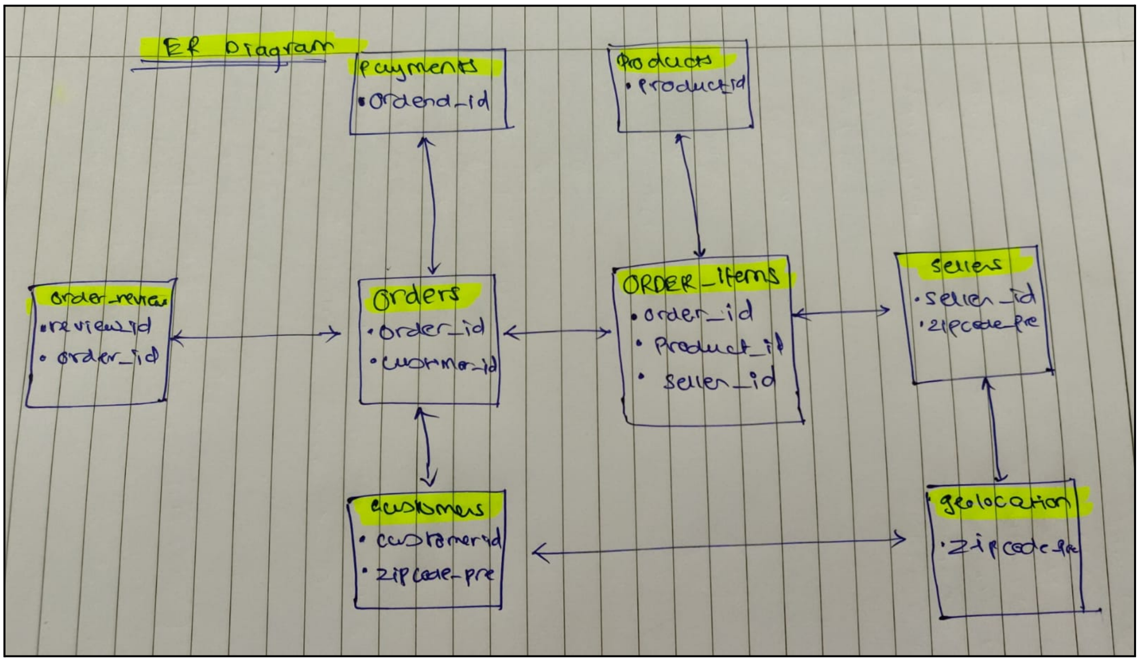
Business Case Study - Target
Using MySQL

1. Exploration of Data:

1.1.ER Diagram:



1.2.Table-wise exploration:

Dtype of data:

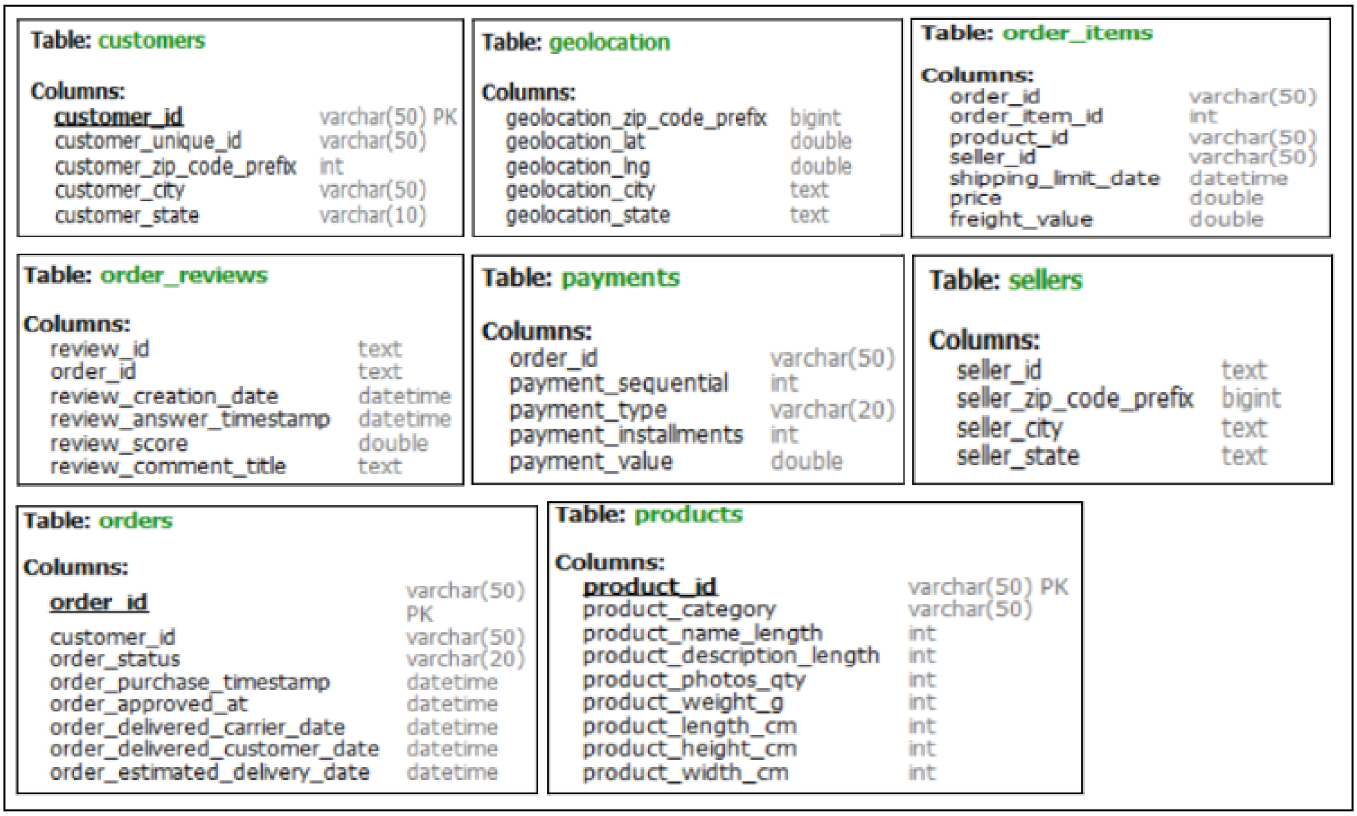


Table: customers

Columns:

customer_id	varchar(50) PK
customer_unique_id	varchar(50)
customer_zip_code_prefix	int
customer_city	varchar(50)
customer_state	varchar(10)

