

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

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

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

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

**Insight :-** In customer table we have 5 columns, and 4 columns are string and 1 column is integer

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

- Time range according to each date per order count
- ```
with cte as (
SELECT
order_id,
customer_id,
EXTRACT(date FROM order_purchase_timestamp) as order_by_date,
EXTRACT(time FROM order_purchase_timestamp) as order_by_time,
FROM `Target.orders`)
```

```
select
order_by_date,
count(order_id) as order_count,
min(order_by_time) as start_time,
max(order_by_time) as end_time
from cte
group by order_by_date
order by order_by_date;
```

| Row | order_by_date | order_count | start_time | end_time |
|-----|---------------|-------------|------------|----------|
| 1   | 2016-09-04    | 1           | 21:15:19   | 21:15:19 |
| 2   | 2016-09-05    | 1           | 00:15:34   | 00:15:34 |
| 3   | 2016-09-13    | 1           | 15:24:19   | 15:24:19 |
| 4   | 2016-09-15    | 1           | 12:16:38   | 12:16:38 |
| 5   | 2016-10-02    | 1           | 22:07:52   | 22:07:52 |
| 6   | 2016-10-03    | 8           | 09:44:50   | 22:51:30 |
| 7   | 2016-10-04    | 63          | 09:06:10   | 23:59:01 |
| 8   | 2016-10-05    | 47          | 00:32:31   | 23:14:34 |
| 9   | 2016-10-06    | 51          | 00:06:17   | 23:49:18 |
| 10  | 2016-10-07    | 46          | 00:54:40   | 23:18:38 |
| 11  | 2016-10-08    | 42          | 01:28:14   | 23:46:06 |
| 12  | 2016-10-09    | 26          | 00:56:52   | 23:55:30 |
| 13  | 2016-10-10    | 39          | 00:01:50   | 18:09:39 |
| 14  | 2016-10-22    | 1           | 08:25:27   | 08:25:27 |
| 15  | 2016-12-23    | 1           | 23:16:47   | 23:16:47 |

**Insight :-** where the order count is 1 then we can see the start time and the end time is same , when the order count is more than 1 then we can see the start time and end time will be different and we get time range

- For entire data set time range where order were placed

SELECT

max(EXTRACT(date FROM order\_purchase\_timestamp )) as  
order\_by\_date,

min(EXTRACT(date FROM order\_purchase\_timestamp )) as  
order\_started\_date

FROM `Target.orders`;

| Row | order_by_date | order_started_date |
|-----|---------------|--------------------|
| 1   | 2018-10-17    | 2016-09-04         |

**Insight :-** we get the time range for entire orders for target company where customer purchase started year is 2016 and customer purchase till 2018 -10 -17

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

- According to customer\_table

select

count(distinct customer\_state) as no\_of\_states,

count(distinct customer\_city) as no\_of\_city

from `Target.customers`;

| Row | no_of_states | no_of_city |
|-----|--------------|------------|
| 1   | 27           | 4119       |

**Insight :-** According to customer\_table we got the states = 27 and cities = 4119

- According to geolocation table

SELECT

count(distinct(geolocation\_state)) as no\_of\_state,

```
count(distinct(geolocation_city)) as no_of_cities
FROM `Target.geolocation`
```

| Row | no_of_state | no_of_cities |
|-----|-------------|--------------|
| 1   | 27          | 8011         |

**Insight :-** According to geolocation table we got the states = 27 and the cities = 8011

- According to sellers table

```
select
count(distinct seller_state) as no_of_seller_state,
count(distinct seller_city) as no_of_seller_city
from `Target.sellers`
```

| Row | no_of_seller_state | no_of_seller_city |
|-----|--------------------|-------------------|
| 1   | 23                 | 611               |

**Insight :-** According to seller table we got the states = 23 and the cities = 611

## 2. In-depth Exploration:

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

```
with cte as (
select
order_on_year,
count(order_id) as count_order
from
(
SELECT *,
EXTRACT(year FROM order_purchase_timestamp ) as order_on_year
FROM `Target.orders`)
group by order_on_year
),
cte2 as (
select
order_on_year,
count_order,
lag(count_order,1) over(order by order_on_year) as previous_year,
from cte)

select
```

```

order_on_year,
count_order,
round((count_order-previous_year)/ previous_year* 100,2) as
increment_order
from cte2
order by 1;

```

| Row | order_on_year | count_order | increment_order |
|-----|---------------|-------------|-----------------|
| 1   | 2016          | 329         | null            |
| 2   | 2017          | 45101       | 13608.51        |
| 3   | 2018          | 54011       | 19.76           |

**Insight :-** In order 2016 to 2017, huge increment in no of orders we can see in 2016 the no of orders 329 and 2017 the no of order 45101 huge increment and the % of increment is 13608.51 , In year 2018 the % of increment order is 19.76 .

