TARGET Business Case Study

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:

- customers.csv
- sellers.csv
- order_items.csv
- geolocation.csv
- payments.csv
- reviews.csv
- orders.csv
- 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

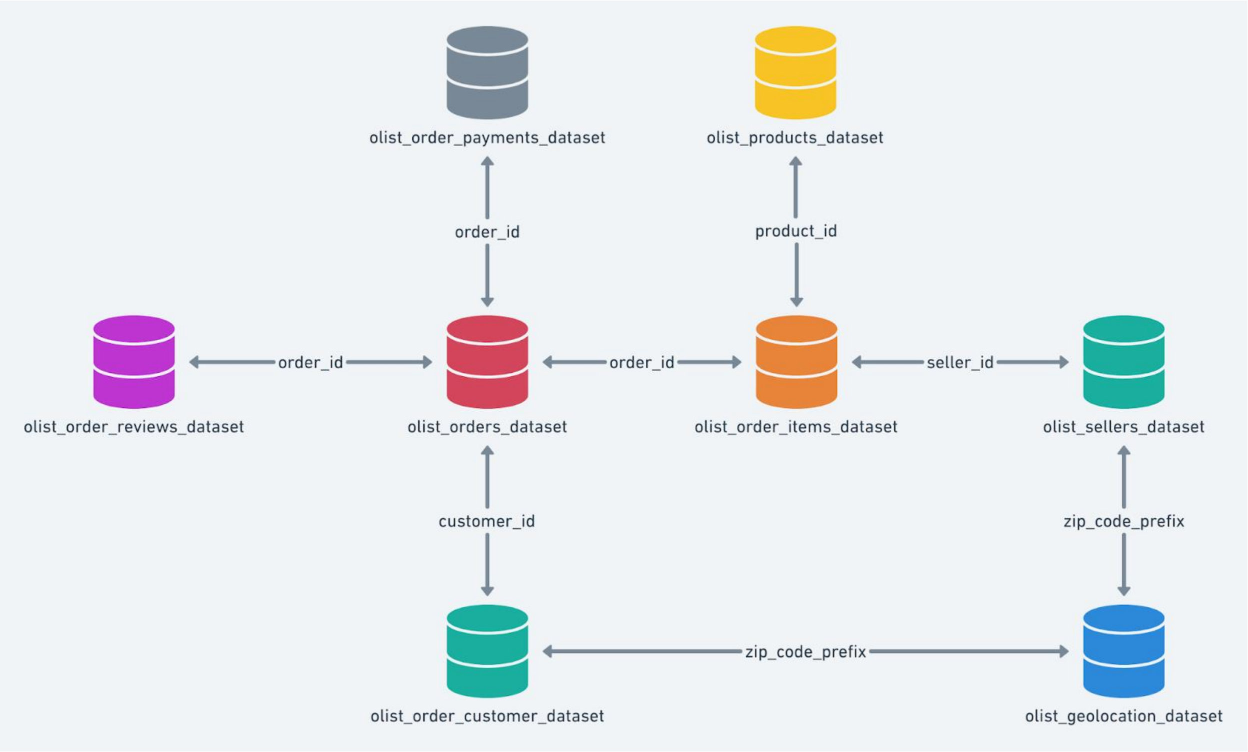
The **reviews.csv** contain following features:

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 ordered
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 portal
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

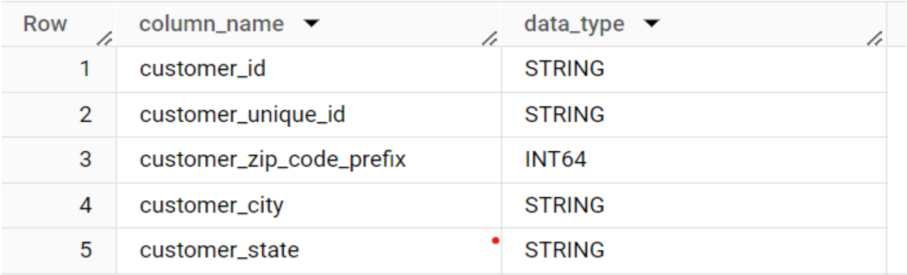
Dataset schema:



Problem Statement:

#Data type of all columns in the "customers" table

```
select column_name,data_type
from `target`.INFORMATION_SCHEMA.COLUMNS
where table_name= 'customers';
```

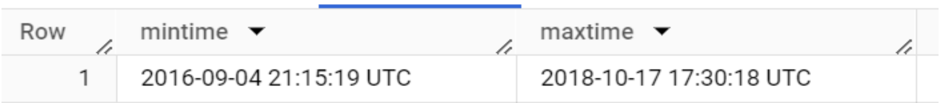


Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

ANALYSIS- All columns present in table are of string data type except customer_zip_code_prefix as integer.

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

```
select
min(order_purchase_timestamp) as mintime,
max(order_purchase_timestamp) as maxtime,
from `target.orders`;
```

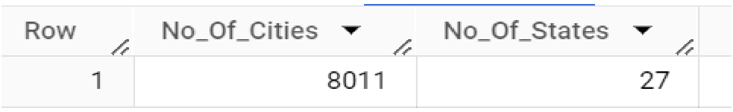


Row	mintime	maxtime
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

ANALYSIS- The given data has minimum time of 2016-09-04 21:15:19 UTC and maxtime is 2018-10-17 17:30:18

#Count the number of Cities and States in our dataset.

```
select
count(distinct geolocation_city) as No_Of_Cities,
count(distinct geolocation_state) as No_Of_States
from `target.geolocation`;
```



