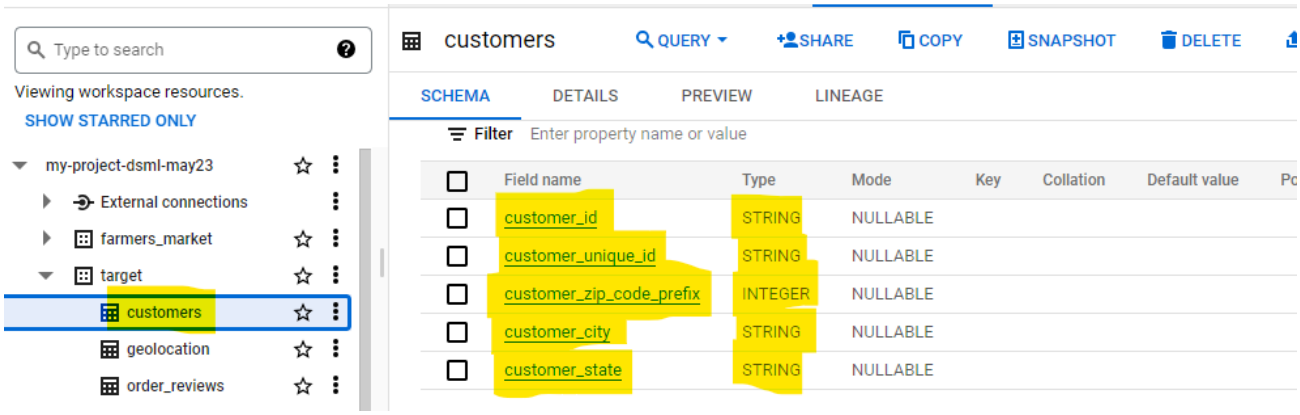
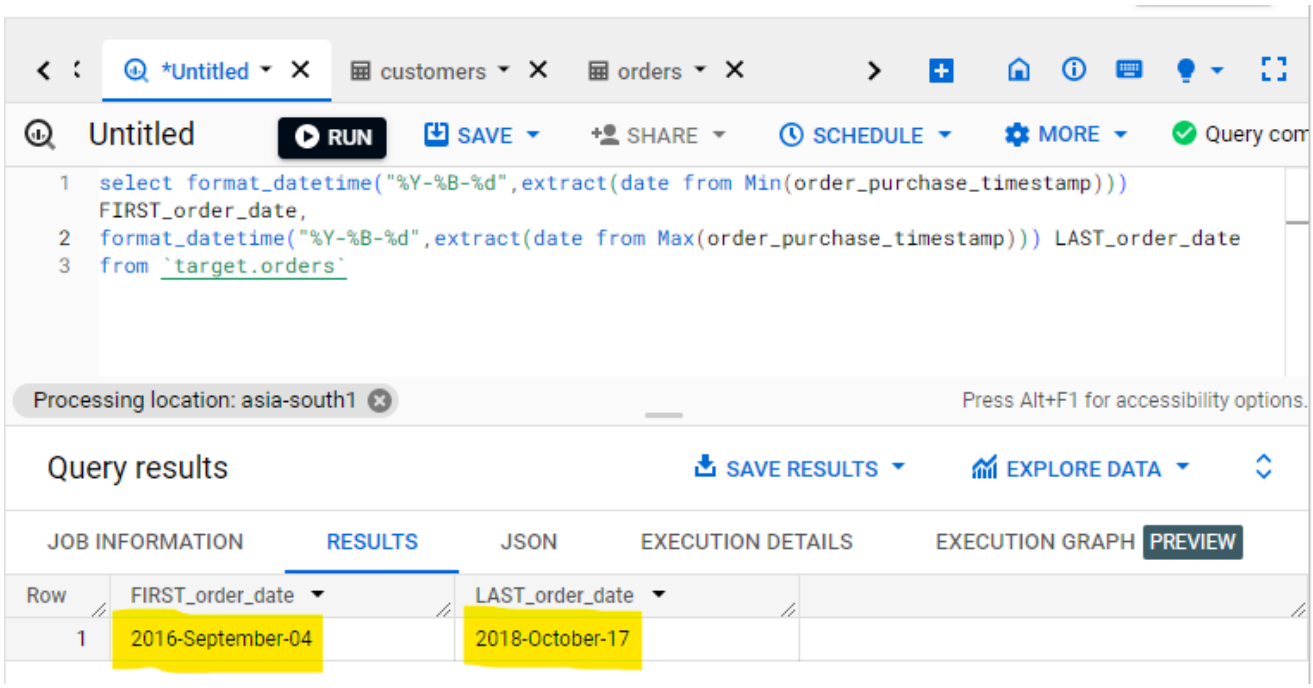
1. **Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:**

