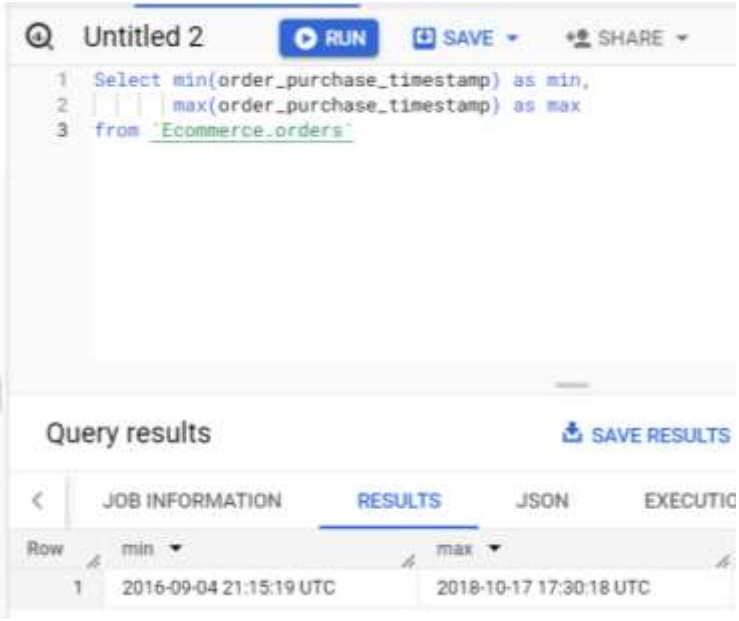
BUSINESS CASE

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.

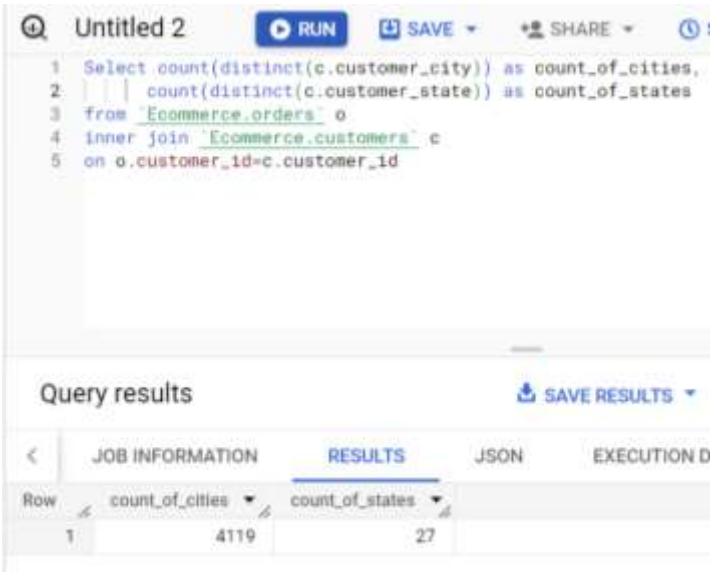
Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

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



From the above result, we can conclude that the **first** purchase of the order was made on **4th September 2016 (9:15 pm)** and the **last** purchase was made on **17th October 2018 (5:30 pm)**

3. Count the Cities & States of customers who ordered during the given period.



Customers from **4119 cities** across **27 states** have purchased in the given period

2. **In-depth Exploration:**

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

```
Select extract(year from order_purchase_timestamp) as year,
       extract(month from order_purchase_timestamp) as month,
       count(order_id) as no_of_orders_placed
from `Ecommerce.orders`
group by 1,2
order by 1,2
```

Row	year	month	no_of_orders_placed
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

From the above result we can see that, during initial months of the Target setup there were lesser number or orders placed and later on its increasing, but not exponentially. We observe that number of orders gets **stable around midyear of 2017**, we also observe that during the month of **November, January number of orders** are the **most** which may be due to

Black Friday sales and holiday season and the orders **decreased** drastically around **October 2018** which may indicate that Target was wrapping up their operations.

2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Row	year	month	no_of_orders_placed
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
14	2017	11	7544
15	2017	12	5673
16	2018	1	7269
17	2018	2	6728

We see some kind of monthly seasonality where customers placed the more number of orders in **November 2017** which may be due to **black Friday sales** and in **January** due to **holiday season**. We also observe that number of orders are being more or less stable during the months of **july and august** of both years **2017 & 2018**

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

```sql
select case when hour between 0 and 6 then "Dawn"
            when hour between 7 and 12 then "Mornings"
            when hour between 13 and 18 then "Afternoon"
            else "Night"
            end as time_of_the_day,
            count(order_id) as no_of_orders_placed
 from (select extract(hour from order_purchase_timestamp) as hour,order_id
from `Ecommerce.orders`)
group by 1
order by 2
```

Press Alt+F1 for Ac

Query results

SAVE RESULTS | EXPLORE DA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHAI

Row	time_of_the_day	no_of_orders_placed
1	Dawn	5242
2	Mornings	27733
3	Night	28331
4	Afternoon	38135