Row	No_Of_Cities	No_Of_States
1	8011	27

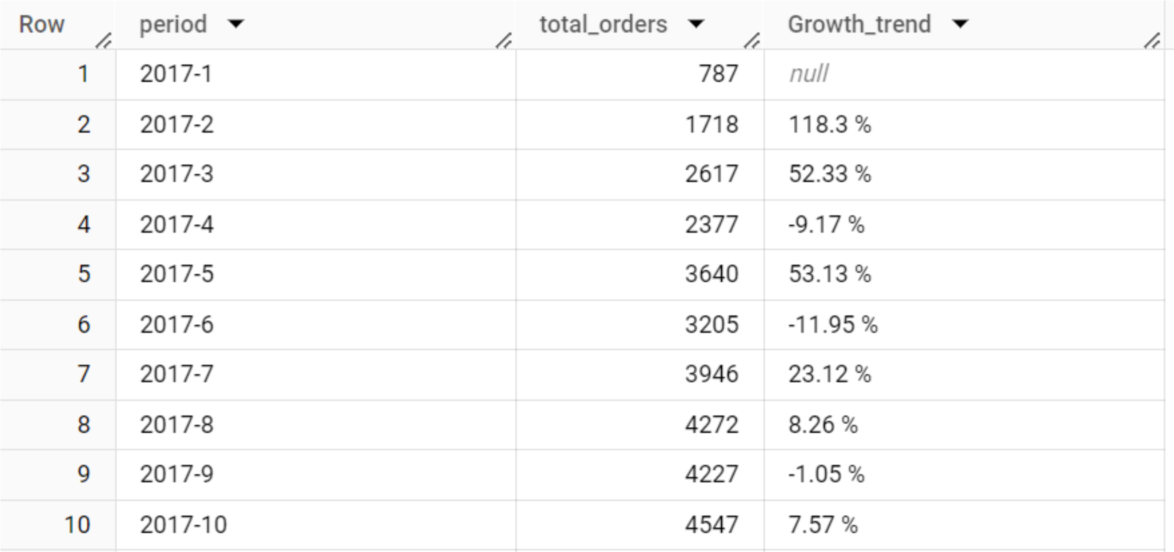
ANALYSIS- Total no of cities present are 8011 and states are 27

Target has established a significant presence across various regions in Brazil. This highlights their extensive reach and strong foothold in the Brazilian market

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

If no. of orders placed has increased gradually in each month, over the years bcuz 2016 incomplete , 2018 sept only 1 order...so go for orders monthwise....

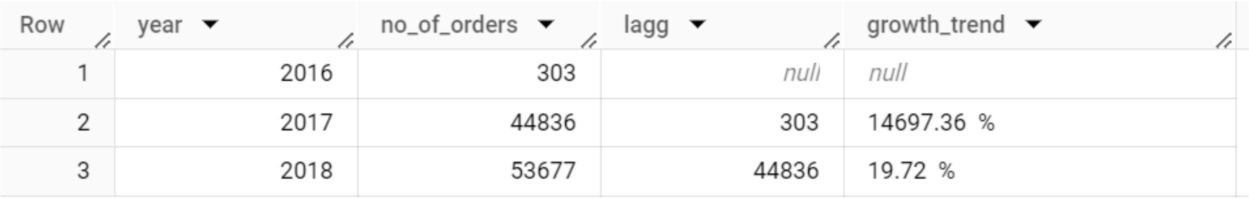
```
with cte as
(select
concat(year,'-',mnth)as period,
sum(no_of_orders)as total_orders,
lag(sum(no_of_orders)) over(order by year,mnth)as lagg
from(select
extract(year from order_purchase_timestamp)as year,
extract(month from order_purchase_timestamp)as mnth,
count(*)as no_of_orders
from `target.orders`
where order_status not in ('canceled','unavailable')
and extract(year from order_purchase_timestamp) >= 2017
group by order_purchase_timestamp,year,mnth
order by year,mnth)
group by year,mnth
order by year,mnth
select
period,
total_orders,
Growth_trend
from
(select * ,
concat(round(((total_orders-lagg)/lagg)*100,2),' %') as Growth_trend
from cte);
```



Row	period	total_orders	Growth_trend
1	2017-1	787	null
2	2017-2	1718	118.3 %
3	2017-3	2617	52.33 %
4	2017-4	2377	-9.17 %
5	2017-5	3640	53.13 %
6	2017-6	3205	-11.95 %
7	2017-7	3946	23.12 %
8	2017-8	4272	8.26 %
9	2017-9	4227	-1.05 %
10	2017-10	4547	7.57 %

# YEARWISE

```
with cte as (
(
  select extract(year from order_purchase_timestamp) as year,
  count(*) as no_of_orders
  from `target.orders`
  where order_status <>'canceled'
  group by 1
  order by 1
)
)

select *,
concat(round(((No_of_Orders-lagg)/lagg)*100,2),'  %') as growth_trend
from
(
  select *,
  lag(no_of_orders) over(order by year)as lagg
  from cte
  order by 1
);
```

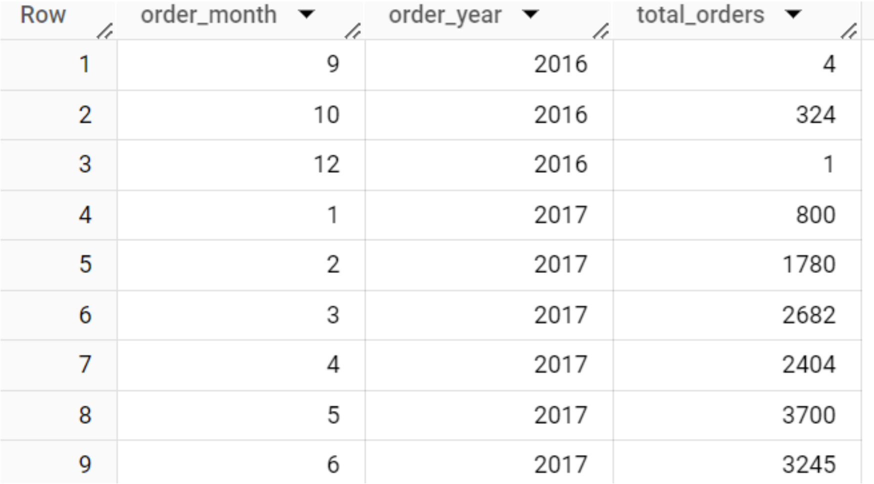


Row	year	no_of_orders	lagg	growth_trend
1	2016	303	null	null
2	2017	44836	303	14697.36 %
3	2018	53677	44836	19.72 %

ANALYSIS- By query analysis, the number of orders placed year-wise and month-wise excluding cancelled and unavailable orders, and considering the "Growth_trend" column indicating order growth rate compared to the previous year, it is evident that there was substantial order growth between 2016 and 2017, followed by a decrease in growth rate in 2018. This analysis helps identify patterns and trends in order placements over time.