Table: geolocation

Columns:

geolocation_zip_code_prefix	bigint
geolocation_lat	double
geolocation_lng	double
geolocation_city	text
geolocation_state	text

Table: order_items

Columns:

order_id	varchar(50)
order_item_id	int
product_id	varchar(50)
seller_id	varchar(50)
shipping_limit_date	datetime
price	double
freight_value	double

Table: order_reviews

Columns:

review_id	text
order_id	text
review_creation_date	datetime
review_answer_timestamp	datetime
review_score	double
review_comment_title	text

Table: payments

Columns:

order_id	varchar(50)
payment_sequential	int
payment_type	varchar(20)
payment_installments	int
payment_value	double

Table: sellers

Columns:

seller_id	text
seller_zip_code_prefix	bigint
seller_city	text
seller_state	text

Table: orders

Columns:

order_id	varchar(50) PK
customer_id	varchar(50)
order_status	varchar(20)
order_purchase_timestamp	datetime
order_approved_at	datetime
order_delivered_carrier_date	datetime
order_delivered_customer_date	datetime
order_estimated_delivery_date	datetime

Table: products

Columns:

product_id	varchar(50) PK
product_category	varchar(50)
product_name_length	int
product_description_length	int
product_photos_qty	int
product_weight_g	int
product_length_cm	int
product_height_cm	int
product_width_cm	int

1.3.Time period of Data:

a). Below are the details on shipping of First order and Last Order in the data:

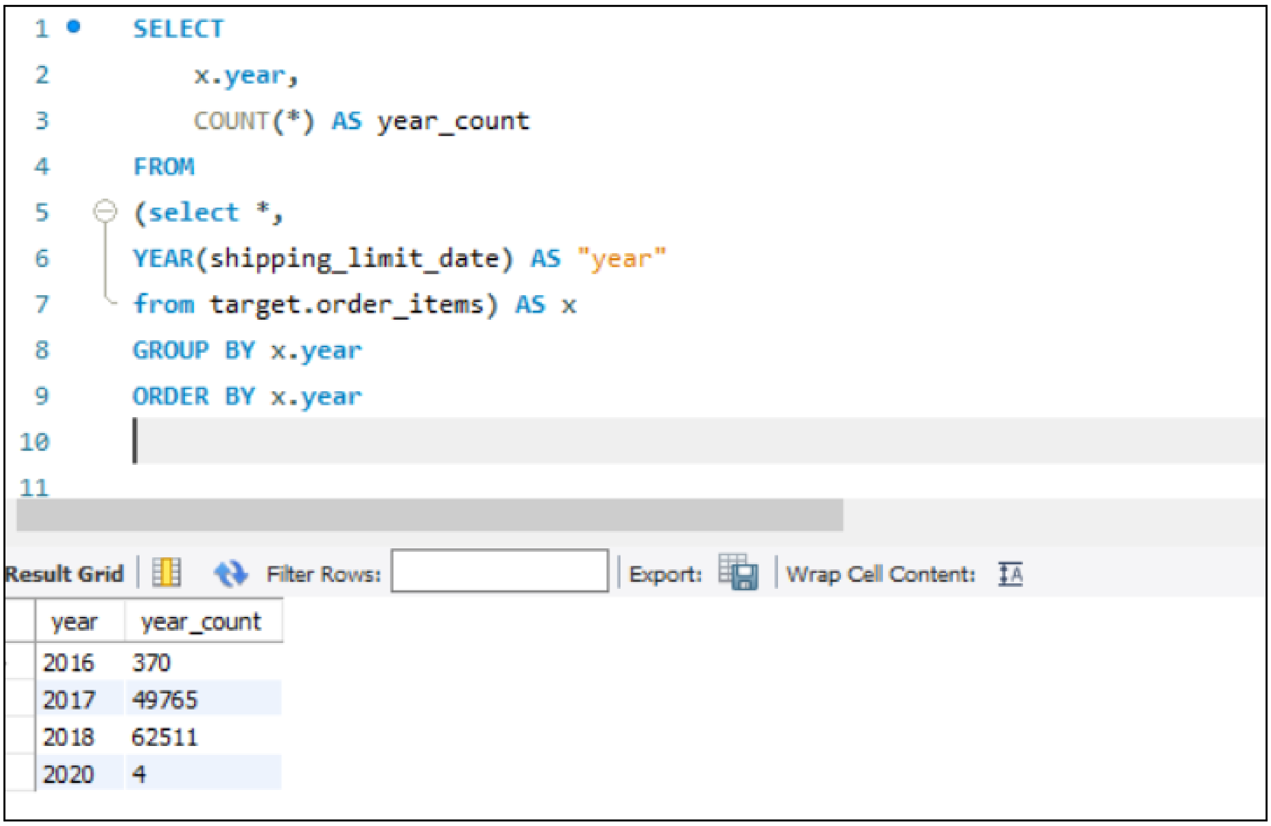
shipping_limit_date — **Start Date** – 19th September 2016 @ 12:15am
Shipping_limit_date — **Last Date** –9th April 2020 @ 10:35pm