From the above result we observe that most of the orders were placed during **afternoon**(i.e,13-18hrs) time of the day

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

1. Get the month on month no. of orders placed in each state.

```sql
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 no_of_orders_placed
from `Ecommerce.orders` o
inner join `Ecommerce.customers` c
on o.customer_id=c.customer_id
group by 1,2,3
order by 2,3,4 desc
```

Row	customer_state	year	month	no_of_orders_placed
1	SP	2016	9	2
2	RR	2016	9	1
3	RS	2016	9	1
4	SP	2016	10	113
5	RJ	2016	10	56
6	MG	2016	10	40
7	RS	2016	10	24
8	PR	2016	10	19
9	SC	2016	10	11
10	GO	2016	10	9
11	CE	2016	10	8
12	PE	2016	10	7
13	DF	2016	10	6
14	BA	2016	10	4

Results per page: 50 1 – 50 of 565

Thus, the number of orders placed across each state on month on month are given in the result. We also observe that number of orders placed in the **state SP** are **more** which provides an incentive to the Target to enhance their operations in a manner where the customers base can be retained and further increased in all the cities in the State SP by providing more discounts.

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

```
1  select customer_state,
2         count(customer_id) as no_of_customers
3  from `Ecommerce.customers`
4  group by 1
5  order by 2 desc
```

Row	customer_state	no_of_customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020
11	PE	1652
12	CE	1336
13	PA	975

Load more

Results per page: 50 1 – 27 of 27

We observe that there are **more** number of customers from the states of **SP, RJ, MG, RS** and **less** number of customers from the states of **RR, AP, AC, AM**. With this given insights

we can enhance Target operations to increase the customer base across all the states by **providing better customer service and support**. It also helps us to find **targeted consumers**.

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

1. 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.

```sql
select t.order_id,
       current_price as cost_of_order,
       round(((new_price-current_price)/current_price)*100,2) as
increasing_cost_percentage
from (select order_id,
       payment_value as current_price,
       lead(payment_value,1) over(order by payment_value) as new_price
from `Ecommerce.payments`
where payment_value>0) t
inner join `Ecommerce.orders` o
on t.order_id=o.order_id
where extract(year from order_purchase_timestamp)>2016
      and extract(month from order_purchase_timestamp) between 1 and 8
order by 2,3
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	order_id	cost_of_order	increasing_cost_perc
1	7db5f2eb8f5f54db9f9e71ba42...	0.01	0.0
2	fb4de3600d359f84927517e78f...	0.01	0.0
3	ca4b9f3ce6fc19e8533501cf8c...	0.01	0.0
4	25b5b0ea53b7d5a2d5712a0d...	0.01	200.0
5	4884bd0f5624b0b7919209656...	0.03	0.0
6	d8c4293616878e2f015351d8f...	0.05	0.0
7	2a4cf847d8e23123b4421ef7f7...	0.05	40.0
8	874904b7cdcd377a298d82e30...	0.08	12.5
9	2122719f6fc72fba16b0b3001d...	0.09	11.11
10	c96a303a722b89bc14a23b7a2...	0.1	10.0
11	33e70b1305003580be0b3447...	0.11	0.0
12	4fb76fa13b108a0d047848342...	0.13	0.0
13	9cd6015fc7c3321b8cfc4459af...	0.13	7.69
14	dfc5c1ee34c30f42aba88ee3a5...	0.14	0.0

We observe that increasing percentage in the cost of orders in the time period between 2017 and 2018(jan to aug)

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

```sql
select c.customer_state,
       round(sum(o1.price),2) as total_order_price,
       round(avg(o1.price),2) as average_order_price
from `Ecommerce.customers` c
inner join `Ecommerce.orders` o
on c.customer_id=o.customer_id
inner join `Ecommerce.order_items` o1
on o.order_id= o1.order_id
group by 1
order by 2 desc, 3 desc
```

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	BA	511349.99	134.6
8	DF	302603.94	125.77
9	GO	294591.95	126.27
10	ES	275037.31	121.91
11	PE	262788.03	145.51
12	CE	227254.71	153.76
13	PA	178947.81	165.69

Load more

Results per page: 50 1 – 27 of 27

From the above result, we observe that states like **SP, RJ, MG** have highest total order price as there were huge number of orders placed in these states. It gives huge opportunity for Target to enhance their operations in these states to increase their revenue.

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

```sql
select c.customer_state,
       round(sum(o1.freight_value),2) as total_freight_value,
       round(avg(o1.freight_value),2) as average_freight_value
from `Ecommerce.customers` c
inner join `Ecommerce.orders` o
on c.customer_id=o.customer_id
inner join `Ecommerce.order_items` o1
on o.order_id= o1.order_id
group by 1
order by 2 desc, 3 desc
```

Row	customer_state	total_freight_value	average_freight_valu
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
11	ES	49764.6	22.06
12	CE	48351.59	32.71
13	PA	38699.3	35.83

Load more

Results per page: 50 1 – 27 of 27

From the above result, we observe, as the orders are more from states like SP, RJ, MG…their freight value is on top. This provides us to **optimize logistics cost.**

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

