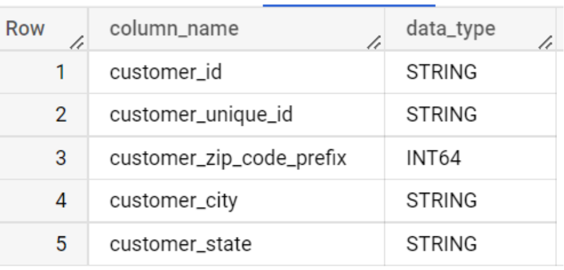
Business Case: SQL

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
    2. Time period for which the data is given
    3. Cities and States covered in the dataset

Solution :

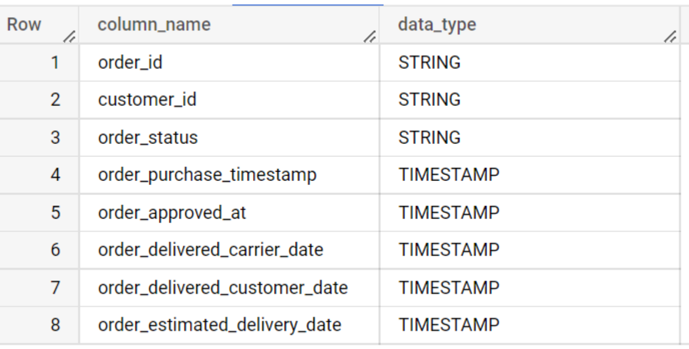
- Data type of columns in a table

```
SELECT column_name, data_type
FROM `<CompanySchema>.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

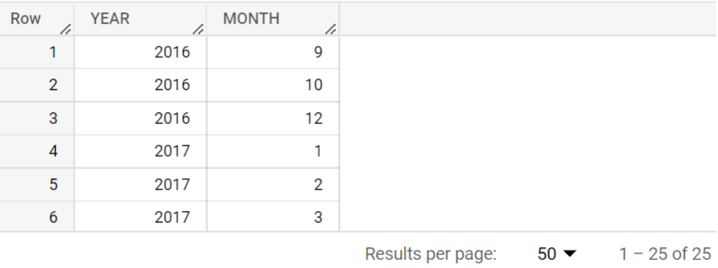
```
SELECT column_name, data_type
FROM `<CompanySchema>.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name="orders";
```



Row	column_name	data_type
1	order_id	STRING
2	customer_id	STRING
3	order_status	STRING
4	order_purchase_timestamp	TIMESTAMP
5	order_approved_at	TIMESTAMP
6	order_delivered_carrier_date	TIMESTAMP
7	order_delivered_customer_date	TIMESTAMP
8	order_estimated_delivery_date	TIMESTAMP

- Time period for which the data is given

```
select distinct EXTRACT(YEAR from order_purchase_timestamp) YEAR,
EXTRACT(MONTH from order_purchase_timestamp) MONTH
from <CompanySchema>.orders
order by YEAR,MONTH;
```

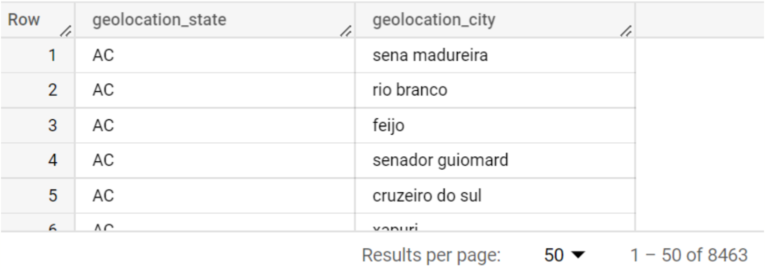


Row	YEAR	MONTH
1	2016	9
2	2016	10
3	2016	12
4	2017	1
5	2017	2
6	2017	3

