TARGET BUSINESS CASE

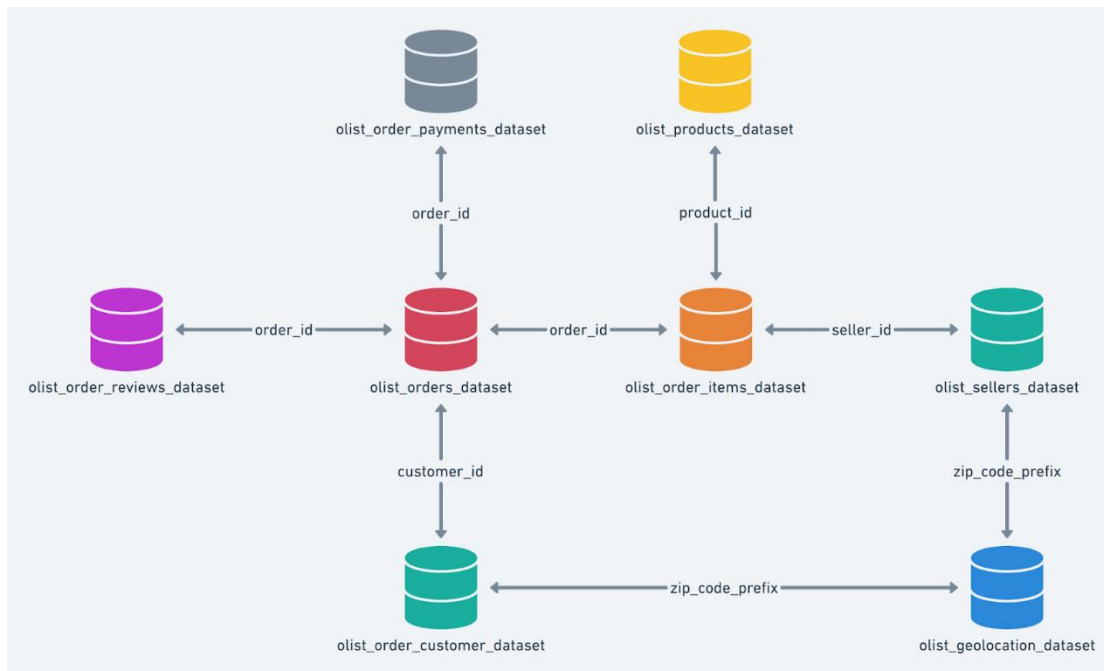**Dataset schema:**

_____
_____

Problem Statement:

Assuming you are a data analyst/ scientist at Target, you have been assigned the task of analyzing 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.
    2. Get the time range between which the orders were placed.
    3. Count the Cities & States of customers who ordered during the given period.
2. **In-depth Exploration:**

    1. Is there a growing trend in the no. of orders placed over the past years?
    2. Can we see some kind of monthly seasonality in terms of the no. of orders being placed?
    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

3. **Evolution of E-commerce orders in the Brazil region:**
    1. Get the month on month no. of orders placed in each state.
    2. How are the customers distributed across all the states?

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

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

6. **Analysis based on the payments:**
    1. Find the month on month no. of orders placed using different payment types.
    2. Find the no. of orders placed on the basis of the payment installments that have been pai

Q1.Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

A.Data type of all columns in the "customers" table.

Ans A.

Select column_name,data_type from target_sql2023.INFORMATION_SCHEMA.COLUMNS where table_name = 'customers'

| JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS |

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 – most of the columns are string types

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

Ans.B. select min(order_purchase_timestamp) as first_order,
max(order_purchase_timestamp) as last_order from `target_sql2023.orders`

| JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS |

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

Insights- this market in brazil instead for 2 years with starting in 2016 and ending in 2018

