I. 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.

```
SELECT

   column_name,
   data_type
FROM scaler-dsml-
sql-1-399511.Biz_case_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

Insights:
> Customers table has 2 different data types String and int64 .

> INT64 can store very large or very small whole numbers so here we used it to store out customer_zip_code_prefix in the above given table .

> STRING can store numeric, alphabetic and special characters in it .

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

```
SELECT
    MIN(order_purchase_timestamp)  start_time,
    MAX(order_purchase_timestamp)  end_time
FROM Biz_case_target.orders;
```

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

Insights:
>All the orders were placed in the span of 2 years that is from 2016-2018 and the starting month is September 2016 and we have data upto October 2018.

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

```
SELECT
    COUNT(DISTINCT customer_city) no_of_cities,
    COUNT(DISTINCT customer_state) no_of_states
FROM Biz_case_target.customers a
JOIN Biz_case_target.orders b
ON a.customer_id = b.customer_id
```

Row	no_of_cities	no_of_states
1	4119	27

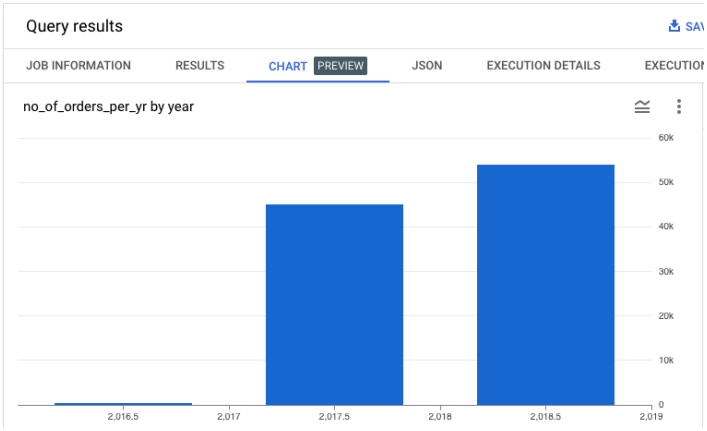
Insights:
> There are total 4119 cities and 27 states .
> So we have large population of customers who may or may not belong to same state and city.

II. In-depth Exploration:

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

```
SELECT year,orders_per_yr
FROM(
      SELECT *,
         EXTRACT(YEAR FROM order_purchase_timestamp) year,
         COUNT(order_id) OVER(PARTITION BY EXTRACT(YEAR FROM
order_purchase_timestamp)) orders_per_yr
      FROM Biz_case_target.orders )a
GROUP BY orders_per_yr,year
ORDER BY year
```

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



Insights:

>The number of orders placed in the year 2016 is around 329.

>When it comes to 2017 there is a marked rise in the number off orders and that's a promising result of good marketing strategies .

>Similarly if we observe in the year 2018 number of orders placed looks consistent With 2017 but there is still a lot of scope to enhance the market.

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

```
SELECT * FROM
(
   SELECT
   EXTRACT(month FROM order_purchase_timestamp) months,
   COUNT(order_id)  no_of_orders_placed_monthly,
   FROM Biz_case_target.orders
   GROUP BY EXTRACT(month FROM order_purchase_timestamp) )a
ORDER BY 2 DESC
```

Row	months	no_of_orders_placed
1	8	10843
2	5	10573
3	7	10318
4	3	9893
5	6	9412
6	4	9343
7	2	8508
8	1	8069
9	11	7544
10	12	5674
11	10	4959
12	9	4305

Insights :

> In the year 2016 there is no consistent growth in the number of order placed which is unfavourable.

>But when we observe 2017 in the month of Jan only single order was placed but from feb to June there was clear efforts in increasing the number of orders ranging from 800 to almost 3700¸ and than from July to October there is consistency in the number of orders being placed in the range of 4026 - 4638.

>November month has the highest number of orders placed that is 7544 followed by December which saw a second highest number of orders 5673 ¸ which indicates holiday season has highest number of orders being placed .

> On the whole 2017 was great in terms of performance when compared to 2016.

>When we observe 2018 right from the start till august there was consistent number of orders being placed in the range of 7269-6512 which is positive while number of orders in September and October is extremely low that is around 16 and 4 that shows a downfall.

> On the whole there is great improvement in numbers but at the end of 2018 graph is extremely low indicating need to ***implement effective market strategies*** *and availing some* ***offers*** *to attract customers* ***and managing inventory with the stock that is on high demand.***

C.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

