Scaler Academy

Shashank Raj

DSML-OCT-22-Adv. Batch

21-Nov-22

—

SQL Business Case

—

Harshit Tyagi

Table of Contents

Questions:

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 of customers ordered during the given period.

Answers:

1.1. Data Type of Columns in a Table.

Customers		
Serial Number	Column Name	Data Type
1.	Customer_id	Varchar
2.	Customer_unique_id	Varchar
3.	customer_zip_code_prefix	Varchar
4.	Customer_city	Varchar
5.	Customer_state	Varchar

Geolocation		
Serial Number	Column Name	Data Type
1	Geolocation_zip_code_prefix	Int
2	Geolocation_lat	REAL
3	Geolocation_lng	REAL
4	Geolocation_city	Varchar
5	Geolocation_state	varchar

Order		
Serial Number	Column Name	Data Type
1	Order_id	Varchar
2	Order_item_id	Int
3	Product_id	Varchar
4	Seller_id	Varchar
5	Shipping_limit_date	Date
6	Price	Int
7	Freight_value	Int

Order		
Serial Number	Column Name	Data Type
1	Review_id	Varchar
2	Order_id	Varchar
3	Review_score	Int
4	Review_comment_title	Varchar
5	Review_creation_date	Date
6	Review_answer_timestamp	TimeStamp

Orders		
Serial Number	Column Name	Data Type
1	Order_id	Varchar
2	Customer_id	Varchar
3	Order_status	Varchar
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

Payments		
Serial Number	Column Name	Data Type
1	Order_id	Varchar
2	Payment_sequential	Int
3	Payment_type	Varchar
4	Payment_installations	Int
5	Payment_value	Float

Products		
Serial Number	Column Name	Data Type
1	Product_id	Varchar
2	Product_category	Varchar
3	Product_name_length	Int
4	Product_description_length	Int
5	Product_photos_qty	Int
6	Product_weight_g	Int
7	Product_length_cm	Int
8	Product_height_cm	Int
9	Product_width_cm	Int

Sellers		
Serial Number	Column Name	Data Type
1	Seller_id	Varchar
2	Seller_zipcode_prefix	Varchar
3	Seller_city	Varchar
4	Seller_state	Varchar

1.2. According to the orders table, the time period of this Dataset is from **2016-09-04 21:15:19** to **2018-10-17 17:30:18**.

This was achieved by using the following query:

```
select
      min(order_purchase_timestamp) as start_dt,
      max(order_purchase_timestamp) as end_dt
from orders
```

1.3. The following query gives us the states and cities from which orders were placed.

27 states and 4119 cities from Brazil.

```
select count(distinct(customer_city))
from customers c
join orders o
on o.customer_id = c.customer_id

select count(distinct(customer_state))
from customers c
join orders o
on o.customer_id = c.customer_id
```

2. In-depth Exploration:

2.1. 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?

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

Answer:

2.1.

From the orders table, we can see that the transaction data is from the range between **2016-2018** by using the following query:

```
select
	year(order_purchase_timestamp) as purchase_year,
	count(order_purchase_timestamp) as purchases
from orders
group by purchase_year
order by purchase_year
```

	123 purchase_year	123 purchases
1	2,016	329
2	2,017	45,101
3	2,018	54,011

Also, we can see that overall purchases have been on a rise, year on year, which is good news.

In order to get the details about monthly sales, we aggregate the timestamp of sales over months, leaving out the **cancelled** orders, we get the following data:

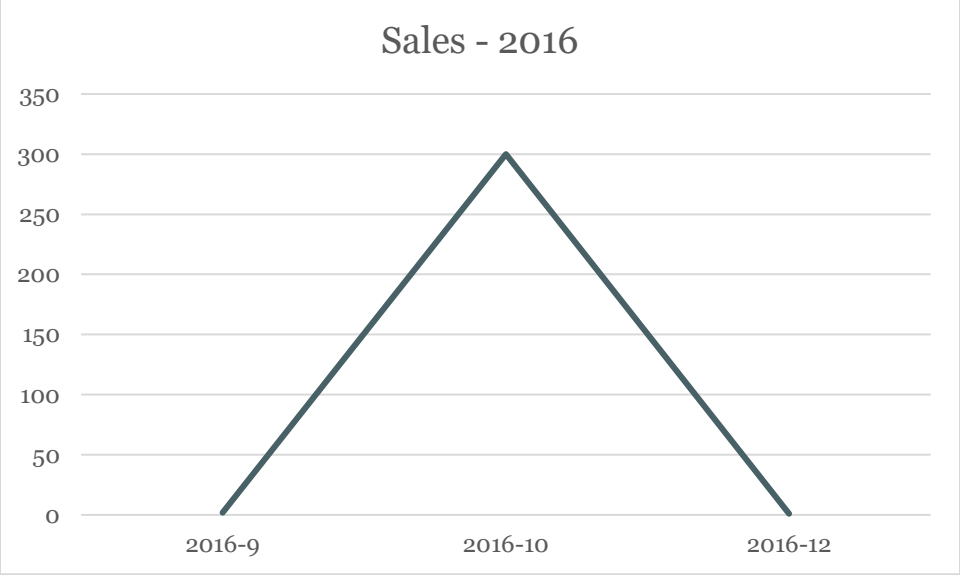
```
-- Purchases By Year, Month
select
	YEAR(order_purchase_timestamp) as shipping_yr,
	MONTH(order_purchase_timestamp) as shipping_month,
	count(order_purchase_timestamp) as orders_total
from orders
where order_status != 'canceled'
group by shipping_yr, shipping_month

order by shipping_yr
```

