Target Analysis

About the dataset:

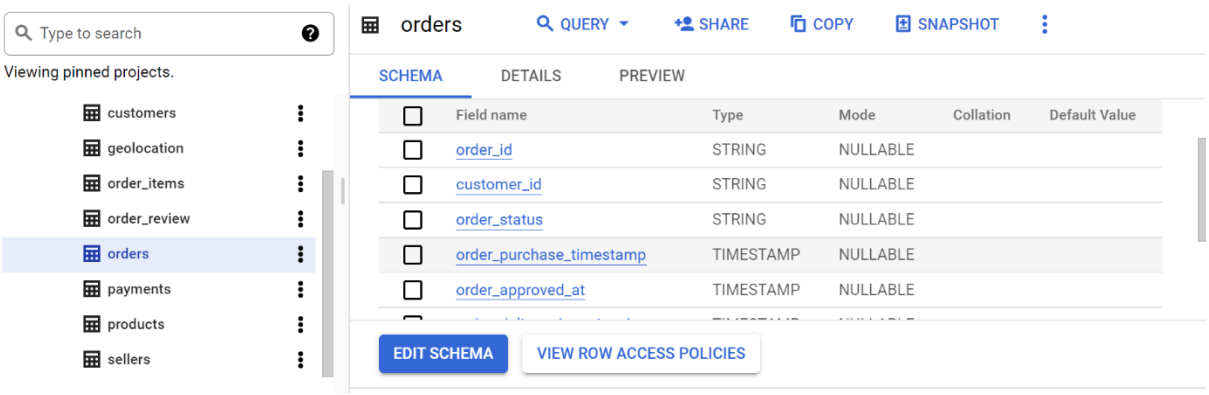
This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allow viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset

A. Data type of columns in a table

Query:

```
Desc `brahmaasmi.Target_Dataset.orders` ;
```



Similarily we can check for other tables as well.

B. Time period for which the data is given

Query:

```
SELECT
 date_diff(
 max(date(order_delivered_customer_date)),min(date(order_p
urchase_timestamp)), year) as year,
date_diff(
 max(date(order_delivered_customer_date)),min(date(order_p
urchase_timestamp)), month) as month,
 date_diff(
 max(date(order_delivered_customer_date)),min(date(order_p
urchase_timestamp)), day) as day FROM `brahmaasmi.Target_D
ataset.orders`;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	year	month	day
1	2	25	773

```
SELECT date(min(order_purchase_timestamp)) as min_time,
date(max(order_delivered_customer_date)) as max_time
 FROM `brahmaasmi.Target_Dataset.orders` ;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | E

Row	min_time	max_time
1	2016-09-04	2018-10-17

C. Cities and States covered in the dataset

```
select distinct c.customer_state
FROM `brahmaasmi.Target_Dataset.customers` c join
`brahmaasmi.Target_Dataset.geolocation` g
 on c.customer_zip_code_prefix=g.geolocation_zip_code_pref
ix;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTI

Row	customer_state
23	PE
24	PR
25	RN
26	RS
27	SC

```
select distinct c.customer_city
FROM `brahmaasmi.Target_Dataset.customers` c join
`brahmaasmi.Target_Dataset.geolocation` g
 on c.customer_zip_code_prefix=g.geolocation_zip_code_pref
ix;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EX

Row	customer_city
1	aracaju
2	riachuelo
3	nossa senhora do socorro
4	barra dos coqueiros
5	itaporanga d'ajuda

2. In-depth Exploration:

A. 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 extract(month from o.order_purchase_timestamp) as M
onth_,
count(o.order_id) as no_of_orders
FROM `brahmaasmi.Target_Dataset.orders` o
group by 1 order by no_of_orders desc ;
```

- August sees maximum number of purchases followed by July & May. Orders dip to their lowest in the month of September then it starts picking up till Jan. All in all September, October, November, December comprise of lowest number of purchases across all the months.

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTIO

Row	Month_	no_of_orders
1	8	10843
2	5	10573
3	7	10318
4	3	9893
5	6	9412

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

```
select extract(hour from o.order_purchase_timestamp) as Ho
urs_of_day,
count(o.order_id) as no_of_orders
FROM
`brahmaasmi.Target_Dataset.orders` o group by 1 order by n
o_of_orders desc ;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTIO

Row	Hours_of_day	no_of_orders
1	16	6675
2	11	6578
3	14	6569
4	13	6518
5	15	6454

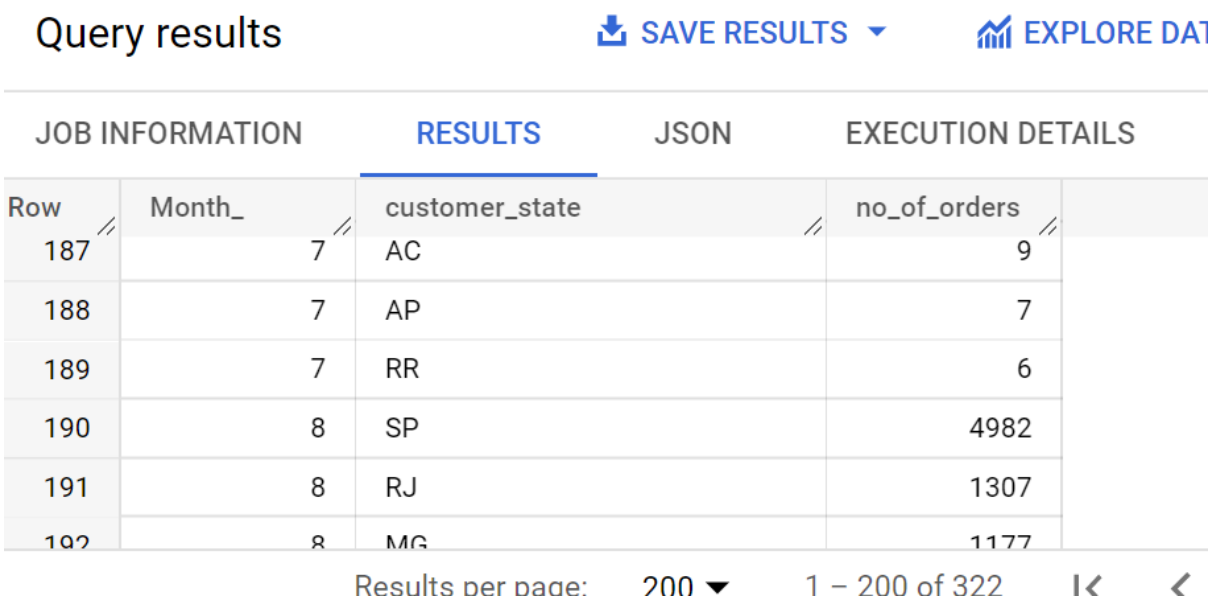
- Brazilians tends to order majorily afternoon

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

A. Get month on month orders by region, states

Month on Month orders By States:

```
select extract(month from o.order_purchase_timestamp) as M
onth_,
p.customer_state,
count(o.order_id) as no_of_orders
FROM `brahmaasmi.Target_Dataset.orders` o
join `brahmaasmi.Target_Dataset.customers` p on o.customer
_id=p.customer_id
group by 1,2 order by month_ ,no_of_orders desc ;
```



Month on Month orders By City:

```sql
select extract(month from o.order_purchase_timestamp) as M
onth_,
p.customer_city,
count(o.order_id) as no_of_orders
FROM `brahmaasmi.Target_Dataset.orders` o
join `brahmaasmi.Target_Dataset.customers` p on o.customer
_id=p.customer_id
group by 1,2 order by month_ ,no_of_orders desc ;
```

Query results SAVE RESULTS EXPLORE D

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	Month_	customer_city	no_of_orders
1	1	sao paulo	1195
2	1	rio de janeiro	545
3	1	belo horizonte	239
4	1	brasilia	151
5	1	curitiba	150

B. How are customers distributed in Brazil

```sql
select
p.customer_city,p.customer_state,
count(p.customer_id) as no_of_customers
from `brahmaasmi.Target_Dataset.customers` p
group by 1,2 order by no_of_customers desc,customer_state
;
```

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_city	customer_state	no_of_orders
1	sao paulo	SP	15540
2	rio de janeiro	RJ	6882
3	belo horizonte	MG	2773
4	brasilia	DF	2131
5	curitiba	PR	1521

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

A. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only)

```
with cte_price_difference as
 (select x.*,
lag(x.total_price,8) over( order by x.year,x.month)
as diff_in_price from
(select Extract(year from o.order_purchase_timestamp) as y
ear,
Extract(month from o.order_purchase_timestamp) as month,
sum(oi.price) as total_price
from `brahmaasmi.Target_Dataset.orders`  o
join `brahmaasmi.Target_Dataset.order_items` as oi on
 o.order_id=oi.order_id
 group by 1,2) x
 where x.year in (2017,2018) and month not in (9,10,11,12)
 order by year desc, month)
 select y.month ,(total_price-
diff_in_price)*100/diff_in_price as
percentage_diff_in_cost
from cte_price_difference y limit 8;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTIO

Row	month	percentage_...
1	1	689.633195...
2	2	241.353983...
3	3	162.649502...
4	4	176.902570...
5	5	96.9125684...
6	6	99.7799526...
7	7	79.8093606...

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