C. Count the Cities & States of customers who ordered during the given period.
Ans.1.3.

```
select
count(customer_city) as Cities,
count(customer_state) as States
from `target_sql2023.customers`
group by customer_city,customer_state;
```

purchase_year by total_count

Chart configuration >|

Chart type
Bar

Dimension (x-axis)
total_count

Measures (y-axis)
purchase_year

Select up to 5 measures

PERSONAL HISTORY   PROJECT HISTORY   C REFRESH ^

JOB INFORMATION   RESULTS   JSON   EXECUTIC

Row	Cities ▾	States ▾
1	3	3
2	8	8
3	2	2
4	4	4
5	3	3

Insights: all of the data is similar in both columns

In-depth Exploration:

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

```sql
select
count(*) as total_count,
extract(YEAR from order_purchase_timestamp) as purchase_year
from `target_sql2023.orders`
group by extract(YEAR from order_purchase_timestamp);
```

| Row	total_count ▾	purchase_year ▾
1	45101	2017
2	54011	2018
3	329	2016

Insights-trends increases year by year

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

Ans B.

```sql
select count(*) as total_count,
extract(year from order_purchase_timestamp) as year,
extract(month from order_purchase_timestamp) as month
from `target_sql2023.orders`
group by extract(year from order_purchase_timestamp),
extract(month from order_purchase_timestamp)
```

Row	total_count	year	month
1	7544	2017	11
2	5673	2017	12
3	6728	2018	2
4	2404	2017	4
5	4026	2017	7
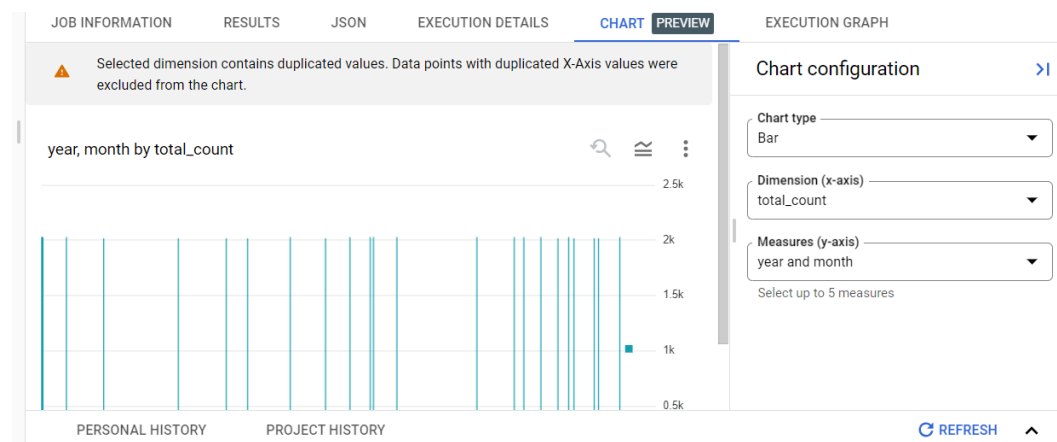
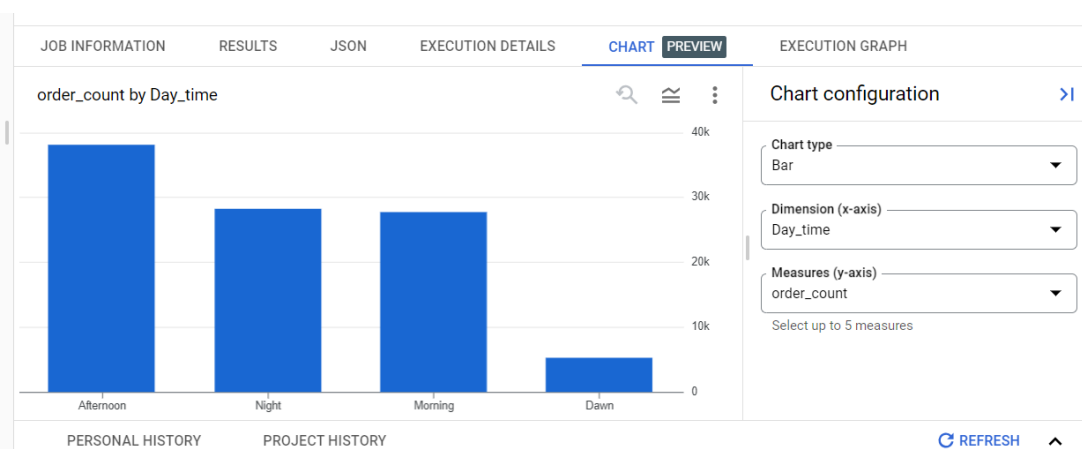Insights – number of orders are placed in months in every year