Query:

```
SELECT *
FROM (SELECT *,
        DATE_FORMAT(shipping_limit_date, '%Y-%m-%d') "Date",
        DATE_FORMAT(shipping_limit_date,'%H:%i:%s') "Time"
        FROM target.order_items) AS x
ORDER BY x.Date
```

b). Order items per year (as per the data):

Highest no.of orders - 2018
Least orders - 2020
No order records for the year 2019



```
SELECT
    x.year,
    COUNT(*) AS year_count
FROM
(select *,
YEAR(shipping_limit_date) AS "year"
from target.order_items) AS x
GROUP BY x.year
ORDER BY x.year
```

year	year_count
2016	370
2017	49765
2018	62511
2020	4

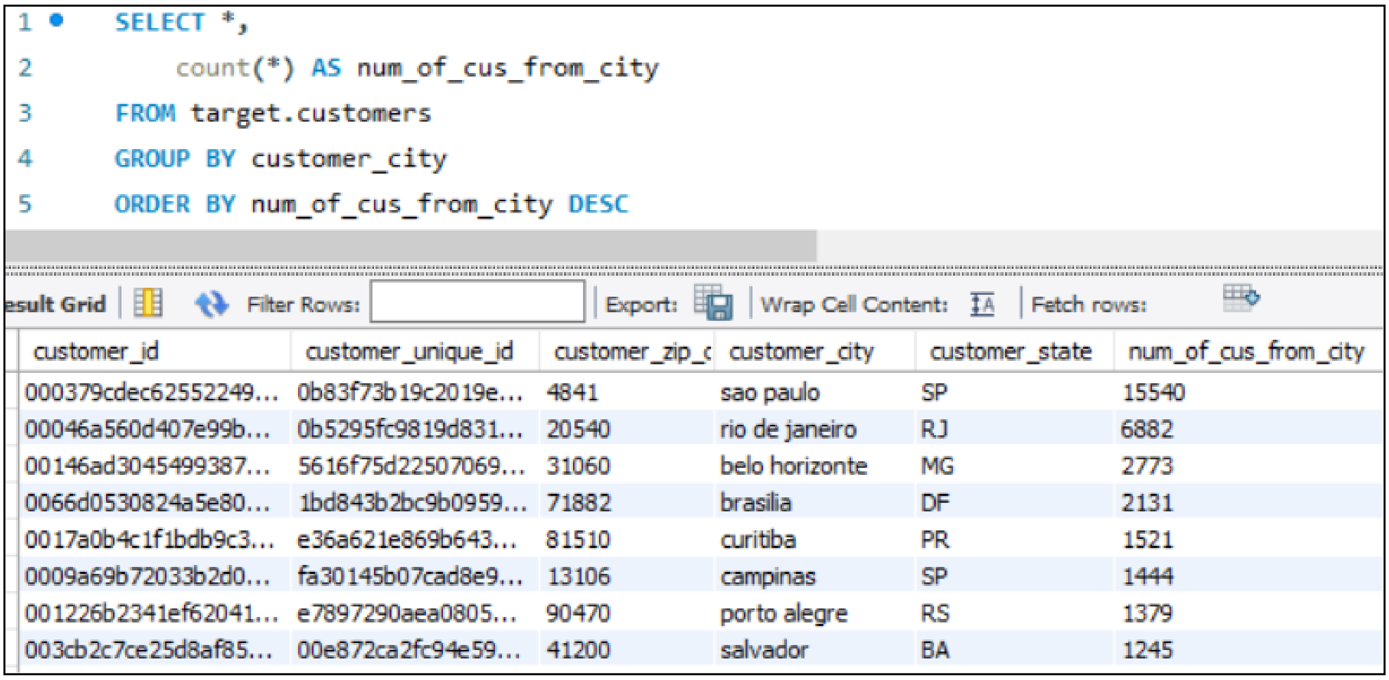
1.4.Location of Data (Cities and States):

Total No. of States – 27

Total no.of cities - 4119

Highest no.of customers are from State → SP - 41746

Highest no.of customers are from city →Sao Paulo, SP - 15540



```
SELECT *,
    count(*) AS num_of_cus_from_city
FROM target.customers
GROUP BY customer_city
ORDER BY num_of_cus_from_city DESC
```

customer_id	customer_unique_id	customer_zip_c	customer_city	customer_state	num_of_cus_from_city
000379cdec62552249...	0b83f73b19c2019e...	4841	sao paulo	SP	15540
00046a560d407e99b...	0b5295fc9819d831...	20540	rio de janeiro	RJ	6882
00146ad3045499387...	5616f75d22507069...	31060	belo horizonte	MG	2773
0066d0530824a5e80...	1bd843b2bc9b0959...	71882	brasilia	DF	2131
0017a0b4c1f1bdb9c3...	e36a621e869b643...	81510	curitiba	PR	1521
0009a69b72033b2d0...	fa30145b07cad8e9...	13106	campinas	SP	1444
001226b2341ef62041...	e7897290aea0805...	90470	porto alegre	RS	1379
003cb2c7ce25d8af85...	00e872ca2fc94e59...	41200	salvador	BA	1245

2. In-depth Exploration:

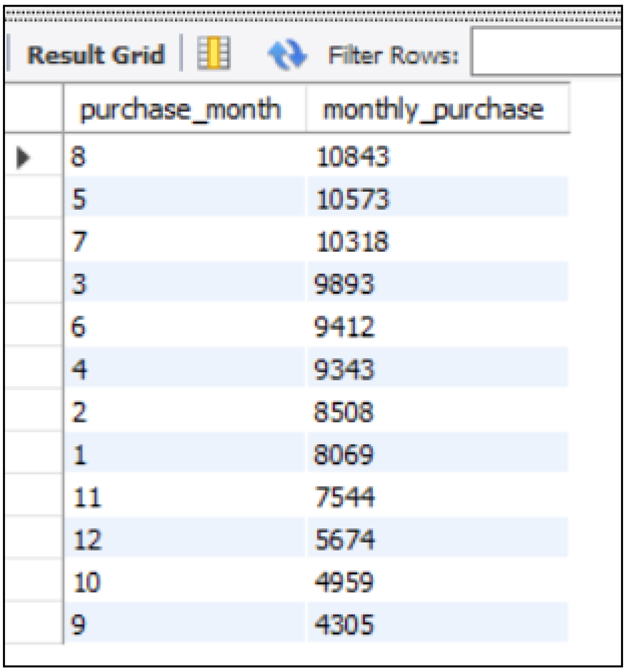
2.1. a). Month-wise purchase Trend:

→ Most of the customers tend to buy in the month of August and least in September irrespective of the year.