```
** Data Time Period : Sept,2016 to Oct,2018
```

- Cities and States covered in the dataset

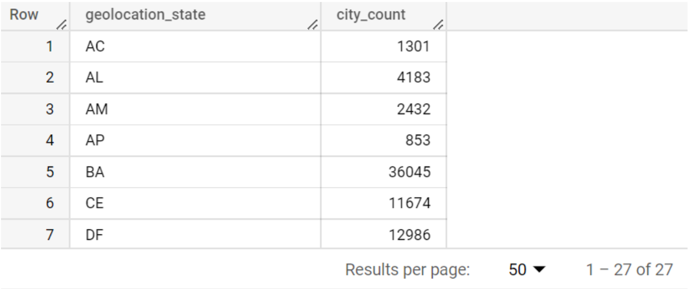
```
select distinct geolocation_state,geolocation_city from <CompanySchema>.geol
ocation
order by geolocation_state;
```



Row	geolocation_state	geolocation_city
1	AC	sena madureira
2	AC	rio branco
3	AC	feijo
4	AC	senador guiomard
5	AC	cruzeiro do sul
6	AC	xapuri

Results per page: 50 ▾ 1 – 50 of 8463

```
select distinct geolocation_state,
count(geolocation_city) as city_count from <CompanySchema>.geolocation
group by geolocation_state
order by geolocation_state;
```



Row	geolocation_state	city_count
1	AC	1301
2	AL	4183
3	AM	2432
4	AP	853
5	BA	36045
6	CE	11674
7	DF	12986

Results per page: 50 ▾ 1 – 27 of 27

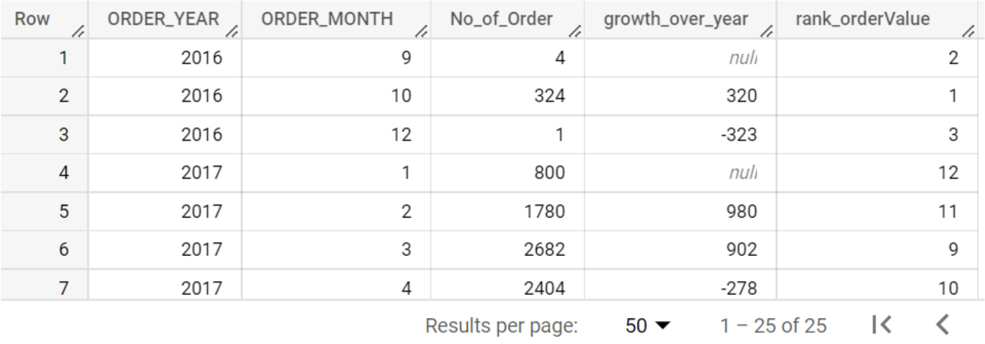
2. In-depth Exploration:

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

Solution :

- 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?