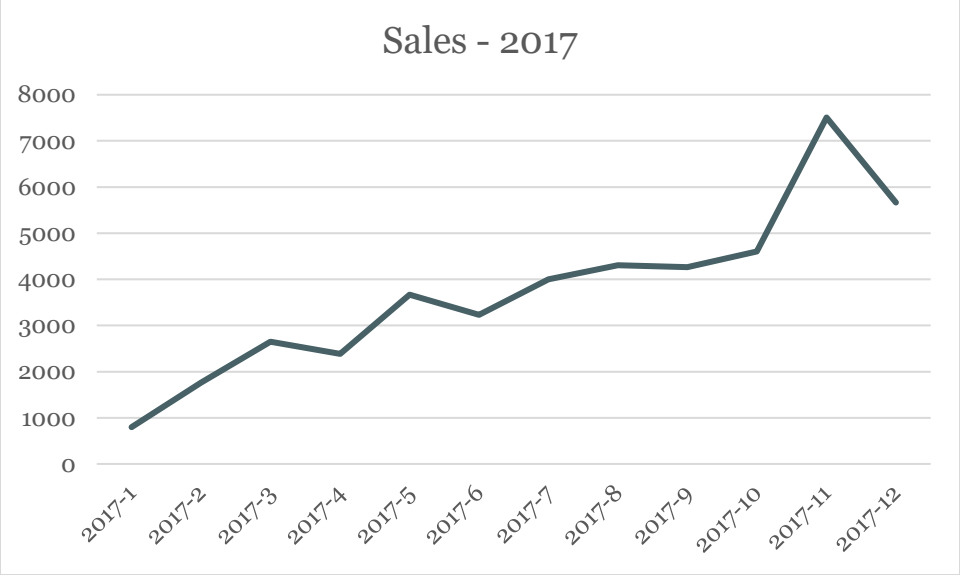
	123 purchase_year	123 purchase_month	123 orders_total
1	2,016	9	2
2	2,016	10	300
3	2,016	12	1
4	2,017	1	797
5	2,017	2	1,763
6	2,017	3	2,649
7	2,017	4	2,386
8	2,017	5	3,671
9	2,017	6	3,229
10	2,017	7	3,998
11	2,017	8	4,304
12	2,017	9	4,265
13	2,017	10	4,605
14	2,017	11	7,507
15	2,017	12	5,662
16	2,018	1	7,235
17	2,018	2	6,655
18	2,018	3	7,185
19	2,018	4	6,924
20	2,018	5	6,849
21	2,018	6	6,149
22	2,018	7	6,251
23	2,018	8	6,428
24	2,018	9	1

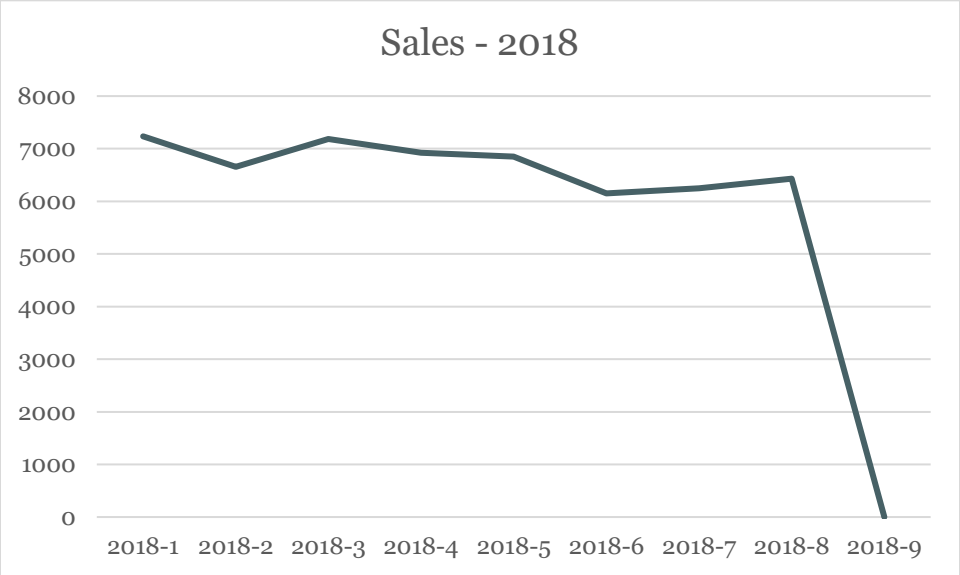
Charing the same, we see the following trend overall:

So, there are bump in sales in March, May and August. Then sharp drops in September, which picks up in November.

Getting the data just by month, we observe the following: (Without Cancellations)







Looking at the data, we can see that in 2016, there was nothing significant, though the number of sales were high in October and dropped back in December. There is no Data for November for sale.

Similarly, data in 2017 show a sharp upward trend, while a significant jump around the final quarter.

Finally, we do not have data to compare for 2018 for the final quarter, but it seems that 2018 was seeing a downward trend.

Here, we see a very interesting statistic, that towards the end of the dataset, the number of cancellations is pretty high.

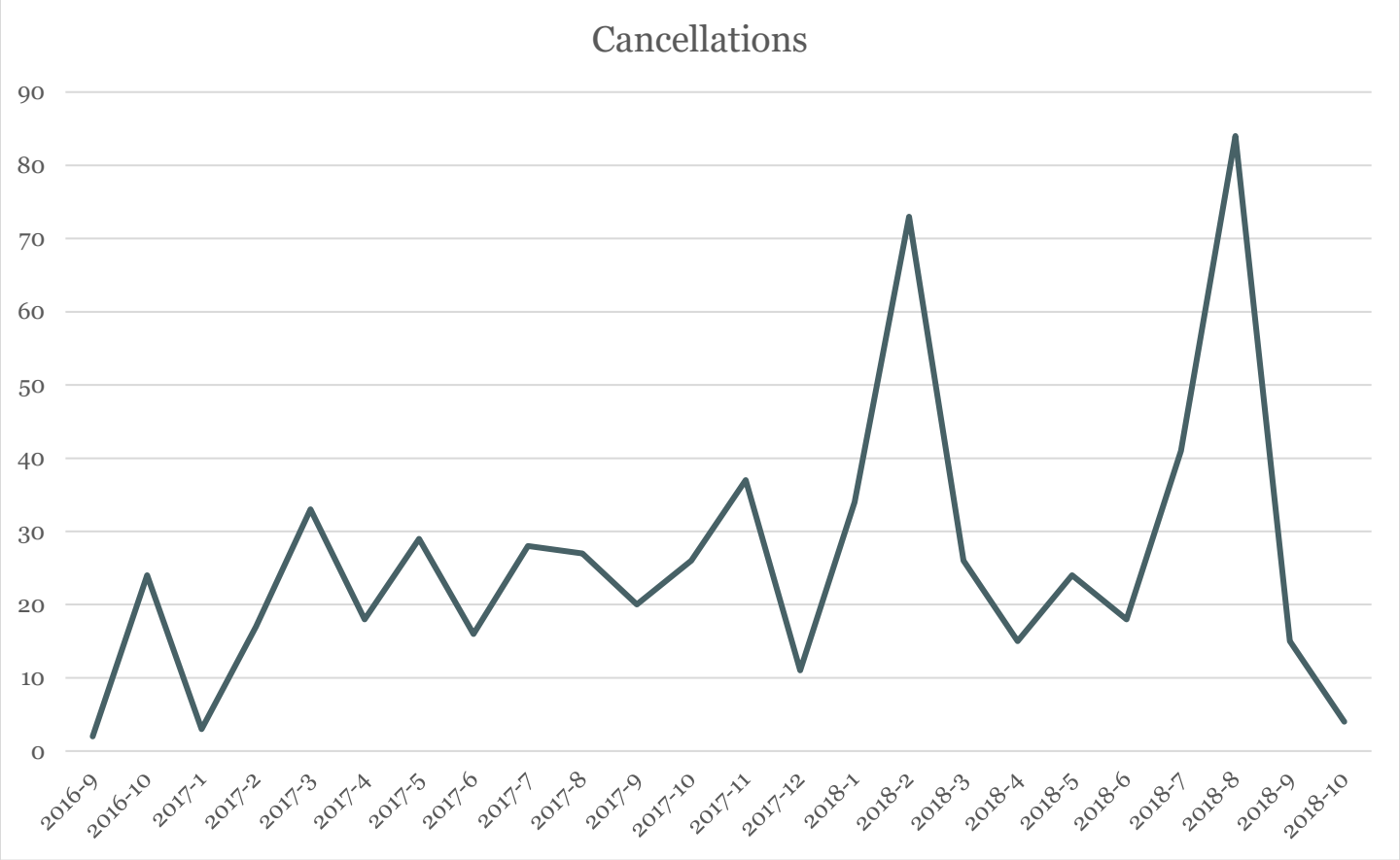
Just investigating the cancellations, we observe the following:

```
select
	year(ords.order_purchase_timestamp) as pr_year,
	month(ords.order_purchase_timestamp) as pr_mnth,
	count(ords.order_purchase_timestamp) as tot_purchase
from orders ords
where ords.order_status = 'canceled'
group by pr_year, pr_mnth

order by pr_year, pr_mnth
```