→ 2nd Quarter has the highest purchase i.e April, May & June, irrespective of the year.

Query:

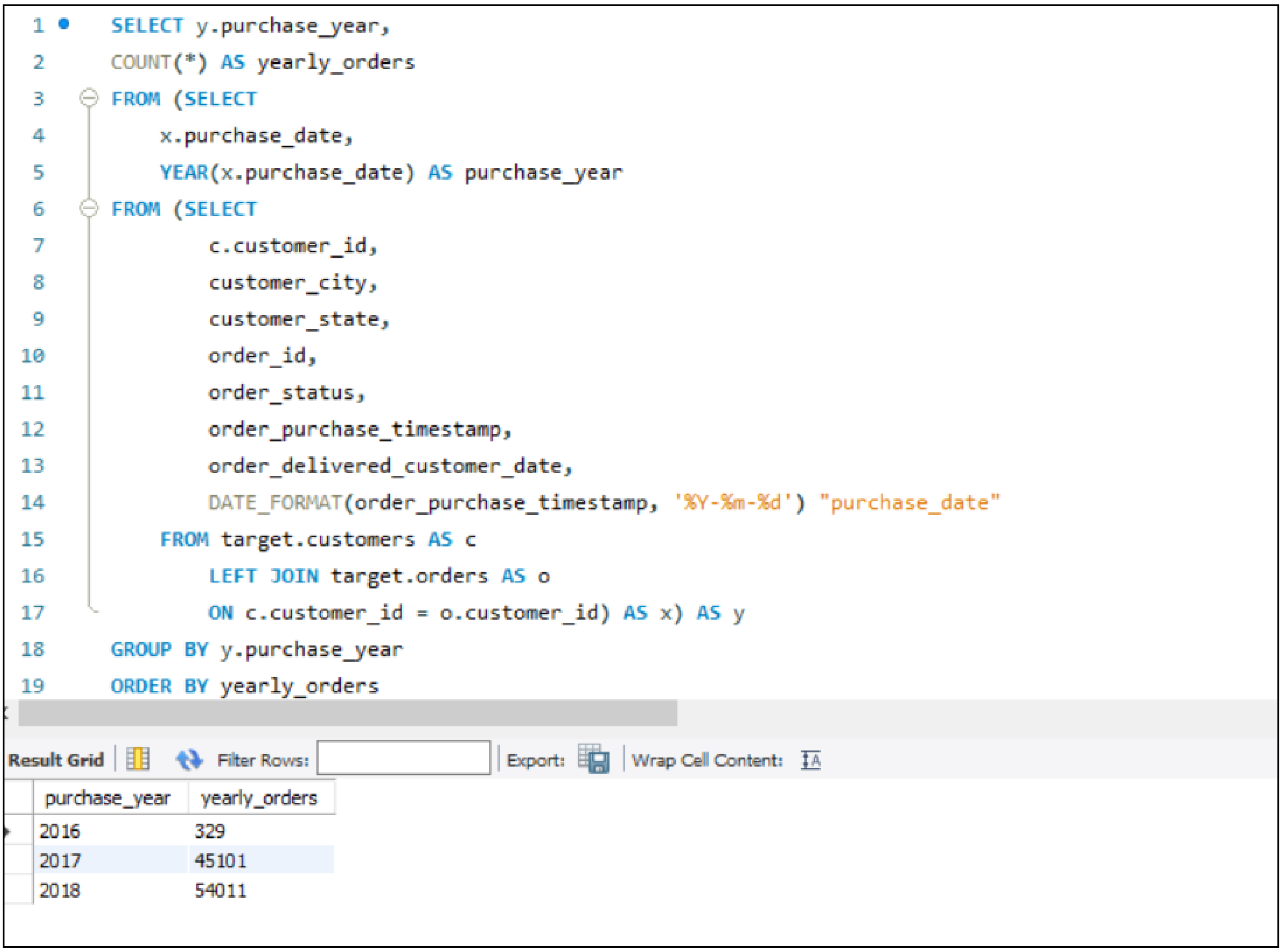
```
SELECT y.purchase_month,
 COUNT(*) AS monthly_purchase
FROM (SELECT
 x.purchase_date,
    MONTH(x.purchase_date) AS purchase_month
FROM (SELECT
  c.customer_id,
  customer_city,
  customer_state,
  order_id,
  order_status,
  order_purchase_timestamp,
  order_delivered_customer_date,
  DATE_FORMAT(order_purchase_timestamp, '%Y-%m-%d') "purchase_date"
 FROM target.customers AS c
  LEFT JOIN target.orders AS o
  ON c.customer_id = o.customer_id) AS x) as y
 GROUP BY y.purchase_month
 ORDER BY monthly_purchase DESC
```



purchase_month	monthly_purchase
8	10843
5	10573
7	10318
3	9893
6	9412
4	9343
2	8508
1	8069
11	7544
12	5674
10	4959
9	4305

2.1. b). Yearly orders placed:

→ As per the trend, the best year is 2017, the growth is more than 120x of the previous year.
→ And in 2018 the growth is only 20%, which is comparatively too low.
→ We could say the growth is stagnant in the year 2018.



```sql
SELECT y.purchase_year,
COUNT(*) AS yearly_orders
FROM (SELECT
  x.purchase_date,
  YEAR(x.purchase_date) AS purchase_year
FROM (SELECT
    c.customer_id,
    customer_city,
    customer_state,
    order_id,
    order_status,
    order_purchase_timestamp,
    order_delivered_customer_date,
    DATE_FORMAT(order_purchase_timestamp, '%Y-%m-%d') "purchase_date"
  FROM target.customers AS c
    LEFT JOIN target.orders AS o
    ON c.customer_id = o.customer_id) AS x) AS y
GROUP BY y.purchase_year
ORDER BY yearly_orders
```

purchase_year	yearly_orders
2016	329
2017	45101
2018	54011

2.1. c). Monthly trend in a year:

In the year 2016:

→ The highest purchase in 2016 was made in the month of October, then it dropped in December.