```
select *,
No_of_Order-
LAG(No_of_Order,1) over(partition by ORDER_YEAR order by ORDER_MONTH asc) as gro
wth_over_year,
dense_rank() over(partition by ORDER_YEAR order by No_of_Order desc) as rank_ord
erValue,
from
(select ORDER_YEAR,ORDER_MONTH,count(order_id) No_of_Order from
(select distinct order_id,EXTRACT(YEAR  from order_purchase_timestamp) as ORDER_
YEAR,
EXTRACT(MONTH  from order_purchase_timestamp) AS ORDER_MONTH
from <CompanySchema>.orders) t
group by ORDER_YEAR,ORDER_MONTH
ORDER BY ORDER_YEAR,ORDER_MONTH) t2
ORDER BY ORDER_YEAR,ORDER_MONTH;
```



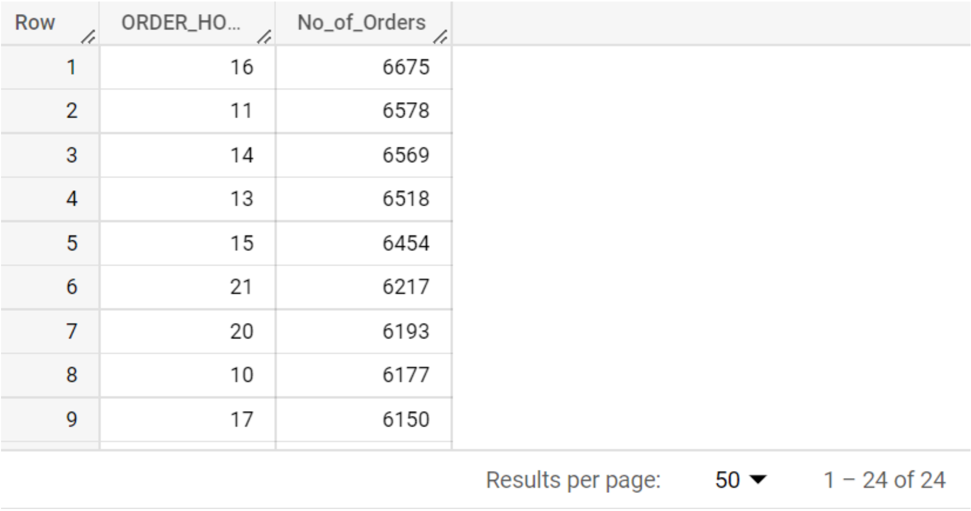
Row	ORDER_YEAR	ORDER_MONTH	No_of_Order	growth_over_year	rank_orderValue
1	2016	9	4	*null*	2
2	2016	10	324	320	1
3	2016	12	1	-323	3
4	2017	1	800	*null*	12
5	2017	2	1780	980	11
6	2017	3	2682	902	9
7	2017	4	2404	-278	10

** No. of orders is surely increasing over months with some exceptional dips for few months in between. Positive growth_over_year shows No. of Orders getting increased over month.

Checking the rank of No. of orders for each month gives us an idea that order has been on peak around September to November month every year, though for 2018 it’s not the same because it looks like data is not complete for 2018 end of the month in this dataset.

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

```
select * from
(select EXTRACT(HOUR  from order_purchase_timestamp) as ORDER_HOUR,
count(distinct order_id) as No_of_Orders
from <CompanySchema>.orders
group by EXTRACT(HOUR  from order_purchase_timestamp))
order by No_of_Orders desc;
```



Row	ORDER_HO...	No_of_Orders
1	16	6675
2	11	6578
3	14	6569
4	13	6518
5	15	6454
6	21	6217
7	20	6193
8	10	6177
9	17	6150

** No. of Orders is highest at around 16:00 hour which is Afternoon time, however, No. of orders pick up at around 10 in the morning and till 22:00 – 23:00 hour the volume of order is high.

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

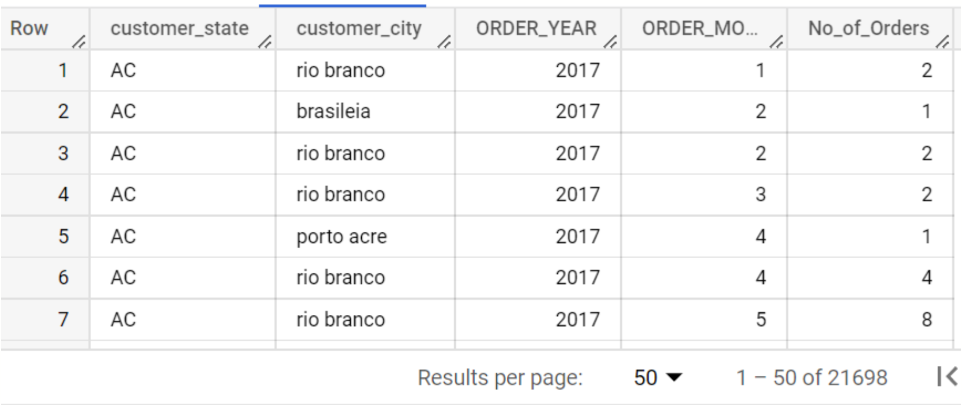
   1. Get month on month orders by region, states

2. How are customers distributed in Brazil

Solution :

- Get month on month orders by region, states