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

```

select
order_on_month,
count(order_id) as count_order
from
(
SELECT *,
EXTRACT(month FROM order_purchase_timestamp ) as order_on_month
FROM `Target.orders`)
group by order_on_month
order by 2 DESC;

```

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

**Insight :-** we can see the no of orders are **dropped in last 4 months** , and **peak** of the no of order is on **June and August Month**

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

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

```

with cte as
(
SELECT
*,
CASE
WHEN EXTRACT(HOUR FROM order_purchase_timestamp AT TIME ZONE
"UTC") BETWEEN 0 AND 6 THEN "Dawn"
WHEN EXTRACT(HOUR FROM order_purchase_timestamp AT TIME ZONE
"UTC") BETWEEN 7 AND 12 THEN "Mornings"
WHEN EXTRACT(HOUR FROM order_purchase_timestamp AT TIME ZONE
"UTC") BETWEEN 13 AND 18 THEN "Afternoon"
WHEN EXTRACT(HOUR FROM order_purchase_timestamp AT TIME ZONE
"UTC") BETWEEN 19 AND 23 THEN "Night"
end as Days
FROM `Target.orders`)

select cte.Days,
       count(*) as no_of_days
from cte

```

```
group by cte.Days
order by 2 DESC ;
```

| Row | Days      | no_of_orders |
|-----|-----------|--------------|
| 1   | Afternoon | 38135        |
| 2   | Night     | 28331        |
| 3   | Mornings  | 27733        |
| 4   | Dawn      | 5242         |

**Insight :-** Afternoon day time is most no orders are coming. And the less no of order come from Dawn day time.

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

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

```
with cte as
(
select *,
extract(month from order_purchase_timestamp) as order_on_month
FROM `Target.orders`)
```

```
select
    c.customer_state,
    order_on_month,
    count(*) as no_of_orders
from `Target.customers` as c
join cte
on c.customer_id = cte.customer_id
group by 1, 2
order by 1 ,2 , 3 DESC;
```

| Row | customer_state | order_on_month | no_of_orders |
|-----|----------------|----------------|--------------|
| 1   | AC             | 1              | 8            |
| 2   | AC             | 2              | 6            |
| 3   | AC             | 3              | 4            |
| 4   | AC             | 4              | 9            |
| 5   | AC             | 5              | 10           |
| 6   | AC             | 6              | 7            |
| 7   | AC             | 7              | 9            |
| 8   | AC             | 8              | 7            |
| 9   | AC             | 9              | 5            |
| 10  | AC             | 10             | 6            |
| 11  | AC             | 11             | 5            |
| 12  | AC             | 12             | 5            |

Results per page: 50 1 – 50 of 322

**Insight :-** we get the no of orders, month on month for each state ex for ac state we  
example:- get the different no of orders for 12 months

```
with cte as
(
select *,
extract(year from order_purchase_timestamp) as order_on_year,
extract(month from order_purchase_timestamp) as order_on_month
FROM `Target.orders`)
select
    c.customer_state,
    order_on_year,
    order_on_month,
    count(*) as no_of_orders
from `Target.customers` as c
join cte
    on c.customer_id = cte.customer_id
group by 1, 2, 3
order by 1, 2, 3, 4 DESC;
```

| Row | customer_state | order_on_year | order_on_month | no_of_orders |
|-----|----------------|---------------|----------------|--------------|
| 1   | AC             | 2017          | 1              | 2            |
| 2   | AC             | 2017          | 2              | 6            |
| 3   | AC             | 2017          | 3              | 6            |
| 4   | AC             | 2017          | 4              | 20           |
| 5   | AC             | 2017          | 5              | 40           |
| 6   | AC             | 2017          | 6              | 24           |
| 7   | AC             | 2017          | 7              | 35           |
| 8   | AC             | 2017          | 8              | 32           |
| 9   | AC             | 2017          | 9              | 45           |
| 10  | AC             | 2017          | 10             | 60           |
| 11  | AC             | 2017          | 11             | 55           |
| 12  | AC             | 2017          | 12             | 60           |