```
SELECT

  n.timing_of_the_day,
  COUNT(n.order_id) AS no_of_orders
FROM
(

SELECT

      order_id,
      order_purchase_timestamp,
      EXTRACT(time FROM order_purchase_timestamp) AS
timeinfo,
      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 "Mornings"
          WHEN EXTRACT(hour FROM order_purchase_timestamp)
BETWEEN
13 AND 18 THEN "Afternoon"
          WHEN EXTRACT(hour from order_purchase_timestamp)
BETWEEN
19 AND 23 THEN "Night"
      END AS timing_of_the_day
FROM Biz_case_target.orders )AS n

GROUP BY timing_of_the_day
ORDER BY no_of_orders DESC
```

Row	timing_of_the_day	no_of_orders
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

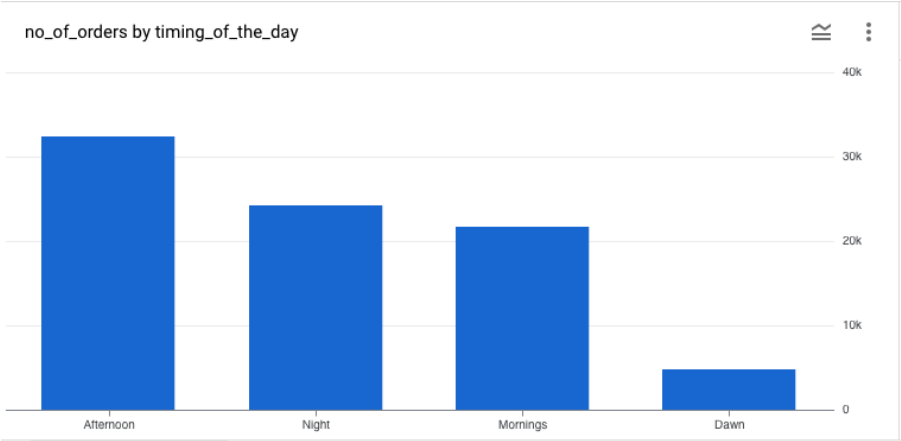
Insights :
>So here we segregated the complete 3 years data and tried to categorise customer activity into 4 categories

>We categorised it based on number of orders placed in a particular duration of the day
> Most of the customer activity around 32370 people placed orders during ***Afternoon*** *so we can call it the* ***peak***

time

>Nights and Mornings having consistent customer activity but when compared to day it's less so to enhance this we can avail some ***flash sales or limited time promotions*** *to encourage customers to buy product on a certain* ***discount price for a limited times slot***

>While dawn shows very less customer activity here we can implement ***Early bird*** *offers with* ***exclusive discounts***



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

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

