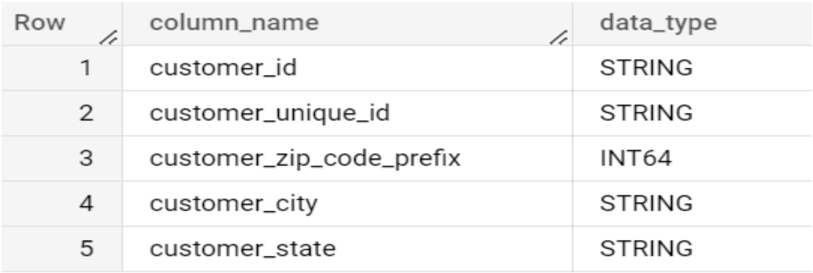
TARGET-SQL

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

1. Data type of columns in a table

- **Query**

```
select column_name, data_type

from `scaler-dsml-381115.Target.INFORMATION_SCHEMA.COLUMNS`

where table_name = 'customers';
```



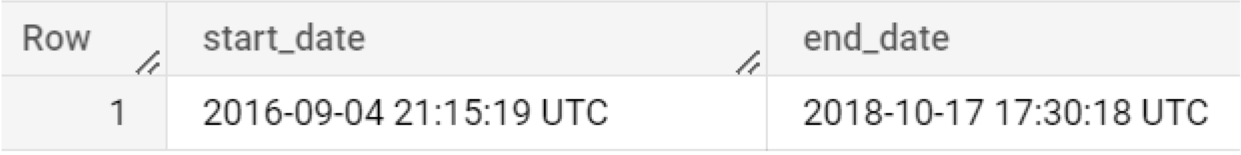
Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

2. Time period for which the data is given

The time period for given data is between year 2016 to 2018. (2016-09-04 to 2018-10-17)

Query:

```
select min(order_purchase_timestamp) as start_date,
max(order_purchase_timestamp) as end_date

FROM `Target.orders`;
```



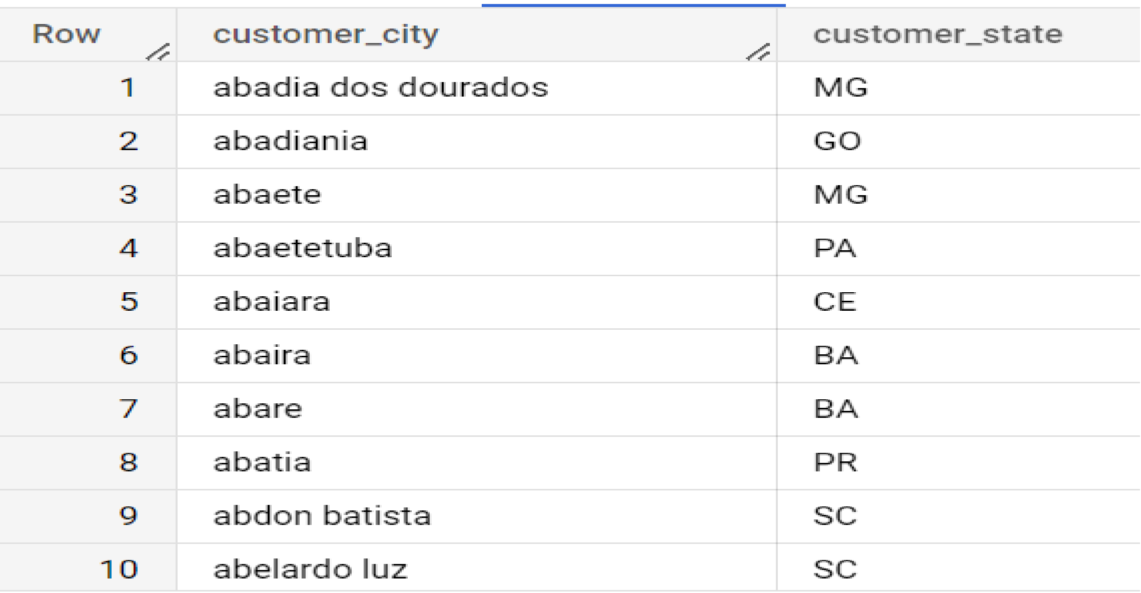
Row	start_date	end_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

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

There are 4119 cities and 27 states from which customers placed orders.