**Insight :-** we get the no of orders, according year month on month for each state for  
Example :- ac state we get the different no of orders for 12 months

## 2. How are the customers distributed across all the states?

```
select
    customer_state,
    count(*) as no_of_customers
from `Target.customers`
group by customer_state
order by 2 DESC;
```

| Row | customer_state | no_of_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            |
| 11  | PE             | 1652            |
| 12  | CE             | 1336            |
| 13  | PA             | 975             |
| 14  | MT             | 907             |
| 15  | MA             | 747             |

**Insight :-** customer are distributed in all 27 state and max no of customers are from sp state , and min no of customers states are from RR state - 46

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

1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).  
You can use the "payment\_value" column in the payments table to get the cost of orders.

```
with cte as (
select *,
extract(year from order_purchase_timestamp) as order_on_year,
extract(month from order_purchase_timestamp) as order_on_month,
FROM `Target.orders`
where extract(year from order_purchase_timestamp) In (2017, 2018) and
        extract(month from order_purchase_timestamp) between 1 and 8
),
cte2 as
(
select
cte.order_on_year,
round(sum(p.payment_value),2) as payment_value_per_year
from `Target.payments` as p
join cte
on cte.order_id = p.order_id
```



```

group by cte.order_on_year
order by 1)
, cte3 as(
select
    cte2.order_on_year,
    cte2.payment_value_per_year,
    lag(cte2.payment_value_per_year,1) over(order by cte2.order_on_year)
as previous_year_value
from cte2)
select
    cte3.order_on_year,
    round((cte3.payment_value_per_year -
previous_year_value)/cte3.payment_value_per_year,2) * 100 as
Percent_increment
from cte3;

```

| Row | order_on_year | Percent_increment |
|-----|---------------|-------------------|
| 1   | 2018          | 57.8              |
| 2   | 2017          | null              |

Insight :- 57.80 is the % increase in the cost of orders from year 2017 to 2018

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

```

with cte as
(select
    o.order_id,
    customer_state
from `Target.customers` as c
join `Target.orders` as o
on o.customer_id = c.customer_id)

select
    cte.customer_state,
    Round(sum(oi.price),2) as Total_price,
    round(avg(oi.price),2) as avg_price
from `Target.order_items` as oi
join cte
on oi.order_id = cte.order_id
group by cte.customer_state
order by 2 DESC

```

| Row | customer_state | Total_price | avg_price |
|-----|----------------|-------------|-----------|
| 1   | SP             | 5202955.05  | 109.65    |
| 2   | RJ             | 1824092.67  | 125.12    |
| 3   | MG             | 1585308.03  | 120.75    |
| 4   | RS             | 750304.02   | 120.34    |
| 5   | PR             | 683083.76   | 119.0     |
| 6   | SC             | 520553.34   | 124.65    |
| 7   | BA             | 511349.99   | 134.6     |
| 8   | DF             | 302603.94   | 125.77    |
| 9   | GO             | 294591.95   | 126.27    |
| 10  | ES             | 275037.31   | 121.91    |
| 11  | PE             | 262788.03   | 145.51    |
| 12  | CE             | 227254.71   | 153.76    |

**Insight :-** From the Q3 most no of orders come from Sp state and here the max total order price is 5202955.05 from sp state , and the avg price value is 109.65 ,

From the Q3 most least no of orders come from RR state and here the min total order price is 7829.09 from state , and the avg price value is 150.57 ,

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

```

with cte as
(select
  o.order_id,
  customer_state
from `Target.customers` as c
join `Target.orders` as o
  on o.customer_id = c.customer_id)

select
  cte.customer_state,
  Round(sum(oi.freight_value),2) as Total_freight_value,
  round(avg(oi.freight_value),2) as avg_freight_value
from `Target.order_items` as oi
join cte
  on oi.order_id = cte.order_id
group by cte.customer_state
order by 2 DESC;

