Context:

Target is a globally renowned brand and a prominent retailer in the United States. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This particular business case focuses on the operations of Target in Brazil and provides insightful information about 100,000 orders placed between 2016 and 2018. The dataset offers a comprehensive view of various dimensions including the order status, price, payment and freight performance, customer location, product attributes, and customer reviews.

By analyzing this extensive dataset, it becomes possible to gain valuable insights into Target's operations in Brazil. The information can shed light on various aspects of the business, such as order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction levels.

Dataset: https://drive.google.com/drive/folders/1TGEc66YKbD443nslRi1bWgVd238gJCnb

The data is available in 8 csv files:

1. customers.csv
2. sellers.csv
3. order_items.csv
4. geolocation.csv
5. payments.csv
6. reviews.csv
7. orders.csv
8. products.csv

The column description for these csv files is given below.

The **customers.csv** contain following features:

Features	**Description**
customer_id	ID of the consumer who made the purchase
customer_unique_id	Unique ID of the consumer
customer_zip_code_prefix	Zip Code of consumer’s location
customer_city	Name of the City from where order is made
customer_state	State Code from where order is made (Eg. são paulo - SP)

The **sellers.csv** contains following features:

Features	**Description**
seller_id	Unique ID of the seller registered
seller_zip_code_prefix	Zip Code of the seller’s location
seller_city	Name of the City of the seller
seller_state	State Code (Eg. são paulo - SP)

The **order_items.csv** contain following features:

Features	**Description**
order_id	A Unique ID of order made by the consumers
order_item_id	A Unique ID given to each item ordered in the order
product_id	A Unique ID given to each product available on the site
seller_id	Unique ID of the seller registered in Target
shipping_limit_date	The date before which the ordered product must be shipped
price	Actual price of the products ordered
freight_value	Price rate at which a product is delivered from one point to another

The **geolocations.csv** contain following features:

Features	**Description**
geolocation_zip_code_prefix	First 5 digits of Zip Code
geolocation_lat	Latitude
geolocation_lng	Longitude
geolocation_city	City
geolocation_state	State

The **payments.csv** contain following features:

Features	**Description**
order_id	A Unique ID of order made by the consumers
payment_sequential	Sequences of the payments made in case of EMI
payment_type	Mode of payment used (Eg. Credit Card)
payment_installments	Number of installments in case of EMI purchase
payment_value	Total amount paid for the purchase order

The **orders.csv** contain following features:

Features	**Description**
order_id	A Unique ID of order made by the consumers
customer_id	ID of the consumer who made the purchase
order_status	Status of the order made i.e. delivered, shipped, etc.
order_purchase_timestamp	Timestamp of the purchase
order_delivered_carrier_date	Delivery date at which carrier made the delivery

order_delivered_customer_date	Date at which customer got the product
order_estimated_delivery_date	Estimated delivery date of the products

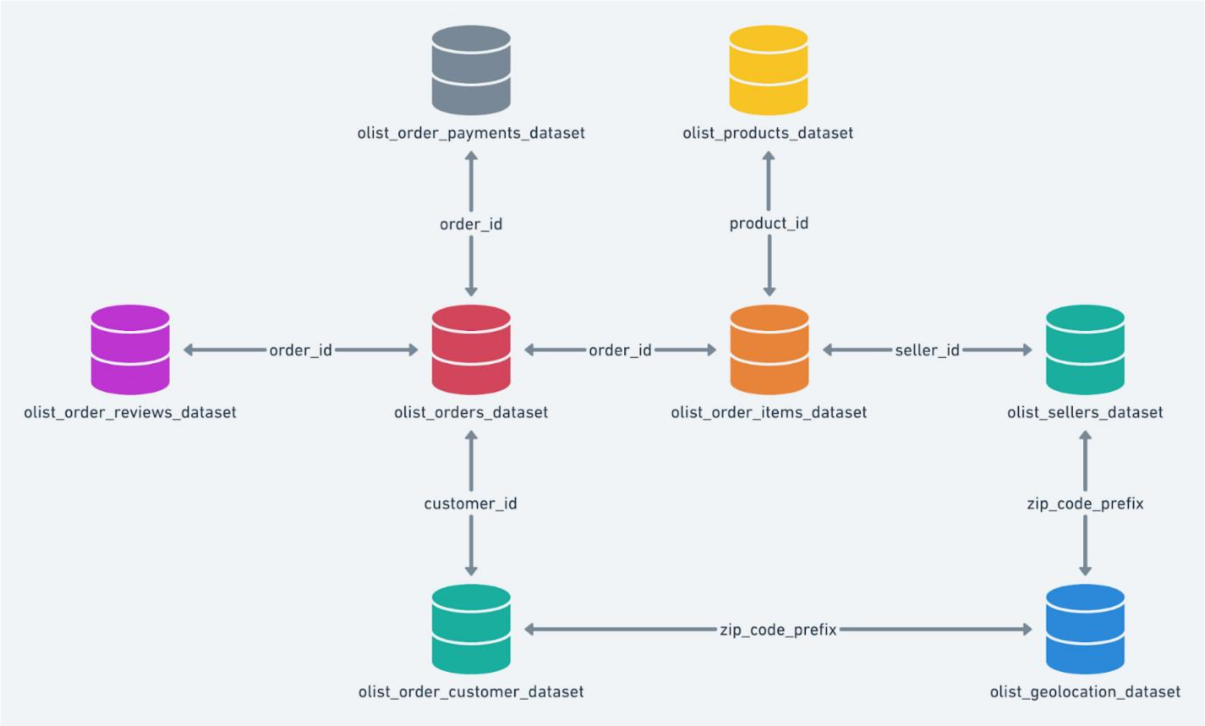
The **reviews.csv** contain following features:

Features	Description
review_id	ID of the review given on the product ordered by the order id
order_id	A Unique ID of order made by the consumers
review_score	Review score given by the customer for each order on a scale of 1-5
review_comment_title	Title of the review
review_comment_message	Review comments posted by the consumer for each order
review_creation_date	Timestamp of the review when it is created
review_answer_timestamp	Timestamp of the review answered