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

```
WITH cte as(
  select *,
  extract(date from order_purchase_timestamp) as order_date,
  extract (year from order_purchase_timestamp) as order_year,
  extract(month from order_purchase_timestamp) as order_month
  from target.orders
)
select order_month, order_year,count(order_id) as total_orders
from cte
group by order_month, order_year
order by order_year, order_month
```



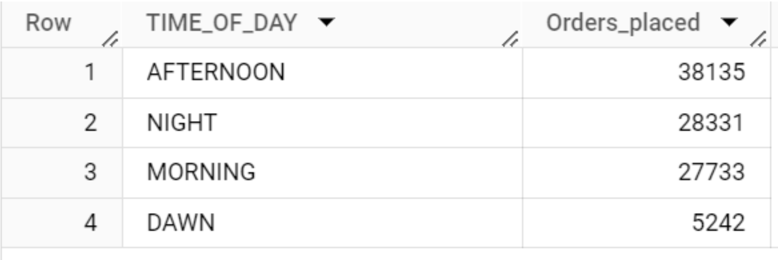
Row	order_month	order_year	total_orders
1	9	2016	4
2	10	2016	324
3	12	2016	1
4	1	2017	800
5	2	2017	1780
6	3	2017	2682
7	4	2017	2404
8	5	2017	3700
9	6	2017	3245

The objective of the query was to calculate the total count orders for each month over the years to get an understanding and managing the business's order patterns. There is a clear monthly seasonal pattern in the number of orders, with increases and decreases throughout the year, perhaps influenced by external factors like holidays or specific shopping periods. We can infer that Peak is at November 2017 shows the highest number of orders, while december 2016 has the lowest number of orders.

# 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

```
with cte as
(
select
case
when extract(hour from order_purchase_timestamp) between 0 and 6 then 'DAWN'
when extract(hour from order_purchase_timestamp) between 7 and 12 then 'MORNING'
when extract(hour from order_purchase_timestamp) between 13 and 18 then 'AFTERNOON'
else 'NIGHT'
end as TIME_OF_DAY,
count(*)as Orders_placed
from `target.orders`
group by TIME_OF_DAY
)

select * from cte
order by 2 desc;
```

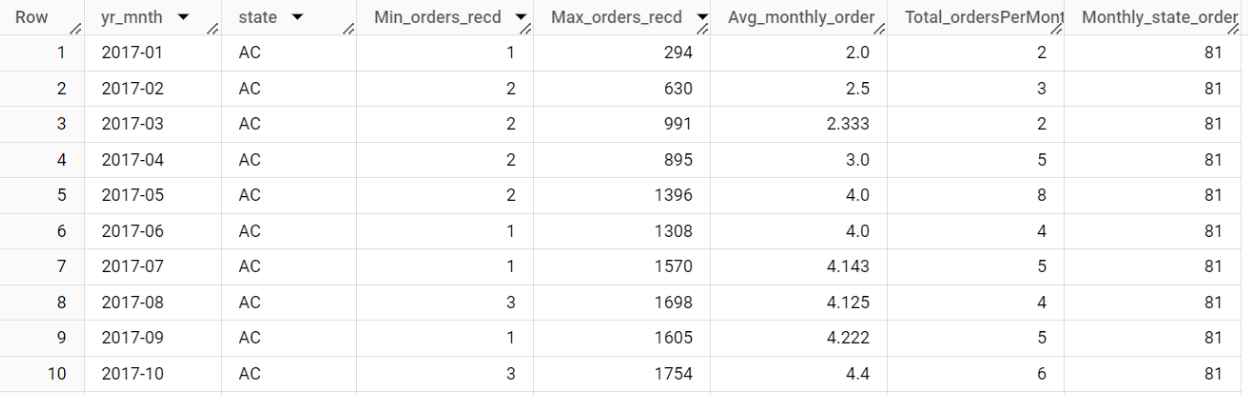


Row	TIME_OF_DAY	Orders_placed
1	AFTERNOON	38135
2	NIGHT	28331
3	MORNING	27733
4	DAWN	5242

ANALYSIS- I have queried to find the apt time as when the most of the orders are being placed. Based on the output, Brazilian customers tend to place the highest number of orders during the afternoon hours, followed by the night hours. Morning hours also show a substantial volume of orders, while dawn hours have fewer orders placed. This suggests that customers in Brazil are actively engaged in online shopping during the afternoon and evening, possibly during leisure time, with a significant number of orders placed in the morning as well.

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

```
select yr_mnth,state,
min(monthly_orders) over(partition by yr_mnth)as Min_orders_recd,
max(monthly_orders) over(partition by yr_mnth) as Max_orders_recd,
round(avg(monthly_orders)  over(partition  by  state  order  by  yr_mnth),3)as
Avg_monthly_order,
monthly_orders as Total_ordersPerMonth,
sum(monthly_orders) over(partition by state)as Monthly_state_order
from
(
select customer_state as state,
format_timestamp("%Y-%m",order_purchase_timestamp) as yr_mnth,
count(*)as monthly_orders
from `target.customers` c
join `target.orders` o
on c.customer_id=o.customer_id
where order_status not in ('canceled','unavailable')
group by customer_state,yr_mnth
order by customer_state,yr_mnth
)
group by monthly_orders,yr_mnth,State
order by state ;
```

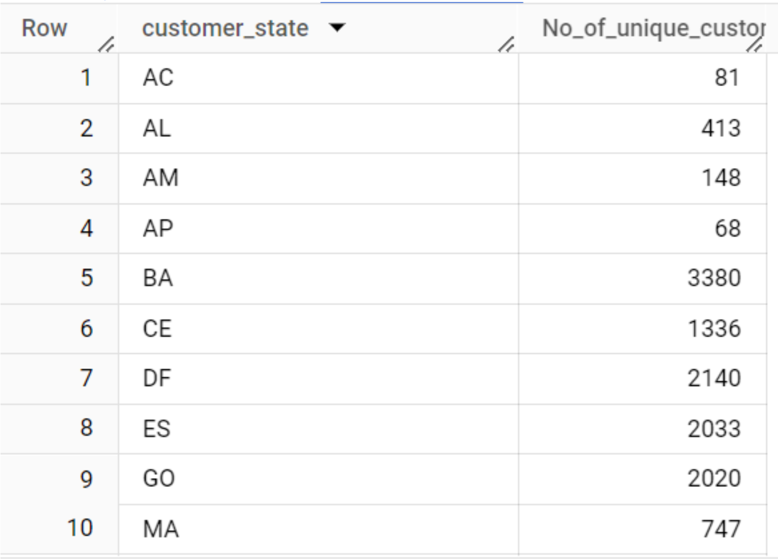


Row	yr_mnth	state	Min_orders_recd	Max_orders_recd	Avg_monthly_order	Total_ordersPerMont	Monthly_state_order
1	2017-01	AC	1	294	2.0	2	81
2	2017-02	AC	2	630	2.5	3	81
3	2017-03	AC	2	991	2.333	2	81
4	2017-04	AC	2	895	3.0	5	81
5	2017-05	AC	2	1396	4.0	8	81
6	2017-06	AC	1	1308	4.0	4	81
7	2017-07	AC	1	1570	4.143	5	81
8	2017-08	AC	3	1698	4.125	4	81
9	2017-09	AC	1	1605	4.222	5	81
10	2017-10	AC	3	1754	4.4	6	81

ANALYSIS- Statewise and month on month analysis help us to understand trend. This insights identifies growth opportunities.