```

| Row | customer_state ▼ | Total_freight_value | avg_freight_value ▼ |
|-----|------------------|---------------------|---------------------|
| 1   | SP               | 718723.07           | 15.15               |
| 2   | RJ               | 305589.31           | 20.96               |
| 3   | MG               | 270853.46           | 20.63               |
| 4   | RS               | 135522.74           | 21.74               |
| 5   | PR               | 117851.68           | 20.53               |
| 6   | BA               | 100156.68           | 26.36               |
| 7   | SC               | 89660.26            | 21.47               |
| 8   | PE               | 59449.66            | 32.92               |
| 9   | GO               | 53114.98            | 22.77               |
| 10  | DF               | 50625.5             | 21.04               |
| 11  | ES               | 49764.6             | 22.06               |
| 12  | CE               | 48351.59            | 32.71               |

**Insight :-** From the Q3 most no of orders come from Sp state and here the max total order price is 5202955.05 from sp state , and the avg price value is 109.65 , And the total\_freight value is 718723.07 and the avg freight value is 15.15

From the Q3 most least no of orders come from RR state and here the min total order price is 7829.09 from state , and the avg price value is 150.57 , And the total\_freight value is 2235.09 and the avg freight value is 42.98

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

- 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_status,
order_purchase_timestamp,
order_estimated_delivery_date,
order_delivered_customer_date,
Actual_delivery_days,
```

```

estimated_delivery_days
from(
select
*,
date_diff(order_delivered_customer_date, order_purchase_timestamp,
day) as Actual_delivery_days,
date_diff(order_estimated_delivery_date,
order_delivered_customer_date,day) as estimated_delivery_days
from
`Target.orders`) as t1
where t1.order_status = "delivered" and Actual_delivery_days is not
null
order by estimated_delivery_days ASC

```

| Row | er_status | order_purchase_timestamp | order_estimated_delivery_date | order_delivered_customer_date | Actual_delivery_days | estimated_delivery_days |
|-----|-----------|--------------------------|-------------------------------|-------------------------------|----------------------|-------------------------|
| 1   | vered     | 2018-02-23 14:57:35 UTC  | 2018-03-15 00:00:00 UTC       | 2018-09-19 23:24:07 UTC       | 208                  | -188                    |
| 2   | vered     | 2017-02-21 23:31:27 UTC  | 2017-03-22 00:00:00 UTC       | 2017-09-19 14:36:39 UTC       | 209                  | -181                    |
| 3   | vered     | 2018-01-03 09:44:01 UTC  | 2018-01-19 00:00:00 UTC       | 2018-07-13 20:51:31 UTC       | 191                  | -175                    |
| 4   | vered     | 2017-03-13 20:17:10 UTC  | 2017-04-05 00:00:00 UTC       | 2017-09-19 17:00:07 UTC       | 189                  | -167                    |
| 5   | vered     | 2017-03-08 22:47:40 UTC  | 2017-04-06 00:00:00 UTC       | 2017-09-19 14:00:04 UTC       | 194                  | -166                    |
| 6   | vered     | 2017-03-07 23:59:51 UTC  | 2017-04-07 00:00:00 UTC       | 2017-09-19 15:12:50 UTC       | 195                  | -165                    |
| 7   | vered     | 2017-03-15 23:23:17 UTC  | 2017-04-10 00:00:00 UTC       | 2017-09-19 17:14:25 UTC       | 187                  | -162                    |
| 8   | vered     | 2017-03-09 13:26:57 UTC  | 2017-04-11 00:00:00 UTC       | 2017-09-19 14:38:21 UTC       | 194                  | -161                    |
| 9   | vered     | 2017-06-12 13:14:11 UTC  | 2017-06-26 00:00:00 UTC       | 2017-12-04 18:36:29 UTC       | 175                  | -161                    |
| 10  | vered     | 2017-03-15 11:24:27 UTC  | 2017-04-13 00:00:00 UTC       | 2017-09-19 14:38:18 UTC       | 188                  | -159                    |

**Insight :-** where actual delivery is 0 mean the delivery time is within 24 hours and where estimated delivery is negative means delivery getting late .

Example :- order purchase is 23 feb , order estimated delivery time is 15 march , but in real the order delivered in 19 september

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

```

(
select
c.customer_state,
round(avg(oi.freight_value),2) as avg_freight_value,
"Top 5 freight value " as sorted
from `Target.customers` as c
join `Target.orders` as o
on o.customer_id = c.customer_id
join `Target.order_items` as oi
on oi.order_id = o.order_id

```

