Problem Statement:

Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analysing the given dataset to extract valuable insights and provide actionable recommendations.

What does 'good' look like?

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.

ANS:
```
SELECT column_name,data_type FROM target_sql.INFORMATION_SCHEMA.COLUMNS
    where table_name = 'customers';
```

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

Insights: INFORMSTION_SCHEMA gives to details of datatype of columns in the dataset. In customers most of the columns are of type string.

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

ANS:

```
select
min(order_purchase_timestamp) as first_order,
max(order_purchase_timestamp) as last_order
from `target_sql.orders`;
```

Row	first_order	last_order
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

Insights: using min() and max() to get the first and last purchase date.
The target has been used to purchase the products from 2016 and 2018 which is around 2 years.

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

ANS:

```
select count(distinct c.customer_city) as num_of_city,
count(distinct c.customer_state) as num_of_state
from target_sql.customers c
JOIN
target_sql.orders o
ON c.customer_id = o.customer_id
where o.order_purchase_timestamp
between (select min(order_purchase_timestamp) from target_sql.orders) and (select
max(order_purchase_timestamp)from target_sql.orders);
```

Row	num_of_city	num_of_state
1	4119	27

Insights: There are total 27 states and 4119 cities from where customers have ordered between the given time period.

2.In-depth Exploration:

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

ANS: yes

```
select count(order_id) as number_of_orders,
t.year
from
(select order_id,
order_purchase_timestamp,
extract (year from order_purchase_timestamp) as year
from target_sql.orders)t
group by t.year
order by t.year;
```

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

INSIGHT: There is huge increase in number of orders placed between year 2016 and 2017 than between 2017 and 2018.Between year 2016 and 2017 the target has reached maximum number of customers.

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

ANS:

```
select
concat(t.month,'-',t.year) as every_month_of_every_year,
count(order_id) as number_of_orders
from
(select order_id,
order_purchase_timestamp,
extract (year from order_purchase_timestamp) as year,
extract (month from order_purchase_timestamp) as month
from target_sql.orders)t
group by t.month,t.year
order by t.year,t.month;
```

every_month_of_every_year	number_of_orders
9-2016	4
10-2016	324
12-2016	1
1-2017	800
2-2017	1780
3-2017	2682
4-2017	2404
5-2017	3700
6-2017	3245
7-2017	4026
8-2017	4331
9-2017	4285
10-2017	4631
11-2017	7544

INSIGHTS:
During 2016 the number of orders placed have been less which is when target has been introduced in the market. Then during 2017 and 2018 the orders placed have a gradual increase and during around festival season (December and January) the number of orders have placed maximum compared to other months.

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

Ans:

```
select time_of_day,
sum(number_of_orders) as total_order
from
(select count(order_id) as number_of_orders,
q1.hour,
case
when q1.hour between 0 and 6 then 'Dawn'
when q1.hour between 7 and 12 then 'Mornings'
when q1.hour between 13 and 18 then 'Afternoon'
when q1.hour between 19 and 23 then 'Night'
end as time_of_day
from
(select order_id,
order_purchase_timestamp,
extract (hour from order_purchase_timestamp) as hour
from target_sql.orders)q1
group by q1.hour
order by q1.hour)q2
group by q2.time_of_day;
```

time_of_day	total_order
Mornings	27733
Dawn	5242
Afternoon	38135
Night	28331

Insights: The maximum number of orders are placed during Afternoon hours. During Dawn the customers are not very active as the less order have placed. During Morning and Night hours there is good number of orders have placed.

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

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

Ans:

```
select
c.customer_state,
o.month,
count(o.order_id) as number_of_orders
from target_sql.customers c
join
(select order_id,
customer_id,
order_purchase_timestamp,
extract (month from order_purchase_timestamp) as month
from target_sql.orders)o
on c.customer_id=o.customer_id
group by c.customer_state,o.month
order by number_of_orders desc;
```

customer_state	month	number_of_orders
SP	8	4982
SP	5	4632
SP	7	4381
SP	6	4104
SP	3	4047
SP	4	3967
SP	2	3357
SP	1	3351
SP	11	3012
SP	12	2357
SP	10	1908
SP	9	1648
RJ	5	1321
RJ	8	1307
RJ	3	1302

Insights:
State SP has high number of orders placed almost for all months where there is large number of customers and very active region.