Query:

```
select distinct customer_city FROM `Target.customers` order by customer_city ;

select distinct customer_state FROM `Target.customers`order by customer_state ;
```



Row	customer_city	customer_state
1	abadia dos dourados	MG
2	abadiania	GO
3	abaete	MG
4	abaetetuba	PA
5	abaiara	CE
6	abaira	BA
7	abare	BA
8	abatia	PR
9	abdon batista	SC
10	abelardo luz	SC

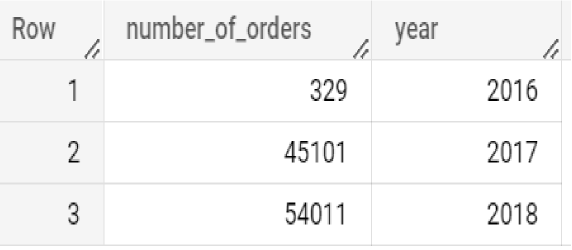
2. In-depth Exploration:

1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

- **There is growing trend on e-commerce in Brazil which can analyzed using increase in number of orders per year.**
- **Number of orders have increased from year 2016 to 2018. (Query 1)**
- **Maximum number of orders are placed between month of January to April.**
- **This is also the period of festivals in Brazil which includes main festival of 'Carnival of Brazil'**
- **Orders are higher in month of July and August which indicates people place more orders in these months. (Query 2)**

Query 1:

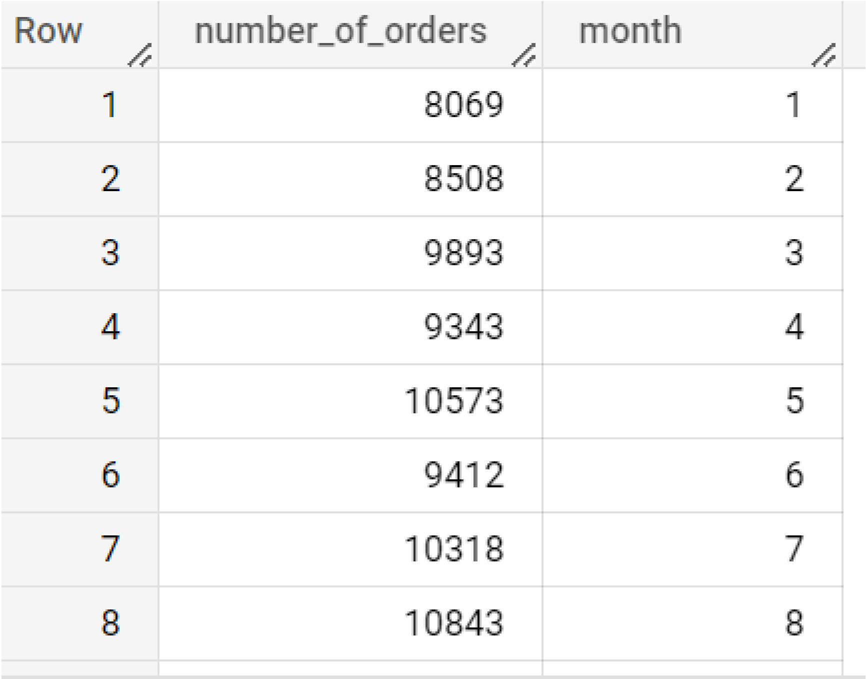
```
select count(order_id) as number_of_orders, extract(year from order_purchase_timestamp) as year

FROM `Target.orders` group by year order by year;
```



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

Query 2:

```
select count(order_id) as number_of_orders,

extract(month from order_purchase_timestamp) as month,

FROM `Target.orders` group by month order by month;
```



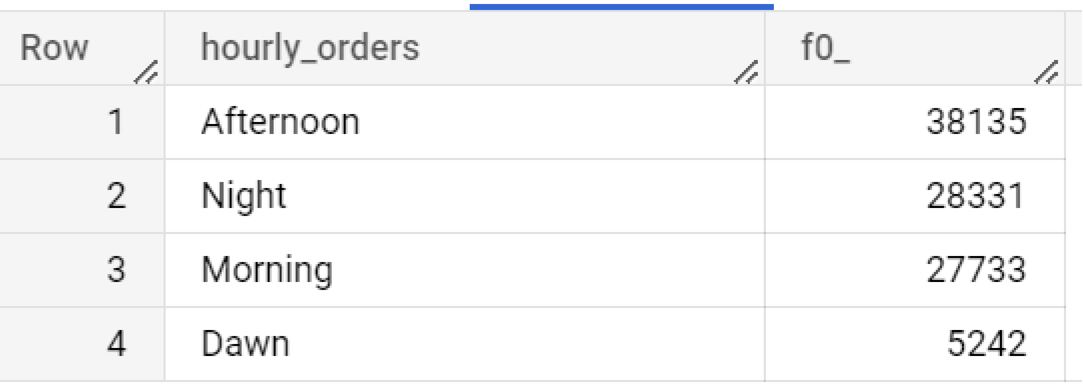
Row	number_of_orders	month
1	8069	1
2	8508	2
3	9893	3
4	9343	4
5	10573	5
6	9412	6
7	10318	7
8	10843	8

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

User tend to buy in afternoon time. Followed by night, morning and dawn respectively.