```

group by c.customer_state
order by 2 DESC limit 5)
union all
(
select
    c.customer_state,
    round(avg(oi.freight_value),2) as avg_freight_value,
    "bottom 5 freight value" as sorted
from `Target.customers` as c
join `Target.orders` as o
    on o.customer_id = c.customer_id
join `Target.order_items` as oi
    on oi.order_id = o.order_id
group by c.customer_state)

```

| Row | customer_state | avg_freight_value | sorted           |
|-----|----------------|-------------------|------------------|
| 1   | RR             | 42.98             | Top 5 high value |
| 2   | PB             | 42.72             | Top 5 high value |
| 3   | RO             | 41.07             | Top 5 high value |
| 4   | AC             | 40.07             | Top 5 high value |
| 5   | PI             | 39.15             | Top 5 high value |
| 6   | SP             | 15.15             | Top 5 low value  |
| 7   | PR             | 20.53             | Top 5 low value  |
| 8   | MG             | 20.63             | Top 5 low value  |
| 9   | RJ             | 20.96             | Top 5 low value  |
| 10  | DF             | 21.04             | Top 5 low value  |

**Insight :-** Top highest value avg\_freight value is 42.98 from RR State.  
 Top lowest value avg\_freight value is 21.04 from DF state.

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

```

(select
    c.customer_state,
    round(avg(date_diff(order_delivered_customer_date,order_purchase_times
tamp,day)),2) as actual_delivery_time,
    "Top 5 State Deliver Time" as sort
from `Target.orders` as o
join `Target.customers` as c
    on o.customer_id = c.customer_id

```

```

group by 1
order by 2 DESC limit 5)
union all
(select
c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_times
tamp,day)),2) as Avg_delivery_time,
"bottom 5 State Deliver Time " as sort
from `Target.orders` as o
join `Target.customers` as c
on o.customer_id = c.customer_id
group by 1
order by 2 asc limit 5)

```

| Row | customer_state | avg_delivery_time | sort                        |
|-----|----------------|-------------------|-----------------------------|
| 1   | RR             | 28.98             | Top 5 State Deliver Time    |
| 2   | AP             | 26.73             | Top 5 State Deliver Time    |
| 3   | AM             | 25.99             | Top 5 State Deliver Time    |
| 4   | AL             | 24.04             | Top 5 State Deliver Time    |
| 5   | PA             | 23.32             | Top 5 State Deliver Time    |
| 6   | SP             | 8.3               | bottom 5 State Deliver Time |
| 7   | PR             | 11.53             | bottom 5 State Deliver Time |
| 8   | MG             | 11.54             | bottom 5 State Deliver Time |
| 9   | DF             | 12.51             | bottom 5 State Deliver Time |
| 10  | SC             | 14.48             | bottom 5 State Deliver Time |

**Insight :-** Top highest value avg\_delivery time is 28.98 from RR State.  
Top lowest value avg\_delivery time is 14.48 from SC state.

**4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.**

**You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.**

```

select
t1.customer_state,
round((t1.Actual_delivery_days- t1.estimated_delivery_days),2) as
fastest_delivery
from
(select
c.customer_state,

```

```

avg(date_diff(order_delivered_customer_date, order_purchase_timestamp,
day)) as Actual_delivery_days,
avg(date_diff(order_estimated_delivery_date,
order_delivered_customer_date,day)) as estimated_delivery_days

```

```

from `Target.orders` as o
join `Target.customers` as c
on c.customer_id = o.customer_id
where o.order_status = "delivered"
group by c.customer_state) as t1
order by 2 ASC
limit 5

```

| Row | customer_state | fastest_delivery |
|-----|----------------|------------------|
| 1   | SP             | -1.84            |
| 2   | PR             | -0.84            |
| 3   | MG             | -0.76            |
| 4   | RO             | -0.22            |
| 5   | AC             | 0.87             |

**Insight :-** fastest delivery in sp state , time is -1.84

## 6. Analysis based on the payments:

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

```

with cte as
(
select *,
extract(year from order_purchase_timestamp) as payment_year,
extract(month from order_purchase_timestamp) as payment_month
FROM `Target.orders`)

select p.payment_type,
       cte.payment_year,
       cte.payment_month,
       count(*) as no_of_orders

```