How are the customers distributed across all the states?

```
select
customer_state,
count(customer_unique_id) AS No_of_unique_customers
from `target.customers`
group by customer_state
order by customer_state;
```



Row	customer_state	No_of_unique_custo
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747

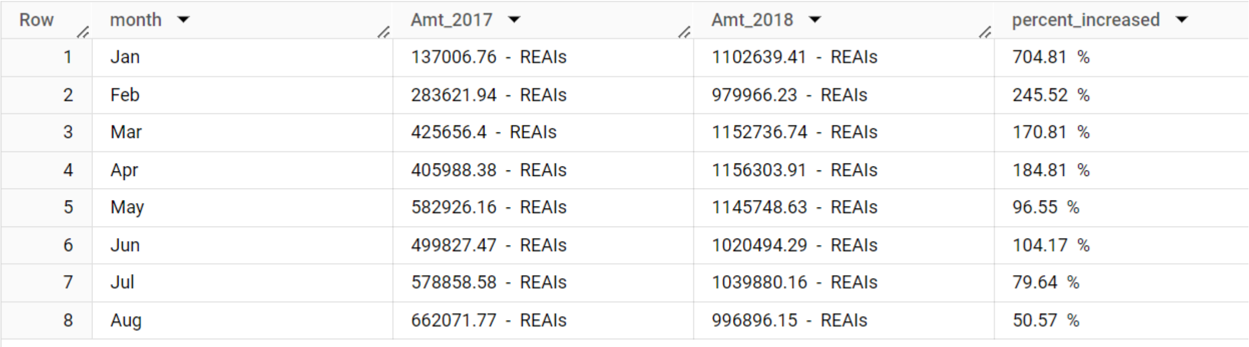
ANALYSIS - The query captures the total number of customers, number of customers whom made more than 1 purchase and total orders placed state wise helps us to do the Geographical Analysis. The state initialized as BA has most number of unique customers

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

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

Monthwise

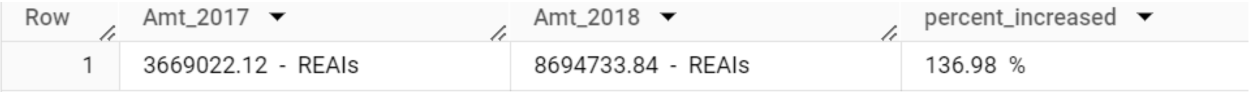
```
with cte as (select month,
round(sum(if(year=2018,amt,0)),2) as cost_2018,
round(sum(if(year=2017,amt,0)),2) as cost_2017
from(select
extract(year from order_purchase_timestamp)as year,
extract(month from order_purchase_timestamp)as mn,
format_timestamp("%b",order_purchase_timestamp) as month,
sum(p.payment_value)as amt
from `target.payments` p
join `target.orders` o
on p.order_id=o.order_id
where order_status not in ('canceled','unavailable')
group by mn,order_purchase_timestamp
order by mn
)as n
where n.mn < 9  and year <> 2016
group by mn,month
order by mn)
select month,
concat(cost_2017,'  -  REAIs')as Amt_2017,
concat(cost_2018,'  -  REAIs')as Amt_2018,
concat(round((((cost_2018-cost_2017)/cost_2017)*100),2),'  %') as percent_increased
from cte;
```



Row	month	Amt_2017	Amt_2018	percent_increased
1	Jan	137006.76 - REAIs	1102639.41 - REAIs	704.81 %
2	Feb	283621.94 - REAIs	979966.23 - REAIs	245.52 %
3	Mar	425656.4 - REAIs	1152736.74 - REAIs	170.81 %
4	Apr	405988.38 - REAIs	1156303.91 - REAIs	184.81 %
5	May	582926.16 - REAIs	1145748.63 - REAIs	96.55 %
6	Jun	499827.47 - REAIs	1020494.29 - REAIs	104.17 %
7	Jul	578858.58 - REAIs	1039880.16 - REAIs	79.64 %
8	Aug	662071.77 - REAIs	996896.15 - REAIs	50.57 %

Yearwise

```
with cte as (select
round(sum(if(year=2018,amt,0)),2) as cost_2018,
round(sum(if(year=2017,amt,0)),2) as cost_2017
from
(select
extract(year from order_purchase_timestamp)as year,
extract(month from order_purchase_timestamp)as mn,
sum(p.payment_value)as amt
from `target.payments` p
join `target.orders` o
on p.order_id=o.order_id
group by order_purchase_timestamp
order by year
)as n
where n.mn < 9  and year <> 2016)
select concat(cost_2017,'   -   REAIs')as Amt_2017,concat(cost_2018,'   -   REAIs')as
Amt_2018,
concat(round((((cost_2018-cost_2017)/cost_2017)*100),2),'  %') as percent_increased
from cte;
```