Query : select hourly_orders,count(*) from

```
(select order_id,extract(hour from order_purchase_timestamp),
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'
  when extract(hour from order_purchase_timestamp)  between 19 and 23 then 'Night'
end as hourly_orders
FROM `Target.orders`) table1
group by  hourly_orders order by count(*) desc;
```



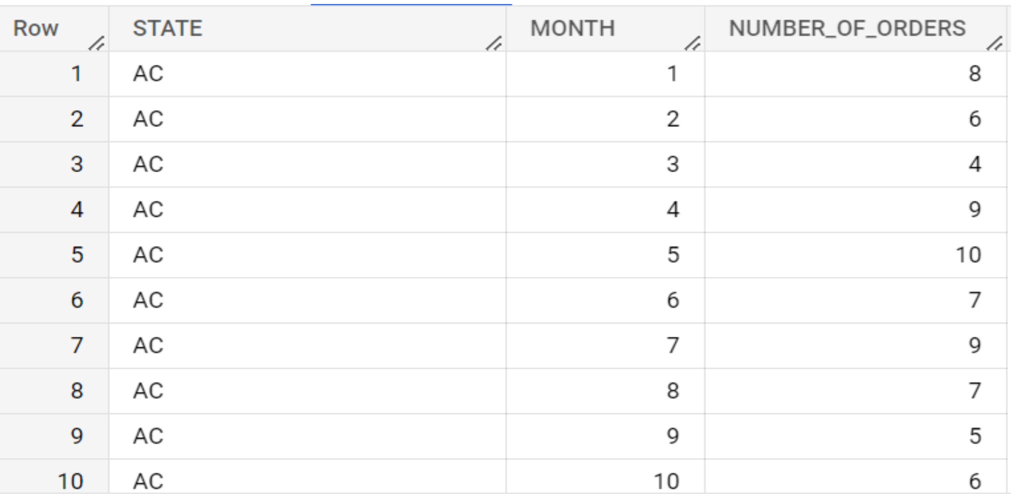
Row	hourly_orders	f0_
1	Afternoon	38135
2	Night	28331
3	Morning	27733
4	Dawn	5242

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

1. Get month on month orders by states

This detail can be found after joining customer and orders table using customer_id field.

```
select state AS STATE, month AS MONTH,count(order_id) as NUMBER_OF_ORDERS from
(select C.customer_id,O.order_id as order_id,C.customer_state as state,
extract(month from o.order_purchase_timestamp) as month
from `Target.customers` C
inner join `Target.orders` O
on C.customer_id = O.customer_id) table1
group by state,month  order by state,month;
```



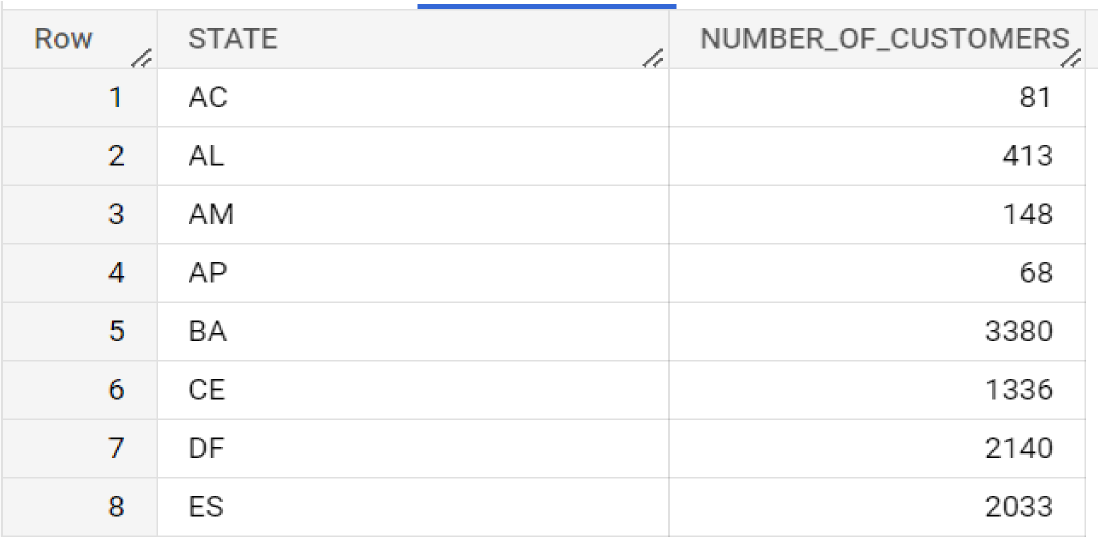
Row	STATE	MONTH	NUMBER_OF_ORDERS
1	AC	1	8
2	AC	2	6
3	AC	3	4
4	AC	4	9
5	AC	5	10
6	AC	6	7
7	AC	7	9
8	AC	8	7
9	AC	9	5
10	AC	10	6