```

from `Target.payments` as p
join cte
on cte.order_id = p.order_id
group by 1, 2, 3
order by 2, 1, 3, 4 DESC;

```

| Row | payment_type | payment_year | payment_month | no_of_orders |
|-----|--------------|--------------|---------------|--------------|
| 1   | UPI          | 2016         | 10            | 63           |
| 2   | credit_card  | 2016         | 9             | 3            |
| 3   | credit_card  | 2016         | 10            | 254          |
| 4   | credit_card  | 2016         | 12            | 1            |
| 5   | debit_card   | 2016         | 10            | 2            |
| 6   | voucher      | 2016         | 10            | 23           |
| 7   | UPI          | 2017         | 1             | 197          |
| 8   | UPI          | 2017         | 2             | 398          |
| 9   | UPI          | 2017         | 3             | 590          |
| 10  | UPI          | 2017         | 4             | 496          |
| 11  | UPI          | 2017         | 5             | 772          |
| 12  | UPI          | 2017         | 6             | 707          |

**Insight :** - no of order totally different on payment type . example for 2016 october month , payment paid by upi is 63, credit card is 254, debit card is 2

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

```

select count(*) as total_no_of_orders
from
(select
order_id,
payment_sequential,
payment_installments,
payment_sequential - payment_installments as installment_paid
from `Target.payments`
where payment_sequential - payment_installments = 0 ) as t2

```

| JOB INFORMATION |                    | RESULTS | JSON | EXECUTED |
|-----------------|--------------------|---------|------|----------|
| Row             | total_no_of_orders |         |      |          |
| 1               | 48290              |         |      |          |



**Insight :-** no of orders which are paid = 48290

## 7. Actionable Insights & Recommendations

**1.Recommendations :-** In the growing trend we are seeing the 2016 to 2017 we get amazing increment in order but now we have very less increment order so Target should need to work on his marketing strategy so they can keep up against the competition .

| Row | order_on_year | count_order | increment_order |
|-----|---------------|-------------|-----------------|
| 1   | 2016          | 329         | null            |
| 2   | 2017          | 45101       | 13608.51        |
| 3   | 2018          | 54011       | 19.76           |

**2.Recommendations :-** we can see from the details that september , october, november and december have low order count .

September to december month includes the holiday seasons , we can apply some strategy .

- Holiday themed promotions
- Social media marketing
- Seasonal Email Campaigns
- Customer Loyalty discounts
- Cross-promotions and collaborations

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

### 3.Recommendations :-

Offer special promotions, discounts, or **limited-time offers specifically tailored for the dawn hours**. This can incentivize customers to place orders during this time by creating a sense of urgency and value.

Collab with local partners that are active during the dawn hours.  
Analyzing the customer behaviour. And Providing good service

| Row | Days      | no_of_orders |
|-----|-----------|--------------|
| 1   | Afternoon | 38135        |
| 2   | Night     | 28331        |
| 3   | Mornings  | 27733        |
| 4   | Dawn      | 5242         |

### 4.Recommendations :-

Some state has few no of orders . So we can increase our website and local presence using search engine optimization . And hire good SEOs who can optimize our site so they can use state specific keywords states whey they search for relevant products or service.

- Giving some State specific Offeres.
- Regional partnerships who has good influence in specific states. They can promote products .
- Providing referral customer programe.
- Personalized email marketing

| Row | customer_state | no_of_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            |
| 11  | PE             | 1652            |
| 12  | CE             | 1336            |
| 13  | PA             | 975             |
| 14  | MT             | 907             |
| 15  | MA             | 747             |

## 5.Recommendations :-

Some state has very high freight value . To reduce freight value by combining multiple shipments into a single delivery .

Reach out to your freight carriers and negotiate better rates based on shipping volume.