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

Ans:

```
select
customer_state,
count(customer_id) as number_of_customers
from target_sql.customers
group by customer_state
order by number_of_customers desc;
```

customer_state	number_of_custome
SP	41746
RJ	12852
MG	11635
RS	5466
PR	5045
SC	3637
BA	3380
DF	2140
ES	2033
GO	2020
PE	1652
CE	1336
PA	975
MT	907
MA	747

Insights:

State SP has highest number of customers and there is huge difference between the number of customers in state SP and RJ. State RR has least number of customers with only 46.

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.

Ans:

```
select *,
round((( q2.total_value_in_month - lag(q2.total_value_in_month,1)over(order by
q2.year,q2.month,q2.total_value_in_month)) /
lag(q2.total_value_in_month,1)over(order by
q2.year,q2.month,q2.total_value_in_month)) * 100,2) as percentage_increase
from
(select
month,
year,
ceil(sum(payment_value)) as total_value_in_month
from
(select p.order_id,
p.payment_value,
extract(month from o.order_purchase_timestamp) as month,
extract(year from o.order_purchase_timestamp) as year
from target_sql.payments p
join
target_sql.orders o
on p.order_id = o.order_id)q1
where q1.month between 1 and 8 and q1.year between 2017 and 2018
group by q1.month,q1.year
order by q1.year,q1.month)q2
order by year,month;
```

Row	month	year	total_value_in_month	percentage_increase
1	1	2017	138489.0	*null*
2	2	2017	291909.0	110.78
3	3	2017	449864.0	54.11
4	4	2017	417789.0	-7.13
5	5	2017	592919.0	41.92
6	6	2017	511277.0	-13.77
7	7	2017	592383.0	15.86
8	8	2017	674397.0	13.84
9	1	2018	1115005.0	65.33
10	2	2018	992464.0	-10.99
11	3	2018	1159653.0	16.85
12	4	2018	1160786.0	0.1
13	5	2018	1153983.0	-0.59
14	6	2018	1023881.0	-11.27
15	7	2018	1066541.0	4.17
16	8	2018	1022426.0	-4.14

Insights:

There is a huge increase in order's cost between 2 month of 2017 and 1 month of 2017. Compared to last Jan-2017, Jan-2018 has maximum order's cost in total. Compared to 2017 the cost of orders has increased in 2018 for all months.

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

Ans:

```
select
q1.customer_state,
total_value,
round((total_value/number_of_orders),2) as average
from
(select
c.customer_state,
round(sum(p.payment_value),2) as total_value,
count(*) as number_of_orders
from target_sql.customers c
join
target_sql.orders o
on o.customer_id = c.customer_id
join
target_sql.payments p
on o.order_id =p.order_id
group by c.customer_state)q1
order by average desc;
```

customer_state	total_value	average
PB	141545.72	248.33
AC	19680.62	234.29
RO	60866.2	233.2
AP	16262.8	232.33
AL	96962.06	227.08
RR	10064.62	218.8
PA	218295.85	215.92
SE	75246.25	208.44
PI	108523.97	207.11
TO	61485.33	204.27
CE	279464.03	199.9
MA	152523.02	198.86
RN	102718.13	196.78
MT	187029.29	195.23
PE	324850.44	187.99

Insights: State PB has highest average of total cost of orders. Whereas total value of orders is high in state SP indicates that SP has large number of orders.

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

Ans:

```
select
q1.customer_state,
total_freight_value,
round((total_freight_value/number_of_orders),2) as average_of_total_value
from
(select
c.customer_state,
round(sum(i.freight_value),2) as total_freight_value,
count(*) as number_of_orders
from target_sql.customers c
join
target_sql.orders o
on o.customer_id = c.customer_id
join
target_sql.order_items i
on o.order_id =i.order_id
group by c.customer_state)q1
order by average_of_total_value desc;
```

customer_state	total_freight_value	average_of_total_value
RR	2235.19	42.98
PB	25719.73	42.72
RO	11417.38	41.07
AC	3686.75	40.07
PI	21218.2	39.15
MA	31523.77	38.26
TO	11732.68	37.25
SE	14111.47	36.65
AL	15914.59	35.84
PA	38699.3	35.83
RN	18860.1	35.65
AP	2788.5	34.01
AM	5478.89	33.21
PE	59449.66	32.92
CE	48351.59	32.71
MT	29715.43	28.17

Insights:

State RR has highest average of total freight of orders. Whereas total value of freight is high in state SP indicates that SP has large number of orders.