```
SELECT
    a.customer_state,
    a.years,a.months,
    no_of_orders
FROM
(SELECT customer_state,
       EXTRACT(year FROM order_purchase_timestamp) years,
       EXTRACT(month FROM order_purchase_timestamp) months,
       COUNT(order_id) no_of_orders
FROM Biz_case_target.orders o
JOIN Biz_case_target.customers c
ON o.customer_id = c.customer_id
GROUP BY EXTRACT(year FROM order_purchase_timestamp),EXTRACT(month
FROM
order_purchase_timestamp),customer_state )a
ORDER BY a.customer_state,a.years,a.months
```

Row	customer_state	years	months	no_of_orders
1	AC	2017	1	2
2	AC	2017	2	3
3	AC	2017	3	2
4	AC	2017	4	5
5	AC	2017	5	8
6	AC	2017	6	4
7	AC	2017	7	5
8	AC	2017	8	4
9	AC	2017	9	5
10	AC	2017	10	6

Insights:

> So in this extracted data you'll find the number of orders placed in each state month-wise In a span of 2016 to 2018

> From the data In the year 2016 AC Brazilian state has NO orders placed at all ¸overall orders placed were also less but among the states SP stands out with 115 orders

>Simalarly SP stands as an outlier for both 2017 and 2018 as with highest number of orders being placed and that too being consistent in every month in placing orders

> In the first quarter of year 2017 and 2018 the number of orders placed were comparatively less than the rest exceptions is noticed in outlier state like SP

>The number of orders placed in the month of November and December across all 2 the years that is 2016 and 2017 is higher than any other months in those respective years but SP stands as an exception because there is no any evident data related to this.

*>We can avail some or **festival offers** in the month of November and December to gain customers count in the year 2018 to **increase the market further***

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

```
SELECT
   COUNT(customer_id) no_of_customers_statewise,
   customer_state
FROM Biz_case_target.customers
GROUP BY customer_state
ORDER BY no_of_customers_statewise
```

Row	no_of_customers_statewise	customer_state
1	46	RR
2	68	AP
3	81	AC
4	148	AM
5	253	RO

Insights:

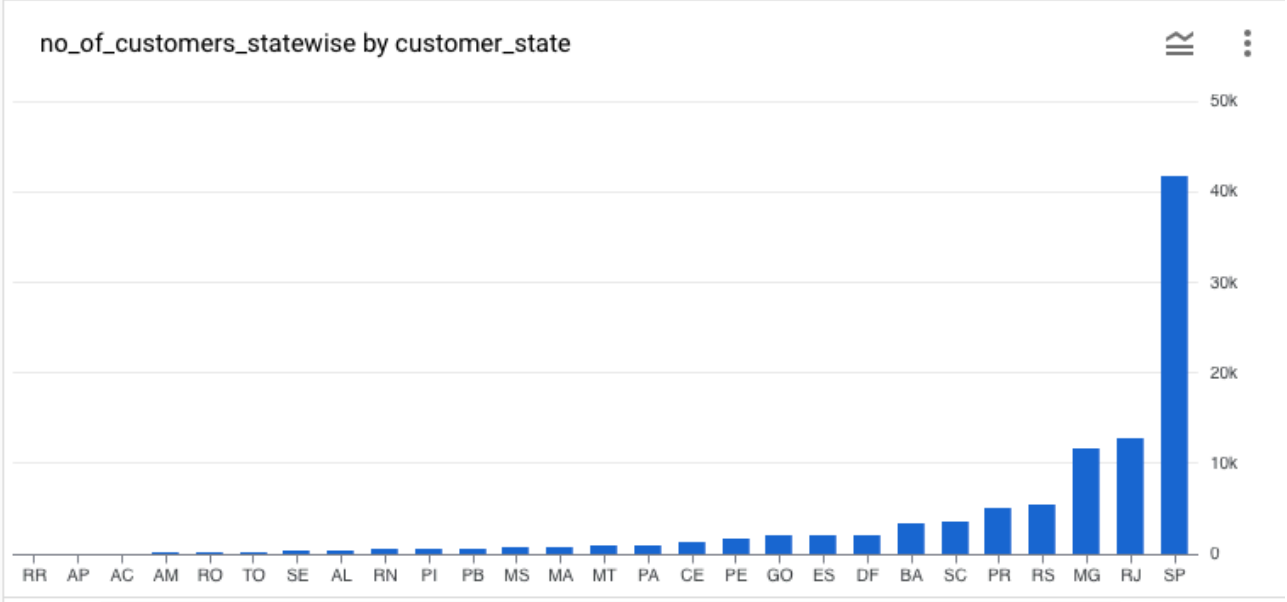
> This data gives us an idea on customers distribution across all the different states.

>To sum it up
3 different states have 1-100 customers,
12 different states have 100-1000 customers , 8 different states have 1000-5500 customers above 5500 we have 3 different states .

Which says the distribution is haphazard .
>It indicates marketing is uneven, need to enhance promotional activities.

>So in states where customer count is high we can take customer reviews and use it for further promotional activities in states where customer count is less .

> Consider states where customer distribution is low provide the supply based on the state needs. >Similarly tailor the requirements of customers based on the age group as well .



IV. Impact on Economy: Analyse 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.