The **products.csv** contain following features:

Features	Description
product_id	A Unique identifier for the proposed project.
product_category_name	Name of the product category
product_name_lenght	Length of the string which specifies the name given to the products o
product_description_lenght	Length of the description written for each product ordered on the site
product_photos_qty	Number of photos of each product ordered available on the shopping
product_weight_g	Weight of the products ordered in grams
product_length_cm	Length of the products ordered in centimeters
product_height_cm	Height of the products ordered in centimeters
product_width_cm	Width of the product ordered in centimeters

Dataset schema:



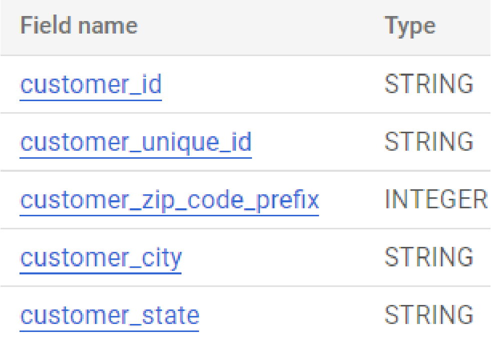
Problem Statement:

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.

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

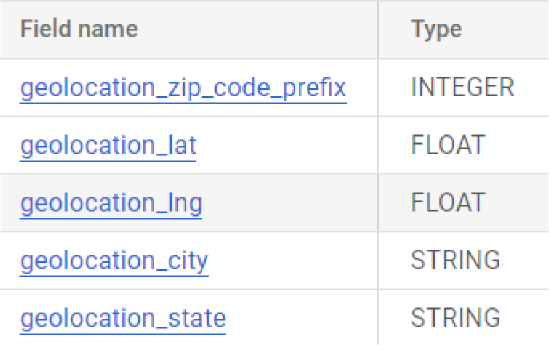
1. Data type of columns in a table

customers :



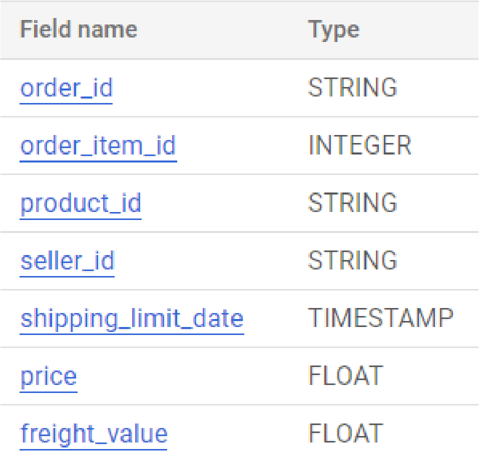
Field name	Type
customer_id	STRING
customer_unique_id	STRING
customer_zip_code_prefix	INTEGER
customer_city	STRING
customer_state	STRING

geolocation :



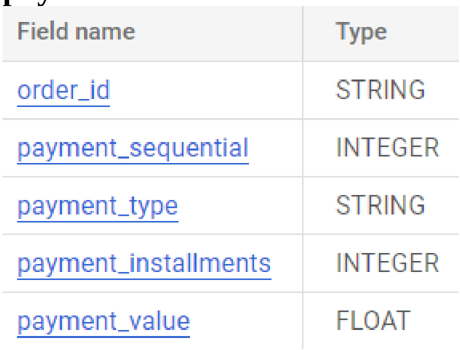
Field name	Type
geolocation_zip_code_prefix	INTEGER
geolocation_lat	FLOAT
geolocation_lng	FLOAT
geolocation_city	STRING
geolocation_state	STRING

order_items :



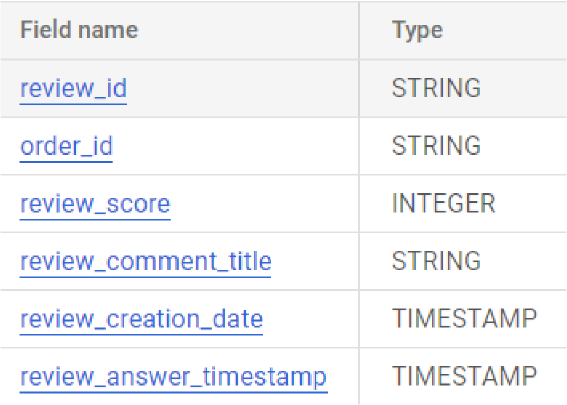
Field name	Type
order_id	STRING
order_item_id	INTEGER
product_id	STRING
seller_id	STRING
shipping_limit_date	TIMESTAMP
price	FLOAT
freight_value	FLOAT

payment :



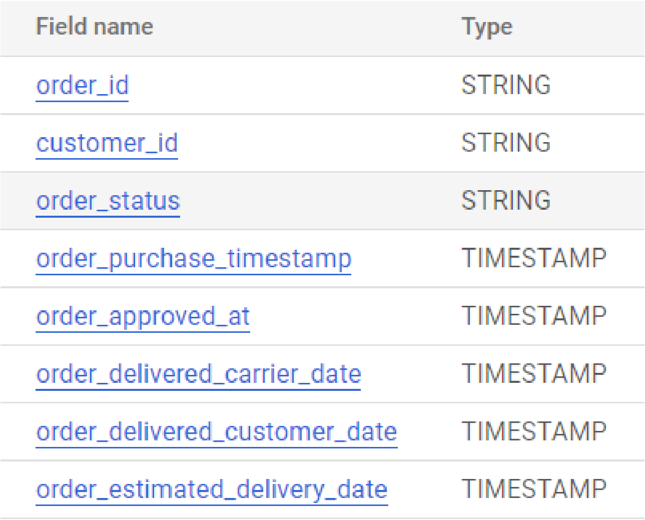
Field name	Type
order_id	STRING
payment_sequential	INTEGER
payment_type	STRING
payment_installments	INTEGER
payment_value	FLOAT

reviews :