2. Distribution of customers across the states in Brazil

State SP has maximum number of customers- 41746

Query :

```
select state AS STATE,count(customer_id) as NUMBER_OF_CUSTOMERS from
(select C.customer_id,O.order_id as order_id,C.customer_state as state,
extract(month from o.order_purchase_timestamp) as month
from `Target.customers` C
inner join `Target.orders` O
on C.customer_id = O.customer_id) table1
group by state order by state;
```



Row	STATE	NUMBER_OF_CUSTOMERS
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033

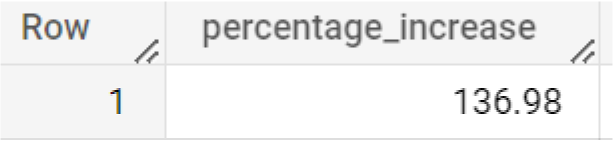
4. Impact on Economy:

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

1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment_value” column in payments table

Query:

```
select

round(((percent_inc_2018-percent_inc_2017)/percent_inc_2017)*100,2) as percentage_increase

from

(select

sum(case

   when month >= 1 and month <= 8 and year =2017 then payment_value

end) as percent_inc_2017,

sum(case

   when month >= 1 and month <= 8 and year =2018 then payment_value

end) as percent_inc_2018

from

(select P.payment_value as payment_value, extract(month FROM O.order_purchase_timestamp) as month,

extract(year FROM O.order_purchase_timestamp) as year

from `Target.orders` O left join `Target.payments` P

on O.order_id = P.order_id

where extract(year FROM O.order_purchase_timestamp) in (2017,2018)) as table1);
```

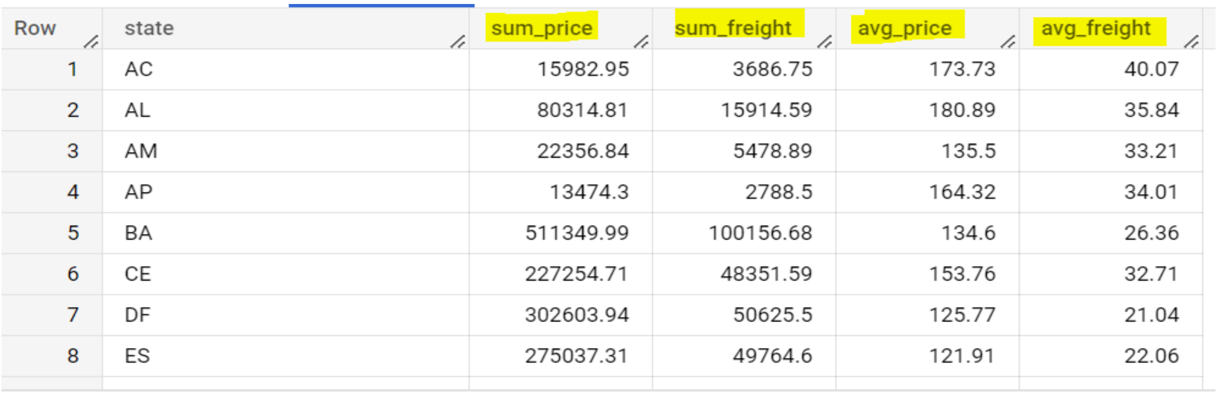


Row	percentage_increase
1	136.98

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

Query:

```
select distinct state, sum_price, sum_freight,a vg_price, avg_freight

from

(select c.customer_state as state,

round(sum(oi.price) over(partition by c.customer_state),2) as sum_price,

round(avg(oi.price) over(partition by c.customer_state),2) as avg_price,

round(sum(oi.freight_value) over(partition by c.customer_state),2) as sum_freight,

round(avg(oi.freight_value) over(partition by c.customer_state),2) as avg_freight,

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) as table1

order by state;
```



