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

1.1) Data type of columns in a table

sol:
```
select column_name, data_type
from `my-project-scaler-381417.target.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

similarly we can just change the "table name" in above query to get the datatypes of required table.

1.2)Time period for which the data is given

sol:
```
SELECT Max(order_purchase_timestamp) as lastorder,MIN((order_purchase_timestamp)) as firstorder FROM `my-project-scaler-381417.target.orders`
```

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

here i tried to find the max timestamp and min timestamp from the orders table so that i could the time period for which the data is given
so the time period in year is "2016-2018".

1.3) Cities and States of customers ordered during the given period

sol:
```
SELECT customer_city,customer_state
FROM `my-project-scaler-381417.target.customers` as c join `my-project-scaler-381417.target.orders` as o
on(o.customer_id = c.customer_id) limit 10
```

Row	customer_city	customer_state
1	acu	RN
2	acu	RN
3	acu	RN
4	ico	CE
5	ico	CE
6	ico	CE
7	ico	CE
8	ico	CE
9	ico	CE
10	ico	CE

customer_city

2. In-depth Exploration:

2.1)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?
sol:

```
SELECT count(order_id) as total_order,extract(year FROM order_purchase_timestamp) as year
FROM `my-project-scaler-381417.target.customers` as c join `my-project-scaler-
381417.target.orders` as o
on(o.customer_id = c.customer_id)
group by year
order by total_order desc
```

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

here I have checked the growth in e-commerce at "year" level, according to the result it seems the growth in ecommerce is strong,

orders are increasing yearly drastically.

```
SELECT count(order_id) as total_order,CASE EXTRACT(MONTH FROM o.order_purchase_timestamp)
    WHEN 1 THEN 'January'
    WHEN 2 THEN 'February'
    WHEN 3 THEN 'March'
    WHEN 4 THEN 'April'
    WHEN 5 THEN 'May'
    WHEN 6 THEN 'June'
    WHEN 7 THEN 'July'
    WHEN 8 THEN 'August'
    WHEN 9 THEN 'September'
    WHEN 10 THEN 'October'
    WHEN 11 THEN 'November'
    WHEN 12 THEN 'December'
  END AS month_name
FROM `my-project-scaler-381417.target.customers` as c join `my-project-scaler-
381417.target.orders` as o
on(o.customer_id = c.customer_id)
group by EXTRACT(MONTH FROM o.order_purchase_timestamp),month_name
order by total_order desc
```

Row	total_order	month_name
1	10843	August
2	10573	May
3	10318	July
4	9893	March
5	9412	June
6	9343	April
7	8508	February
8	8069	January
9	7544	November
10	5674	December

above query is to get the insights on the monthly orders placed, so result shows
top 3 months where order are placed more are "August,May,July" resp and least orders are
placed in the month of "September"
so there is a peak in order in the month of "August" followed by the other two mentioned
above.

2.2) What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?
sol:

```
SELECT
  CASE
    WHEN extract(time from order_purchase_timestamp) < TIME '06:00:00' THEN 'Dawn'
    WHEN extract(time from order_purchase_timestamp) >= TIME '06:00:00' AND extract(time fro
m order_purchase_timestamp) < TIME '12:00:00' THEN 'Morning'
    WHEN extract(time from order_purchase_timestamp) >= TIME '12:00:00' AND extract(time fro
m order_purchase_timestamp) < TIME '18:00:00' THEN 'Afternoon'
    ELSE 'Night'
  END AS period,
  COUNT(*) AS orders_count
FROM `my-project-scaler-381417.target.orders`
GROUP BY period
Order by orders_count desc
```

Row	period	orders_count
1	Afternoon	38361
2	Night	34100
3	Morning	22240
4	Dawn	4740

above query is to find at what time do brazilians tend to buy more, so the result shows that
they tends buy more in afternoon, followed by
night,morning, dawn resp.

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

3.1)Get month on month orders by states

Sol:

```
select extract(Month from o.order_purchase_timestamp)as Month,extract(Year from o.order_purcha
se_timestamp) as Year,c.customer_state,count(distinct o.order_id)as No_of_Orders
```

```
from `target.orders` as o join `target.customers` as c on c.customer_id=o.customer_id
group by Month,Year,c.customer_state
order by Year,Month
```

Row	Month	Year	customer_state	No_of_Orders
1	9	2016	RS	1
2	9	2016	RR	1
3	9	2016	SP	2
4	10	2016	SP	113
5	10	2016	MG	40
6	10	2016	GO	9
7	10	2016	CE	8
8	10	2016	SC	11
9	10	2016	RJ	56
10	10	2016	RS	24

3.2) Distribution of customers across the states in Brazil

Sol:

```
SELECT customer_state, COUNT(*) AS total_customers
FROM `target.customers`
GROUP BY customer_state
ORDER BY total_customers DESC
```

Row	customer_state	total_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

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

4.1)Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use "payment_value" column in payments table

Sol:
```
SELECT
  EXTRACT(YEAR FROM o.order_purchase_timestamp) AS year,
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  AVG(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 THEN p.payment_value END)
 AS avg_2017_payment_value,
  AVG(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018 THEN p.payment_value END)
 AS avg_2018_payment_value,
  (AVG(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2018 THEN p.payment_value END
) -
   AVG(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 THEN p.payment_value END
)) /
  AVG(CASE WHEN EXTRACT(YEAR FROM o.order_purchase_timestamp) = 2017 THEN p.payment_value END)
 AS percent_increase
FROM
```

```
  `my-project-scaler-381417.target.orders` AS o
  JOIN `my-project-scaler-381417.target.payments` AS p ON o.order_id = p.order_id
WHERE
  EXTRACT(YEAR FROM o.order_purchase_timestamp) IN (2017, 2018) AND EXTRACT(MONTH FROM o.order
_purchase_timestamp) BETWEEN 1 AND 8
GROUP BY
  year, month
ORDER BY

  year ASC, month ASC
```

Row	year	month	avg_2017_paym	avg_2018_paym	percent_increas
1	2017	1	162.927105...	null	null
2	2017	2	154.776251...	null	null
3	2017	3	158.570179...	null	null
4	2017	4	162.500206...	null	null
5	2017	5	150.334386...	null	null
6	2017	6	148.799877...	null	null
7	2017	7	137.220968...	null	null
8	2017	8	148.218971...	null	null
9	2018	1	null	147.428821...	null
10	2018	2	null	142.759398...	null

4.2)Mean & Sum of price and freight value by customer state

Sol: SELECT

```
  c.customer_state,
  AVG(p.payment_value) AS mean_payment_value,
  SUM(p.payment_value) AS total_payment_value,
  AVG(oi.freight_value) AS mean_fulfillment,
  SUM(oi.freight_value) AS total_fulfillment
FROM
  `my-project-scaler-381417.target.order_items` AS oi
  JOIN `my-project-scaler-381417.target.orders` AS o ON oi.order_id = o.order_id
  JOIN `my-project-scaler-381417.target.customers` AS c ON o.customer_id = c.customer_id
  JOIN `my-project-scaler-381417.target.payments` AS p ON o.order_id = p.order_id
GROUP BY
  c.customer_state
ORDER BY
  total_payment_value DESC
```

Row	customer_state	mean_payment_	total_payment_v	mean_fulfillmen	total_fulfillment_
1	SP	153.274616...	7597209.66...	15.1989504...	753351.179...
2	RJ	180.684246...	2769347.43...	21.1009297...	323413.950...
3	MG	170.563985...	2326151.63...	20.6262875...	281301.310...
4	RS	176.885137...	1147276.99...	21.8285060...	141579.690...
5	PR	178.564909...	1064603.98...	20.5752583...	122669.690...
6	BA	196.988725...	797410.359...	26.3188290...	106538.619...
7	SC	182.785613...	786343.710...	21.4356950...	92216.3600...
8	GO	211.472839...	513879.000...	22.7314938...	55237.5299...
9	DF	174.938831...	432623.730...	21.0751475...	52118.8399...
10	ES	173.569435...	405805.340...	21.9814242...	51392.5699...

5. . Analysis on sales, freight and delivery time

5.1) Calculate days between purchasing, delivering and estimated delivery

Sol: SELECT