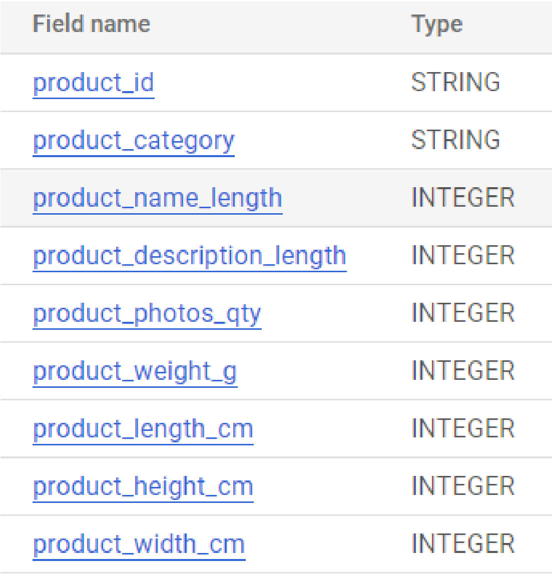
Field name	Type
review_id	STRING
order_id	STRING
review_score	INTEGER
review_comment_title	STRING
review_creation_date	TIMESTAMP
review_answer_timestamp	TIMESTAMP

orders :



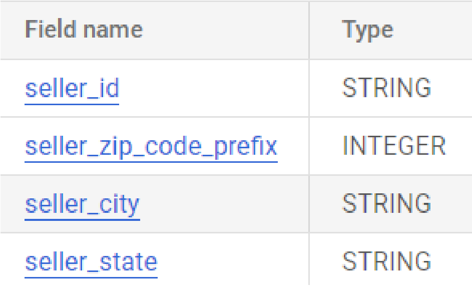
Field name	Type
order_id	STRING
customer_id	STRING
order_status	STRING
order_purchase_timestamp	TIMESTAMP
order_approved_at	TIMESTAMP
order_delivered_carrier_date	TIMESTAMP
order_delivered_customer_date	TIMESTAMP
order_estimated_delivery_date	TIMESTAMP

products :



Field name	Type
product_id	STRING
product_category	STRING
product_name_length	INTEGER
product_description_length	INTEGER
product_photos_qty	INTEGER
product_weight_g	INTEGER
product_length_cm	INTEGER
product_height_cm	INTEGER
product_width_cm	INTEGER

sellers :

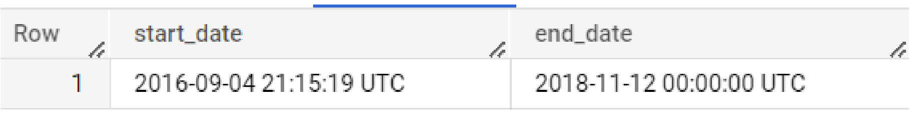


Field name	Type
seller_id	STRING
seller_zip_code_prefix	INTEGER
seller_city	STRING
seller_state	STRING

2. Time period for which the data is given

assumption - max and min based on `order purchase timestamp` .

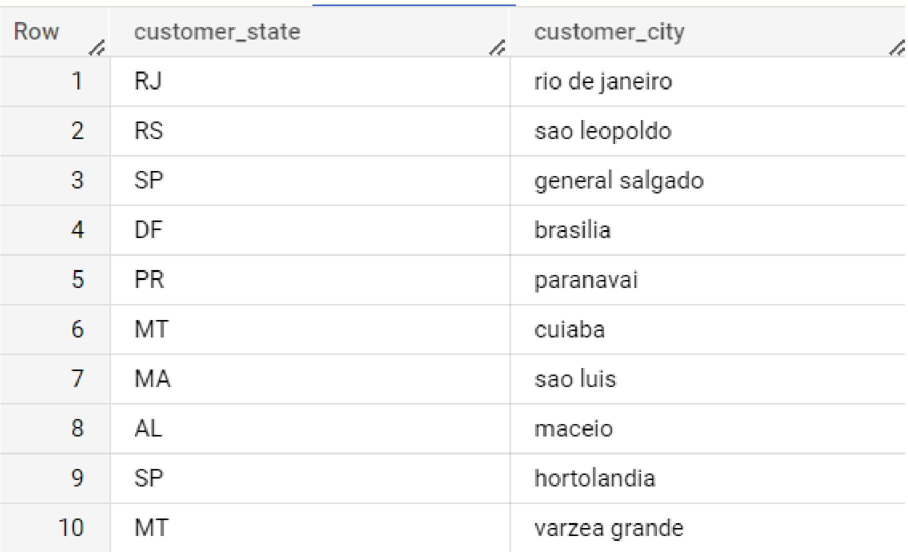
```
 select min(order_purchase_timestamp) start_date ,
 max(order_estimated_delivery_date) end_date
from `Target_Data.orders` ;
```



Row	start_date	end_date
1	2016-09-04 21:15:19 UTC	2018-11-12 00:00:00 UTC

3. Cities and States of customers ordered during the given period

```
SELECT
  DISTINCT customers.customer_state,
  customers.customer_city
FROM
  `Target_Data.customers` customers
JOIN
  `Target_Data.orders` orders
ON
  customers.customer_id = orders.customer_id ;
```



Row	customer_state	customer_city
1	RJ	rio de janeiro
2	RS	sao leopoldo
3	SP	general salgado
4	DF	brasilia
5	PR	paranavai
6	MT	cuiaba
7	MA	sao luis
8	AL	maceio
9	SP	hortolandia
10	MT	varzea grande

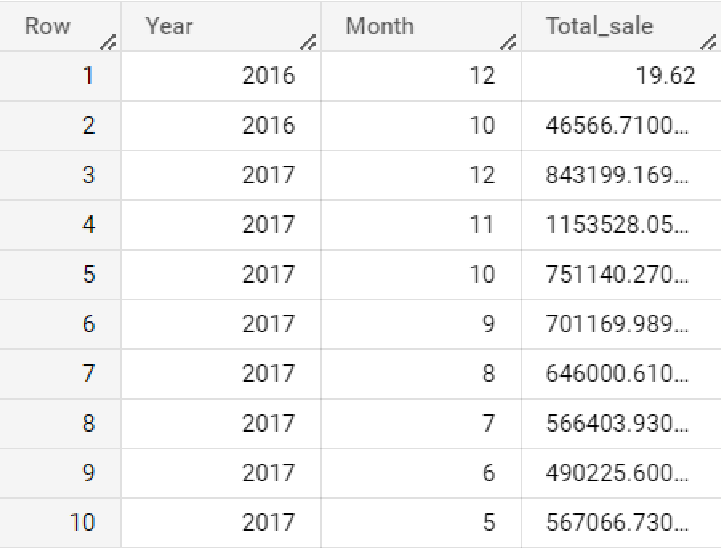
Q2.) In-depth Exploration:

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?