Row	state	sum_price	sum_freight	avg_price	avg_freight
1	AC	15982.95	3686.75	173.73	40.07
2	AL	80314.81	15914.59	180.89	35.84
3	AM	22356.84	5478.89	135.5	33.21
4	AP	13474.3	2788.5	164.32	34.01
5	BA	511349.99	100156.68	134.6	26.36
6	CE	227254.71	48351.59	153.76	32.71
7	DF	302603.94	50625.5	125.77	21.04
8	ES	275037.31	49764.6	121.91	22.06

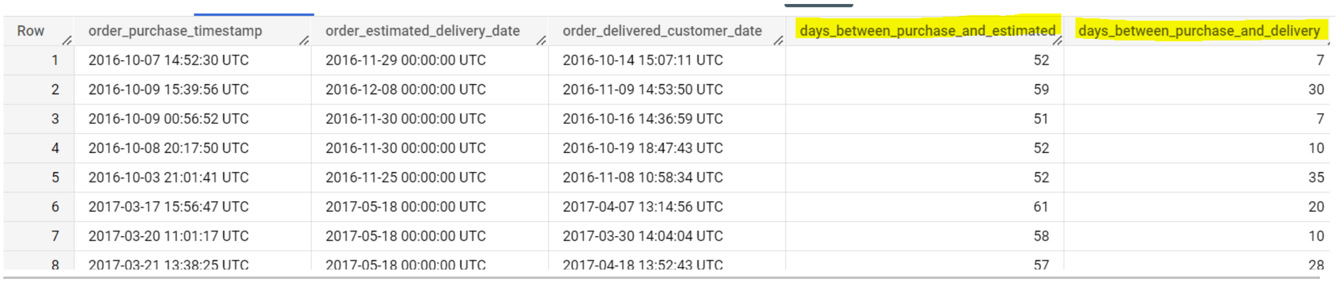
5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery.

Condition order_delivered_customer_date is not null can be added to check only those orders which are delivered.

Query:

```
select order_purchase_timestamp, order_estimated_delivery_date,order_delivered_customer_date,

date_diff(order_estimated_delivery_date, order_purchase_timestamp,day) as days_between_purchase_and_estimated,

date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as days_between_purchase_and_delivery,

from `Target.orders`;

where order_delivered_customer_date is not null;
```



Row	order_purchase_timestamp	order_estimated_delivery_date	order_delivered_customer_date	days_between_purchase_and_estimated	days_between_purchase_and_delivery
1	2016-10-07 14:52:30 UTC	2016-11-29 00:00:00 UTC	2016-10-14 15:07:11 UTC	52	7
2	2016-10-09 15:39:56 UTC	2016-12-08 00:00:00 UTC	2016-11-09 14:53:50 UTC	59	30
3	2016-10-09 00:56:52 UTC	2016-11-30 00:00:00 UTC	2016-10-16 14:36:59 UTC	51	7
4	2016-10-08 20:17:50 UTC	2016-11-30 00:00:00 UTC	2016-10-19 18:47:43 UTC	52	10
5	2016-10-03 21:01:41 UTC	2016-11-25 00:00:00 UTC	2016-11-08 10:58:34 UTC	52	35
6	2017-03-17 15:56:47 UTC	2017-05-18 00:00:00 UTC	2017-04-07 13:14:56 UTC	61	20
7	2017-03-20 11:01:17 UTC	2017-05-18 00:00:00 UTC	2017-03-30 14:04:04 UTC	58	10
8	2017-03-21 13:38:25 UTC	2017-05-18 00:00:00 UTC	2017-04-18 13:52:43 UTC	57	28

2. Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

- time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
- diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

```
select order_purchase_timestamp,order_estimated_delivery_date,order_delivered_customer_date,

date_diff(order_delivered_customer_date,order_purchase_timestamp,day) as time_to_delivery,

date_diff(order_delivered_customer_date,order_estimated_delivery_date,day) as diff_estimated_delivery

from `Target.orders`

where order_delivered_customer_date is not null;
```