```
SELECT DISTINCT *,
    LAG(total_payment_of_year,1) OVER(ORDER BY years)
previous_years_total ,
    ROUND((total_payment_of_year - LAG(total_payment_of_year,1)
OVER(ORDER BY
years))/LAG(total_payment_of_year,1) OVER(ORDER BY years) *
    100, 2) percentage_increase
FROM
(
SELECT DISTINCT
  years,
  SUM(payment_value) OVER(PARTITION BY years)
total_payment_of_year
  FROM
(
SELECT
     EXTRACT(year FROM order_purchase_timestamp) years,
     EXTRACT(month FROM order_purchase_timestamp) months,
     payment_value
FROM Biz_case_target.payments p
JOIN Biz_case_target.orders o
ON p.order_id = o.order_id
WHERE EXTRACT(year FROM order_purchase_timestamp) BETWEEN 2017 AND
2018 AND EXTRACT(month FROM order_purchase_timestamp) <=8 )p
)q
```

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTI

Row	years	total_payment_of_year	previous_years_total	percentage_increase
1	2017	3669022.12	*null*	*null*
2	2018	8694733.84	3669022.12	136.98

Insights:

> This data gives us an idea about percentage increase in the cost of orders from 2017 to 2018 considering months between Jan to Aug only from both the years

>So total payment of 2017 is 3669022.12 and 2018 is 8694733.84
>Percentage increase is 136.98% it can ***act like a goal setting for further years going ahead*** *> This indicates a* ***substantial growth***

>Considering the ***contributing factors*** *like* ***customer activity, states involved, requirement of goods etc and focusing mainly*** *on areas where* ***growth potential is high can enhance business in future***

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

```
with cte as
(SELECT
   c.customer_state,
   ROUND(SUM(price),2) total_price,
   COUNT(DISTINCT o.order_id) as no_of_orders
FROM Biz_case_target.orders o
JOIN Biz_case_target.order_items i
ON o.order_id = i.order_id
JOIN Biz_case_target.customers c
ON o.customer_id = c.customer_id
GROUP BY customer_state)
SELECT

customer_state,
total_price,
ROUND(total_price/no_of_orders,2) as avg_tot_price
from cte
ORDER BY 2 desc
```

Row	customer_state	total_price	avg_tot_price
1	SP	5202955.05	125.75
2	RJ	1824092.67	142.93
3	MG	1585308.03	137.33
4	RS	750304.02	138.13
5	PR	683083.76	136.67
6	SC	520553.34	144.12
7	BA	511349.99	152.28
8	DF	302603.94	142.4
9	GO	294591.95	146.78
10	ES	275037.31	135.82

Insights:
>so here lets observe state SP which has high total price **5202955** and highest customer count **47449** but average seems low that's around **125.75**

>In this case these customers fall under high volume customers here you can try ***upselling or cross selling.***

>This is an instance to understand the ***high value market*** *in this case you should* ***retain the market*** *and* ***upsell the products to the existing customers***

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