```
select order_id,
       date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
delivery_time,
       date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_estimated_delivery
from `Ecommerce.orders`
order by 1,2,3
```

Row	order_id	days_to_deliver	diff_estimated_deliv
1	00010242fe8c5a6d1ba2dd792...	7	8
2	00018f77f2f0320c557190d7a1...	16	2
3	000229ec398224ef6ca0657da...	7	13
4	00024acbcdf0a6daa1e931b03...	6	5
5	00042b26cf59d7ce69dfabb4e...	25	15
6	00048cc3ae777c65dbb7d2a06...	6	14
7	00054e8431b9d7675808bcb8...	8	16
8	000576fe39319847cbb9d288c...	5	15
9	0005a1a1728c9d785b8e2b08b...	9	0
10	0005f50442cb953dcd1d21e1f...	2	18

From the above result, it provides us with an insight to reduce delivery time on orders where it takes more number of days to deliver, which in turn increase the sales of Target provided with better customer service and satisfaction.

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

```
select c.customer_state,
       round(avg(o1.freight_value),2) as average_freight_value
from `Ecommerce.customers` c
inner join `Ecommerce.orders` o
on c.customer_id=o.customer_id
inner join `Ecommerce.order_items` o1
on o.order_id= o1.order_id
group by 1
order by 2 desc
limit 5
```

Row	customer_state	average_freight_valu
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

Row	customer_state	average_freight_valu
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

It provides us with states having highest and lowest average freight value. It helps to focus on these states to enhance their operations by optimizing the freight cost.

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

```
select c.customer_state,
       round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),
2) as avg_delivery_time
from `Ecommerce.orders` o
inner join `Ecommerce.customers` c
on o.customer_id=c.customer_id
group by 1
order by 2 desc
limit 5
```

Press Alt+F1 for Accessibility Optic

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART | PREVIEW

Row	customer_state	avg_delivery_time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

```
select c.customer_state,
       round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),
2) as avg_delivery_time
from `Ecommerce.orders` o
inner join `Ecommerce.customers` c
on o.customer_id=c.customer_id
group by 1
order by 2
limit 5
```

Press Alt+F1 for Accessibility Optic

Query results

SAVE RESULTS | EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | CHART | PREVIEW

Row	customer_state	avg_delivery_time
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

It provides us with states having highest and lowest delivery time, which helps Target to focus on states with higher delivery time, where it can either **localize warehousing** near these states or **outsource it to third party for better service**.

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.

```
select c.customer_state,
       round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,
day)),2) as avg_diff_estimated_delivery
from `Ecommerce.orders` o
inner join `Ecommerce.customers` c
on o.customer_id=c.customer_id
group by 1
order by 2 desc
limit 5
```

Row	customer_state	avg_diff_estimated_d
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

In these states, delivery is fast compared to estimated date of delivery.

6. **Analysis based on the payments:**

1. Find the month on month no. of orders placed using different payment types.

```
select p.payment_type,
       extract(year from o.order_purchase_timestamp) as year,
       extract(month from o.order_purchase_timestamp) as month,
       count(o.order_id) as no_of_orders_placed
from `Ecommerce.orders` o
inner join `Ecommerce.payments` p
on o.order_id=p.order_id
group by 1,2,3
order by 1,2,3
```

Row	payment_type	year	month	no_of_orders_placed
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	UPI	2018	1	1518

Results per page: 50 1 – 50 of 90

From the above result, it provides us with number of orders placed using different payments mode like debit card, credit card, UPI, and voucher. We observe that huge number of orders placed using UPI on the months of November, January.

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

```
select p.payment_installments,
       count(o.order_id) as no_of_orders_placed
from `Ecommerce.orders` o
inner join  `Ecommerce.payments` p
on o.order_id = p.order_id
where p.payment_installments >=1
group by 1
order by 1
```

Row	payment_installment	no_of_orders_placed
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	5	5239
6	6	3920
7	7	1626
8	8	4268
9	9	644
10	10	5328
11	11	23
12	12	133
13	13	16

Load more

Results per page: 50 1 – 23 of 23

These are the number of orders placed on the basis of the payment installments that have been paid. We see that huge number of orders are placed where customers have paid on first payment installment

7.Actionable Insights & Recommendations

The following recommendations are given to optimize the sales of Target:

- Data analysis have provided with better opportunity to know who the customers and from which location which helps us to develop buyer persona.
- To improve the brand awareness of Target, focus on quality of products and improve marketing tactics to increase the customer base, where the insights provided in this business case can help to know our targeted consumers
- Delivery time of products can be optimized by providing warehouses near certain locations where the customer base is more. It can also be outsourced for efficiency.
- Providing better customer support service can make them re-engaged to buy even more products.
- Can make use of social media (digital marketing) to drive new visitors to the store and make them into paying consumers in the states where there are lower number of orders placed
- Freight cost can be optimized by outsourcing transportation