```
  oi.order_id,
  oi.order_item_id,
  oi.product_id,
  oi.seller_id,
  oi.shipping_limit_date,
  oi.price,
  oi.freight_value,
  o.order_purchase_timestamp,
  o.order_delivered_customer_date,
  o.order_estimated_delivery_date,
  DATE_DIFF(o.order_purchase_timestamp, o.order_delivered_customer_date, DAY) AS days_to_delivery,
  DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY) AS
diff_estimated_delivery
FROM
  `my-project-scaler-381417.target.order_items` AS oi
  JOIN `my-project-scaler-381417.target.orders` AS o ON oi.order_id = o.order_id
```

Row		shipping_limit_date	price	freight_value	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	days_to_delivery	diff_estimated_d
1	84e8de603b16...	2018-07-09 13:31:36 UTC	3.0	12.79	2018-07-03 12:37:02 UTC	2018-07-10 10:24:34 UTC	2018-07-24 00:00:00 UTC	-6	13
2	84e8de603b16...	2018-08-14 14:04:44 UTC	3.0	15.23	2018-08-10 13:47:16 UTC	2018-08-22 23:03:23 UTC	2018-08-23 00:00:00 UTC	-12	0
3	b9c6440f124f...	2017-05-12 19:05:20 UTC	3.5	8.72	2017-05-01 18:58:54 UTC	null	2017-06-06 00:00:00 UTC	null	null
4	884e7742faac...	2018-06-28 01:30:49 UTC	3.5	7.39	2018-06-21 20:29:25 UTC	2018-07-04 14:04:53 UTC	2018-07-12 00:00:00 UTC	-12	7
5	884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23	2018-06-06 18:49:33 UTC	2018-06-22 15:05:01 UTC	2018-07-12 00:00:00 UTC	-15	19
6	884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23	2018-06-06 18:49:33 UTC	2018-06-22 15:05:01 UTC	2018-07-12 00:00:00 UTC	-15	19
7	884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23	2018-06-06 18:49:33 UTC	2018-06-22 15:05:01 UTC	2018-07-12 00:00:00 UTC	-15	19
8	884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23	2018-06-06 18:49:33 UTC	2018-06-22 15:05:01 UTC	2018-07-12 00:00:00 UTC	-15	19
9	884e7742faac...	2018-06-12 19:15:14 UTC	3.5	18.23	2018-06-06 18:49:33 UTC	2018-06-22 15:05:01 UTC	2018-07-12 00:00:00 UTC	-15	19
10	1d1c43cde1c5...	2017-10-20 14:50:12 UTC	4.5	11.85	2017-10-16 14:29:26 UTC	2017-10-25 09:42:40 UTC	2017-11-01 00:00:00 UTC	-8	6

5.2) Find time_to_delivery & diff_estimated_delivery. Formula for the same given below

time_to_delivery = order_purchase_timestamp-order_delivered_customer_date

diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

sol: SELECT

order_id,

order_purchase_timestamp,

order_delivered_customer_date,

order_estimated_delivery_date,

TIMESTAMP_DIFF(order_purchase_timestamp, order_delivered_customer_date, DAY) AS time_to_delivery,

TIMESTAMP_DIFF(order_estimated_delivery_date, order_delivered_customer_date, DAY) AS diff_estimated_delivery

FROM

`my-project-scaler-381417.target.orders`

Row	order_id	order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_delivery	diff_estimated_c
1	7a4df5d8cff4090e541401a20a...	2017-11-25 11:10:33 UTC	null	2017-12-12 00:00:00 UTC	null	null
2	35de4050331c6c644cddc86f4...	2017-12-05 01:07:58 UTC	null	2018-01-08 00:00:00 UTC	null	null
3	b5359909123fa03c50bdb0cfe...	2017-12-05 01:07:52 UTC	null	2018-01-11 00:00:00 UTC	null	null
4	dba5062fbda3af4fb6c33b1e04...	2018-02-09 17:21:04 UTC	null	2018-03-07 00:00:00 UTC	null	null
5	90ab3e7d52544ec7bc3363c82...	2017-11-06 13:12:34 UTC	null	2017-12-01 00:00:00 UTC	null	null
6	fa65dad1b0e818e3ccc5cb0e3...	2017-04-20 12:45:34 UTC	null	2017-05-18 00:00:00 UTC	null	null
7	1df2775799eecdf9dd8502425...	2017-07-13 11:03:05 UTC	null	2017-08-14 00:00:00 UTC	null	null
8	6190a94657e1012983a274b8...	2017-07-11 13:36:30 UTC	null	2017-08-14 00:00:00 UTC	null	null
9	58ce513a55c740a3a81e8c8b7...	2017-07-29 18:05:07 UTC	null	2017-08-14 00:00:00 UTC	null	null
10	088683f795a3d30bfd61152c4f...	2017-07-13 10:02:47 UTC	null	2017-08-14 00:00:00 UTC	null	null

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

Sol: SELECT

c.customer_state,