123 pr_year	123 pr_mnth	123 Cancellations
2,016	9	2
2,016	10	24
2,017	1	3
2,017	2	17
2,017	3	33
2,017	4	18
2,017	5	29
2,017	6	16
2,017	7	28
2,017	8	27
2,017	9	20
2,017	10	26
2,017	11	37
2,017	12	11
2,018	1	34
2,018	2	73
2,018	3	26
2,018	4	15
2,018	5	24
2,018	6	18
2,018	7	41
2,018	8	84
2,018	9	15
2,018	10	4

There seems to be an alarming increase in the number of order cancellations towards the end of 2018, which is probably worth investigating.

This may also be affecting the sales, which sees a downward trend in the previous chart.



There seems to not be any patterns in the cancellations, however, the peaks in cancellations are something that can be looked at.

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

3.1. Get month on month orders by states

3.2. Distribution of customers across the states in Brazil.

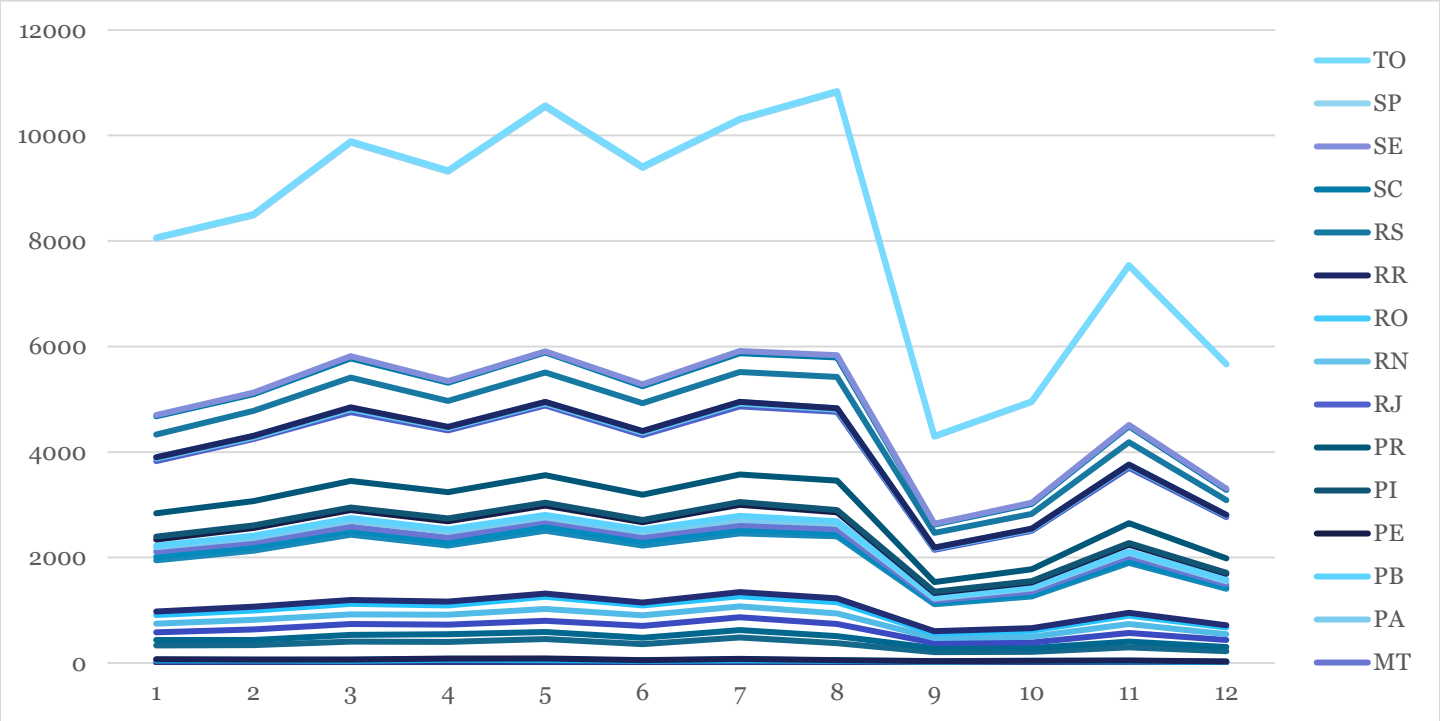
Answer:

3.1.

Below is the query to get the sales of states by months.

```
select
c.customer_state,
       month(o.order_purchase_timestamp) as ord_month,
       count(o.order_purchase_timestamp) as cust_ord
from customers c
join orders o
on c.customer_id = o.customer_id
group by c.customer_state,ord_month
order by c.customer_state, ord_month
```

Attached is the chart to visualize the output:

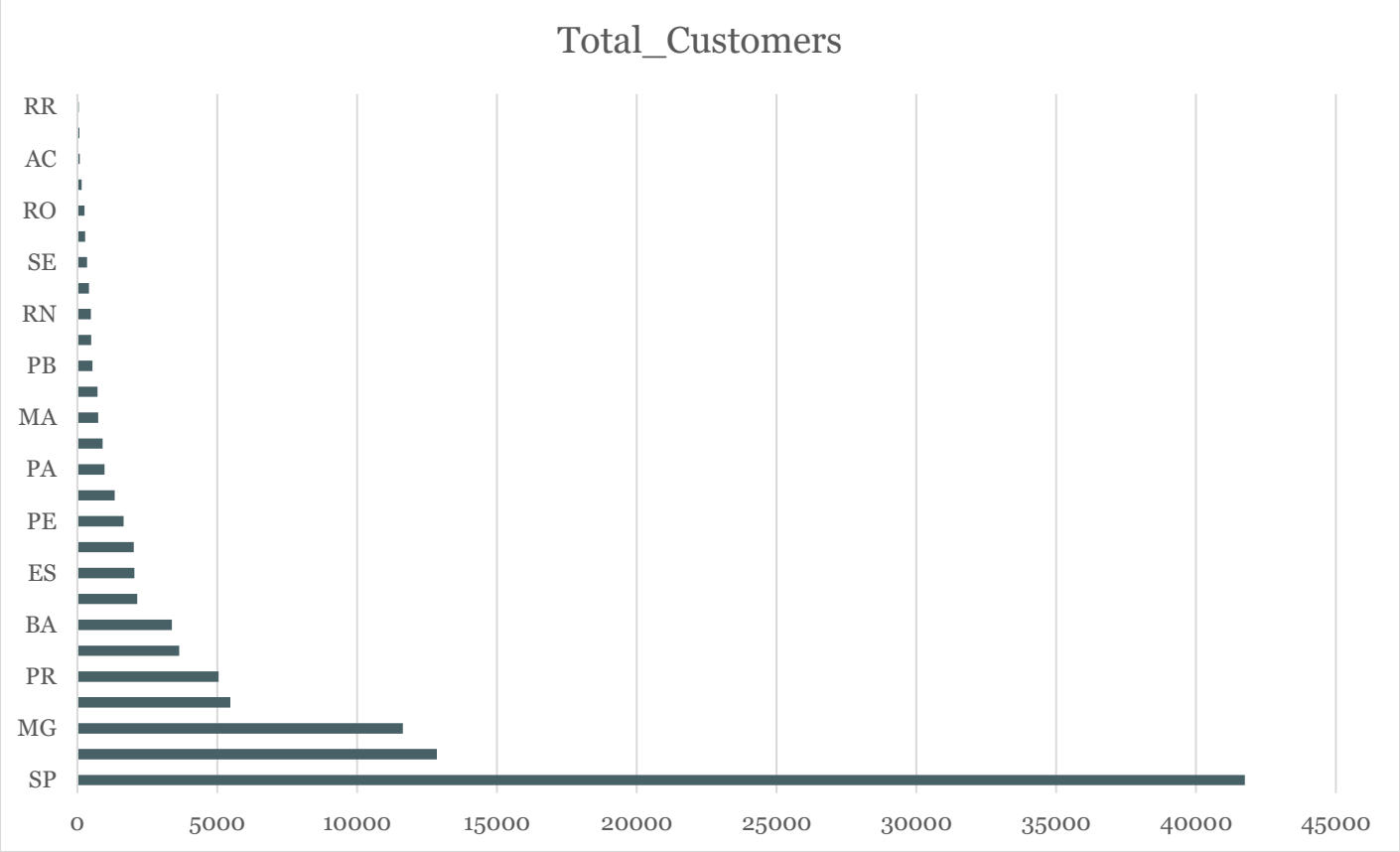


3.2

Distribution of Customers in Brazil, overall, based on the customer table:

```
-- Distribution of customer across states in Brazil

select customer_state,
       count(distinct(customer_id)) as total_customers
from customers
group by customer_state
```

	customer_state	total_customers
1	SP	41,746
2	RJ	12,852
3	MG	11,635
4	RS	5,466
5	PR	5,045
6	SC	3,637
7	BA	3,380
8	DF	2,140
9	ES	2,033
10	GO	2,020
11	PE	1,652
12	CE	1,336
13	PA	975
14	MT	907
15	MA	747
16	MS	715
17	PB	536
18	PI	495
19	RN	485
20	AL	413
21	SE	350
22	TO	280
23	RO	253
24	AM	148
25	AC	81
26	AP	68
27	RR	46


Total_Customers
RR
AC
RO
SE
RN
PB
MA
PA
PE
ES
BA
PR
MG
SP
0
5000
10000
15000
20000
25000
30000
35000
40000
45000

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

4.1. 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

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

Answers:

4.1.

```
-- Percentage Change in Sales Year over Year over Months

select *,
	ROUND(((Y.Total_Yearly_Sale-Y.PY_Val)/Y.Total_Yearly_Sale)*100,2) as PCT_Chng
from (
select
*,
LAG(X.Total_Yearly_Sale) over(order by X.Purchase_year) as PY_Val
from (
SELECT
	Year(o.order_purchase_timestamp) as Purchase_year,
