day) as diff_
estimated_delivery

from `Project_01.orders`
```

Row	order_id	time_to_delivery	diff_estimated_delivery
1	1950d777989f6a877539f5379...	30	16
2	2c45c33d2f9cb8ff8b1c86cc28...	30	54
3	65d1e226dfaeb8cdc42f66542...	35	30
4	635c894d068ac37e6e03dc54e...	30	20
5	3b97562c3aee8bdedcb5c2e45...	32	30
6	68f47f50f04c4cb6774570cfde...	29	30
7	276e9ec344d3bf029ff83a161c...	43	22
8	54e1a3c2b97fb0809da548a59...	40	35
9	fd04fa4105ee8045f6a0139ca5...	37	28
10	302bb8109d097a9fc6e9cefc5...	33	21

3.Top 5 states with lowest average freight value:

```
select c.customer_state, avg(freight_value) as avg_freightvalue

from `Project_01.order_items` as o
join `Project_01.orders` as r
on o.order_id = r.order_id
join `Project_01.customers` as c
on r.customer_id = c.customer_id
group by c.customer_state
order by avg_freightvalue asc
limit 5
```

Row	customer_state	avg_freightvalue
1	SP	15.1472753...
2	PR	20.5316515...
3	MG	20.6301668...
4	RJ	20.9609239...
5	DF	21.0413549...

4.Top 5 states with highest average freight value:

```
select c.customer_state, avg(freight_value) as avg_freightvalue,

from `Project_01.order_items` as o
join `Project_01.orders` as r
on o.order_id = r.order_id
join `Project_01.customers` as c
on r.customer_id = c.customer_id
```

```
group by c.customer_state
order by avg_freightvalue desc
limit 5
```

Row	customer_state	avg_freightvalue
1	RR	42.9844230...
2	PB	42.7238039...
3	RO	41.0697122...
4	AC	40.0733695...
5	PI	39.1479704...

5.Top 5 states with lowest average time to delivery:

```
select customer_state,avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day
)) as avg_time_to_delivery

from `Project_01.orders` as o
join `Project_01.customers` as c
on o.customer_id = c.customer_id
group by c.customer_state
order by avg_time_to_delivery
limit 5
```

Row	customer_state	avg_time_to_delivery
1	SP	8.298061489072...
2	PR	11.52671135486...
3	MG	11.54381329810...
4	DF	12.50913461538...
5	SC	14.47956019171...

6. Top 5 states where delivery is really fast compared to estimated date:

```
select customer_state,avg(date_diff(order_estimated_delivery_date,order_delivered_customer_dat
e,day)) as avg_diff_estimated_delivery

from `Project_01.orders` as o
join `Project_01.customers` as c
on o.customer_id = c.customer_id
group by c.customer_state
order by avg_diff_estimated_delivery
limit 5
```

Row	customer_state	avg_diff_estimated_delivery
1	AL	7.9471032745591943
2	MA	8.76847977684797
3	SE	9.1731343283582127
4	ES	9.6185463659147885
5	BA	9.93488943488941

6.Payment type analysis

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

```
select payment_type,extract(month from o.order_purchase_timestamp) as month,
       count(o.order_id) as count_of_orders
from `Project_01.payments` as p
join `Project_01.orders` as o
on p.order_id = o.order_id
group by payment_type,month
order by payment_type,month
```

Row	payment_type	month	count_of_orders
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056

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

```
select payment_installments,count(order_id) as count_of_orders

from `Project_01.payments`
group by payment_installments
order by payment_installments
```

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

7. Actionable insights and Recommendations:

- From the above analysis, we could see a remarkable growth of Target in e-commerce in Brazil.
- Focusing on customer's order reviews and trying to resolve their issues at the earliest would lead to a better growth.
- Maintaining lower average of delivery time for all the states as far as possible would also level up customer's purchases.
- Focusing on the states where customer's distribution is low and thus taking the necessary action would help grow the business countrywide.
- Continuing to accept the payment_type where most of the customers use for purchasing would also help Target to be a customer friendly retailer.