   a. Data type of all columns in the "customers" table.



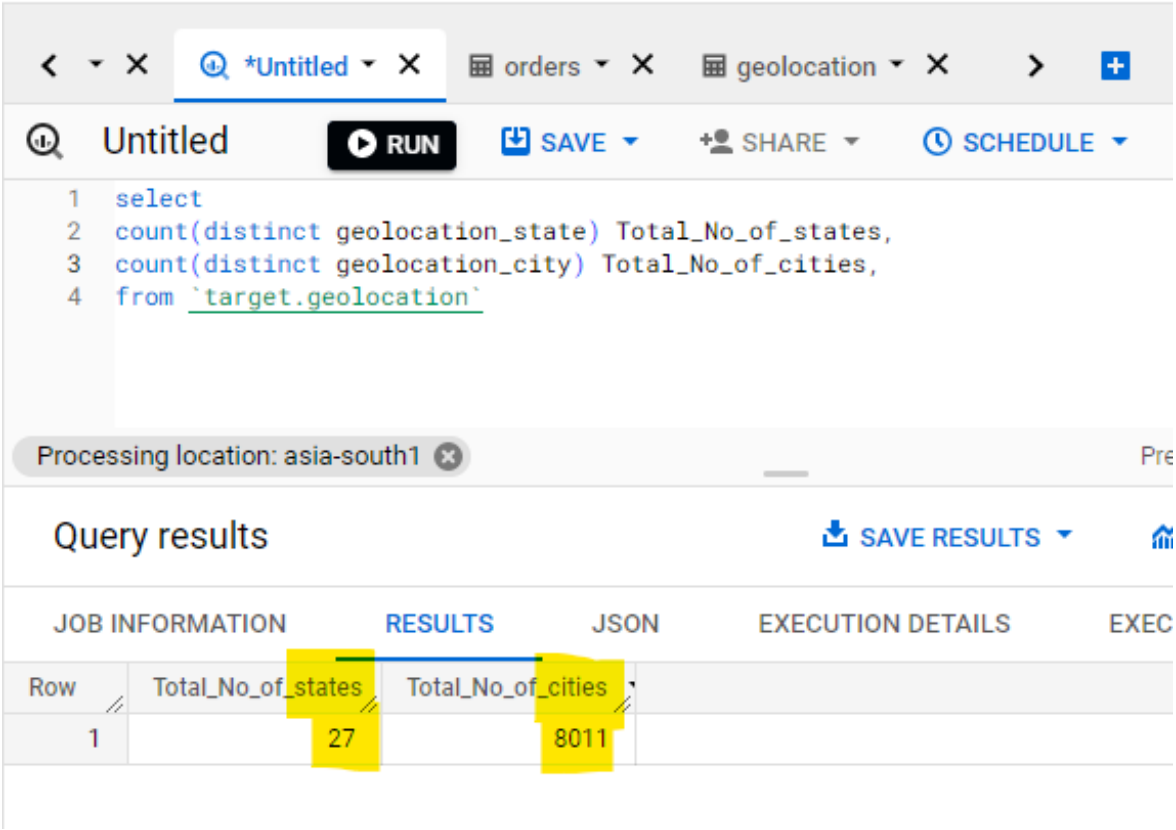
   b. Get the time range between which the orders were placed.



- The orders in the data set were placed in the time interval - 4th september , 2016 to 17th October,2018.