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

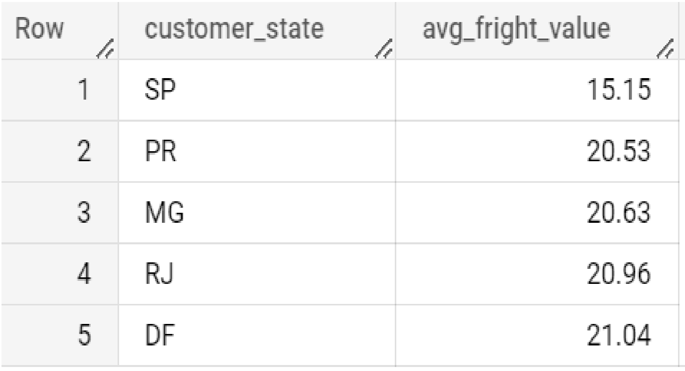
Query:

```
select  c.customer_state,

round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2)as time_to_delivery,

round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)),2) as diff_estimated_delivery,

round(avg(oi.freight_value),2) avg_fright_value

from `Target.orders` o inner join `Target.customers` c

on c.customer_id = o.customer_id

inner join `Target.order_items` oi on

oi.order_id = o.order_id

group by c.customer_state;
```

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

a) lowest average freight value

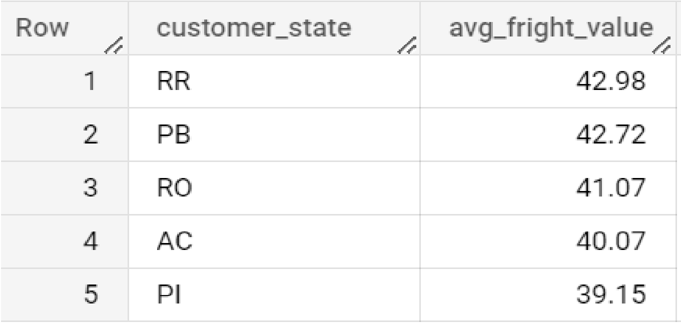
```
select  c.customer_state as state,

round(avg(oi.freight_value),2) avg_fright_value

from `Target.orders` o inner join `Target.customers` c

on c.customer_id = o.customer_id

inner join `Target.order_items` oi on

oi.order_id = o.order_id

group by c.customer_state

order by avg_fright_value desc limit 5;
```



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

b) highest average freight value

```
select  c.customer_state as state, round(avg(oi.freight_value),2) avg_fright_value

from `Target.orders` o inner join `Target.customers` c

on c.customer_id = o.customer_id inner join `Target.order_items` oi on

oi.order_id = o.order_id group by c.customer_state order by avg_fright_value limit 5;
```

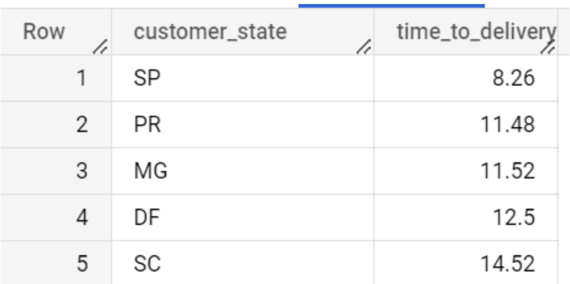


Row	customer_state	avg_fright_value
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

Top 5 states with highest/lowest average time to delivery

a) Highest average time to delivery

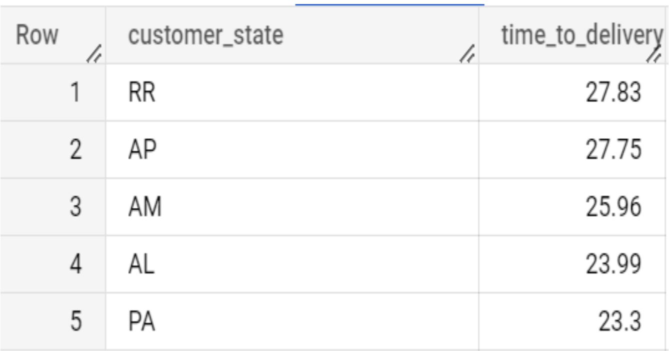
```
select  c.customer_state as state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2)as time_to_
delivery from `Target.orders` o inner join `Target.customers` c  on c.customer_id = o.customer_id

inner join `Target.order_items` oi on oi.order_id = o.order_id

group by c.customer_state order by time_to_delivery limit 5;
```



Row	customer_state	time_to_delivery
1	SP	8.26
2	PR	11.48
3	MG	11.52
4	DF	12.5
5	SC	14.52

b) Lowest average time to delivery

```
select  c.customer_state as state,

round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2)as time_to_delivery

from `Target.orders` o inner join `Target.customers` c

on c.customer_id = o.customer_id

inner join `Target.order_items` oi on

oi.order_id = o.order_id

group by c.customer_state

order by time_to_delivery desc limit 5;
```