Assumption : total sales only on the basis of product status "DELIVERED"

```
SELECT
```

```
  EXTRACT(year
  FROM
    order_purchase_timestamp) Year,
  EXTRACT(month
  FROM
    order_purchase_timestamp) Month,
  SUM(payment_value) Total_sale
FROM
  `Target_Data.orders` o
JOIN
  `Target_Data.payments` p
ON
  o.order_id = p.order_id
WHERE
  order_status = "delivered"
GROUP BY
  1 , 2
ORDER BY
  1, 2 ;
```



Row	Year	Month	Total_sale
1	2016	12	19.62
2	2016	10	46566.7100...
3	2017	12	843199.169...
4	2017	11	1153528.05...
5	2017	10	751140.270...
6	2017	9	701169.989...
7	2017	8	646000.610...
8	2017	7	566403.930...
9	2017	6	490225.600...
10	2017	5	567066.730...

Insights - a) There is a growing trend in sales year on year.
b) NOV , JAN , FEB , MARCH are the months where sudden increase in sales can be seen in above table.

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

Assumption : Between 0 to 6 - "DAWN"
Between 7 to 12 - "MORNING"

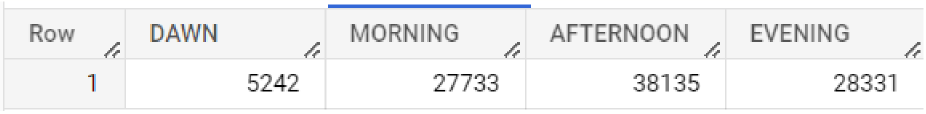
Between 13 to 18 - "AFTERNOON"
Between 19 to 23 - "NIGHT"

```
with query1 as(
select
extract(hour from order_purchase_timestamp ) hour ,
count(distinct order_id) all_order
from `Target_Data.orders`
group by 1

)

select sum(case when hour between 0 and 6  then all_order end) DAWN ,
sum(case when hour between 7 and 12  then all_order end) MORNING ,
sum(case when hour between 13 and 18  then all_order end) AFTERNOON ,
sum(case when hour between 19 and 23  then all_order end) EVENING

from query1
```



Row	DAWN	MORNING	AFTERNOON	EVENING
1	5242	27733	38135	28331

```
  Insight- Brazilian customers buy most in the afternoon.
```

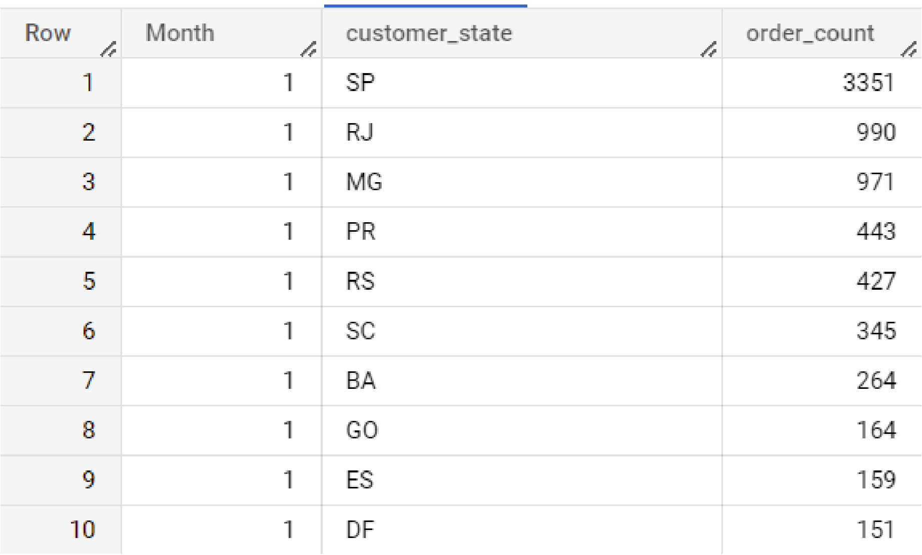
Q3) Evolution of E-commerce orders in the Brazil region

1. Get month on month orders by states

Assumption - order count with all order status.

```
SELECT
  EXTRACT(month
  FROM
    order_purchase_timestamp) Month,
  customer_state,
  COUNT(order_id) order_count
FROM
  `Target_Data.orders` o
JOIN
  `Target_Data.customers` c
ON
```

```
  o.customer_id = c.customer_id
GROUP BY
  1,
  2
ORDER BY
  1,
  3 DESC
```

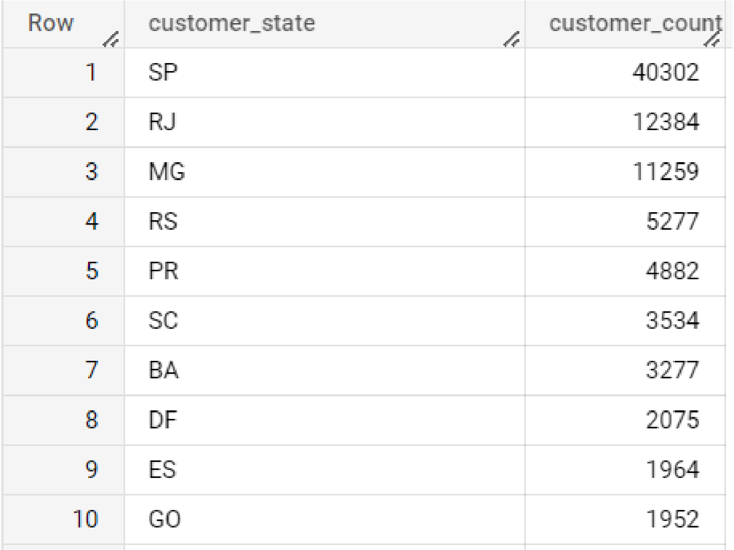


Row	Month	customer_state	order_count
1	1	SP	3351
2	1	RJ	990
3	1	MG	971
4	1	PR	443
5	1	RS	427
6	1	SC	345
7	1	BA	264
8	1	GO	164
9	1	ES	159
10	1	DF	151

Insight - maximum order is coming from state "SP" for the month of JAN and for the whole table SP is the state with maximum orders and RR with lowest order.