C. During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

- o   0-6 hrs : Dawn
- o   7-12 hrs : Mornings
- o   13-18 hrs : Afternoon
- o   19-23 hrs : Night

Ans C.

```
 select

count(order_id) as order_count,Day_time
from (select order_id,order_purchase_timestamp,
case
when(purchase_hour between 0 and 6) then 'Dawn'
when(purchase_hour between 7 and 12) then 'Morning'
when(purchase_hour between 13 and 18) then 'Afternoon'
else 'Night'
end as Day_time
from
(select order_id,order_purchase_timestamp,
extract(hour from order_purchase_timestamp) as purchase_hour
from `target_sql2023.orders`))
group by Day_time
order by order_count desc;
```

Row	order_count ▼	Day_time ▼
1	38135	Afternoon
2	28331	Night
3	27733	Morning
4	5242	Dawn

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

order_count by Day_time

Chart configuration

Chart type
Bar

Dimension (x-axis)
Day_time

Measures (y-axis)
order_count

Select up to 5 measures

PERSONAL HISTORY   PROJECT HISTORY   ↻ REFRESH   ∧

3.Evolution of E-commerce orders in the Brazil region:
   A. Get the month on month no. of orders placed in each state.
      1ans.

```sql
select
extract(month from o.order_purchase_timestamp) as month,
c.customer_state,
count(o.order_id) as order_count
from `target_sql2023.orders` o
join `target_sql2023.customers` c
on o.customer_id = c.customer_id
group by 1,2
order by 1,2;
```