→ Again the purchases grew in the month of January (2017).

In the year 2017:

→ Purchases dropped in the month of April, June, September, and December compared to the previous month’s purchases.

→ In December, the purchase’s dropped the most compared to other months in comparison to the previous month’s purchases.

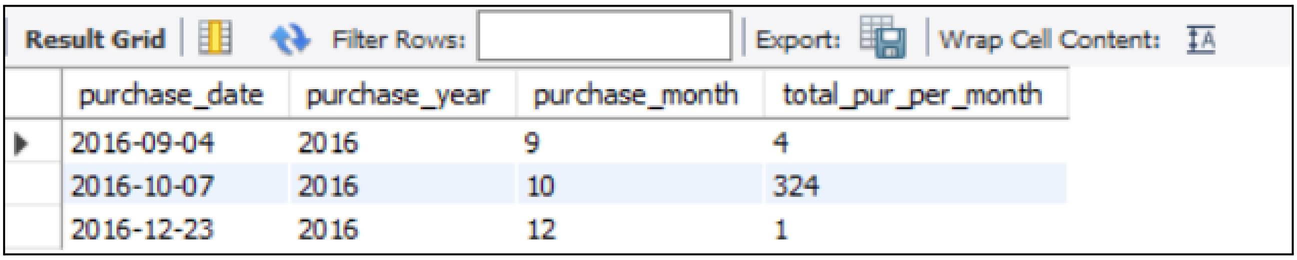
→ Highest growth rate is seen in January 2017

→ Highest growth was seen in the month of November purchases made compared to the previous month.

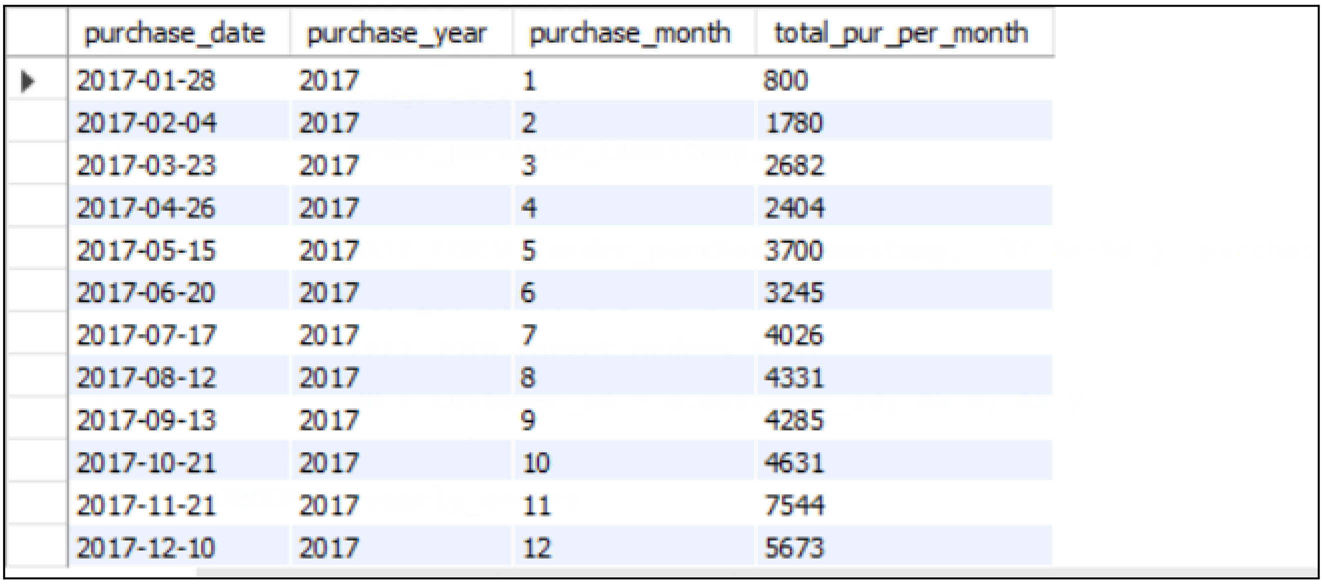
In the year 2018:

→ The biggest fall in purchases is seen in September.