```
select distinct customer_state, customer_city, ORDER_YEAR, ORDER_MONTH,
count(distinct order_id) over(partition by customer_state, customer_city,
ORDER_YEAR,ORDER_MONTH) No_of_Orders FROM
(select c.customer_state, c.customer_city, order_id,
EXTRACT(YEAR  from order_purchase_timestamp) as ORDER_YEAR,
EXTRACT(MONTH  from order_purchase_timestamp) AS ORDER_MONTH
from <CompanySchema>.customers c
left join <CompanySchema>.orders o
on c.customer_id=o.customer_id)
order by customer_state,ORDER_YEAR,ORDER_MONTH,customer_city;
```

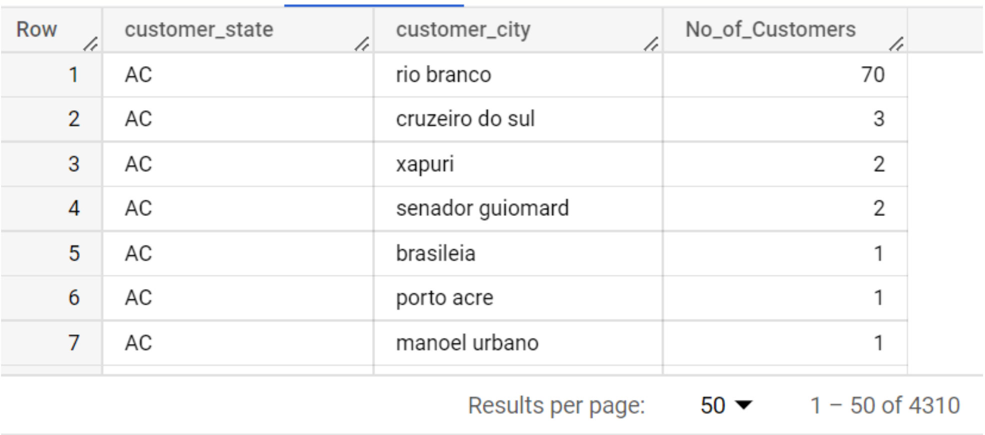


Row	customer_state	customer_city	ORDER_YEAR	ORDER_MO...	No_of_Orders
1	AC	rio branco	2017	1	2
2	AC	brasileia	2017	2	1
3	AC	rio branco	2017	2	2
4	AC	rio branco	2017	3	2
5	AC	porto acre	2017	4	1
6	AC	rio branco	2017	4	4
7	AC	rio branco	2017	5	8

Results per page: 50 ▼ 1 – 50 of 21698 |<

- How are customers distributed in Brazil

```
select customer_state,customer_city,
count(distinct customer_id) No_of_Customers
from <CompanySchema>.customers
group by customer_state,customer_city
order by customer_state,No_of_Customers desc;
```



Row	customer_state	customer_city	No_of_Customers
1	AC	rio branco	70
2	AC	cruzeiro do sul	3
3	AC	xapuri	2
4	AC	senador guiomard	2
5	AC	brasileia	1
6	AC	porto acre	1
7	AC	manoel urbano	1

Results per page: 50 ▼ 1 – 50 of 4310

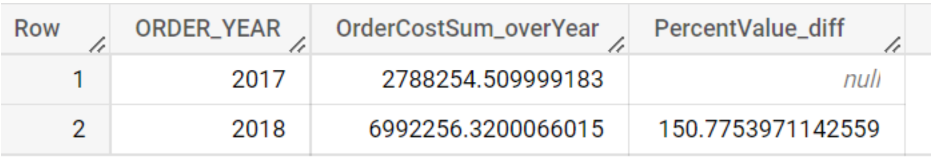
4. Impact on Economy: Analyze the money movemented by e-commerce by looking at order prices, freight and others.
    1. Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only)

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

Solution :

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