```
  AVG(oi.freight_value) AS mean_freight,

  AVG(DATE_DIFF(o.order_purchase_timestamp, o.order_delivered_customer_date, DAY)) AS
mean_days_to_delivery,

  AVG(DATE_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)) AS
mean_diff_estimated_delivery

FROM

  `my-project-scaler-381417.target.order_items` AS oi

  JOIN `my-project-scaler-381417.target.orders` AS o ON oi.order_id = o.order_id

  JOIN `my-project-scaler-381417.target.customers` AS c ON o.customer_id = c.customer_id

GROUP BY

  c.customer_state
```

Row	customer_state	mean_freight	mean_days_to_d	mean_diff_estim
1	MT	28.1662843...	-17.5081967...	13.6393442...
2	MA	38.2570024...	-21.2037500...	9.10999999...
3	AL	35.8436711...	-23.9929742...	7.97658079...
4	SP	15.1472753...	-8.25960855...	10.2655943...
5	MG	20.6301668...	-11.5155221...	12.3971510...
6	PE	32.9178626...	-17.7920962...	12.5521191...
7	RJ	20.9609239...	-14.6893821...	11.1444931...
8	DF	21.0413549...	-12.5014861...	11.2747346...
9	RS	21.7358043...	-14.7082993...	13.2030001...
10	SE	36.6531688...	-20.9786666...	9.16533333...

7.) top 5 states with highest/lowest average freight value - sort in desc/asc limit 5

Sol:

```
SELECT

  c.customer_state,

  AVG(oi.freight_value) AS mean_freight

FROM

  `my-project-scaler-381417.target.order_items` AS oi

  JOIN `my-project-scaler-381417.target.orders` AS o ON oi.order_id = o.order_id
```

```
  JOIN `my-project-scaler-381417.target.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state
ORDER BY
  mean_freight DESC
LIMIT
  5
```

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

```
SELECT
  c.customer_state,
  AVG(oi.freight_value) AS mean_freight
FROM
  `my-project-scaler-381417.target.order_items` AS oi
  JOIN `my-project-scaler-381417.target.orders` AS o ON oi.order_id = o.order_id
  JOIN `my-project-scaler-381417.target.customers` AS c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state
ORDER BY
```

```
  mean_freight ASC
LIMIT
  5
```

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...

8.) Top 5 states with lowest average time to delivery

Sol:

```
SELECT
  c.customer_state,
  AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY)) AS mea
n_time_to_delivery
FROM
  `my-project-scaler-381417.target.orders` AS o
JOIN
  `my-project-scaler-381417.target.customers` AS c
ON
  o.customer_id = c.customer_id
GROUP BY
  c.customer_state
ORDER BY
  mean_time_to_delivery ASC
LIMIT 5
```

Row	customer_state	mean_time_to_d
1	SP	8.29806148...
2	PR	11.5267113...
3	MG	11.5438132...
4	DF	12.5091346...
5	SC	14.4795601...

9.) Top 5 states with highest average time to delivery

Sol: SELECT

```
  c.customer_state,
  AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY)) AS mea
n_time_to_delivery
FROM
  `my-project-scaler-381417.target.orders` AS o
JOIN
  `my-project-scaler-381417.target.customers` AS c
ON
  o.customer_id = c.customer_id
GROUP BY
  c.customer_state
ORDER BY
  mean_time_to_delivery DESC
LIMIT
  5
```

Row	customer_state	mean_time_to_d
1	RR	28.9756097...
2	AP	26.7313432...
3	AM	25.9862068...
4	AL	24.0403022...
5	PA	23.3160676...

9.) Top 5 states where delivery is really fast/ not so fast compared to estimated date

Sol: SELECT

```
  c.customer_state,
  AVG(TIMESTAMP_DIFF(o.order_delivered_customer_date, o.order_purchase_timestamp, DAY)) AS mea
n_time_to_delivery,
  AVG(TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_delivered_customer_date, DAY)) A
S mean_diff_estimated_delivery,
  AVG(TIMESTAMP_DIFF(o.order_estimated_delivery_date, o.order_purchase_timestamp, DAY)) AS mea
n_estimated_delivery
FROM
  `my-project-scaler-381417.target.orders` o
  JOIN `my-project-scaler-381417.target.customers` c ON o.customer_id = c.customer_id
GROUP BY
  c.customer_state
HAVING
  mean_time_to_delivery < mean_diff_estimated_delivery AND mean_time_to_delivery < mean_estima
ted_delivery
ORDER BY
  mean_time_to_delivery ASC
LIMIT
  5
```

Row	customer_state	mean_time_to_d	mean_diff_estim	mean_estimated
1	SP	8.29806148...	10.1353253...	18.8091074...
2	PR	11.5267113...	12.3642088...	24.2517343...
3	MG	11.5438132...	12.2969616...	24.2241512...
4	RO	18.9135802...	19.1316872...	38.4071146...

``

10.) Payment type analysis:

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

Sol: SELECT

```
  EXTRACT(MONTH FROM o.order_purchase_timestamp) AS month,
  c.customer_state,
  o.order_purchase_timestamp
FROM
  `my-project-scaler-381417.target.customers` AS c
  JOIN `my-project-scaler-381417.target.orders` AS o ON o.customer_id = c.customer_id
GROUP BY
  EXTRACT(MONTH FROM o.order_purchase_timestamp),
  c.customer_state,
  o.order_purchase_timestamp;
```

Row	month	customer_state	order_purchase_timestamp
1	1	RN	2018-01-26 22:12:04 UTC
2	1	SP	2018-01-13 11:23:50 UTC
3	1	SP	2018-01-17 06:17:02 UTC
4	1	SP	2017-01-19 13:38:16 UTC
5	1	SP	2018-01-15 14:41:21 UTC
6	1	SP	2018-01-01 15:20:38 UTC
7	1	SP	2018-01-21 16:32:05 UTC
8	1	SP	2018-01-03 11:17:39 UTC
9	1	SP	2018-01-09 10:16:34 UTC
10	1	SP	2018-01-16 10:05:17 UTC

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

Sol:

```
SELECT
  payment_installments,
  COUNT(order_id) AS order_count
FROM
  `my-project-scaler-381417.target.payments`
GROUP BY
  payment_installments
ORDER BY
  payment_installments;
```

Row	payment_installr	order_count
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