```
with cte as
(SELECT
       customer_state,
       ROUND(SUM(i.freight_value),2)  total_freight_value,
       COUNT(DISTINCT o.order_id) no_of_orders
   FROM Biz_case_target.orders o
   JOIN Biz_case_target.customers c
   ON c.customer_id = o.customer_id
   JOIN Biz_case_target.order_items i
   ON i.order_id = o.order_id
GROUP BY customer_state
ORDER BY customer_state)
SELECT

    customer_state,
    total_freight_value,
    ROUND(total_freight_value/no_of_orders,2) as
avg_freight_val
FROM cte
ORDER BY 2,3 DESC
```

Row	customer_state	total_freight_value	avg_freight_val
1	RR	2235.19	48.59
2	AP	2788.5	41.01
3	AC	3686.75	45.52
4	AM	5478.89	37.27
5	RO	11417.38	46.22
6	TO	11732.68	42.05
7	SE	14111.47	40.9
8	AL	15914.59	38.72
9	RN	18860.1	39.13
10	MS	19144.03	27.0

Insights:

>If the average freight value is higher than ***try to reduce the consignments/ shipments***

>Select the ***most suitable among the competitors***

>Select the ***most optimal way of transportation route in the states with highest total freight cost and avg is high***

v. 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**

```
SELECT

    order_id,
    order_purchase_timestamp AS order_placed,
    order_delivered_customer_date AS order_delivered,
    order_estimated_delivery_date AS estimated_delivery,
    DATE_DIFF(order_delivered_customer_date,
order_purchase_timestamp,DAY) del_time,
    DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date,DAY) diff_est_del,
FROM Biz_case_target.orders
WHERE order_status = "delivered"
ORDER BY - diff_est_del DESC
```

Row	order_id	order_placed	order_delivered	estimated_delivery	del_time	diff_est_del
1	1b3190b2dfa9d789e1f14c05b...	2018-02-23 14:57:35 UTC	2018-09-19 23:24:07 UTC	2018-03-15 00:00:00 UTC	208	-188
2	ca07593549f1816d26a572e06...	2017-02-21 23:31:27 UTC	2017-09-19 14:36:39 UTC	2017-03-22 00:00:00 UTC	209	-181
3	47b40429ed8cce3aee9199792...	2018-01-03 09:44:01 UTC	2018-07-13 20:51:31 UTC	2018-01-19 00:00:00 UTC	191	-175
4	2fe324febf907e3ea3f2aa9650...	2017-03-13 20:17:10 UTC	2017-09-19 17:00:07 UTC	2017-04-05 00:00:00 UTC	189	-167
5	285ab9426d6982034523a855f...	2017-03-08 22:47:40 UTC	2017-09-19 14:00:04 UTC	2017-04-06 00:00:00 UTC	194	-166
6	440d0d17af552815d15a9e41a...	2017-03-07 23:59:51 UTC	2017-09-19 15:12:50 UTC	2017-04-07 00:00:00 UTC	195	-165
7	c27815f7e3dd0b926b5855262...	2017-03-15 23:23:17 UTC	2017-09-19 17:14:25 UTC	2017-04-10 00:00:00 UTC	187	-162
8	0f4519c5f1c541ddec9f21b3bd...	2017-03-09 13:26:57 UTC	2017-09-19 14:38:21 UTC	2017-04-11 00:00:00 UTC	194	-161
9	d24e8541128cea179a11a6517...	2017-06-12 13:14:11 UTC	2017-12-04 18:36:29 UTC	2017-06-26 00:00:00 UTC	175	-161
10	2d7561026d542c8dbd8f0daea...	2017-03-15 11:24:27 UTC	2017-09-19 14:38:18 UTC	2017-04-13 00:00:00 UTC	188	-159

Insights:
> Based on the data if the time taken to deliver is more than estimated time than that would show a **negative**
impact** on **customers** and **there order numbers
*>To avoid this take some precautionary measures to keep an eye on the **inventory and its availability near the***

location
*>Make sure **inform customers in prior** about the **effects due to seasonal changes** for example heavy rains or*

etc

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

TOP5