```
select ORDER_YEAR, sum(Order_cost) OrderCostSum_overYear,
(sum(Order_cost)-
LAG(sum(Order_cost)) over(order by sum(Order_cost)))*100/LAG(sum(Order_cost)
) over(order by sum(Order_cost)) PercentValue_diff
from
(select distinct order_id,
EXTRACT(YEAR from shipping_limit_date) ORDER_YEAR, price as Order_cost
from <CompanySchema>.order_items
where EXTRACT(MONTH from shipping_limit_date) between 1 and 8
and EXTRACT(YEAR from shipping_limit_date) in (2017,2018))
group by ORDER_YEAR
order by ORDER_YEAR;
```

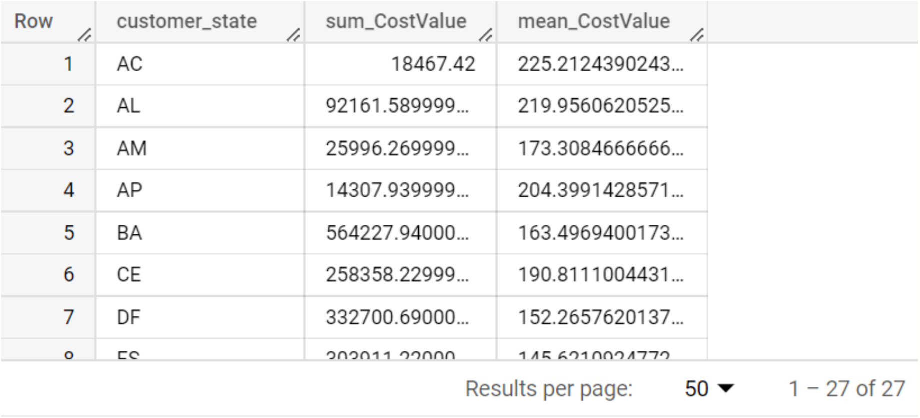


Row	ORDER_YEAR	OrderCostSum_overYear	PercentValue_diff
1	2017	2788254.509999183	null
2	2018	6992256.3200066015	150.7753971142559

```
** For the months between Jan to July, % increase in sum of cost of orders
from 2017 to 2018 is around 150%.
```

- Mean & Sum of price and freight value by customer state

```
select customer_state,sum(price+freight_value) sum_CostValue,
avg(price+freight_value) mean_CostValue from
(select distinct customer_state,oi.order_id,price,freight_value from
<CompanySchema>.customers c inner join <CompanySchema>.orders o
on c.customer_id=o.customer_id
inner join <CompanySchema>.order_items oi
on o.order_id=oi.order_id)
group by customer_state
order by customer_state;
```



Row	customer_state	sum_CostValue	mean_CostValue
1	AC	18467.42	225.2124390243...
2	AL	92161.589999...	219.9560620525...
3	AM	25996.269999...	173.3084666666...
4	AP	14307.939999...	204.3991428571...
5	BA	564227.94000...	163.4969400173...
6	CE	258358.22999...	190.8111004431...
7	DF	332700.69000...	152.2657620137...
8	ES	303911.22000...	145.6210924772...

Results per page: 50 ▼ 1 – 27 of 27