2. Distribution of customers across the states in Brazil

```
SELECT
  customer_state,
  COUNT(DISTINCT customer_unique_id) customer_count
FROM
  `Target_Data.customers`
GROUP BY
  1
ORDER BY
  2 DESC ;
```



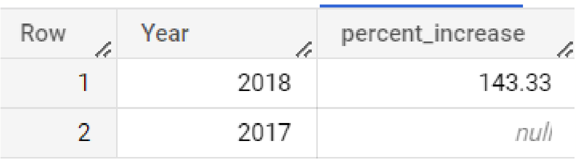
Row	customer_state	customer_count
1	SP	40302
2	RJ	12384
3	MG	11259
4	RS	5277
5	PR	4882
6	SC	3534
7	BA	3277
8	DF	2075
9	ES	1964
10	GO	1952

Insight - "SP" has the highest customer count and "RR" has lowest customer count .

Q4) Impact on Economy: Analyze the money movement by e-commerce by looking at order prices, freight and others.

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

```
WITH
  query1 AS (
  SELECT
    Year,
    Total_sale,
    LEAD(Total_sale) OVER(ORDER BY year DESC) next_sale
  FROM (
    SELECT
      EXTRACT(year
      FROM
        order_purchase_timestamp) Year,
      SUM(payment_value) Total_sale
    FROM
      `Target_Data.payments` p
    JOIN
      `Target_Data.orders` o
    ON
      p.order_id = o.order_id
    WHERE
      order_status = "delivered"
      AND EXTRACT(month
      FROM
        order_purchase_timestamp) BETWEEN 1
      AND 8
    GROUP BY
      1 ) tab1 )
SELECT
  Year,
  round(((Total_sale - next_sale)/next_sale ) * 100 , 2) percent_increase
FROM
  query1
ORDER BY
  1 DESC ;
```



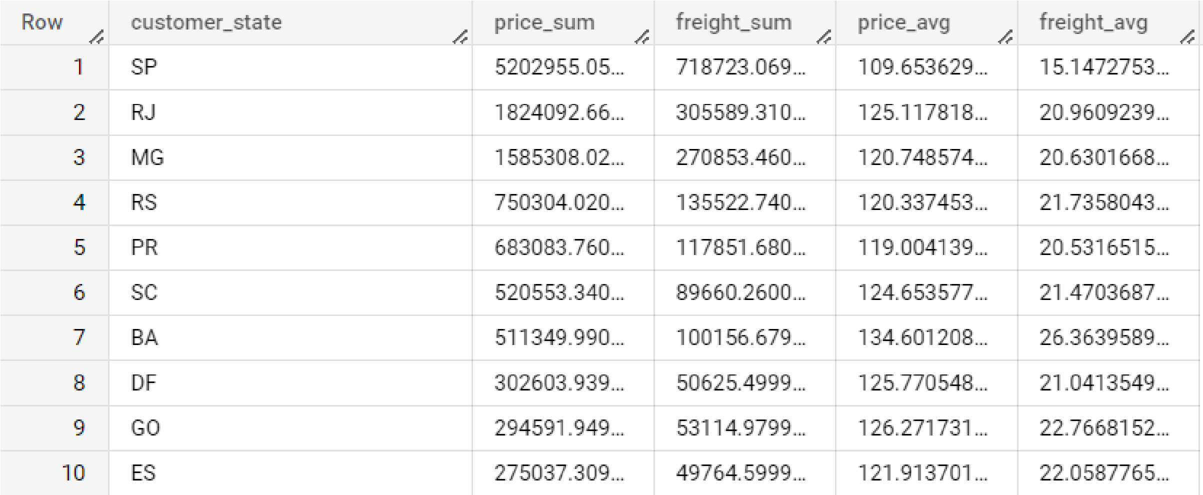
Row	Year	percent_increase
1	2018	143.33
2	2017	null

Insight - There is 143.33% increase in sales in 2018 from 2017 , so we can also say from this that there is a upward trend in the business.

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

```
SELECT
  customer_state,
  SUM(price) AS price_sum,
  SUM(freight_value) AS freight_sum,
  AVG(price) AS price_avg,
  AVG(freight_value) AS freight_avg
FROM
  `Target_Data.orders` o
JOIN
  `Target_Data.order_items` oi
```

```
ON
  o.order_id = oi.order_id
JOIN
  `Target_Data.customers` c
ON
  c.customer_id = o.customer_id
GROUP BY
  1
ORDER BY
  2 DESC,
  3 DESC
LIMIT
  10 ;
```



Row	customer_state	price_sum	freight_sum	price_avg	freight_avg
1	SP	5202955.05...	718723.069...	109.653629...	15.1472753...
2	RJ	1824092.66...	305589.310...	125.117818...	20.9609239...
3	MG	1585308.02...	270853.460...	120.748574...	20.6301668...
4	RS	750304.020...	135522.740...	120.337453...	21.7358043...
5	PR	683083.760...	117851.680...	119.004139...	20.5316515...
6	SC	520553.340...	89660.2600...	124.653577...	21.4703687...
7	BA	511349.990...	100156.679...	134.601208...	26.3639589...
8	DF	302603.939...	50625.4999...	125.770548...	21.0413549...
9	GO	294591.949...	53114.9799...	126.271731...	22.7668152...
10	ES	275037.309...	49764.5999...	121.913701...	22.0587765...