```
with cte as
(SELECT
       customer_state,
       ROUND(SUM(i.freight_value),2)  total_freight_value,
       COUNT(DISTINCT o.order_id) no_of_orders
   FROM Biz_case_target.orders o
   JOIN Biz_case_target.customers c
   ON c.customer_id = o.customer_id
   JOIN Biz_case_target.order_items i
   ON i.order_id = o.order_id
GROUP BY customer_state
ORDER BY customer_state)
SELECT

    customer_state,
    total_freight_value,
    ROUND(total_freight_value/no_of_orders,2) as avg_freight_val
FROM cte
ORDER BY 3 DESC
LIMIT 5
```

Row	customer_state	total_freight_value	avg_freight_val
1	RR	2235.19	48.59
2	PB	25719.73	48.35
3	RO	11417.38	46.22
4	AC	3686.75	45.52
5	PI	21218.2	43.04

BOTTOM 5

```
with cte as
(SELECT
       customer_state,
       ROUND(SUM(i.freight_value),2)  total_freight_value,
       COUNT(DISTINCT o.order_id) no_of_orders
   FROM Biz_case_target.orders o
   JOIN Biz_case_target.customers c
   ON c.customer_id = o.customer_id
   JOIN Biz_case_target.order_items i
   ON i.order_id = o.order_id
GROUP BY customer_state
ORDER BY customer_state)
SELECT

    customer_state,
    total_freight_value,
    ROUND(total_freight_value/no_of_orders,2) as avg_freight_val
FROM cte
ORDER BY 3
LIMIT 5
```

Row	customer_state	total_freight_value	avg_freight_val
1	SP	718723.07	17.37
2	MG	270853.46	23.46
3	PR	117851.68	23.58
4	DF	50625.5	23.82
5	RJ	305589.31	23.95

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

TOP5

```
with cte as
(SELECT
  customer_state,
DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,D
AY) AS del_time,
  o.order_id
FROM Biz_case_target.orders o
JOIN Biz_case_target.customers c
ON o.customer_id = c.customer_id
WHERE order_status = "delivered"
GROUP BY 1,2,3)
SELECT
   customer_state,
```

```
   ROUND(SUM(del_time)/COUNT(DISTINCT order_id),2) as avg_del_time
FROM cte
GROUP BY customer_state
ORDER BY 2 desc
LIMIT 5
```

Row	customer_state	avg_del_time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

BOTTOM 5

```
with cte as
(SELECT
  customer_state,
DATE_DIFF(order_delivered_customer_date,order_purchase_timestamp,D
AY) AS del_time,
  o.order_id
FROM Biz_case_target.orders o
JOIN Biz_case_target.customers c
ON o.customer_id = c.customer_id
WHERE order_status = "delivered"
GROUP BY 1,2,3)
SELECT
   customer_state,
   ROUND(SUM(del_time)/COUNT(DISTINCT order_id),2) as avg_del_time
FROM cte
GROUP BY customer_state
ORDER BY 2 LIMIT 5
```

Row	customer_state	avg_del_time
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

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.

FASTEST 5

```
with cte as (

SELECT

  customer_state,
  DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date,DAY) diff_est_del,
  o.order_id
FROM Biz_case_target.orders o
JOIN Biz_case_target.customers c
ON o.customer_id = c.customer_id
WHERE order_status = "delivered"
GROUP BY 1,2,3
)
SELECT
   customer_state,
   ROUND(SUM(diff_est_del)/COUNT(DISTINCT order_id),2) as
avg_diff_est_del
FROM cte
GROUP BY customer_state
ORDER BY 2
LIMIT 5
```

Row	customer_state	avg_diff_est_del
1	AL	7.95
2	MA	8.77
3	SE	9.17
4	ES	9.62
5	BA	9.93

SLOWEST 5