Partnering with third party agent for the shipment by storing the inventory

| Row | customer_state | Total_freight_value | avg_freight_value |
|-----|----------------|---------------------|-------------------|
| 1   | SP             | 718723.07           | 15.15             |
| 2   | RJ             | 305589.31           | 20.96             |
| 3   | MG             | 270853.46           | 20.63             |
| 4   | RS             | 135522.74           | 21.74             |
| 5   | PR             | 117851.68           | 20.53             |
| 6   | BA             | 100156.68           | 26.36             |
| 7   | SC             | 89660.26            | 21.47             |
| 8   | PE             | 59449.66            | 32.92             |
| 9   | GO             | 53114.98            | 22.77             |
| 10  | DF             | 50625.5             | 21.04             |
| 11  | ES             | 49764.6             | 22.06             |
| 12  | CE             | 48351.59            | 32.71             |

## 6.Recommendations :-

### To improve delivery time

- Implement robust inventory management systems to accurately track stock levels and avoid stockouts. Having real-time visibility into your inventory helps you fulfill orders promptly without delays.
- Enable order tracking capabilities so that customers can monitor the progress of their shipments. Integrate tracking services directly into your website or provide links to carrier tracking pages.
- Strategically locating warehouse
- Optimizing the shipment route

| Row | er_status | order_purchase_timestamp | order_estimated_delivery_date | order_delivered_customer_date | Actual_delivery_days | estimated_delivery_c |
|-----|-----------|--------------------------|-------------------------------|-------------------------------|----------------------|----------------------|
| 1   | vered     | 2018-02-23 14:57:35 UTC  | 2018-03-15 00:00:00 UTC       | 2018-09-19 23:24:07 UTC       | 208                  | -188                 |
| 2   | vered     | 2017-02-21 23:31:27 UTC  | 2017-03-22 00:00:00 UTC       | 2017-09-19 14:36:39 UTC       | 209                  | -181                 |
| 3   | vered     | 2018-01-03 09:44:01 UTC  | 2018-01-19 00:00:00 UTC       | 2018-07-13 20:51:31 UTC       | 191                  | -175                 |
| 4   | vered     | 2017-03-13 20:17:10 UTC  | 2017-04-05 00:00:00 UTC       | 2017-09-19 17:00:07 UTC       | 189                  | -167                 |
| 5   | vered     | 2017-03-08 22:47:40 UTC  | 2017-04-06 00:00:00 UTC       | 2017-09-19 14:00:04 UTC       | 194                  | -166                 |
| 6   | vered     | 2017-03-07 23:59:51 UTC  | 2017-04-07 00:00:00 UTC       | 2017-09-19 15:12:50 UTC       | 195                  | -165                 |
| 7   | vered     | 2017-03-15 23:23:17 UTC  | 2017-04-10 00:00:00 UTC       | 2017-09-19 17:14:25 UTC       | 187                  | -162                 |
| 8   | vered     | 2017-03-09 13:26:57 UTC  | 2017-04-11 00:00:00 UTC       | 2017-09-19 14:38:21 UTC       | 194                  | -161                 |
| 9   | vered     | 2017-06-12 13:14:11 UTC  | 2017-06-26 00:00:00 UTC       | 2017-12-04 18:36:29 UTC       | 175                  | -161                 |
| 10  | vered     | 2017-03-15 11:24:27 UTC  | 2017-04-13 00:00:00 UTC       | 2017-09-19 14:38:18 UTC       | 188                  | -159                 |

## 7.Recommendations :-

To increase the payment value collab with different different backs so they can provide some offers so customer has access to pay payment

| payment_year ▼ | payment_month ▼ | no_of_orders ▼ |
|----------------|-----------------|----------------|
| 2016           | 10              | 63             |
| 2016           | 9               | 3              |
| 2016           | 10              | 254            |
| 2016           | 12              | 1              |
| 2016           | 10              | 2              |
| 2016           | 10              | 23             |
| 2017           | 1               | 197            |
| 2017           | 2               | 398            |
| 2017           | 3               | 590            |
| 2017           | 4               | 496            |
| 2017           | 5               | 772            |
| 2017           | 6               | 707            |

## 8.additional analysis:-

Q1: Find the no of orders of each product\_category for each state