Dataset does not consists of data for entire year for 2016, 2018. For these years , only sparse data corresponding to few months have been provided.

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



There are 27 States and 8011 Cities in the dataset.

2. **In-depth Exploration:**

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

customer | *Untitled | orders

Untitled RUN SAVE SHARE SCHEDULE

```
select * from (select
extract(year from order_purchase_timestamp) Year,
count(order_id) No_of_Orders
from `target.orders`
group by extract(year from order_purchase_timestamp)) x
order by x.year
```

Processing location: asia-south1 Press Alt

Query results SAVE RESULTS EXP

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTIO

Row	Year	No_of_Orders
1	2016	329
2	2017	45101
3	2018	54011

There is a **fast growing trend** observed in the no.of orders placed over the past years.

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

orders *Untitled

Untitled RUN SAVE SHARE SCHEDULE MORE

```
select extract(year from order_purchase_timestamp) Year,
extract(month from order_purchase_timestamp) Month,
count(order_id) No_of_Orders
from `target.orders`
group by 1,2
order by 1,2
```

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH

Row	Year	Month	No_of_Orders
13	2017	10	4631
14	2017	11	7544
15	2017	12	5673
16	2018	1	7269
17	2018	2	6728

Results per page:

There was a mothly seasonality in the no.of orders , which shows an increasing trend and reached to its peak with with 7544 orders in the month of November,2017 and there by there is a slight drop in the seasonality till August 2018 and there was steep drop thereafter.

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

Untitled RUN SAVE SHARE SCHEDULE

```
with table1 as (select case
when Hour between 0 and 6 then 'Dawn'
when Hour between 7 and 12 then 'Mornings'
when Hour between 13 and 18 then 'Afternoon'
Else 'Night'
end Timings from
(select
extract(HOUR FROM order_purchase_timestamp) Hour
from `target.orders`
order by extract(HOUR FROM order_purchase_timestamp)) X )

select  Timings , count(Timings) No_of_Orders from table1
group by Timings
order by 2 desc
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	Timings	No_of_Orders
1	Afternoon	38135
2	Night	28331
3	Mornings	27733
4	Dawn	5242

At **Afternoon** Times, the orders that were placed were **high**.
At **Mornings and Nights** , the orders were optimum.
At **Dawns** the orders placed were **low.**

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

```
with table1 as (select extract(Year from o.order_purchase_timestamp) Year,
extract(month from o.order_purchase_timestamp) Month,c.customer_state State,
o.order_id orders,
from target.customers c
inner join target.orders o
on c.customer_id=o.customer_id)
select Year,Month,State, count(orders) No_of_Orders from table1 group by 1,2,3 order by 1,2,3
```

Processing location: asia-south1

Query results

Row	Year	Month	State	No_of_Orders
22	2016	10	SC	11
23	2016	10	SE	3
24	2016	10	SP	113
25	2016	12	PR	1
26	2017	1	AC	2

In 2016, Highest no.of orders(113) were placed from SP state in October(10) month.
In 2017, Highest no.of orders(2357) were placed from SP state in December(12) month.
In 2018, Highest no.of orders(3253) were placed from SP state in August(8) month.
Of all the states SP is placing highest no.of orders each year.

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

Untitled RUN SAVE SHARE

```
select
customer_state State,
count(customer_id) No_of_Customers
from `target.customers`
group by 1 order by 2 desc
```

Processing location: asia-south1

Query results SAVE R

JOB INFORMATION RESULTS JSON EXECUTION DET

Row	State	No_of_Customers
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045

SP state has highest no.of customers - 41746
RR state has lowest no.of customers - 46

4. **Impact on Economy: Analyze 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.

customers | payments | *Untitled 2

Untitled 2 — RUN | SAVE | SHARE | SCHEDULE | MORE | Query