```
** For different customer states, sum and mean value for Cost (Price +
Freight value) is calculated as above.
```

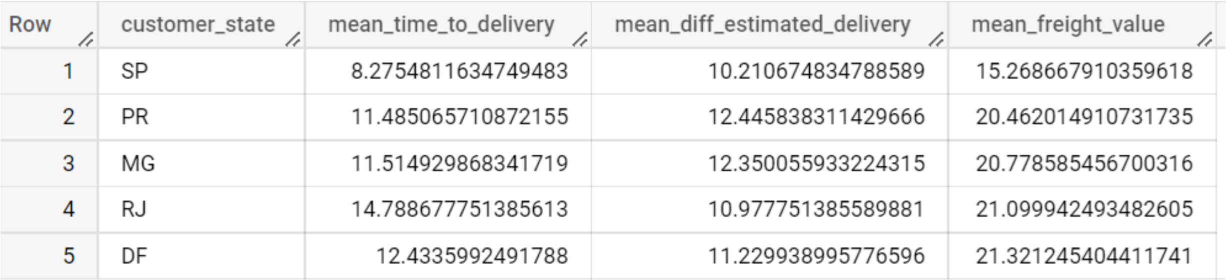
5. Analysis on sales, freight and delivery time
   1. Calculate days between purchasing, delivering and estimated delivery
   2. Create columns:
      - time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
      - diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date
   3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
   4. Sort the data to get the following:

      1. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
      2. Top 5 states with highest/lowest average time to delivery
      3. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Solution :

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

```
select customer_state, avg(time_to_delivery) mean_time_to_delivery,
avg(diff_estimated_delivery) mean_diff_estimated_delivery,
avg(freight_value) mean_freight_value from
(select distinct t.*,freight_value from
(select distinct customer_state,o.order_id,order_purchase_timestamp,order_es
timated_delivery_date,order_delivered_customer_date,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) t
ime_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date,D
AY) diff_estimated_delivery
from <CompanySchema>.orders o
join <CompanySchema>.customers c
on o.customer_id=c.customer_id) t
join <CompanySchema>.order_items oi
on t.order_id=oi.order_id)
group by customer_state
order by mean_freight_value
limit 5;
```



Row	customer_state	mean_time_to_delivery	mean_diff_estimated_delivery	mean_freight_value
1	SP	8.2754811634749483	10.210674834788589	15.268667910359618
2	PR	11.485065710872155	12.445838311429666	20.462014910731735
3	MG	11.514929868341719	12.350055933224315	20.778585456700316
4	RJ	14.788677751385613	10.977751385589881	21.099942493482605
5	DF	12.4335992491788	11.229938995776596	21.321245404411741

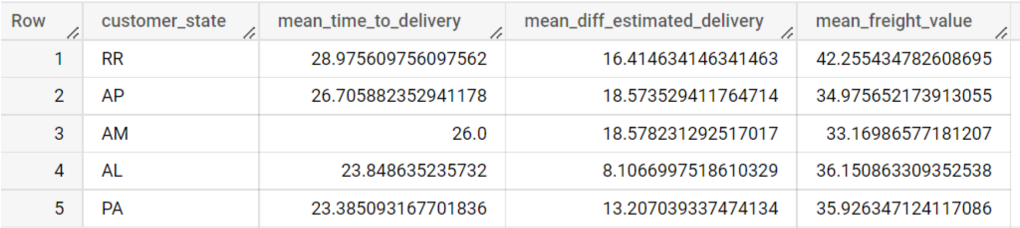
```
** Mean time to delivery is calculated as average of ‘order delivered to customer’ date minus ‘order
purchase date’ over different customer state.
```

Mean difference of estimated delivery is calculated as average of 'order estimated delivery date' minus 'order delivered customer date' over different customer state.

Also, Mean Freight value is calculated over each customer state and is ordered by the same in ascending and data limited to count 5.

- Top 5 states with highest/lowest average time to delivery

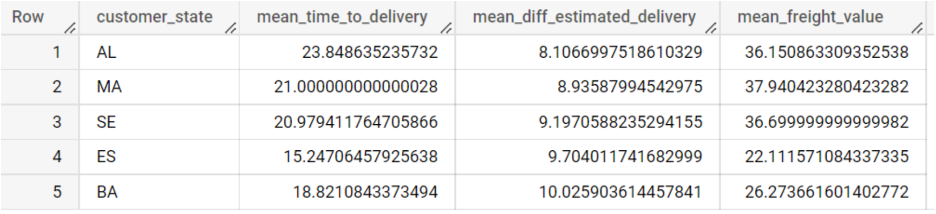
```
select customer_state, avg(time_to_delivery) mean_time_to_delivery,
avg(diff_estimated_delivery) mean_diff_estimated_delivery,avg(freight_value)
 mean_freight_value from
(select distinct t.*,freight_value from
(select distinct customer_state,o.order_id,order_purchase_timestamp,order_es
timated_delivery_date,order_delivered_customer_date,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) t
ime_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date,D
AY) diff_estimated_delivery
from <CompanySchema>.orders o
join <CompanySchema>.customers c
on o.customer_id=c.customer_id) t
join <CompanySchema>.order_items oi
on t.order_id=oi.order_id)
group by customer_state
order by mean_time_to_delivery desc
limit 5;
```



Row	customer_state	mean_time_to_delivery	mean_diff_estimated_delivery	mean_freight_value
1	RR	28.975609756097562	16.414634146341463	42.255434782608695
2	AP	26.705882352941178	18.573529411764714	34.975652173913055
3	AM	26.0	18.578231292517017	33.16986577181207
4	AL	23.848635235732	8.1066997518610329	36.150863309352538
5	PA	23.385093167701836	13.207039337474134	35.926347124117086

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