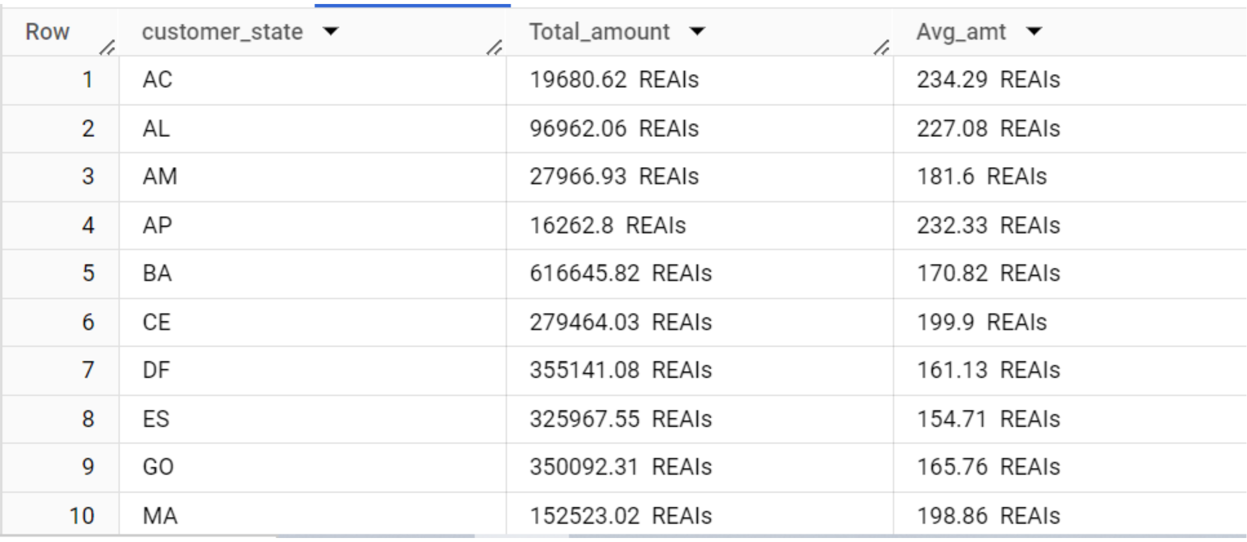
Row	Amt_2017	Amt_2018	percent_increased
1	3669022.12 - REAIs	8694733.84 - REAIs	136.98 %

ANALYSIS-

The query shows the 136.98% increase in the total cost year wise which signifies the increase in order costs of products suggests a pricing adjustment or changes in product offerings, influenced by factors like procurement, raw material costs, and demand-supply dynamics.

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

```
select customer_state ,
concat(round(sum(payment_value),2),'  REAIs') as Total_amount,
concat(round(avg(payment_value),2),'  REAIs') as Avg_amt
from `target.customers` c
join `target.orders` o
on c.customer_id=o.customer_id
join`target.payments` p
on p.order_id = o.order_id
group by customer_state
order by customer_state;
```

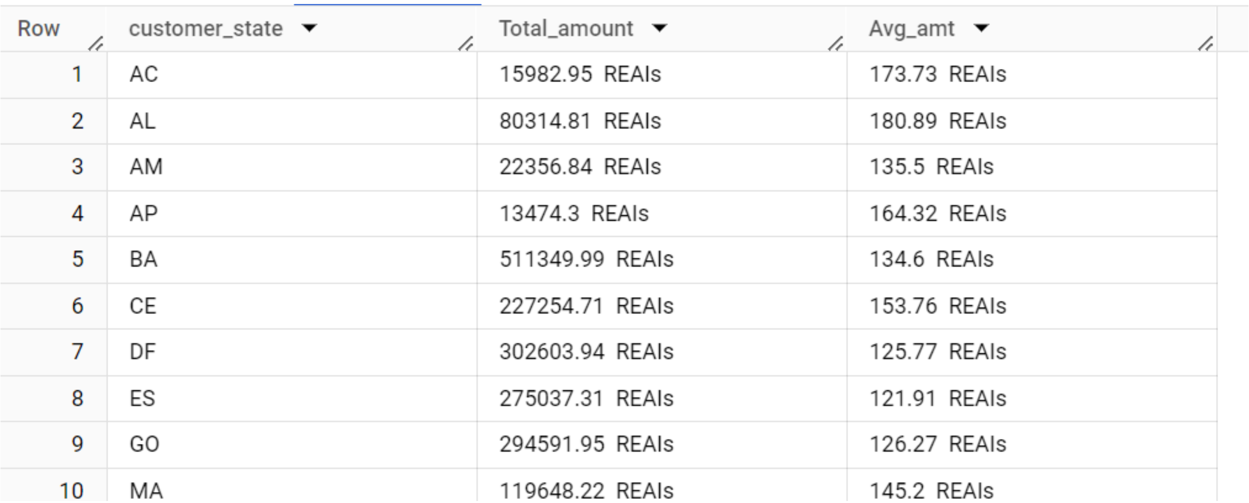


Row	customer_state	Total_amount	Avg_amt
1	AC	19680.62 REAIs	234.29 REAIs
2	AL	96962.06 REAIs	227.08 REAIs
3	AM	27966.93 REAIs	181.6 REAIs
4	AP	16262.8 REAIs	232.33 REAIs
5	BA	616645.82 REAIs	170.82 REAIs
6	CE	279464.03 REAIs	199.9 REAIs
7	DF	355141.08 REAIs	161.13 REAIs
8	ES	325967.55 REAIs	154.71 REAIs
9	GO	350092.31 REAIs	165.76 REAIs
10	MA	152523.02 REAIs	198.86 REAIs

Insights- This information enables targeted marketing campaigns and provides insights into regional customer behavior and preferences.

Using order price

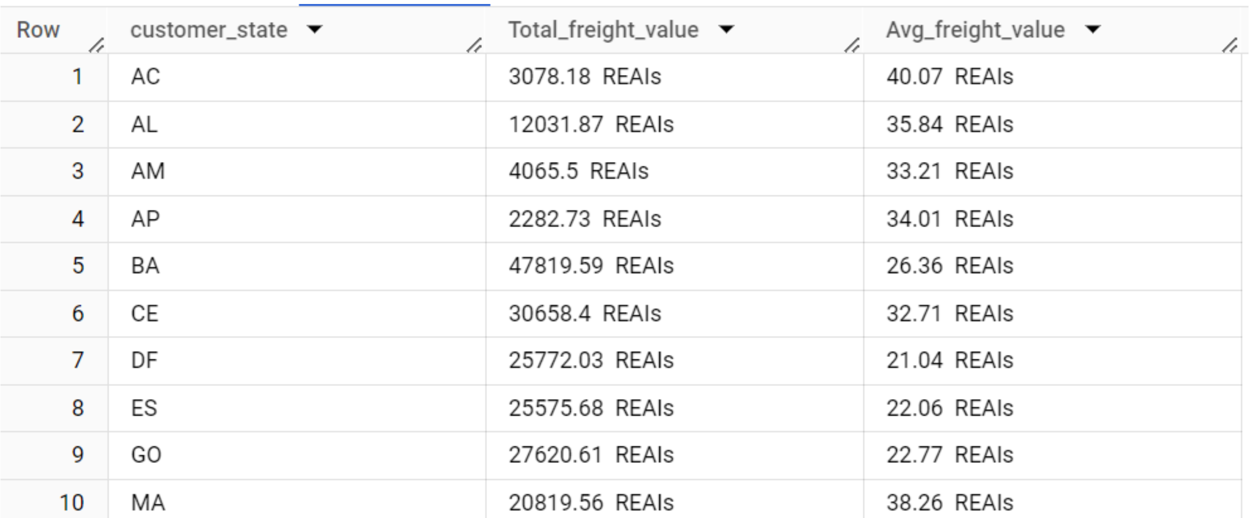
```
select customer_state ,
concat(round(sum(oi.price),2),'  REAIs') as Total_amount,
concat(round(avg(oi.price),2),'  REAIs') as Avg_amt
from `target.customers` c
join `target.orders` o
on c.customer_id=o.customer_id
join`target.order_items`oi
on oi.order_id = o.order_id
group by customer_state
order by customer_state;
```