```
with table1 as (select extract(year from order_purchase_timestamp ) Year,
extract(month from order_purchase_timestamp ) Month,
payment_value
from `target.orders` o
inner join target.payments p
on o.order_id=p.order_id)
select Year, round(sum(payment_value)) sales
from table1
where Month between 01 and  08
group by table1.year
order by 1
```

Press Alt+F1 for accessibility optio

Query results — SAVE RESULTS | EXPLORE DATA

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

Row	Year	sales
1	2017	3669022.0
2	2018	8694734.0

The sales has more than doubled in 2018 as compared to that of 2017.

Untitled

RUN SAVE SHARE SCHEDULE MORE

```
with table1 as (select extract(year from order_purchase_timestamp ) Year,
extract(month from order_purchase_timestamp ) Month,
payment_value
from `target.orders` o
inner join target.payments p
on o.order_id=p.order_id),
table2 as (select Year, round(sum(payment_value)) sales
from table1
where Month between 01 and  08
group by table1.year
order by 1),
table3 as  (select (select sales  from table2 where table2.year=2017) s1,(select sales  f
table2 where table2.year=2018) s2 from table2),
table4 as (select  distinct s1,s2 from table3)
select round(((s2-s1)/s1)*100) Percentage_increase_in_sales from table4
```

Processing location: asia-south1

Press Alt+F1 for accessibi

Query results

SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREV

Row	Percentage_increase_in_sales
1	137.0

The sales increased by 137 percent for year 2018 compared to as that of 2017.

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

Untitled RUN SAVE SHARE SCHEDULE

```
select customer_state State,
round(sum(payment_value)) Total_Sales_from_eachState ,
round(avg(payment_value)) Avg_Sales_from_eachState
from `target.customers` c
inner join `target.orders` o
on c.customer_id=o.customer_id
inner join target.payments p
on o.order_id=p.order_id
group by customer_state
order by 2 desc
```

Processing location: asia-south1

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXEC

Row	State	Total_Sales_from_eachState	Avg_Sales_from_eachState
1	SP	5998227.0	138.0
2	RJ	2144380.0	159.0
3	MG	1872257.0	155.0
4	RS	890899.0	157.0

SP state has the highest Total sales , with total sales = 5998227.
RR state has the Lowest Total sales , with total sales = 10065.

Untitled RUN SAVE SHARE

```
select customer_state State,
round(sum(payment_value)) Total_Sales_from_eachState ,
round(avg(payment_value)) Avg_Sales_from_eachState
from `target.customers` c
inner join `target.orders` o
on c.customer_id=o.customer_id
inner join target.payments p
on o.order_id=p.order_id
group by customer_state
order by 3 desc
```

Processing location: asia-south1

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DET

Row	State	Total_Sales_from_eachState	Avg_Sales_from_eachState
1	PB	141546.0	248.0
2	AC	19681.0	234.0
3	RO	60866.0	233.0
4	AP	16263.0	232.0

PB state has the highest Average sales , with Average sales = 248.0.
SP state has the Lowest Average sales , with Average sales = 138.0.

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

Untitled RUN SAVE SHARE SCHEDULE

```
select customer_state State,
round(sum(freight_value)) Total_freight_from_eachState ,
round(avg(freight_value)) Avg_freight_from_eachState
from `target.customers` c
inner join `target.orders` o
on c.customer_id=o.customer_id
inner join target.order_items   oi
on o.order_id=oi.order_id
group by customer_state
order by 2 desc
```

Processing location: asia-south1

Query results SAVE RESULTS

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECU

Row	State	Total_freight_from_eachState	Avg_freight_from_eachState
1	SP	718723.0	15.0
2	RJ	305589.0	21.0
3	MG	270853.0	21.0

Results per page: 50 1 – 27 of 27

SP state has the highest Total freight , with total freight = 718723.0.
RR state has the Lowest Total freight, with total freight = 2235.0.

Untitled RUN SAVE SHARE SCHEDULE