```
select customer_state, avg(time_to_delivery) mean_time_to_delivery,
avg(diff_estimated_delivery) mean_diff_estimated_delivery,
avg(freight_value) mean_freight_value from
(select distinct t.*,freight_value from
(select distinct customer_state,o.order_id,order_purchase_timestamp,order_es
timated_delivery_date,order_delivered_customer_date,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) t
ime_to_delivery,
TIMESTAMP_DIFF(order_estimated_delivery_date,order_delivered_customer_date,D
AY) diff_estimated_delivery
from <CompanySchema>.orders o
join <CompanySchema>.customers c
on o.customer_id=c.customer_id) t
join <CompanySchema>.order_items oi
on t.order_id=oi.order_id)
group by customer_state
order by mean_diff_estimated_delivery
limit 5;
```



Row	customer_state	mean_time_to_delivery	mean_diff_estimated_delivery	mean_freight_value
1	AL	23.848635235732	8.1066997518610329	36.150863309352538
2	MA	21.000000000000028	8.93587994542975	37.940423280423282
3	SE	20.979411764705866	9.1970588235294155	36.699999999999982
4	ES	15.24706457925638	9.704011741682999	22.111571084337335
5	BA	18.8210843373494	10.025903614457841	26.273661601402772

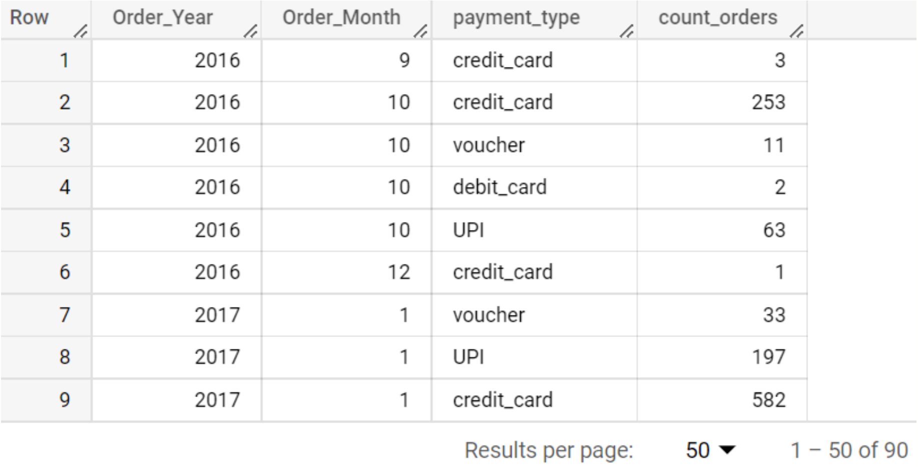
```
** AL customer state has least mean difference of delivered order from
estimated order.
```

6. Payment type analysis:
    1. Month over Month count of orders for different payment types
    2. Distribution of payment installments and count of orders

Solution :

- Month over Month count of orders for different payment types