Row	customer_state	Total_amount	Avg_amt
1	AC	15982.95 REAIs	173.73 REAIs
2	AL	80314.81 REAIs	180.89 REAIs
3	AM	22356.84 REAIs	135.5 REAIs
4	AP	13474.3 REAIs	164.32 REAIs
5	BA	511349.99 REAIs	134.6 REAIs
6	CE	227254.71 REAIs	153.76 REAIs
7	DF	302603.94 REAIs	125.77 REAIs
8	ES	275037.31 REAIs	121.91 REAIs
9	GO	294591.95 REAIs	126.27 REAIs
10	MA	119648.22 REAIs	145.2 REAIs

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

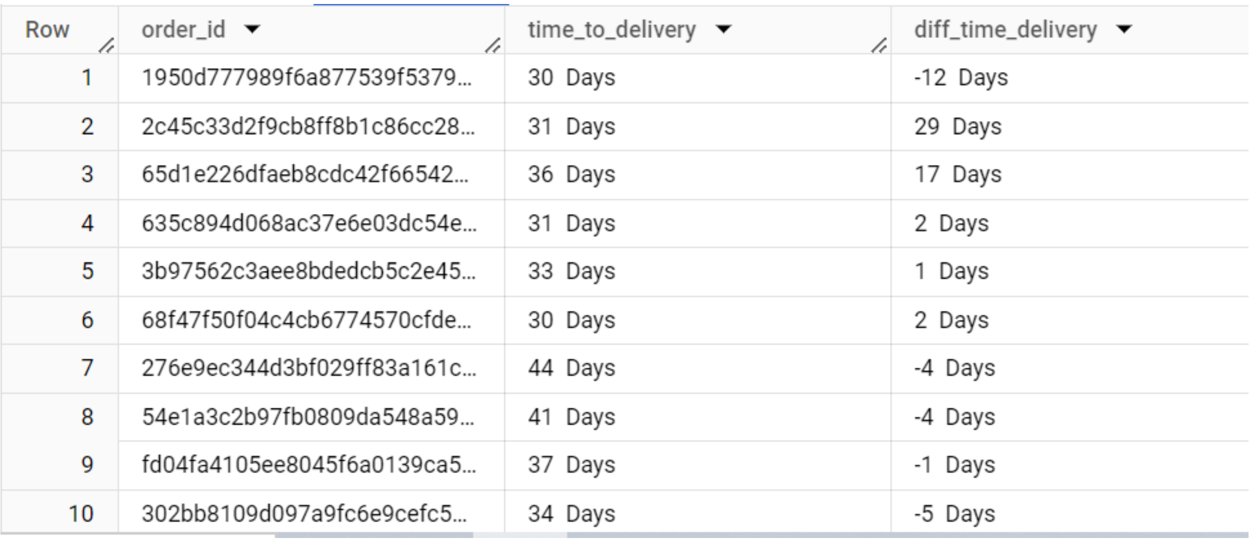
```
select customer_state ,
concat(round(sum(distinct freight_value),2),'  REAIs') as Total_freight_value,
concat(round(avg(freight_value),2),'  REAIs') as Avg_freight_value
from `target.customers` c
join `target.orders` o on c.customer_id=o.customer_id
join`target.order_items` oi on oi.order_id = o.order_id
group by customer_state
order by customer_state;
```



Row	customer_state	Total_freight_value	Avg_freight_value
1	AC	3078.18 REAIs	40.07 REAIs
2	AL	12031.87 REAIs	35.84 REAIs
3	AM	4065.5 REAIs	33.21 REAIs
4	AP	2282.73 REAIs	34.01 REAIs
5	BA	47819.59 REAIs	26.36 REAIs
6	CE	30658.4 REAIs	32.71 REAIs
7	DF	25772.03 REAIs	21.04 REAIs
8	ES	25575.68 REAIs	22.06 REAIs
9	GO	27620.61 REAIs	22.77 REAIs
10	MA	20819.56 REAIs	38.26 REAIs

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.

```
with cte as (
select order_id,
extract(date from order_purchase_timestamp)as order_purchase_date,
extract(date from order_delivered_customer_date)as delivered_customer_date,
extract(date from order_estimated_delivery_date)as estimated_delivery_date
from `target.orders`
)
select order_id,
concat(date_diff(cte.delivered_customer_date,cte.order_purchase_date,day),'  Days') as
time_to_delivery,
concat(date_diff(cte.estimated_delivery_date,cte.delivered_customer_date,day),'  Days')
as diff_time_delivery
from cte;
select
order_id,
datetime_diff(order_delivered_customer_date,order_purchase_timestamp,day) as
delivery_time,
datetime_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as
diff_estimated_delivery
from
`target.orders`;
```