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

Ans:

```
select order_id,
date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
time_taken_to_deliver,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_in_estimated_delivery
from target_sql.orders
order by time_taken_to_deliver desc,diff_in_estimated_delivery desc;
```

order_id	time_taken_to_delive	diff_in_estimated_de
ca07593549f1816d26a572e06...	209	-181
1b3190b2dfa9d789e1f14c05b...	208	-188
440d0d17af552815d15a9e41a...	195	-165
2fb597c2f772eca01b1f5c561b...	194	-155
0f4519c5f1c541ddec9f21b3bd...	194	-161
285ab9426d6982034523a855f...	194	-166
47b40429ed8cce3aee9199792...	191	-175
2fe324febf907e3ea3f2aa9650...	189	-167
2d7561026d542c8dbd8f0daea...	188	-159
437222e3fd1b07396f1d9ba8c...	187	-144
c27815f7e3dd0b926b5855262...	187	-162
dfe5f68118c2576143240b8d7...	186	-153
6e82dcfb5eada6283dba34f16...	182	-155
2ba1366baecad3c3536f27546...	181	-152

Insights:
Maximum time taken to deliver the order is 209 which huge considering when the order was purchased. There is huge deviation from the estimated time of delivery than when the order is delivered.

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

Ans:

```
select
top5_state_with_high_avg,
top5_state_with_low_avg
from
(select
customer_state as top5_state_with_high_avg,
row_number()over(order by average_of_total_value desc) as high_5
from
(select
*,
row_number()over(order by average_of_total_value desc ) as top_5
from
(select
q1.customer_state,
total_freight_value,
round((total_freight_value/number_of_orders),2) as average_of_total_value
from
(select
c.customer_state,
round(sum(i.freight_value),2) as total_freight_value,
count(*) as number_of_orders
from target_sql.customers c
join
target_sql.orders o
on o.customer_id = c.customer_id
join
target_sql.order_items i
on o.order_id =i.order_id
group by c.customer_state)q1
order by average_of_total_value desc))q2
where q2.top_5 <=5)
join
(select
customer_state as top5_state_with_low_avg,
row_number()over(order by average_of_total_value asc) as low_5
from
(select
*,
row_number()over(order by average_of_total_value asc ) as top_5
from
(select
q1.customer_state,
total_freight_value,
round((total_freight_value/number_of_orders),2) as average_of_total_value
from
(select
c.customer_state,
round(sum(i.freight_value),2) as total_freight_value,
count(*) as number_of_orders
from target_sql.customers c
join
target_sql.orders o
on o.customer_id = c.customer_id
join target_sql.order_items i on o.order_id =i.order_id
```

```
group by c.customer_state)q1
order by average_of_total_value asc))q2
where q2.top_5 <=5)
on high_5 = low_5;

with using CTE

WITH StateAverage AS (
select
c.customer_state,
round(sum(i.freight_value),2) as total_freight_value,
count(*) as number_of_orders,
round((sum(i.freight_value)/count(*)),2) as average_of_total_value
from target_sql.customers c
join
target_sql.orders o
on o.customer_id = c.customer_id
join
target_sql.order_items i
on o.order_id =i.order_id
group by c.customer_state
)
, RankedStates AS (
SELECT
customer_state,
average_of_total_value,
ROW_NUMBER() OVER (ORDER BY average_of_total_value DESC) AS high_rank,
ROW_NUMBER() OVER (ORDER BY average_of_total_value ASC) AS low_rank
FROM StateAverage
)
SELECT
h.customer_state AS top5_state_with_high_avg,
h.average_of_total_value as high_average,
l.customer_state AS top5_state_with_low_avg,
l.average_of_total_value as low_average
FROM RankedStates h
JOIN
RankedStates l
ON h.high_rank = l.low_rank
WHERE h.high_rank <= 5;
```

top5_state_with_high_avg	high_average	top5_state_with_low_avg	low_average
RR	42.98	SP	15.15
PB	42.72	PR	20.53
RO	41.07	MG	20.63
AC	40.07	RJ	20.96
PI	39.15	DF	21.04

Insights:
The state RR holds the high average as it has low number of customers and has reasonable freight values. The state SP has low average as it holds the maximum number of customers compared to other states.