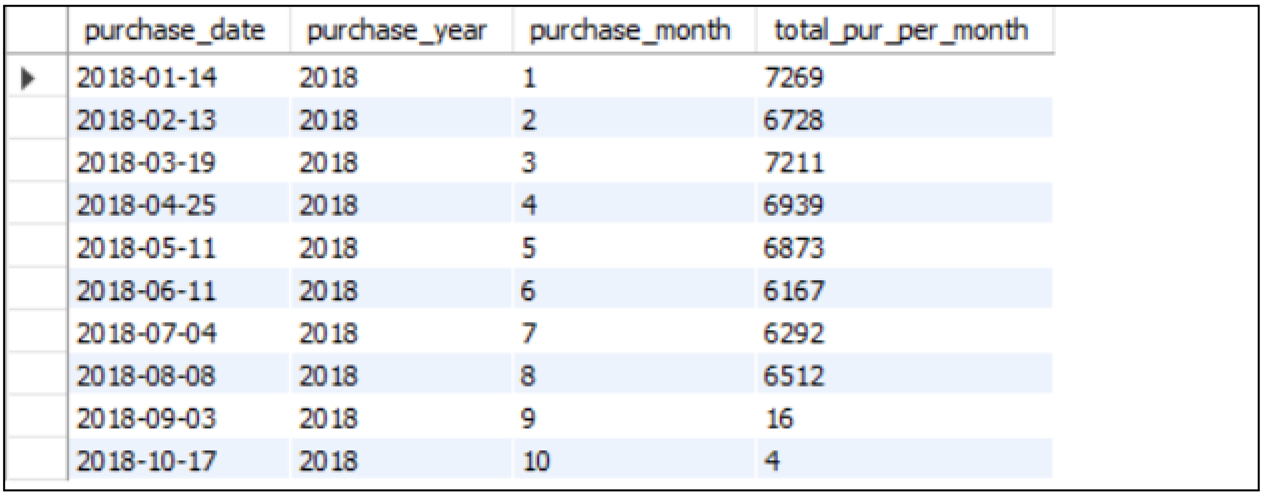
→ Highest purchases were done in January.



purchase_date	purchase_year	purchase_month	total_pur_per_month
2016-09-04	2016	9	4
2016-10-07	2016	10	324
2016-12-23	2016	12	1



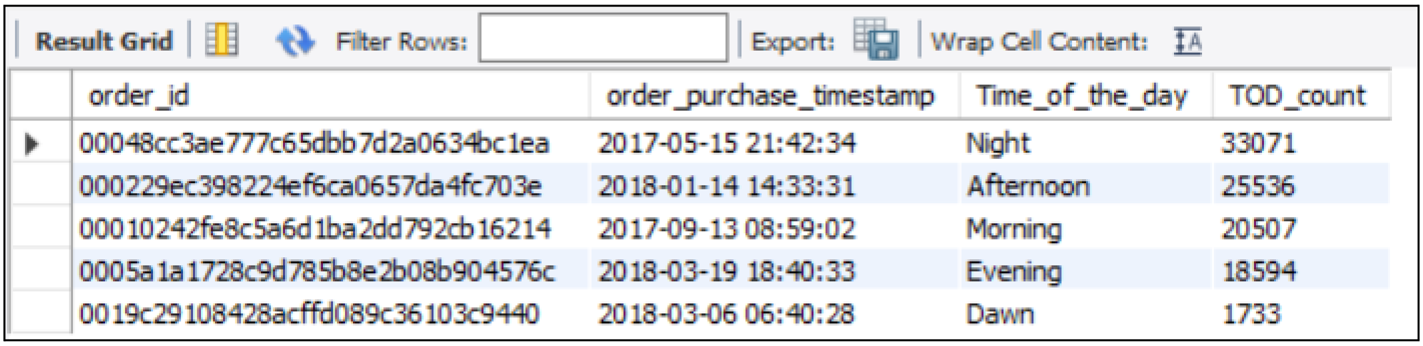
purchase_date	purchase_year	purchase_month	total_pur_per_month
2017-01-28	2017	1	800
2017-02-04	2017	2	1780
2017-03-23	2017	3	2682
2017-04-26	2017	4	2404
2017-05-15	2017	5	3700
2017-06-20	2017	6	3245
2017-07-17	2017	7	4026
2017-08-12	2017	8	4331
2017-09-13	2017	9	4285
2017-10-21	2017	10	4631
2017-11-21	2017	11	7544
2017-12-10	2017	12	5673



purchase_date	purchase_year	purchase_month	total_pur_per_month
2018-01-14	2018	1	7269
2018-02-13	2018	2	6728
2018-03-19	2018	3	7211
2018-04-25	2018	4	6939
2018-05-11	2018	5	6873
2018-06-11	2018	6	6167
2018-07-04	2018	7	6292
2018-08-08	2018	8	6512
2018-09-03	2018	9	16
2018-10-17	2018	10	4

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

—> Most of the orders are placed at Night (Between 19:00 - 05:59 hrs)
—> Second highest orders are placed in the Afternoon (Between 12:00 - 15:59 hrs)
—> Least orders are placed @ Dawn (Between 05:00 - 06:59 hrs)



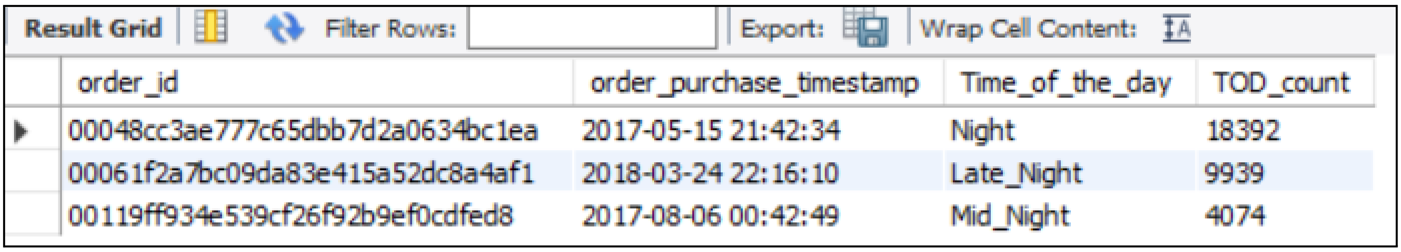
order_id	order_purchase_timestamp	Time_of_the_day	TOD_count
00048cc3ae777c65dbb7d2a0634bc1ea	2017-05-15 21:42:34	Night	33071
000229ec398224ef6ca0657da4fc703e	2018-01-14 14:33:31	Afternoon	25536
00010242fe8c5a6d1ba2dd792cb16214	2017-09-13 08:59:02	Morning	20507
0005a1a1728c9d785b8e2b08b904576c	2018-03-19 18:40:33	Evening	18594
0019c29108428acffd089c36103c9440	2018-03-06 06:40:28	Dawn	1733