```
select x.customer_state,
sum(oi.freight_value) as total_freight_value,
avg(oi.freight_value) as avg_freight_value,
sum(oi.price) as total_price,
avg(oi.price) as avg_price
from
(
select o.order_id,c.customer_state
from `brahmaasmi.Target_Dataset.orders`  o
join  `brahmaasmi.Target_Dataset.customers` c on
o.customer_id=c.customer_id
) x join
`brahmaasmi.Target_Dataset.order_items` oi on
x.order_id=oi.order_id
 group by 1;
```

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	total_freight...	avg_freight_...	total_price	avg_price
1	MT	29715.4300...	28.1662843...	156453.529...	148.297184...
2	MA	31523.7700...	38.2570024...	119648.219...	145.204150...
3	AL	15914.5899...	35.8436711...	80314.81	180.889211...
4	SP	718723.069...	15.1472753...	5202955.05...	109.653629...
5	MG	270853.460...	20.6301668...	1585308.02...	120.748574...

5. Analysis on sales, freight and delivery time

A. Create columns:

- time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

```
SELECT
 date_diff(
 date(order_delivered_customer_date),date(order_purchase_t
imestamp),day)
as time_to_delivery,
 date_diff(
 date(order_estimated_delivery_date),date(order_delivered_
customer_date),day)
as diff_estimated_delivery
 FROM `brahmaasmi.Target_Dataset.orders`;
```

Query results SAVE RESULTS EXPLO

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAIL

Row	time_to_delivery	diff_estimated_delivery
1	30	-12
2	31	29
3	36	17
4	31	2
5	33	1

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

```
with cte_freight as (SELECT
 date_diff(
 date(order_delivered_customer_date),date(order_purchase_t
imestamp),day) as time_to_delivery,
 date_diff(
 date(order_estimated_delivery_date),date(order_delivered_
customer_date),day) as diff_estimated_delivery,
oi.freight_value,
o.customer_id
FROM `brahmaasmi.Target_Dataset.orders` o join
`brahmaasmi.Target_Dataset.order_items` oi on oi.order_id=
o.order_id)

select c.customer_state,
sum(x.diff_estimated_delivery) as mean_estimated_del,
sum(x.time_to_delivery) as mean_time_del ,
sum(x.freight_value) as mean_freight,
from cte_freight x join
`brahmaasmi.Target_Dataset.customers` c on
c.customer_id=x.customer_id
group by 1;
```

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	customer_state	mean_estimated_del	mean_time_del	mean_freight
1	MT	15111	18570	29715.4300...
2	MA	7925	17272	31523.7700...
3	AL	3730	10439	15914.5899...
4	SP	520529	402301	718723.069...
5	MG	172347	153980	270853.460...

C. Sort the data to get the following:

With the help of same CTE function mentioned above we can answer below questions:

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

Top 5 states with highest average freight value:

```
select c.customer_state,
avg(x.freight_value) as mean_freight,
from cte_freight x join
`brahmaasmi.Target_Dataset.customers` c on
c.customer_id=x.customer_id
group by 1 order by mean_freight desc LIMIT 5;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON

Row	customer_state	mean_freight
1	RR	42.984423076923093
2	PB	42.723803986710941
3	RO	41.069712230215842
4	AC	40.073369565217405
5	PI	39.147970479704767

Top 5 states with lowest average freight value:

```
select c.customer_state,
avg(x.freight_value) as mean_freight,
from cte_freight x join
`brahmaasmi.Target_Dataset.customers` c on
c.customer_id=x.customer_id
group by 1 order by mean_freight LIMIT 5;
```

Query results SAVE

JOB INFORMATION | RESULTS | JSOI

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

- Top 5 states with highest/lowest average time to delivery

Top 5 states with highest average time to delivery:

```
select c.customer_state,
sum(x.time_to_delivery) as mean_time_del
from cte_freight x join
`brahmaasmi.Target_Dataset.customers` c on
c.customer_id=x.customer_id
group by 1 order by mean_time_del desc limit 5;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | E

Row	customer_state	mean_time_del
1	AP	28.222222222222218
2	RR	28.173913043478258
3	AM	26.337423312883427
4	AL	24.447306791569098
5	PA	23.702087286527469

Top 5 states with lowest average time to delivery:

```
select c.customer_state,
avg(x.time_to_delivery) as mean_time_del
from cte_freight x join
`brahmaasmi.Target_Dataset.customers` c on
c.customer_id=x.customer_id
group by 1 order by mean_time_del limit 5;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON

Row	customer_state	mean_time_del
1	SP	8.66225265379071
2	PR	11.893078420959467
3	MG	11.920724626461224
4	DF	12.893842887473479
5	SC	14.950219619326486

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

Top 5 states where delievery is fast:

```
select c.customer_state,
avg(x.diff_estimated_delivery) as mean_estimated_del
from cte_freight x join
`brahmaasmi.Target_Dataset.customers` c on
c.customer_id=x.customer_id
group by 1 order by mean_estimated_del limit 5;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EX

Row	customer_state	mean_estimated_del
1	AL	8.73536299765808
2	MA	9.9062499999999929
3	SE	10.002666666666677
4	ES	10.646292134831446
5	BA	10.98262286179745

Top 5 states with slow delivery:

```
select c.customer_state,
avg(x.diff_estimated_delivery) as mean_estimated_del
from cte_freight x join
`brahmaasmi.Target_Dataset.customers` c on
c.customer_id=x.customer_id
group by 1 order by mean_estimated_del desc limit 5;
```

Query results SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXECUTION

ow	customer_state	mean_estimated_del
1	AC	20.978021978021971
2	RO	20.040293040293058
3	AM	19.932515337423315
4	AP	18.395061728395063
5	RR	18.326086956521742

6. Payment type analysis:

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

```
select extract(month from o.order_purchase_timestamp) as M
onth_,
p.payment_type,
count(o.order_id) as no_of_orders
FROM `brahmaasmi.Target_Dataset.orders` o
join `brahmaasmi.Target_Dataset.payments` p on o.order_id=
p.order_id
group by 1,2 order by month_,no_of_orders desc ;
```

Query results

SAVE RESULTS

EXPLORI

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	Month_	payment_type	no_of_orders
3	1	voucher	477
4	1	debit_card	118
5	2	credit_card	6609
6	2	UPI	1723
7	2	voucher	424

B. Distribution of payment installments and count of orders

```
select
p.payment_installments,
count(p.order_id) as no_of_orders
from `brahmaasmi.Target_Dataset.payments` p
group by 1 order by no_of_orders desc ;
```

Query results

SAVE RESULTS

JOB INFORMATION | RESULTS | JSON | EXE

Row	payment_in...	no_of_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328

Actionable Insights:

- Maximum number of orders are coming from Afternoon 1 PM to 4 PM. Among this patch there is a pattern too. Order starts increasing from 1 PM and reaches to max by 4 PM.
- Least number of orders are coming from morning 1 AM to 7 AM. Among this patch there is a pattern too. Order starts decreasing from 1 AM and reaches to its low by 5 PM. Then from 6 AM onwards it starts picking up and by 11 AM it reached to 2nd max no of orders.
- Credit_Card is the most used Payment type followed by UPI & Debit_Card is the least used Payment Type.
- August sees maximum number of purchases followed by July & May. Orders dip to their lowest in the month of September then it starts picking up till Jan. All in all September, October, November, December comprise of lowest number of purchases across all the months.
- Most used payment_installment is 1 followed by 2,3 & 4. People are rarely using installments above 9
- % increase in cost of orders from 2017 to 2018 shows decreasing trend from Jan to Aug
- SP,RJ,MG are leading states with maximum total price & freight value when AC,AP,RR being states with lowest total price & freight value
- Among states AL being the state with fastest delivery of orders followed by MA,SE

Recommendations:

- Debit_Card being the least used Payment Type, Coupons, discounts and other lucrative offers can be availed to the customers to encourage the usage of this Pyament_Type
- Maximum numbers of orders are coming from August, July & May month. Sale/ discounts can be offered to increase orders in other months as well. December being one of the lowest months for orders Christmas sales, new year sale/discount can be lucrative option to attract customers
- Percentage price rise is lowest in the month of August and maximum in the month of January and we see most orders are coming from August and January being one of lowest order receiving month. With this trend conclusion can be drawn that price rise is one of the most important contributing factors for customers. Multiple Market decisions and strategies can be adopted to cater this huge price rise in mentioned month to attract customers in months where we are receiving less orders.
- Campaigns or other marketing strategies can be used to increase penetration in states like AC, AP, RR from where we are receiving least sum of price/freight value
- There are multiple instances where estimated_delievery time is quite longer. Ways can be adopted to shorten this mismatch between actual delivery & estimated delivery to improve the customer experience.