```sql
select Order_Year,Order_Month,payment_type,count(order_id) count_orders from
(select distinct
EXTRACT(YEAR from order_purchase_timestamp) Order_Year,EXTRACT(MONTH from order_purchase_timestamp) Order_Month,payment_type,o.order_id
from <CompanySchema>.orders o
inner join <CompanySchema>.payments p
on o.order_id=p.order_id)
group by Order_Year,Order_Month,payment_type
order by Order_Year,Order_Month;
```



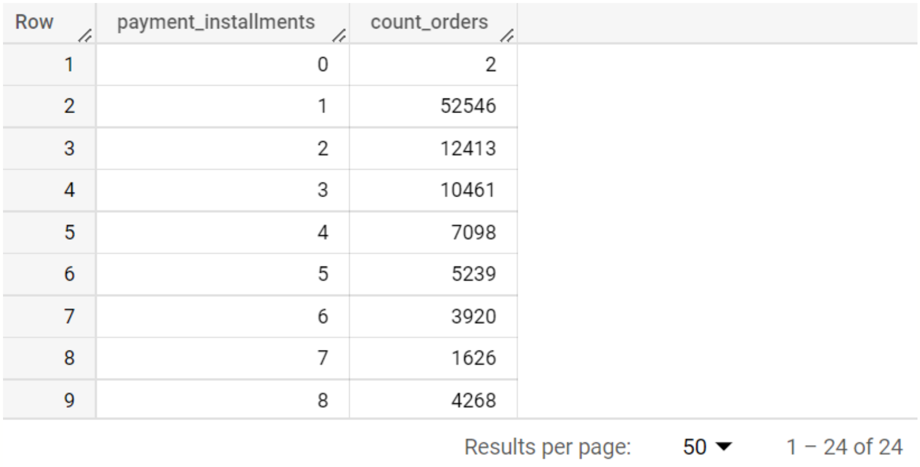
Row	Order_Year	Order_Month	payment_type	count_orders
1	2016	9	credit_card	3
2	2016	10	credit_card	253
3	2016	10	voucher	11
4	2016	10	debit_card	2
5	2016	10	UPI	63
6	2016	12	credit_card	1
7	2017	1	voucher	33
8	2017	1	UPI	197
9	2017	1	credit_card	582

Results per page: 50 ▼ 1 – 50 of 90

```
** From above analysis, we can see that for different payment type over
different month of 2016 year and onwards, the count of orders are varying.
```

- Distribution of payment installments and count of orders

```sql
select payment_installments, count(order_id) count_orders
from <CompanySchema>.payments
group by payment_installments
order by payment_installments;
```



Row	payment_installments	count_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

Results per page: 50 ▾ 1 – 24 of 24

** Count of orders calculation for different payment installments.

7. Actionable Insights

- As we can see that more orders are placed at around 16:00 hour afternoon, so we need to make sure products are available in plenty and should not go out of stock
- Focus on states where time of delivery is more to bring down the No. of days to be taken to deliver the product
- Increase delivery frequency/agents to bring down estimated time of delivery for orders
- For the states, where average freight value is more, we need to further break down the cost to be bear by customers and bring it down
- Based on the review score, improve the service and quality of products and delivery experience for customers

8. Recommendations

- Around September to November of every year, the customer engagement is more, so we can provide benefits like discounts, vouchers, offers, etc to retain the customers and attract more customers
- Credit card seems to be used of more often as payment mode, so we can come up with good offers on credit card payments.
- Also, we can collaborate with more credit card banks for allow payments for customers
- Looking at number of customers for different cities each state, we can see the engagement and focus on bringing more customers for other cities
- Looking into count of sellers per city for different states, we can bring in more sellers to better meet demand-supply

***** End of Document *****