Query:

```
SELECT *,
        COUNT(*) AS TOD_count
FROM (SELECT
        order_id,
        order_purchase_timestamp,
        CASE
        WHEN HOUR(TIME(order_purchase_timestamp)) BETWEEN 6 AND 7 THEN "Dawn"
        WHEN HOUR(TIME(order_purchase_timestamp)) BETWEEN 8 AND 11 THEN "Morning"
        WHEN HOUR(TIME(order_purchase_timestamp)) BETWEEN 12 AND 15 THEN "Afternoon"
        WHEN HOUR(TIME(order_purchase_timestamp)) BETWEEN 16 AND 18 THEN "Evening"
        ELSE "Night"
        END Time_of_the_day
        FROM target.orders) As x
GROUP BY x.Time_of_the_day
ORDER BY TOD_count DESC
```

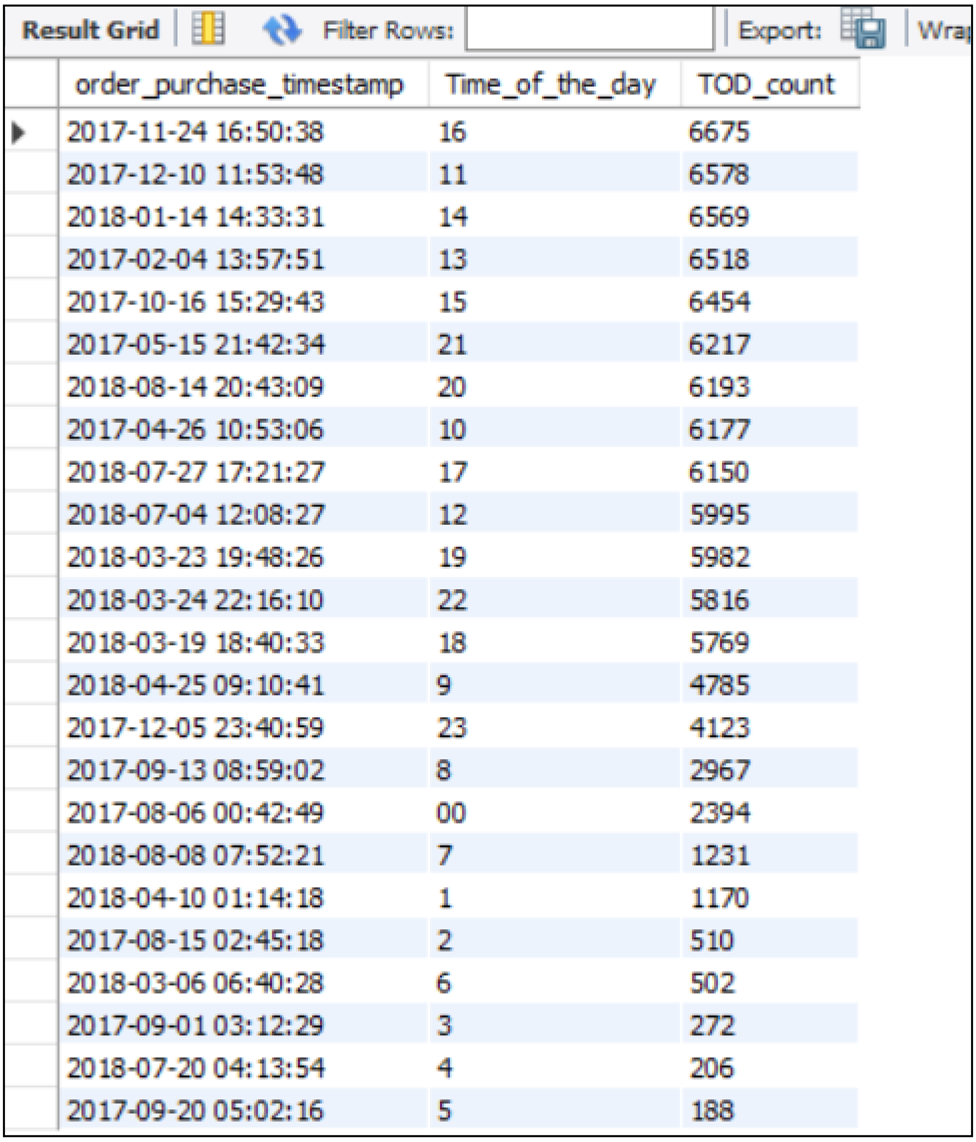
Further deep-diving into Night hours, below are the analysis:

→ Most of the orders placed in the nighttime are placed between 19:00 - 21:59 hrs.



order_id	order_purchase_timestamp	Time_of_the_day	TOD_count
00048cc3ae777c65dbb7d2a0634bc1ea	2017-05-15 21:42:34	Night	18392
00061f2a7bc09da83e415a52dc8a4af1	2018-03-24 22:16:10	Late_Night	9939
00119ff934e539cf26f92b9ef0cdfed8	2017-08-06 00:42:49	Mid_Night	4074

The hourly analysis says that most of the orders are placed between 13:00 - 16:59 hrs.