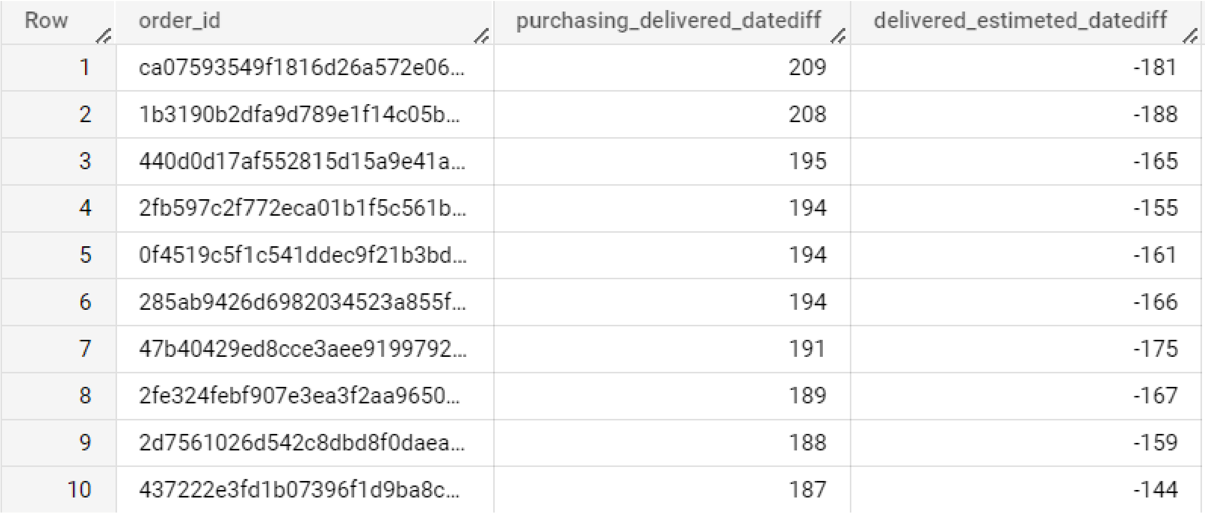
Q5) Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery

Assumption - taking delivering date as order delivered customer date and order status as "DELIVERED"

```
SELECT
  order_id,
  DATE_DIFF((order_delivered_customer_date),(order_purchase_timestamp), day) purchasing_del
ivered_datediff,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day) delivered_es
timeted_datediff
FROM
```

```
  `Target_Data.orders`
WHERE
  order_status = "delivered"
ORDER BY
  2 DESC,
  3 DESC ;
```



Row	order_id	purchasing_delivered_datediff	delivered_estimeted_datediff
1	ca07593549f1816d26a572e06...	209	-181
2	1b3190b2dfa9d789e1f14c05b...	208	-188
3	440d0d17af552815d15a9e41a...	195	-165
4	2fb597c2f772eca01b1f5c561b...	194	-155
5	0f4519c5f1c541ddec9f21b3bd...	194	-161
6	285ab9426d6982034523a855f...	194	-166
7	47b40429ed8cce3aee9199792...	191	-175
8	2fe324febf907e3ea3f2aa9650...	189	-167
9	2d7561026d542c8dbd8f0daea...	188	-159
10	437222e3fd1b07396f1d9ba8c...	187	-144

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

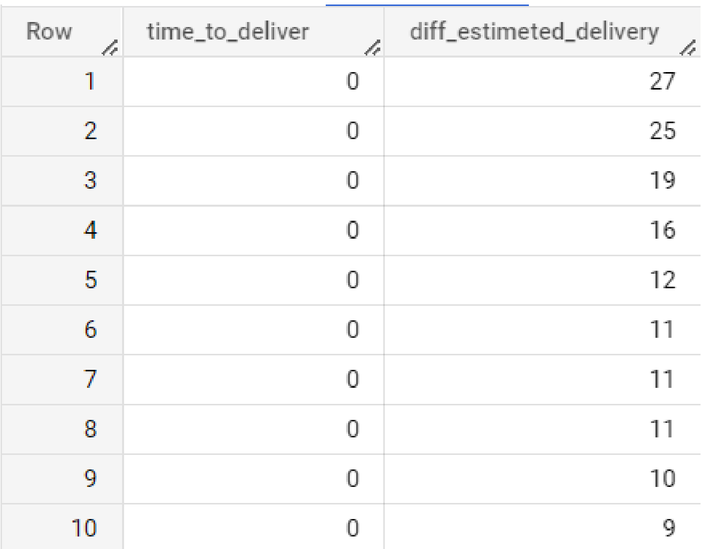
Assumption - taking order status "delivered"

```
 SELECT

  DATE_DIFF((order_purchase_timestamp),(order_delivered_customer_date), day) time_to_deliver,
  DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day) diff_estimeted_delivery
FROM
  `Target_Data.orders`
WHERE
  order_status = "delivered"
```

```
ORDER BY
  1 DESC,
  2 DESC

LIMIT
  10;
```