```
with cte as (

SELECT

  customer_state,
  DATE_DIFF(order_estimated_delivery_date,
order_delivered_customer_date,DAY) diff_est_del,
  o.order_id
FROM Biz_case_target.orders o
JOIN Biz_case_target.customers c
ON o.customer_id = c.customer_id
WHERE order_status = "delivered"
GROUP BY 1,2,3
)
SELECT
   customer_state,
   ROUND(SUM(diff_est_del)/COUNT(DISTINCT order_id),2) as
avg_diff_est_del
FROM cte
GROUP BY customer_state
ORDER BY 2 DESC
LIMIT 5
```

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

VI. Analysis based on the payments

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

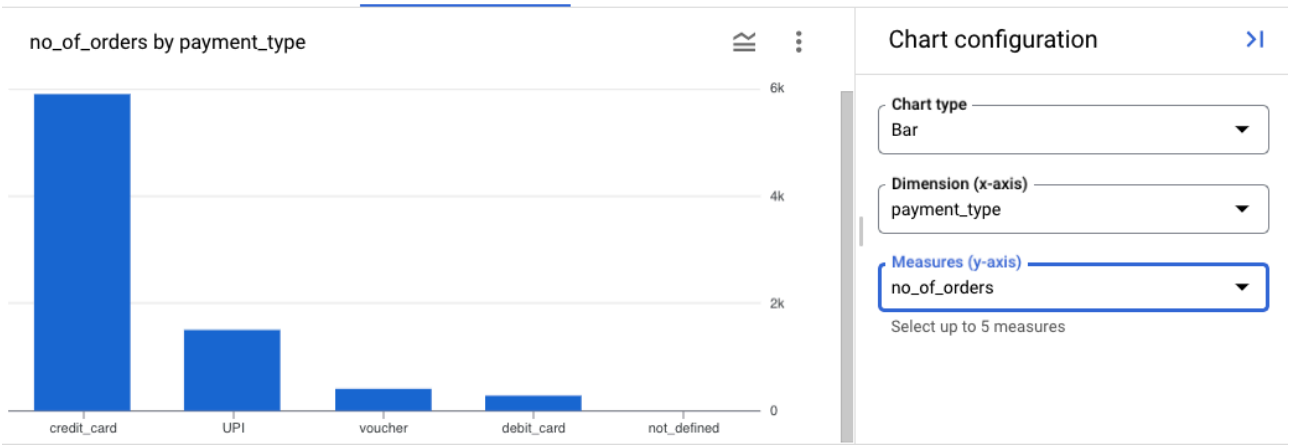
```
SELECT * FROM
(
   SELECT payment_type,
          EXTRACT(year FROM order_purchase_timestamp) AS years,
          EXTRACT(month FROM order_purchase_timestamp) AS
months,
          COUNT(o.order_id) AS no_of_orders
   FROM Biz_case_target.orders o
   JOIN Biz_case_target.payments p
   ON o.order_id = p.order_id
```

```
   GROUP BY EXTRACT(year FROM
order_purchase_timestamp),EXTRACT(month FROM
   order_purchase_timestamp),payment_type
)a
ORDER BY payment_type,years,months
```

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAIL

Row	payment_type	years	months	no_of_orders
1	UPI	2016	10	63
2	UPI	2017	1	197
3	UPI	2017	2	398
4	UPI	2017	3	590
5	UPI	2017	4	496
6	UPI	2017	5	772
7	UPI	2017	6	707
8	UPI	2017	7	845
9	UPI	2017	8	938
10	UPI	2017	9	903

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



Insights:

*>There are **different payment options** which are seen in the above graph that gives us an **idea of flexibility in***

customers payment modes** which is an **added factor to enhance customer activity

>So we have diverse customers every year (2016-2018) there are payments made from all the different modes every year

*>From 2017 Jan to 2018 Aug there is a **significant and constant increase in the credit card usage** while paying and the no of orders placed across all these months **range from 500 to 5000***

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

```
SELECT *
FROM
(
SELECT

          payment_installments,
          COUNT(order_id) AS no_of_orders
     FROM Biz_case_target.payments
     GROUP BY payment_installments
)
WHERE payment_installments >= 1
Order by 2 DESC
```

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