	Month(o.order_purchase_timestamp) as Purchase_Mnth,
	round(sum(p.payment_value),2) Total_Yearly_Sale
from orders o
join payments p
on o.order_id = p.order_id
where Month(o.order_purchase_timestamp) not in (9,10,11,12)
group by Purchase_year,Purchase_Mnth
order by Purchase_year
) as X)

as Y
```

Purchase_year	Purchase_Mnth	Total_Yearly_Sale	PY_Val	PCT_Chng
2,017	1	138,488.04	[NULL]	[NULL]
2,017	2	291,908.01	138,488.04	52.56
2,017	3	449,863.6	291,908.01	35.11
2,017	4	417,788.03	449,863.6	-7.68
2,017	5	592,918.82	417,788.03	29.54
2,017	6	511,276.38	592,918.82	-15.97
2,017	7	592,382.92	511,276.38	13.69
2,017	8	674,396.32	592,382.92	12.16
2,018	1	1,115,004.18	674,396.32	39.52
2,018	2	992,463.34	1,115,004.18	-12.35
2,018	3	1,159,652.12	992,463.34	14.42
2,018	4	1,160,785.48	1,159,652.12	0.1
2,018	5	1,153,982.15	1,160,785.48	-0.59
2,018	6	1,023,880.5	1,153,982.15	-12.71
2,018	7	1,066,540.75	1,023,880.5	4
2,018	8	1,022,425.32	1,066,540.75	-4.31

```
-- Percentage Change in Sales Year over Year in Total


select *,
       ROUND(((Y.Total_Yearly_Sale-Y.PY_Val)/Y.Total_Yearly_Sale)*100,2) as PCT_Chng
from (
select
*,
LAG(X.Total_Yearly_Sale) over(order by X.Purchase_year) as PY_Val
from (
SELECT
       Year(o.order_purchase_timestamp) as Purchase_year,
       round(sum(p.payment_value),2) Total_Yearly_Sale
from orders o
join payments p
on o.order_id = p.order_id
where Month(o.order_purchase_timestamp) not in (9,10,11,12)
group by Purchase_year
order by Purchase_year
) as X)
as Y
```

Purchase_year	Total_Yearly_Sale	PY_Val	PCT_Chng
2,017	3,669,022.12	[NULL]	[NULL]
2,018	8,694,733.84	3,669,022.12	57.8