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

Ans:

```
with avg_time_taken_by_state as
(select
t.customer_state,
avg(t.time_taken_to_deliver) as average_time
from
(select
c.customer_state,
o.order_id,
date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as
time_taken_to_deliver
from target_sql.orders o
join
target_sql.customers c
on o.customer_id = c.customer_id
order by time_taken_to_deliver desc)t
group by t.customer_state) ,
top_5_states as
(select
customer_state,
average_time,
row_number()over(order by average_time desc) as high_avg,
row_number()over(order by average_time asc) as low_avg
from avg_time_taken_by_state)
select
h.customer_state as state_with_high_avg_del_time,
ceil(h.average_time) as top_5_high_avg_days,
l.customer_state as state_with_low_avg_del_time,
ceil(l.average_time) as top_5_low_avg_days
from top_5_states h
join top_5_states l
on h.high_avg = l.low_avg
where h.high_avg <=5;
```

state_with_high_avg_del_time	top_5_high_avg_days	state_with_low_avg_del_time	top_5_low_avg_days
RR	29.0	SP	9.0
AP	27.0	PR	12.0
AM	26.0	MG	12.0
AL	25.0	DF	13.0
PA	24.0	SC	15.0

Insights:

The average time taken to deliver an order with in the state is low for State SP which is around 9 days which is reasonable time to deliver an order. But the states RR, AP, AM, AL, has taken an average of almost a month to deliver an order.

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.

Ans:

```
select
t.customer_state,
ceil(avg(time_taken_to_deliver)) as time_taken,
ceil(avg(diff_in_estimated_delivery)) estimated_time
from
(select
c.customer_state,
o.order_id,
date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as
time_taken_to_deliver,
date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as
diff_in_estimated_delivery
from target_sql.orders o
join
target_sql.customers c
on o.customer_id = c.customer_id
order by time_taken_to_deliver desc,diff_in_estimated_delivery desc)t
group by t.customer_state
having avg(time_taken_to_deliver) <= avg(diff_in_estimated_delivery);
```

customer_state	time_taken	estimated_time
SP	9.0	11.0
PR	12.0	13.0
MG	12.0	13.0
RO	19.0	20.0

Insights:

Only 4 states out of 27 that have fastest delivery time than the estimated delivery time.

6.Analysis based on the payments:

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

Ans:

```
select
concat(t.month,'-',t.year) as every_month_of_every_year,
count(order_id) as number_of_orders,
t.payment_type
from
(select o.order_id,
o.order_purchase_timestamp,
extract (year from o.order_purchase_timestamp) as year,
extract (month from o.order_purchase_timestamp) as month,
p.payment_type
from target_sql.orders o
join
target_sql.payments p
on o.order_id = p.order_id)t
group by t.month,t.year,t.payment_type
order by t.year,t.month,number_of_orders desc;
```

every_month_of_every_year	number_of_orders	payment_type
9-2016	3	credit_card
10-2016	254	credit_card
10-2016	63	UPI
10-2016	23	voucher
10-2016	2	debit_card
12-2016	1	credit_card
1-2017	583	credit_card
1-2017	197	UPI
1-2017	61	voucher
1-2017	9	debit_card
2-2017	1356	credit_card
2-2017	398	UPI
2-2017	119	voucher
2-2017	13	debit_card
3-2017	2016	credit_card

Insights:

For every month starting from year 2016, most of the customers chooses credit card option for their payments followed by UPI options and vouchers.
Debit card option is not preferred by most of the customers.

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

Ans:

```sql
select
count(distinct t.order_id) as number_of_orders,
t.payment_installments
from
(select o.order_id,
o.order_purchase_timestamp,
p.payment_type,
p.payment_installments,
p.payment_sequential
from target_sql.orders o
join
target_sql.payments p
on o.order_id = p.order_id)t
where t.payment_sequential = t.payment_installments
group by t.payment_installments
order by number_of_orders desc;
```

number_of_orders	payment_installment
48236	1
53	2
1	3

Insights:

Large number of customers were placed an order with full payment (not preferring EMI). There are still large number of customers who are not yet completed their full payment.