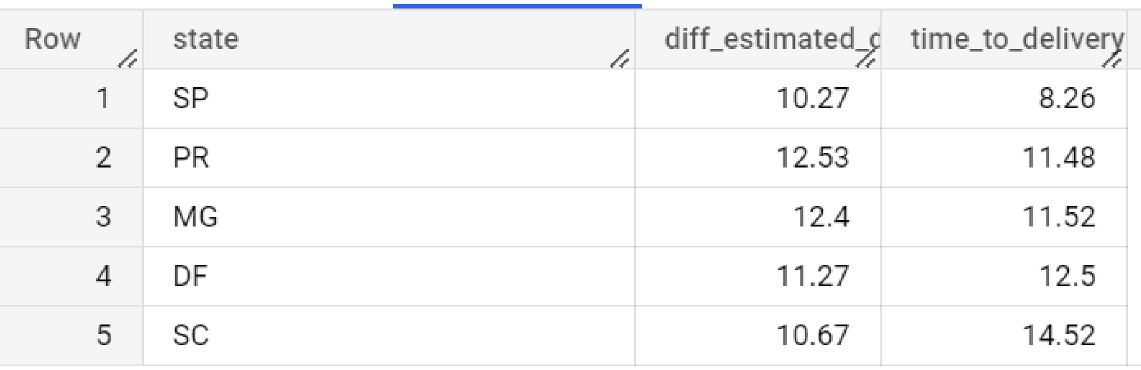


Row	customer_state	time_to_delivery
1	RR	27.83
2	AP	27.75
3	AM	25.96
4	AL	23.99
5	PA	23.3

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

So fast delivery compared to estimated date

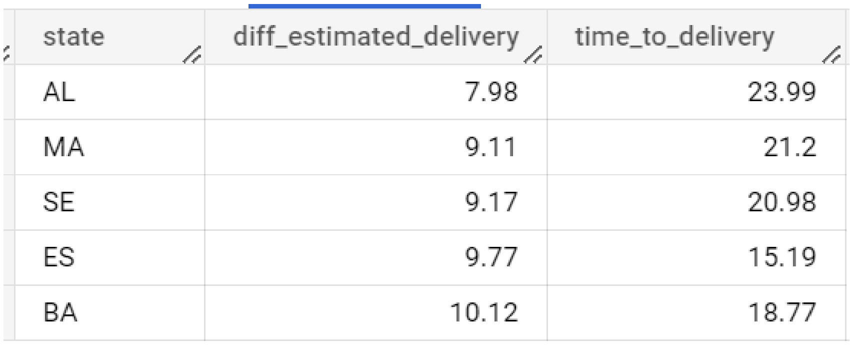
```
select  c.customer_state as state,

round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)),2) as diff_estimated_delivery,

round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2)as time_to_delivery

from `Target.orders` o inner join `Target.customers` c

on c.customer_id = o.customer_id inner join `Target.order_items` oi on

oi.order_id = o.order_id

group by c.customer_state order by time_to_delivery  limit 5;
```



Row	state	diff_estimated_c	time_to_delivery
1	SP	10.27	8.26
2	PR	12.53	11.48
3	MG	12.4	11.52
4	DF	11.27	12.5
5	SC	10.67	14.52

c) Not so fast delivery compared to estimated date

```
select  c.customer_state as state,

round(avg(date_diff(order_estimated_delivery_date,order_delivered_customer_date,day)),2) as diff_estimated_delivery

from `Target.orders` o inner join `Target.customers` c  on c.customer_id = o.customer_id

inner join `Target.order_items` oi on oi.order_id = o.order_id

group by c.customer_state order by diff_estimated_delivery desc limit 5;
```