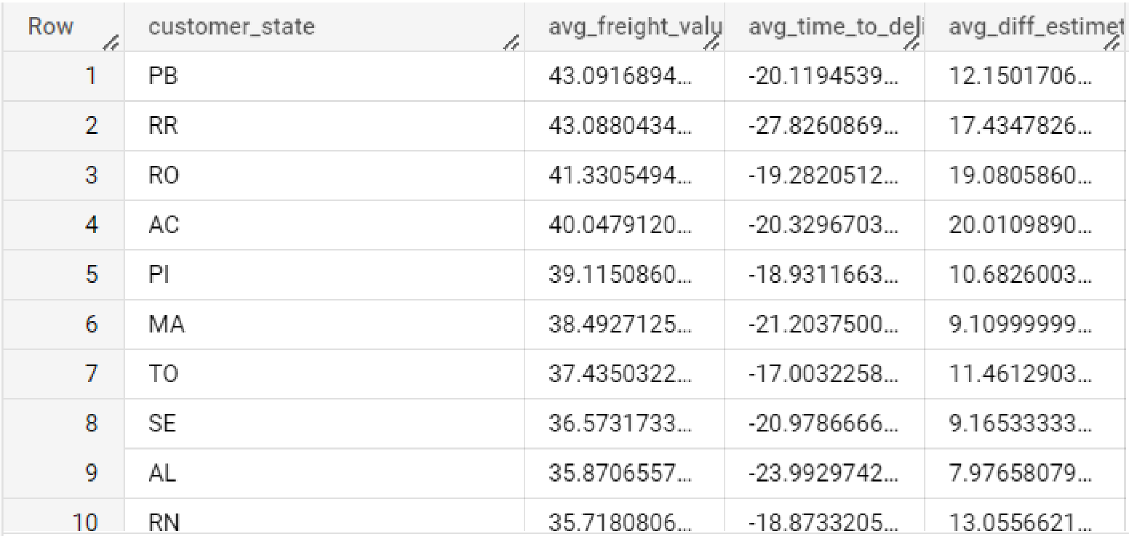
Row	time_to_deliver	diff_estimeted_delivery
1	0	27
2	0	25
3	0	19
4	0	16
5	0	12
6	0	11
7	0	11
8	0	11
9	0	10
10	0	9

```
 Insight - As it can be seen from above table only few orders got delivered on time and
there is a big gap between purchase and
           deliver date
 Recommendation - company needs to look into there time_to_delivery in order to improve
customer relation which will lead to increase
                  in overall sales although most of the orders has less delivery time as
compared to estimeted_delivery but still
                  there is a scope of minimizing the delivery time.
```

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

Assumption - taking order status as "delivered"

```
 SELECT
  customer_state,
  AVG(freight_value) avg_freight_value,
  AVG(DATE_DIFF((order_purchase_timestamp),(order_delivered_customer_date), day)) avg_time_
to_deliver,
  AVG(DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day)) avg_dif
f_estimeted_delivery
FROM
  `Target_Data.orders` o
JOIN
  `Target_Data.order_items` oi
ON
  oi.order_id = o.order_id
JOIN
  `Target_Data.customers` c
ON
  c.customer_id = o.customer_id
WHERE
  order_status = "delivered"
GROUP BY
  customer_state
ORDER BY
  2 DESC
  limit
  10 ;
```



Row	customer_state	avg_freight_valu	avg_time_to_deli	avg_diff_estimet
1	PB	43.0916894...	-20.1194539...	12.1501706...
2	RR	43.0880434...	-27.8260869...	17.4347826...
3	RO	41.3305494...	-19.2820512...	19.0805860...
4	AC	40.0479120...	-20.3296703...	20.0109890...
5	PI	39.1150860...	-18.9311663...	10.6826003...
6	MA	38.4927125...	-21.2037500...	9.10999999...
7	TO	37.4350322...	-17.0032258...	11.4612903...
8	SE	36.5731733...	-20.9786666...	9.16533333...
9	AL	35.8706557...	-23.9929742...	7.97658079...
10	RN	35.7180806...	-18.8733205...	13.0556621...

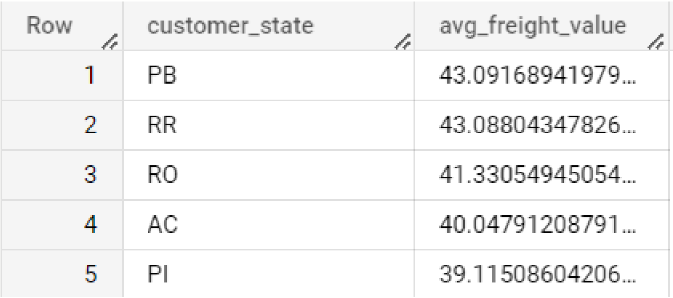
```
Insights - highest average freight value is in "PB", "RR" , "RO" , "AC" , "PI" and these
states also one of the highest
           avg_time_to_delivery.
```

4. Sort the data to get the following:

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

```
SELECT
  customer_state,
  AVG(freight_value) avg_freight_value

FROM
  `Target_Data.orders` o
JOIN
  `Target_Data.order_items` oi
ON
  oi.order_id = o.order_id
JOIN
  `Target_Data.customers` c
ON
  c.customer_id = o.customer_id
WHERE
  order_status = "delivered"
GROUP BY
  customer_state
ORDER BY
  2 DESC
  limit 5 ;
```

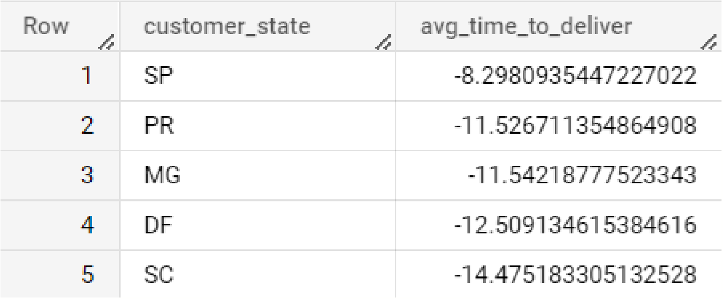


Row	customer_state	avg_freight_value
1	PB	43.09168941979...
2	RR	43.08804347826...
3	RO	41.33054945054...
4	AC	40.04791208791...
5	PI	39.11508604206...