```
select customer_state State,
round(sum(freight_value)) Total_freight_from_eachState ,
round(avg(freight_value)) Avg_freight_from_eachState
from `target.customers` c
inner join `target.orders` o
on c.customer_id=o.customer_id
inner join target.order_items   oi
on o.order_id=oi.order_id
group by customer_state
order by 3 desc
```

Processing location: asia-south1

Query results SAVE RESULTS

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	State	Total_freight_from_e	Avg_freight_from_ea
1	PB	25720.0	43.0
2	RR	2235.0	43.0
3	RO	11417.0	41.0

Results per page: 50 1 – 27

PB, RR states has the **highest** Average freight , with Average freight = 43.0.
SP state has the **Lowest** Average freight, with Average freight = 15.0

5. **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

*Untitled | orders | payments | *Untitled 2

Untitled RUN SAVE SHARE SCHEDULE MORE

```
select order_id,
extract(date from order_purchase_timestamp) Ordered_date,
extract(date from order_estimated_delivery_date) Expected_Delivery_date,
extract(date from order_delivered_customer_date) Actual_Delivery_date,
date_diff(order_delivered_customer_date,order_purchase_timestamp, day) time_to_deliver ,
date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) as diff_estimated_delivery
from `target.orders`
order by 6 desc
```

Processing location: asia-south1

Query results SAVE RESULTS EXP

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

Row	order_id	Ordered_date	Expected_Delivery_date	Actual_Delivery_date	time_to_deliver	diff_estimated_delivery
1	1b3190b2dfa9d78...	2018-02-23	2018-03-15	2018-09-19	208	188
2	ca07593549f1816...	2017-02-21	2017-03-22	2017-09-19	209	181
3	47b40429ed8cce3...	2018-01-03	2018-01-19	2018-07-13	191	175
4	2fe324febf907e3e...	2017-03-13	2017-04-05	2017-09-19	189	167
5	285ab9426d6982...	2017-03-08	2017-04-06	2017-09-19	194	166

The maximum no.of days taken to deliver an order was 208 and the difference between expected and actual date of delivery is maximum for the same order which is 188 days.

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

Untitled RUN SAVE SHARE SCHEDULE

```
select * from
(   select customer_state,
 round(avg(freight_value),2) Avg_freight_value,
dense_rank() over  (order by avg(freight_value) desc)  DR
from `target.customers` c
inner join `target.orders` o
on c.customer_id=o.customer_id
inner join `target.order_items` oi
on o.order_id=oi.order_id
group by customer_state) x
where DR between 1 and 5
order by DR
```

Query results SAV

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTIC

Row	customer_state	Avg_freight_value	DR
1	RR	42.98	1
2	PB	42.72	2
3	RO	41.07	3
4	AC	40.07	4
5	PI	39.15	5

RP, PB, RO, AC, PI are top 5 states with the **highest average** freight values (42.98, 42.72, 41.07, 40.07, 39.15) respectively.

Untitled RUN SAVE SHARE SCHEDULE

```
select * from
(   select customer_state,
 round(avg(freight_value),2) Avg_freight_value,
dense_rank() over  (order by avg(freight_value) asc)  DR
from `target.customers` c
inner join `target.orders` o
on c.customer_id=o.customer_id
inner join `target.order_items` oi
on o.order_id=oi.order_id
group by customer_state) x
where DR between 1 and 5
order by DR
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EX

Row	customer_state	Avg_freight_value	DR
1	SP	15.15	1
2	PR	20.53	2
3	MG	20.63	3
4	RJ	20.96	4
5	DF	21.04	5

SP, PR, MG, RJ, DF are top 5 states with the **LOWEST average** freight values (15.15, 20.53, 20.63, 20.96, 21.04) respectively.

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

Untitled RUN SAVE SHARE SCHEDULE MORE This query will process 8.32 M

```
select * from
(   select customer_state State,
 round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2) Avg_DeliveryTime,
dense_rank() over (order by avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)) desc)  DR
from `target.customers` c
inner join `target.orders` o
on c.customer_id=o.customer_id
group by customer_state) x
where DR between 1 and 5
order by DR
```

Press Alt+F1 for accessibility

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	State	Avg_DeliveryTime	DR
1	RR	28.98	1
2	AP	26.73	2
3	AM	25.99	3
4	AL	24.04	4
5	PA	23.32	5

RP, AP, AM, AL, PA are top 5 states with the **highest average** DELIVERY TIME with values (28.98, 26.73, 25.99, 24.04, 23.32) respectively.