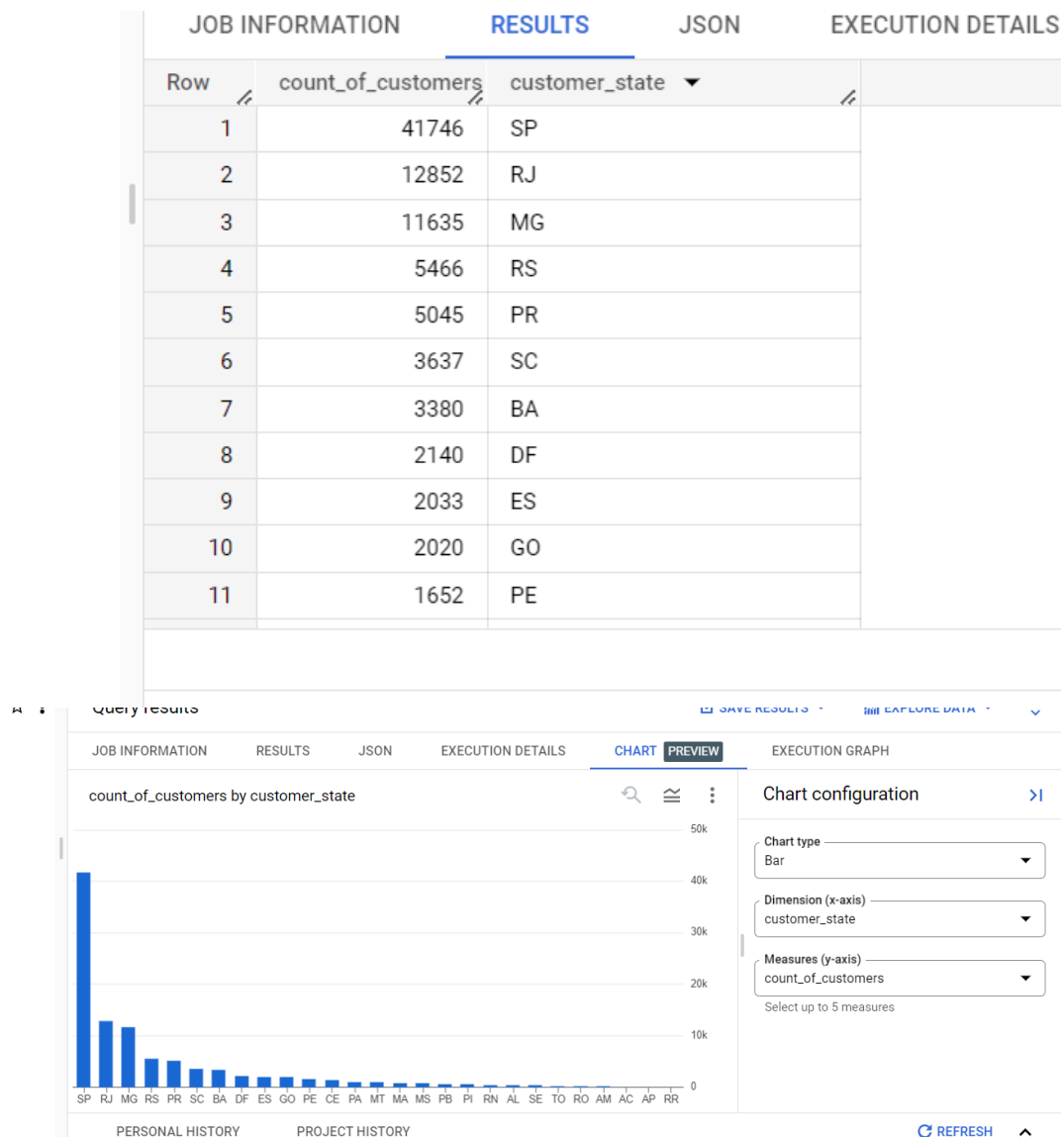| Row	month	customer_state	order_count
1	1	AC	8
2	1	AL	39
3	1	AM	12
4	1	AP	11
5	1	BA	264
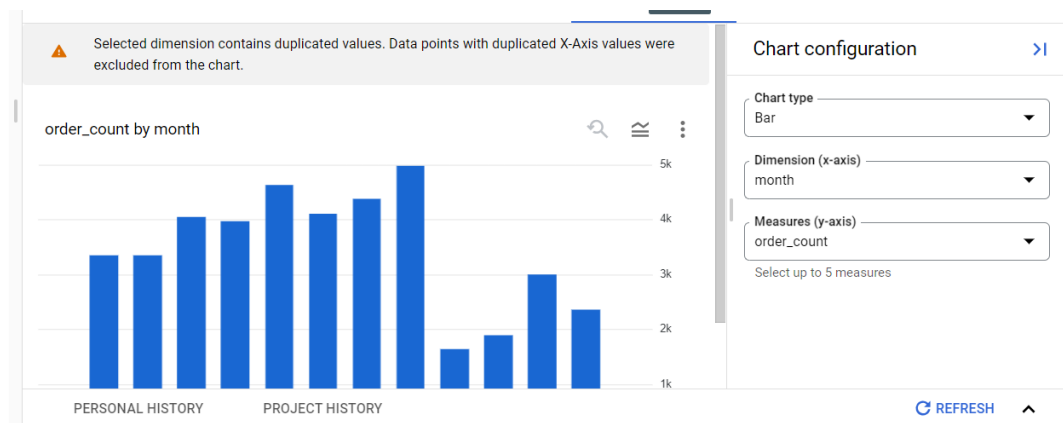
Insights – orders are placed in each states of brazil.