```
with cte1 as (select oi.order_id,
                  o.customer_id,
                  p.product_category
from `Target.order_items` as oi
join `Target.products` as p
on p.product_id = oi.product_id
join `Target.orders` as o
on oi.order_id = o.order_id)

select c.customer_state,
       if(product_category is Null, "Anonymous_products", product_category) as
product_category,
       count(*) as No_of_orders
from cte1
join `Target.customers` as c
on c.customer_id = cte1.customer_id
group by 1, 2
order by 3 DESC
```

| Row | customer_state ▼ | product_category ▼   | No_of_orders ▼ |
|-----|------------------|----------------------|----------------|
| 1   | SP               | bed table bath       | 5235           |
| 2   | SP               | HEALTH BEAUTY        | 4204           |
| 3   | SP               | sport leisure        | 3667           |
| 4   | SP               | Furniture Decoration | 3531           |
| 5   | SP               | housewares           | 3265           |
| 6   | SP               | computer accessories | 3170           |
| 7   | SP               | Watches present      | 2281           |
| 8   | SP               | automotive           | 1747           |
| 9   | SP               | toys                 | 1712           |
| 10  | SP               | telephony            | 1646           |
| 11  | RJ               | bed table bath       | 1644           |
| 12  | SP               | Garden tools         | 1559           |
| 13  | SP               | perfumery            | 1477           |

**Insight :** we can see for sp state product\_category is different

Q2: Find the no of product for each category

```

with cte1 as (select oi.order_id,
                    o.customer_id,
                    p.product_category
from `Target.order_items` as oi
join `Target.products` as p
on p.product_id = oi.product_id
join `Target.orders` as o
on oi.order_id = o.order_id)

select
    if(product_category is Null, "Anonymous_products", product_category) as
product_category,
    count(*) as No_of_product_category
from cte1

group by 1
order by 2 DESC;

```

| Row | product_category ▼   | No_of_product_cate |
|-----|----------------------|--------------------|
| 1   | bed table bath       | 11115              |
| 2   | HEALTH BEAUTY        | 9670               |
| 3   | sport leisure        | 8641               |
| 4   | Furniture Decoration | 8334               |
| 5   | computer accessories | 7827               |
| 6   | housewares           | 6964               |
| 7   | Watches present      | 5991               |
| 8   | telephony            | 4545               |
| 9   | Garden tools         | 4347               |
| 10  | automotive           | 4235               |
| 11  | toys                 | 4117               |
| 12  | Cool Stuff           | 3796               |
| 13  | perfumery            | 3419               |

Results per page:

Q 3 find the no of review for each product category

```

select
p.product_category,
count(ors.review_score) as no_of_review_score
from `Target.order_reviews` as ors
join `Target.order_items` as oi
on ors.order_id = oi.order_id
join `Target.products` as p
on p.product_id = oi.product_id
group by 1
order by 1

```

| Row | product_category ▼              | no_of_review_score |
|-----|---------------------------------|--------------------|
| 1   | <i>null</i>                     | 1598               |
| 2   | Agro Industria e Comercio       | 212                |
| 3   | Art                             | 207                |
| 4   | Arts and Crafts                 | 24                 |
| 5   | Bags Accessories                | 1088               |
| 6   | Blu Ray DVDs                    | 63                 |
| 7   | CITTE AND UPHACK FURNITURE      | 38                 |
| 8   | CONSTRUCTION SECURITY TO...     | 193                |
| 9   | Casa Construc o                 | 600                |
| 10  | Christmas articles              | 146                |
| 11  | Construction Tools Construction | 926                |
| 12  | Construction Tools Garden       | 240                |

Q: 4 How are the sellers distributed across all the states?

```

select
    seller_state,
    count(*) as no_of_sellers
from `Target.sellers`
group by 1
order by 2 DESC;

```

| Row | seller_state | no_of_sellers |
|-----|--------------|---------------|
| 1   | SP           | 1849          |
| 2   | PR           | 349           |
| 3   | MG           | 244           |
| 4   | SC           | 190           |
| 5   | RJ           | 171           |
| 6   | RS           | 129           |
| 7   | GO           | 40            |
| 8   | DF           | 30            |
| 9   | ES           | 23            |
| 10  | BA           | 19            |
| 11  | CE           | 13            |
| 12  | PE           | 9             |
| 13  | RR           | 6             |

**Q:5 Find the no of reviews for each review\_score**

```

SELECT
    review_score,
    COUNT(*) AS NO_OF_REVIEWS
FROM `Target.order_reviews`
GROUP BY
    review_score
ORDER BY
    Review_score

```

| Row | review_score | NO_OF_REVIEWS |
|-----|--------------|---------------|
| 1   | 1            | 11424         |
| 2   | 2            | 3151          |
| 3   | 3            | 8179          |
| 4   | 4            | 19142         |
| 5   | 5            | 57328         |