4.2.

```sql
select
	c.customer_state,
	round(sum(oi.freight_value),2) as 'Total_Freight',
	round(AVG(oi.freight_value),2) as 'Mean_Freight',
	round(sum(oi.price),2) as 'Total_Price',
	round(AVG(oi.price),2) as 'Mean_Price'
from order_items oi
join orders o
on oi.order_id = o.order_id
join customers c
on o.customer_id = c.customer_id
group by c.customer_state

ORDER by c.customer_state
```

	customer_state	Total_Freight	Mean_Freight	Total_Price	Mean_Price
1	AC	3,686.75	40.07	15,982.95	173.73
2	AL	15,914.59	35.84	80,314.81	180.89
3	AM	5,478.89	33.21	22,356.84	135.5
4	AP	2,788.5	34.01	13,474.3	164.32
5	BA	100,156.68	26.36	511,349.99	134.6
6	CE	48,351.59	32.71	227,254.71	153.76
7	DF	50,625.5	21.04	302,603.94	125.77
8	ES	49,764.6	22.06	275,037.31	121.91
9	GO	53,114.98	22.77	294,591.95	126.27
10	MA	31,523.77	38.26	119,648.22	145.2
11	MG	270,853.46	20.63	1,585,308.03	120.75
12	MS	19,144.03	23.37	116,812.64	142.63
13	MT	29,715.43	28.17	156,453.53	148.3
14	PA	38,699.3	35.83	178,947.81	165.69
15	PB	25,719.73	42.72	115,268.08	191.48
16	PE	59,449.66	32.92	262,788.03	145.51
17	PI	21,218.2	39.15	86,914.08	160.36
18	PR	117,851.68	20.53	683,083.76	119
19	RJ	305,589.31	20.96	1,824,092.67	125.12
20	RN	18,860.1	35.65	83,034.98	156.97
21	RO	11,417.38	41.07	46,140.64	165.97
22	RR	2,235.19	42.98	7,829.43	150.57
23	RS	135,522.74	21.74	750,304.02	120.34
24	SC	89,660.26	21.47	520,553.34	124.65
25	SE	14,111.47	36.65	58,920.85	153.04
26	SP	718,723.07	15.15	5,202,955.05	109.65
27	TO	11,732.68	37.25	49,621.74	157.53

5. Analysis on sales, freight and delivery time

1. Calculate days between purchasing, delivering and estimated delivery
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
3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
4. Sort the data to get the following:
5. Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
6. Top 5 states with highest/lowest average time to delivery
7. Top 5 states where delivery is really fast/ not so fast compared to estimated date

Answers:

5.1

```
select
	day(order_purchase_timestamp) as Purchase_day,
	day(order_estimated_delivery_date) as Est_Delivery_Day,
	day(order_delivered_customer_date) as Actual_Delivery_Day,
	DATEDIFF(order_estimated_delivery_date,order_purchase_timestamp) as 'Est_Wait_Time',
	DATEDIFF(order_delivered_customer_date,order_purchase_timestamp) as 'Act_Wait_Time'
from orders
group by order_id
```

Sample output:

123 Purchase_day	123 Est_Delivery_Day	123 Actual_Delivery_Day	123 Est_Wait_Time	123 Act_Wait_Time
13	29	20	16	7
26	15	12	19	16
14	5	22	22	8
8	20	14	12	6
4	17	1	41	25
15	6	22	22	7
10	4	18	25	8
4	25	9	21	5
19	29	29	10	10
2	23	4	21	2
24	9	29	16	5
27	7	7	11	11
24	22	31	29	7
13	6	26	21	13
14	28	22	14	8

5.2

```
select
	DATEDIFF(order_estimated_delivery_date,order_delivered_customer_date
) as 'diff_estimated_delivery',
	DATEDIFF(order_delivered_customer_date,order_purchase_timestamp
) as 'time_to_delivery'
from orders
```

Sample Output:

123 diff_estimated_delivery	123 time_to_delivery
9	7
3	16
14	8
6	6
16	25
15	7
17	8
16	5
0	10

5.3

```
select
	c.customer_state,
	round(AVG(oi.freight_value),2) as 'Mean_Freight',
	round(avg(DATEDIFF(o.order_delivered_customer_date,o.order_purchase_timestamp)),2) as
'time_to_delivery',
	round(avg(DATEDIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date)),2) as
'diff_estimated_delivery'
from order_items oi
join orders o
on oi.order_id = o.order_id
join customers c
on o.customer_id = c.customer_id
group by c.customer_state
ORDER by c.customer_state
```