Untitled RUN SAVE SHARE SCHEDULE MORE Query

```
select * from
(   select customer_state State,
 round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2) Avg_DeliveryTime,
dense_rank() over (order by avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)) )  DR
from `target.customers` c
inner join `target.orders` o
on c.customer_id=o.customer_id
group by customer_state) x
where DR between 1 and 5
order by DR
```

Press Alt+F1 for accessi

Query results SAVE RESULTS EXPLORE DATA

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREVIEW

Row	State	Avg_DeliveryTime	DR
1	SP	8.3	1
2	PR	11.53	2
3	MG	11.54	3
4	DF	12.51	4
5	SC	14.48	5

SP, PR, MG, DF, SC are top 5 states with the **LOWEST average** DELIVERY TIME with values (8.3, 11.53, 11.54, 12.51, 14.48) respectively.

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.

Untitled RUN SAVE SHARE SCHEDULE MORE

```
with table1 as (select * from
(    select customer_state State,
 date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) Dt
from `target.customers` c
inner join `target.orders` o
on c.customer_id=o.customer_id) x
where Dt < 0),
table2 as (select State, round(avg(abs(Dt)),2) Avg_time_ofDeliveryBeforeExpected,
dense_rank() over (order by round(avg(abs(Dt)),2) desc) Dr
from table1
group by table1.State
order by 2 desc)
select table2.State,Avg_time_ofDeliveryBeforeExpected,Dr
from table2
where Dr<=5
order by 3
```

Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS EXECUTION GRAPH PREV

Row	State	Avg_time_ofDeliveryBeforeExpected	Dr
1	RR	23.75	1
2	AP	21.88	2
3	AC	21.54	3
4	AM	20.28	4
5	RO	20.03	5

RR, AP, AC, AM, RO are the top 5 states where the order delivery is really fast as compared to the estimated date of delivery with average time of delivery before expected is (23.75, 21.88, 21.54, 20.28, 20.03) days respectively.