   B.How are the customers distributed across all the states?
      ANS B.

```sql
select count(*)
as count_of_customers,
customer_state
from `target_sql2023.customers`
group by customer_state
order by count_of_customers desc;
```

Insights- customers are distributed state wise

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

A. 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 A.

```
select
payment_sequential as year,
```

```
    payment_sequential as month,
100 * (sum(case when  payment_sequential = 2018 THEN payment_value end) -
sum(case when payment_sequential = 2017 THEN payment_value END)) /
sum(case when payment_sequential = 2017 THEN payment_value END) AS
cost_increase
from `target_sql2023.payments`
where payment_type = 'credit_card' and payment_sequential between 1 and 8
group by 1, 2
order by 1,2
```

Row	year ▼	month ▼	cost_increase ▼
1	1	1	null
2	2	2	null
3	3	3	null

Insights – increase of percentage in 2017 to 2018 (January to august)

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

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

```
SELECT c.customer_state, AVG(oi.price) AS avg_price, SUM(oi.price) as
total_avg_price
FROM `target_sql2023.order_items` oi
JOIN `target_sql2023.orders` o ON oi.order_id = o.order_id
JOIN `target_sql2023.customers` c ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2 DESC;
```

Row	customer_state ▼	avg_price ▼	total_avg_price ▼
1	PB	191.4752159468…	115268.0799999…
2	AL	180.8892117117…	80314.80999999…
3	AC	173.7277173913…	15982.94999999…
4	RO	165.9735251798…	46140.64000000…
5	PA	165.6924166666…	178947.8099999…
6	AP	164.3207317073…	13474.29999999…
7	PI	160.3580811808…	86914.07999999…
8	TO	157.5293333333…	49621.74000000…
9	RN	156.9659357277…	83034.97999999…
10	CE	153.7582611637…	227254.7099999…

Results per page: 50

avg_price, total_avg_price by customer_state

Chart configuration ›I

Chart type
Bar

Dimension (x-axis)
customer_state

Measures (y-axis)
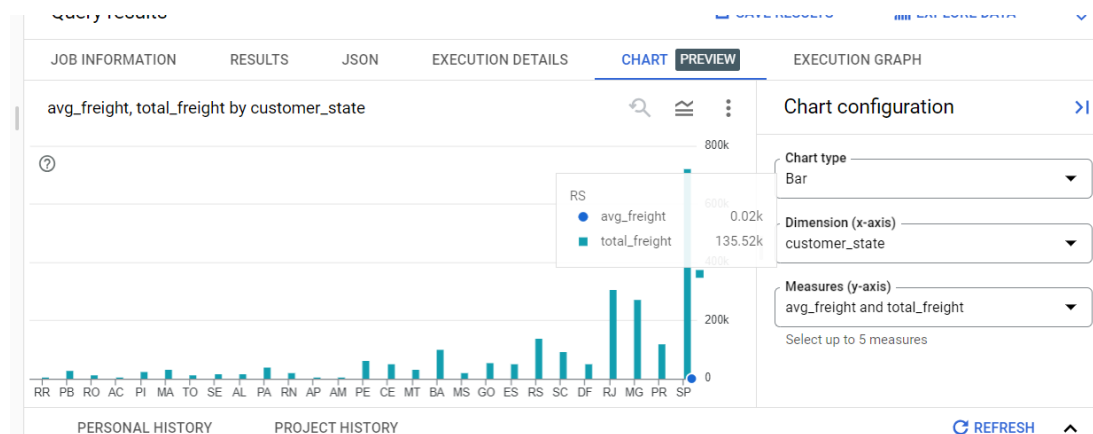avg_price and total_avg_price
Select up to 5 measures

Insights – total average of price of each states

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

```
SELECT c.customer_state,AVG(oi.freight_value) AS avg_freight,
SUM(oi.freight_value) AS total_freight
FROM `target_sql2023.order_items` oi
JOIN `target_sql2023.orders` o ON oi.order_id = o.order_id
JOIN `target_sql2023.customers` c ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2 DESC;
```

Insights – total average of freight of each state

5.Analysis based on sales, freight and delivery time.