6. Top 5 states with highest/lowest average time to delivery

```
SELECT
  customer_state,
  AVG(DATE_DIFF((order_purchase_timestamp),(order_delivered_customer_date), day)) avg_time_
to_deliver

FROM
  `Target_Data.orders` o
JOIN
  `Target_Data.customers` c
ON
  c.customer_id = o.customer_id
WHERE
  order_status = "delivered"
GROUP BY
  customer_state
ORDER BY
  2 DESC
  limit 5 ;
```



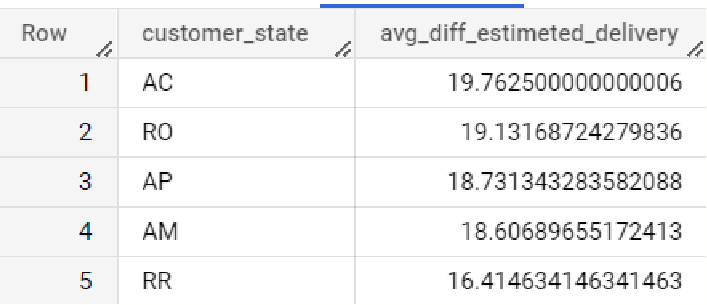
Row	customer_state	avg_time_to_deliver
1	SP	-8.2980935447227022
2	PR	-11.526711354864908
3	MG	-11.54218777523343
4	DF	-12.509134615384616
5	SC	-14.475183305132528

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

```
SELECT
  customer_state,
  AVG(DATE_DIFF(order_estimated_delivery_date, order_delivered_customer_date, day)) avg_dif
f_estimeted_delivery

FROM
  `Target_Data.orders` o
JOIN
  `Target_Data.customers` c
ON
  c.customer_id = o.customer_id
WHERE
  order_status = "delivered"
GROUP BY
  customer_state
ORDER BY
```

```
  2 DESC
  limit 5 ;
```



Row	customer_state	avg_diff_estimeted_delivery
1	AC	19.762500000000006
2	RO	19.13168724279836
3	AP	18.731343283582088
4	AM	18.60689655172413
5	RR	16.414634146341463

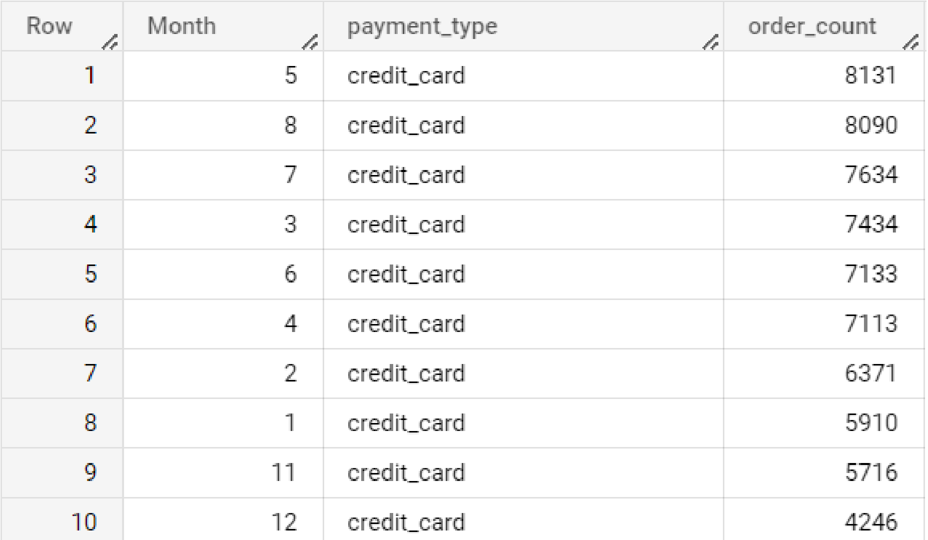
Insights - fastest delivery state as compared to estimated date is "AC" .

Q6) Payment type analysis:

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

Assumption - 1. order status is delivered .
2. and ignoring similar order id for different payment types

```
SELECT
  EXTRACT(month
  FROM
    order_purchase_timestamp) Month,
  payment_type,
  COUNT(o.order_id) order_count
FROM
  `Target_Data.orders` o
JOIN
  `Target_Data.payments` p
ON
  o.order_id = p.order_id
  where order_status = "delivered"
GROUP BY
  1, 2
ORDER BY
  3 DESC
LIMIT
  10 ;
```



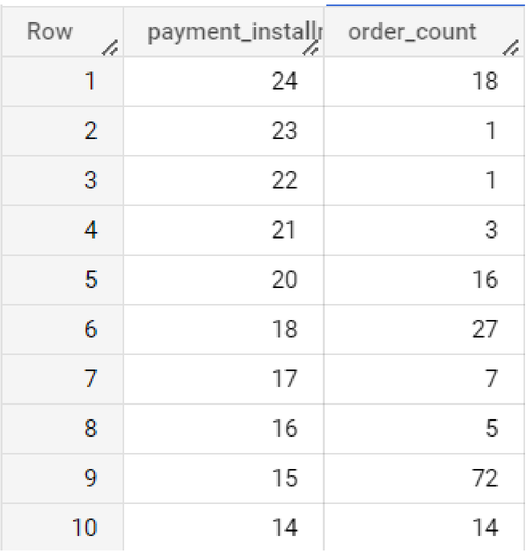
Row	Month	payment_type	order_count
1	5	credit_card	8131
2	8	credit_card	8090
3	7	credit_card	7634
4	3	credit_card	7434
5	6	credit_card	7133
6	4	credit_card	7113
7	2	credit_card	6371
8	1	credit_card	5910
9	11	credit_card	5716
10	12	credit_card	4246

Insight - maximum payments are done by credit card.

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

Assumption - order status is delivered.

```
SELECT
payment_installments ,
  COUNT(o.order_id) order_count
FROM
  `Target_Data.orders` o
JOIN
  `Target_Data.payments` p
ON
  o.order_id = p.order_id
  where order_status = "delivered"
GROUP BY
  1
ORDER BY
  1 DESC
LIMIT
  10 ;
```



Row	payment_install	order_count
1	24	18
2	23	1
3	22	1
4	21	3
5	20	16
6	18	27
7	17	7
8	16	5
9	15	72
10	14	14

Actionable Insights - 1. Although there is an increase in overall sales year on year but states like "SP" , "RJ" , "MG" are having maximum customers , maximum orders , lowest avg freight value and lowest avg delivery time and doing half of the revenue in Brazil.
2. States like "PB" , "RR" , "AC" are having lowest customers , lowest orders , highest avg freight value and highest avg delivery time.

Recommendation - 1. In order to improve sales in low performing states we need to decrease freight value and delivery time if possible, in order to acquire more customers. as we have seen in high performing states more number of customers lead to more sales in respective states.