6. **Analysis based on the payments:**

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

Untitled RUN SAVE SHARE SCHEDUL

```
select
extract(month from order_purchase_timestamp) Order_month,
payment_type,
count(o.order_id) No_of_orders
from `target.orders` o
inner join `target.payments` p
on o.order_id=p.order_id
group by 1,2
order by 1,3 desc
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	Order_month	payment_type	No_of_orders
1	1	credit_card	6103
2	1	UPI	1715
3	1	voucher	477
4	1	debit_card	118
5	2	credit_card	6609
6	2	UPI	1723
7	2	voucher	424
8	2	debit_card	82
9	3	credit_card	7707
10	3	UPI	1942

Every month the credit card payments are high and debit card payments are low.

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

Untitled RUN SAVE

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

Query results

JOB INFORMATION RESULTS JSON

Row	payment_installments	No_of_orders
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	5	5234
6	6	3916
7	7	1623

Total **49060** orders have been placed on the basis of payment installments and out of which **18** orders have finished the 24 installments

DATA SET ANALYSIS:

1. Initial exploration of structure and charecteristics of the data

The schema consists of of 8 tables namely:

- Customers -> consists of customer information such as name, id, state , city etc.
- Orders -> consists of orders related information such as order_id, order date and time, order status, expected delivery time, actual delivery time etc.
- Payments -> consists of all the payments related information such as payment value, payment mode, payment installments etc.
- Products -> consists of all the information related to products such as product id, product name, product description, product dimensions etc
- Sellers -> consists of all the information corresponding to to sellers such as seller id, city, state and zipcode of the seller.
- Order_reviews -> consists of the information corresponding to product such as review id, creation date , score , actual comment information etc.
- Order_items -> consists of all the information corresponding to items that were ordered like order id, order item id, product id, seller id, freight value, shipping limit period etc.
- Geolocation -> It consists of all the information corresponding to cities and states of Brazil such as names of cities and states, their zipcode and latitude and longitude infromation.
- Order table is connected to Customers, order items, order reviews and payments.
- Order items table is further connected to sellers and products.
- Sellers is further connected to geolocations.
- Geolocations is connected to customers.

In-depth Exploration:

- The data set consists of information of the Target retail company from 4th september , 2016 to 17th October,2018.

- There are 27 States and 8011 Cities in the dataset.
- There is a fast growing trend observed in the no.of orders placed over the past years.
- There was a mothly seasonality in the no.of orders , which shows an increasing trend and reached to its peak with with 7544 orders in the month of November,2017 and there by there is a slight drop in the seasonality till August 2018 and there was steep drop thereafter.
- At Afternoon Times, the orders that were placed were high.
- At Mornings and Nights , the orders were optimum.
- At Dawns the orders placed were low.

Evolution of E-commerce orders in the Brazil region:

- The evolution of e-commerce started with some humble figures in the 3rd quarter of 2016 and has shown an exponentially increasing trends in the following years of 2017, 2018.
- The sales more than doubled in 2018, compared to those in 2017.
- In 2016, Highest no.of orders(113) were placed from SP state in October(10) month.
- In 2017, Highest no.of orders(2357) were placed from SP state in December(12) month.
- In 2018, Highest no.of orders(3253) were placed from SP state in August(8) month.
- Of all the states SP is placing highest no.of orders each year.
- SP state has highest no.of customers - 41746
- RR state has lowest no.of customers - 46

Impact on Economy:

- Target Retails has done a business which is equal to 16008872 Brazilian currency, which is a whopping amount.

Analysis on sales, freight, and delivery time:

- The totals sales is highest in SP state as there are huge no.of customers.

- The sales increased by 137 percent for year 2018 compared to as that of 2017.
- SP state has the highest Total sales , with total sales = 5998227.
- RR state has the Lowest Total sales , with total sales = 10065.
- PB state has the highest Average sales , with Average sales = 248.0.
- SP state has the Lowest Average sales , with Average sales = 138.0.
- SP state has the highest Total freight , with total freight = 718723.0.
- RR state has the Lowest Total freight, with total freight = 2235.0.
- PB, RR states has the highest Average freight , with Average freight = 43.0.
- SP state has the Lowest Average freight, with Average freight = 15.0
- The maximum no.of days taken to deliver an order was 208 and the difference between expected and actual date of delivery is maximum for the same order which is 188 days.
- RP, PB, RO, AC, PI are top 5 states with the highest average freight values (42.98, 42.72, 41.07, 40.07, 39.15) respectively.
- SP, PR, MG, RJ, DF are top 5 states with the LOWEST average freight values (15.15, 20.53, 20.63, 20.96, 21.04) respectively.
- RP, AP, AM, AL, PA are top 5 states with the highest average DELIVERY TIME with values (28.98, 26.73, 25.99, 24.04, 23.32) respectively.
- SP, PR, MG, DF, SC are top 5 states with the LOWEST average DELIVERY TIME with values (8.3, 11.53, 11.54, 12.51, 14.48) respectively.

- RR, AP, AC, AM, RO are the top 5 states where the order delivery is really fast as compared to the estimated date of delivery with average time of delivery before expected is (23.75, 21.88, 21.54, 20.28, 20.03) days respectively.

Analysis based on the payments:

- Purchases are done using 4 payment types:
- 1.credit_card
- 2.voucher

- **3.debit_card**
- **4.UPI**
- **Every month the credit card payments are high .**
- **Debit card payments are low.**
- **Total no.of orders in the dataset is 99441.**
- **Total 49060 orders have been placed on the basis of payment installments and out of which 18 orders have finished the 24 installments.**

Actionable Insights & Recommendations:

- **A short survey with less than 15 seconds could be sent on the mobile or email or e-commerce site , to know the most popular products, new products that customers want to see on the site , recommendations from customers on the areas of improvement.**
- **The delivery time could be reduced by having a disscussion with the vendors and sellers.**
- **Offers can be rolled out to increase the single shot payment as we could see approx half of the orders were paid in installments.**
- **At each quarter , if there could be a sale period connecting to the festivals of the country could boost the sales by huge amount.**