Sample Output:

	customer_state	Mean_Freight	Avg_time_to_delivery	Avg_diff_estimated_delivery
1	AC	40.07	20.68	20.98
2	AL	35.84	24.45	8.74
3	AM	33.21	26.34	19.93
4	AP	34.01	28.22	18.4
5	BA	26.36	19.19	10.98
6	CE	32.71	20.92	11.1
7	DF	21.04	12.89	12.2
8	ES	22.06	15.59	10.65
9	GO	22.77	15.34	12.29
10	MA	38.26	21.59	9.91
11	MG	20.63	11.92	13.34
12	MS	23.37	15.46	11.23
13	MT	28.17	17.91	14.57
14	PA	35.83	23.7	14.25
15	PB	42.72	20.55	13.04

5.4

The Next Sections will be solved by changing the order value and order type. Hence, I will write the common code here and provide the answers below:

```
select
	c.customer_state,
	round(AVG(oi.freight_value),2) as 'Mean_freight',
	round(avg(DATEDIFF(o.order_delivered_customer_date,o.order_purchase_timestamp)),2) as
'Avg_time_to_delivery',
	round(avg(DATEDIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date)),2) as
'Avg_diff_estimated_delivery'
from order_items oi
join orders o
on oi.order_id = o.order_id
join customers c
on o.customer_id = c.customer_id
group by c.customer_state
ORDER by Mean_freight DESC
LIMIT 5
```

5.5

Top 5 States with the Highest Average Freight Value are:

1. RR
2. PB
3. RO
4. AC
5. PI

Top 5 States with the Lowest Average Freight Value are:

1. SP
2. PR
3. MG
4. RJ
5. DF

5.6

Top 5 States with the Highest Average Time to Delivery:

1. AP
2. RR
3. AM
4. AL
5. PA

Top 5 States with the Lowest Average Time to Delivery:

1. SP
2. PR
3. MG
4. DF
5. SC

5.7

Top 5 states where delivery is really fast compared to estimated date are:

1. SP
2. PR
3. MG
4. RO
5. AC

Top 5 states where delivery is not so fast compared to estimated date are:

```
select
	*,
	(X.Avg_time_to_delivery-X.Avg_diff_estimated_delivery) as Del_Diff
from (
select
	c.customer_state,
	round(AVG(oi.freight_value),2) as 'Mean_freight',
	round(avg(DATEDIFF(o.order_delivered_customer_date,o.order_purchase_timestamp)),2) as
'Avg_time_to_delivery',
	round(avg(DATEDIFF(o.order_estimated_delivery_date,o.order_delivered_customer_date)),2) as
'Avg_diff_estimated_delivery'
from order_items oi
join orders o
on oi.order_id = o.order_id
join customers c
on o.customer_id = c.customer_id
group by c.customer_state
ORDER by Avg_time_to_delivery
) as X
order by Del_Diff DESC
LIMIT 5
```

1. AL
2. MA
3. SE
4. RR
5. AP

6. Payment type analysis:

1. Month over Month count of orders for different payment types.
2. Count of orders based on the no. of payment instalments.

Answer:

6.1.

```
select
       Year(o.order_purchase_timestamp) as 'Ord_Year',
       Month(o.order_purchase_timestamp) as 'Ord_Month',
       p.payment_type as 'Mode_of_Payment',
       count(o.order_id) as 'Orders_placed'
from payments p
join orders o
on o.order_id = p.order_id
group by Ord_Year,Ord_Month, p.payment_type
order by Ord_Year,Ord_Month, p.payment_type
```

Sample Solution:

	Ord_Year	Ord_Month	Mode_of_Payment	Orders_placed
1	2,016	9	credit_card	3
2	2,016	10	credit_card	254
3	2,016	10	debit_card	2
4	2,016	10	UPI	63
5	2,016	10	voucher	23
6	2,016	12	credit_card	1
7	2,017	1	credit_card	583
8	2,017	1	debit_card	9
9	2,017	1	UPI	197
10	2,017	1	voucher	61
11	2,017	2	credit_card	1,356
12	2,017	2	debit_card	13
13	2,017	2	UPI	398
14	2,017	2	voucher	119
15	2,017	3	credit_card	2,016
16	2,017	3	debit_card	31
17	2,017	3	UPI	590
18	2,017	3	voucher	200

6.2.

```
select
	p.payment_installments as 'Payment_Installments',
	count(o.order_id) as 'Orders_placed'
from payments p
join orders o
on o.order_id = p.order_id
group by p.payment_installments
order by p.payment_installments
```

	Payment_Installments	Orders_placed
1	0	2
2	1	52,546
3	2	12,413
4	3	10,461
5	4	7,098
6	5	5,239
7	6	3,920
8	7	1,626
9	8	4,268
10	9	644
11	10	5,328
12	11	23
13	12	133
14	13	16
15	14	15
16	15	74
17	16	5
18	17	8
19	18	27
20	20	17
21	21	3
22	22	1
23	23	1
24	24	18

THE END.