A. 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

A.ans.

```sql
SELECT
order_id,order_purchase_timestamp,order_delivered_carrier_date,order_deliver
ed_customer_date, order_estimated_delivery_date,
  DATE_DIFF(order_purchase_timestamp,order_delivered_carrier_date,day) AS
time_to_delivery,
  DATE_DIFF(order_estimated_delivery_date,order_delivered_customer_date,day)
AS diff_estimated_delivery
FROM `target_sql2023.orders`
where order_status = 'delivered';
```

Row	order_id	order_purchase_timestamp	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_
1	635c894d068ac37e6e03dc54e...	2017-04-15 15:37:38 UTC	2017-04-27 16:06:59 UTC	2017-05-16 14:49:55 UTC	2017-05-18 00:00
2	f79bd0b3bacc5142f90f81a15b...	2017-04-22 13:55:16 UTC	2017-05-04 18:42:45 UTC	2017-05-12 13:55:55 UTC	2017-05-18 00:00
3	276e9ec344d3bf029ff83a161c...	2017-04-08 21:20:24 UTC	2017-04-25 10:53:00 UTC	2017-05-22 14:11:31 UTC	2017-05-18 00:00
4	cec8f5f7a13e5ab934a486ec9e...	2017-03-17 15:56:47 UTC	2017-04-04 10:53:37 UTC	2017-04-07 13:14:56 UTC	2017-05-18 00:00
5	81727d69b41f367b7d1e9e305...	2017-04-19 11:44:34 UTC	2017-05-04 14:03:47 UTC	2017-05-15 12:54:02 UTC	2017-05-18 00:00
6	7752cfbc93594ff73fd8b59c61...	2017-04-26 13:31:30 UTC	2017-05-16 12:18:15 UTC	2017-05-18 07:30:04 UTC	2017-05-18 00:00
7	10ed5499d1623638ee810eff1...	2017-03-21 13:38:25 UTC	2017-04-04 16:30:16 UTC	2017-04-18 13:52:43 UTC	2017-05-18 00:00
8	1b57522ebd11e3c3cdd09a554...	2017-07-07 22:55:15 UTC	2017-07-20 20:39:47 UTC	2017-07-27 18:55:49 UTC	2017-08-14 00:00
9	011b8dd4f8c36fd4fcf50643ce...	2017-07-07 22:00:51 UTC	2017-08-01 13:17:35 UTC	2017-08-04 20:57:27 UTC	2017-08-14 00:00
10	a6ee9db107e8fd0fc14d99be8...	2018-05-18 23:13:16 UTC	2018-06-04 16:03:00 UTC	2018-06-05 20:04:41 UTC	2018-06-06 00:00
11	a01f50d51f398895df76f09531...	2018-05-12 14:38:31 UTC	2018-05-28 14:51:00 UTC	2018-05-30 14:22:05 UTC	2018-06-06 00:00

Results per page: 50 ▼   1 – 50 of 96478

Insights –the number of days of delivery time as time to deliver and estimated time

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

Ans.B
```sql
SELECT c.customer_state,round(AVG(oi.freight_value),2) AS
avg_highest_freight,
FROM `target_sql2023.order_items` oi
JOIN `target_sql2023.orders` o ON oi.order_id = o.order_id
JOIN `target_sql2023.customers` c ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2 DESC
limit 5;
```

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

Insights -top 5 average highest freight customer state are RR,PB,RO,AC,PI

```sql
SELECT c.customer_state,round(AVG(oi.freight_value),2) AS
avg_lowest_freight,
FROM `target_sql2023.order_items` oi
JOIN `target_sql2023.orders` o ON oi.order_id = o.order_id
JOIN `target_sql2023.customers` c ON o.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2
limit 5;
```

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

PERSONAL HISTORY   PROJECT HISTORY

Insights – 5 lowest average freight of customer state are SP,PR,MG,RJ,DF

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

ans.c.

```sql
select
avg(time_to_deliver) as avg_time_delivered,
avg(diff_estimated_delivery) as avg_diff_estimated_delivery,
  customer_state from
(select order_id, customer_state,
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, Day) AS
time_to_deliver,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, Day)
AS diff_estimated_delivery
from
(select o.order_id, o.order_status, o.order_purchase_timestamp,
o.order_delivered_customer_date,
o.order_estimated_delivery_date,c.customer_state
from `target_sql2023.orders` as o
left join `target_sql2023.customers` as c
on o.customer_id=c.customer_id
where o.order_status = 'delivered'))
group by customer_state
order by avg_time_delivered desc
limit 5;
```