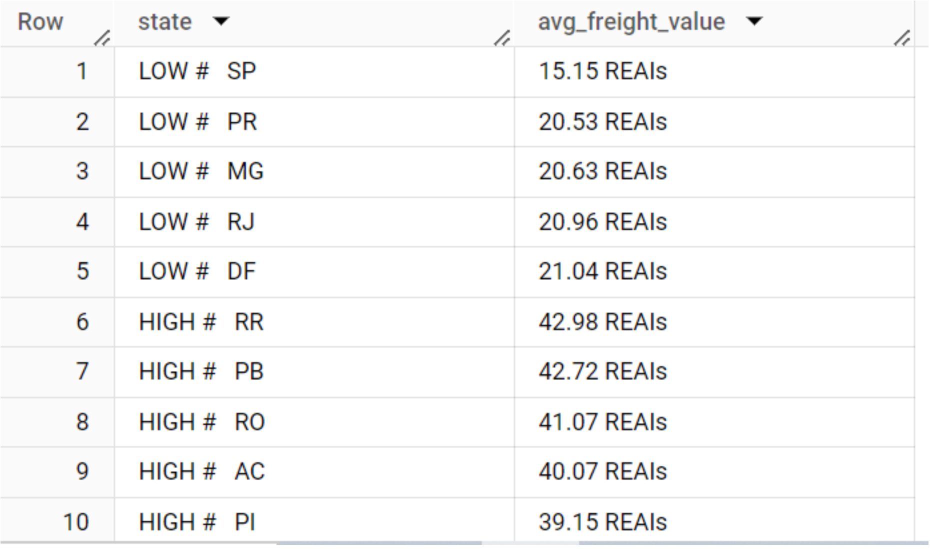


Row	order_id	time_to_delivery	diff_time_delivery
1	1950d777989f6a877539f5379...	30 Days	-12 Days
2	2c45c33d2f9cb8ff8b1c86cc28...	31 Days	29 Days
3	65d1e226dfaeb8cdc42f66542...	36 Days	17 Days
4	635c894d068ac37e6e03dc54e...	31 Days	2 Days
5	3b97562c3aee8bdedcb5c2e45...	33 Days	1 Days
6	68f47f50f04c4cb6774570cfde...	30 Days	2 Days
7	276e9ec344d3bf029ff83a161c...	44 Days	-4 Days
8	54e1a3c2b97fb0809da548a59...	41 Days	-4 Days
9	fd04fa4105ee8045f6a0139ca5...	37 Days	-1 Days
10	302bb8109d097a9fc6e9cefc5...	34 Days	-5 Days

From Query Analysis, The "Time_to_deliver" column represents the number of days taken to deliver an order to the customer from the purchase date, while the "diff_estimated_delivery" column indicates the difference between the estimated delivery date and the actual delivery date.

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

```
select state ,avg_freight_value from
((select concat('HIGH #   ',customer_state) as state,
max(freight_value) as High_freight_value,
concat(round(avg(freight_value),2),' REAIs') as avg_freight_value
from `target.customers` c
join `target.orders` o on c.customer_id=o.customer_id
join `target.order_items` as p on o.order_id = p.order_id
group by customer_state
order by 3 desc
limit 5 )
union all(select concat('LOW #   ',customer_state) as state,
min(freight_value) as low_freight_value,
concat(round(avg(freight_value),2),' REAIs') as avg_freight_value
from `target.customers` c
join `target.orders` o on c.customer_id=o.customer_id
join `target.order_items` as p on o.order_id = p.order_id
group by customer_state
order by 3 asc
limit 5))as t;
```

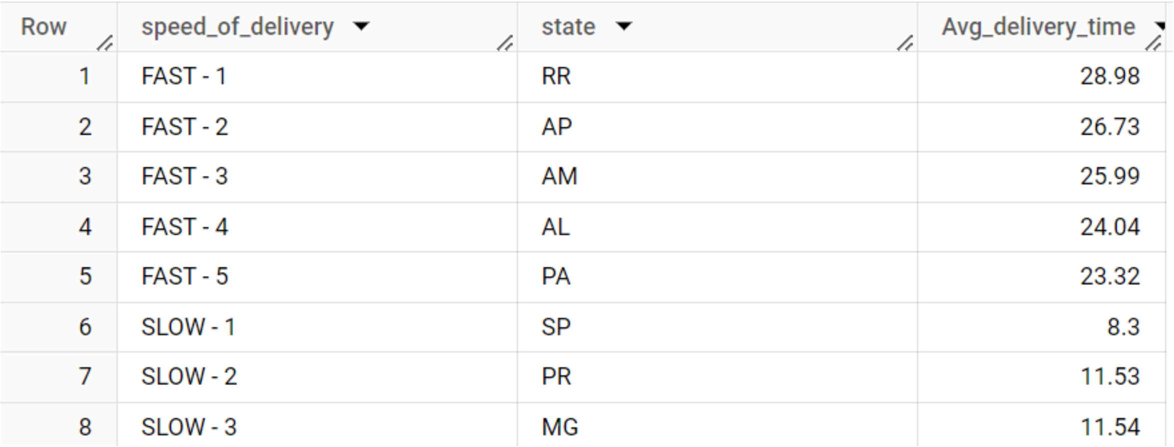


Row	state	avg_freight_value
1	LOW # SP	15.15 REAIs
2	LOW # PR	20.53 REAIs
3	LOW # MG	20.63 REAIs
4	LOW # RJ	20.96 REAIs
5	LOW # DF	21.04 REAIs
6	HIGH # RR	42.98 REAIs
7	HIGH # PB	42.72 REAIs
8	HIGH # RO	41.07 REAIs
9	HIGH # AC	40.07 REAIs
10	HIGH # PI	39.15 REAIs

Insights- query mention lowest and highest states in avg freight values.

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

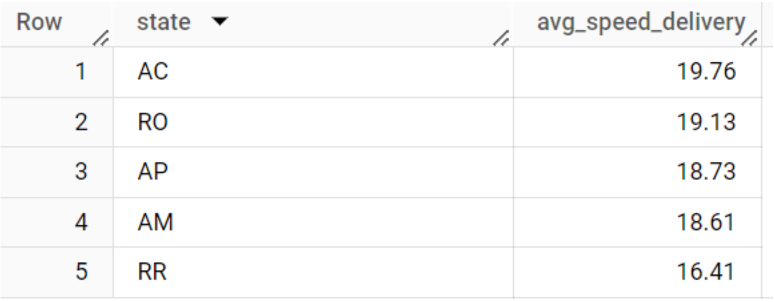
```
with cte as
(
select state,'FAST'as val,avg(delivery_time) as avg_delivery_time,
dense_rank() over (order by avg(delivery_time) desc) as rnk
from
(
select customer_state as state,
datetime_diff(order_delivered_customer_date,order_purchase_timestamp,day)           as
delivery_time,
from `target.customers` as c
join `target.orders` as o on c.customer_id = o.customer_id
group by state,order_delivered_customer_date,order_purchase_timestamp,delivery_time
) nt1
group by state

union all

select state,'SLOW'as val,avg(delivery_time) as avg_delivery_time,
dense_rank() over (order by avg(delivery_time) asc) as rnk
from
(
select customer_state as state,
datetime_diff(order_delivered_customer_date,order_purchase_timestamp,day)          as
delivery_time,
from `target.customers` as c
join `target.orders` as o on c.customer_id = o.customer_id
group by state,order_delivered_customer_date,order_purchase_timestamp,delivery_time
) nt2
group by state
)

select  concat(val," - ",rnk)  as  speed_of_delivery,state,round(avg_delivery_time,2)as
Avg_delivery_time
from cte
where rnk<=5
order by 1;
```