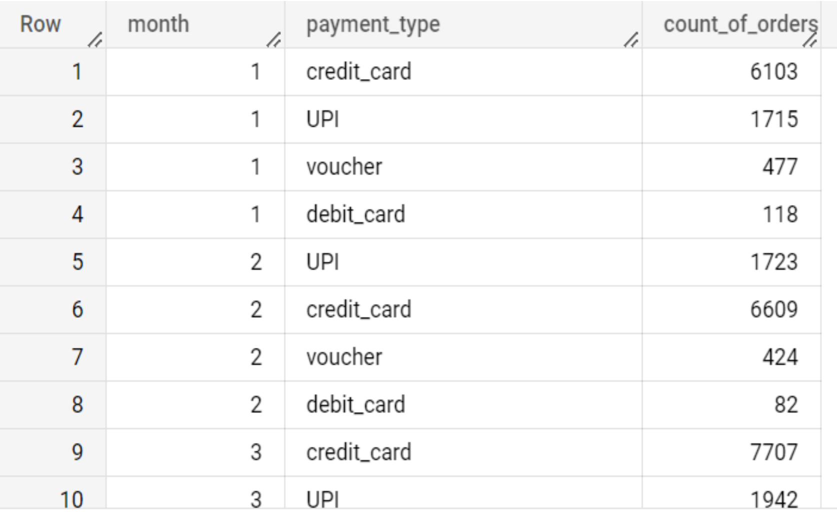
state	diff_estimated_delivery	time_to_delivery
AL	7.98	23.99
MA	9.11	21.2
SE	9.17	20.98
ES	9.77	15.19
BA	10.12	18.77

6. Payment type analysis:

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

Query:

```
select month,payment_type,count(order_id) as count_of_orders from

(select o.order_id as order_id,extract( month FROM o.order_purchase_timestamp) as month,p.payment_type as payment_type

from `Target.orders` o

inner join `Target.payments` p

on o.order_id = p.order_id)

group by month,payment_type

order by month;
```

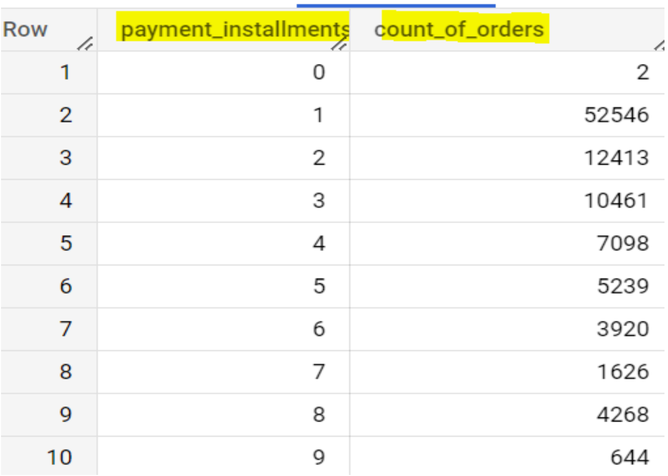


Row	month	payment_type	count_of_orders
1	1	credit_card	6103
2	1	UPI	1715
3	1	voucher	477
4	1	debit_card	118
5	2	UPI	1723
6	2	credit_card	6609
7	2	voucher	424
8	2	debit_card	82
9	3	credit_card	7707
10	3	UPI	1942

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

Query:

```
select payment_installments, count(order_id) as count_of_orders from `Target.payments`

group by payment_installments;
```



Row	payment_installments	count_of_orders
1	0	2
2	1	52546
3	2	12413
4	3	10461
5	4	7098
6	5	5239
7	6	3920
8	7	1626
9	8	4268
10	9	644

7. Actionable Insights

- Data shared contains total 99441 customers who placed order between 2016-2018.
- Customers are from 4119 different cities and 27 different states.

- Insurance and services, Fashion Children's Clothing, PC Gamer are low selling categories. Product belonging to this category can been given more discount to increase sell.

- People are buying products in 0-3 installments. Bank offers should be provided on credit card purchases.

- After festival season people are shopping less. During that period more discount should be given on products to attract customers. Clearance sales should be organized during this period.

- During June to August period new product should be launched. In month

- São Paulo state has the highest numbers of customers -41746.
  Roraima, Amapa has lowest sells. So, need to focus on these states to increase sales.

- Customer tend to buy more in afternoon as compared night, morning. Low selling products can be shown in to recommendation to customers with discounts in afternoon time period.

- Product category PC's have highest average freight value followed by ELECTRICES 2.
- Product categories bed table bath, HEALTH BEAUTY, sport leisure, Furniture Decoration, computer accessories are top selling categories.

- People make payment mostly from credit card followed by UPI and vouchers.
- There are 32951 different products available in Target with 73 product categories.

8. Recommendations

- The average delivery days are 16 days which can be reduced as people will prefer faster ecommerce platform.

- States like Roraima, Amapa, Acre are from north Brazil region which have low customers.
  Increasing business in north Brazil region would improve product sell.

- It can be seen that sales are decreasing after August to December. So, giving additional discounts on products will attract customers to buy products and increase sell.

- The average approval time is 0.26 after order is placed. This can be reduced so that final delivery time is minimized.