Row	avg_time_delivered	avg_diff_estimated_d	customer_state ▼
1	28.97560975609…	16.41463414634…	RR
2	26.73134328358…	18.73134328358…	AP
3	25.98620689655…	18.60689655172…	AM
4	24.04030226700…	7.947103274559…	AL
5	23.31606765327…	13.19027484143…	PA

Insights -top 5 states of highest and lowest delivery time are RR,AP,AM,AL,PA

```sql
select
avg(time_to_deliver) as avg_time_delivered,
avg(diff_estimated_delivery) as avg_diff_estimated_delivery,
  customer_state from
(select order_id, customer_state,
```

```
DATE_DIFF(order_delivered_customer_date, order_purchase_timestamp, Day) AS
time_to_deliver,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, Day)
AS diff_estimated_delivery
from
(select o.order_id, o.order_status, o.order_purchase_timestamp,
o.order_delivered_customer_date,
o.order_estimated_delivery_date,c.customer_state
from `target_sql2023.orders` as o
left join `target_sql2023.customers` as c
on o.customer_id=c.customer_id
where o.order_status = 'delivered'))
group by customer_state
order by avg_time_delivered
limit 5;
```

Row	avg_time_delivered	avg_diff_estimated_c	customer_state ▼
1	8.298093544722…	10.13448906010…	SP
2	11.52671135486…	12.36420881576…	PR
3	11.54218777523…	12.29910163818…	MG
4	12.50913461538…	11.11875000000…	DF
5	14.47518330513…	10.60095882684…	SC

Insights – top 5 lowest delivery time of states are SP,PR,MG,DF,SC

D. 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 D.
```
select
avg(diff_estimated_delivery) as avg_diff_estimated_delivery,
  customer_state from
(select order_id, customer_state,
DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, Day)
AS diff_estimated_delivery
from
```

```
(select o.order_id, o.order_status, o.order_purchase_timestamp,
o.order_delivered_customer_date, o.order_estimated_delivery_date,
c.customer_state
from `target_sql2023.orders` as o
left join `target_sql2023.customers` as c
on o.customer_id=c.customer_id
where o.order_status = 'delivered'))
group by customer_state
order by avg_diff_estimated_delivery DESC
limit 5;
```

Row	avg_diff_estimated_c	customer_state ▼
1	19.76250000000...	AC
2	19.13168724279...	RO
3	18.73134328358...	AP
4	18.60689655172...	AM
5	16.41463414634...	RR

Insights – the 5 states where the delivery Is very fast are AC,RO,AP,AM,RR

5.Analysis based on the payments:
A. Find the month on month no. of orders placed using different payment types.

ANS A.

```
select count(order_id) as count_order, payment_type,
extract(month from order_purchase_timestamp) as month from
(select o.order_id,o.order_purchase_timestamp, p.payment_type
from `target_sql2023.payments` as p
left join `target_sql2023.orders` as o
on p.order_id=o.order_id)
group by month,payment_type
order by month, payment_type;
```

	JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	CHART P

Row	count_order ▼	payment_type ▼	month ▼	
1	1715	UPI	1	
2	6103	credit_card	1	
3	118	debit_card	1	
4	477	voucher	1	
5	1723	UPI	2	
6	6609	credit_card	2	
7	82	debit_card	2	
8	424	voucher	2	
9	1942	UPI	3	

Query results   ⬆ SAVE RESULTS ▾   📊 EXPLORE DATA ▾   ↕

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS	CHART PREVIEW	EXECUTION GRAPH

month by count_order

Chart configuration

Chart type
Bar

Dimension (x-axis)
count_order

Measures (y-axis)
month

Select up to 5 measures

Insights — different payments types are upi,debit card,voucher and credit card

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

ANS B.

```
select
count(order_id) as total_count,payment_installments
from `target_sql2023.payments`
where payment_installments != 0
group by payment_installments;
```

Row	total_count	payment_installment
1	52546	1
2	12413	2
3	10461	3
4	7098	4
5	5239	5
6	3920	6
7	1626	7
8	4268	8
9	644	9
10	5328	10

Re

Insights -number of installments are paid