Row	speed_of_delivery	state	Avg_delivery_time
1	FAST - 1	RR	28.98
2	FAST - 2	AP	26.73
3	FAST - 3	AM	25.99
4	FAST - 4	AL	24.04
5	FAST - 5	PA	23.32
6	SLOW - 1	SP	8.3
7	SLOW - 2	PR	11.53
8	SLOW - 3	MG	11.54

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.

```
select customer_state as state,
round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,da
y)),2)as avg_speed_delivery
from `target.customers` as c
join `target.orders` as o on c.customer_id = o.customer_id
group by state
order by avg_speed_delivery desc
limit 5;
```

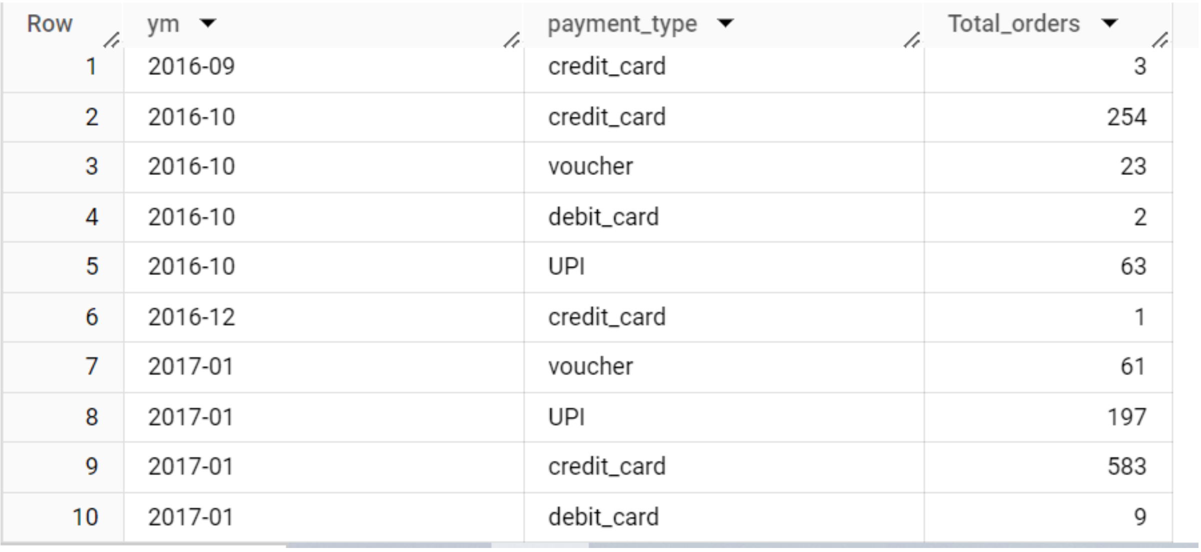


Row	state	avg_speed_delivery
1	AC	19.76
2	RO	19.13
3	AP	18.73
4	AM	18.61
5	RR	16.41

Analysis based on the payments:

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

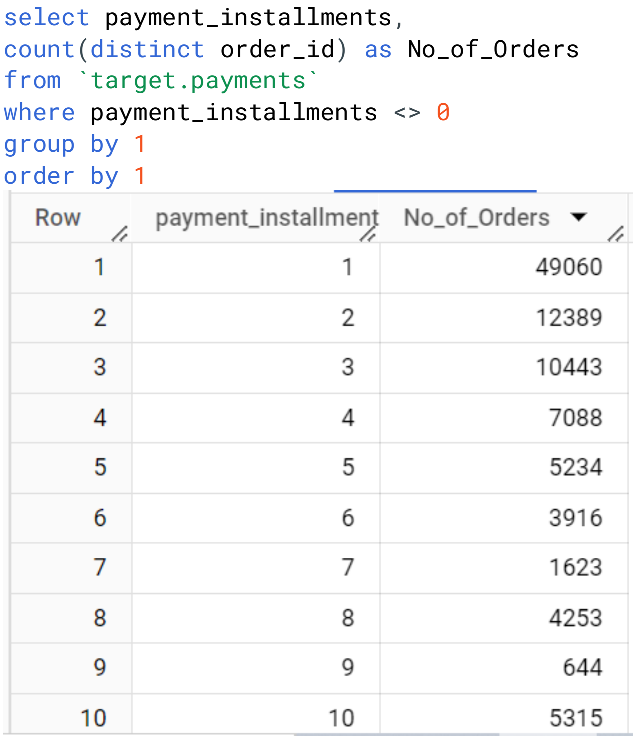
```
select ym,
payment_type,
sum(cnt)as Total_orders
from
(
select
format_timestamp("%Y-%m",order_purchase_timestamp) as ym,
payment_type,
count(p.order_id)as cnt
from `target.payments` p
join `target.orders` o
on p.order_id=o.order_id
group by order_purchase_timestamp,payment_type
)
where payment_type is not null
group by payment_type,ym
order by ym;
```



Row	ym	payment_type	Total_orders
1	2016-09	credit_card	3
2	2016-10	credit_card	254
3	2016-10	voucher	23
4	2016-10	debit_card	2
5	2016-10	UPI	63
6	2016-12	credit_card	1
7	2017-01	voucher	61
8	2017-01	UPI	197
9	2017-01	credit_card	583
10	2017-01	debit_card	9

The query provides insights into customer payment preferences by analyzing the monthly distribution of payment types. By analyzing the monthly distribution of payment types, businesses can gain insights into customer payment preferences, identify trends in payment methods, This information can be valuable for optimizing the checkout experience, expanding payment options and improving overall customer satisfaction

Find the no. of orders placed on the basis of the payment installments that have been paid.(0 zero installment means down payment)



```
select payment_installments,
count(distinct order_id) as No_of_Orders
from `target.payments`
where payment_installments <> 0
group by 1
order by 1
```

Row	payment_installment	No_of_Orders
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	5	5234
6	6	3916
7	7	1623
8	8	4253
9	9	644
10	10	5315

Query Results shows the count of distinct orders for each payment instalment option. It indicates the distribution of customers who choose different instalment plans for their payments. Mostly preferred instalment is 1 or we can say that atleast 1 of many has been paid. The data indicates a notable customer preference for a particular focus on the 9 -10 instalment range.