order_purchase_timestamp	Time_of_the_day	TOD_count
2017-11-24 16:50:38	16	6675
2017-12-10 11:53:48	11	6578
2018-01-14 14:33:31	14	6569
2017-02-04 13:57:51	13	6518
2017-10-16 15:29:43	15	6454
2017-05-15 21:42:34	21	6217
2018-08-14 20:43:09	20	6193
2017-04-26 10:53:06	10	6177
2018-07-27 17:21:27	17	6150
2018-07-04 12:08:27	12	5995
2018-03-23 19:48:26	19	5982
2018-03-24 22:16:10	22	5816
2018-03-19 18:40:33	18	5769
2018-04-25 09:10:41	9	4785
2017-12-05 23:40:59	23	4123
2017-09-13 08:59:02	8	2967
2017-08-06 00:42:49	00	2394
2018-08-08 07:52:21	7	1231
2018-04-10 01:14:18	1	1170
2017-08-15 02:45:18	2	510
2018-03-06 06:40:28	6	502
2017-09-01 03:12:29	3	272
2018-07-20 04:13:54	4	206
2017-09-20 05:02:16	5	188

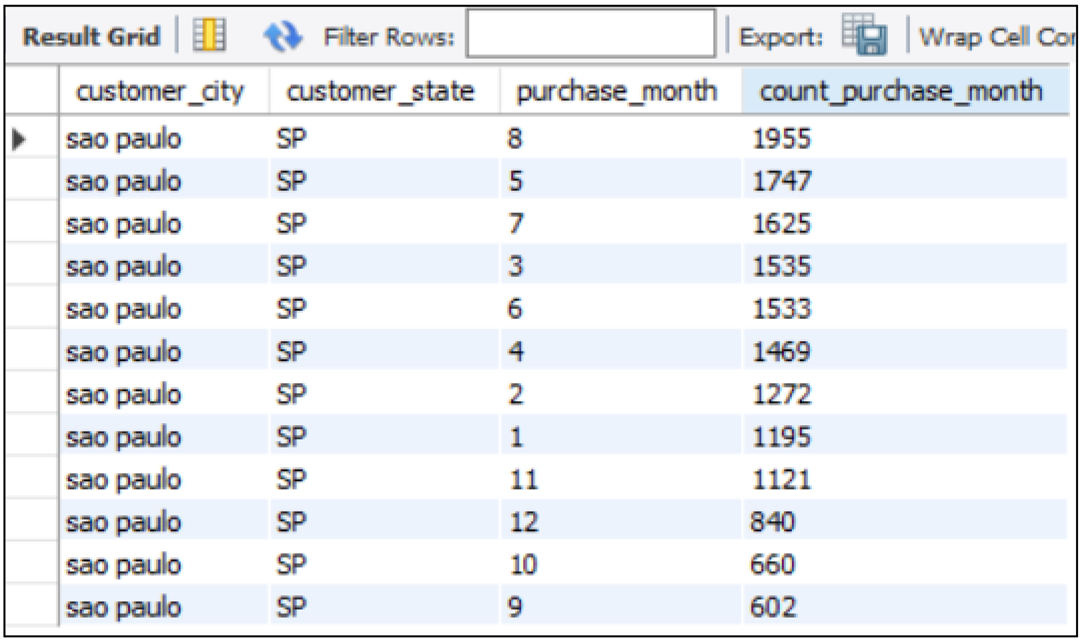
3. Evolution of E-commerce orders in the Brazil region:

3.1 Get month on month orders by region, states

Sao Paulo, SP:

—> In the city Sao Paulo,SP highest orders are placed in the month of August.

—> Least orders are placed in the month of September

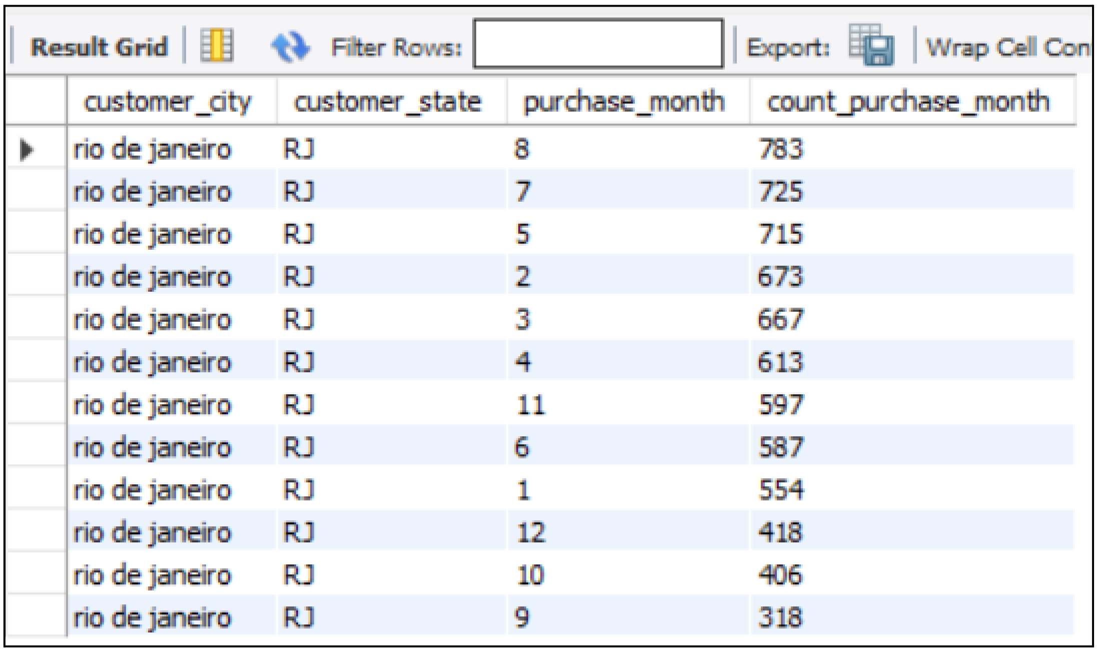


customer_city	customer_state	purchase_month	count_purchase_month
sao paulo	SP	8	1955
sao paulo	SP	5	1747
sao paulo	SP	7	1625
sao paulo	SP	3	1535
sao paulo	SP	6	1533
sao paulo	SP	4	1469
sao paulo	SP	2	1272
sao paulo	SP	1	1195
sao paulo	SP	11	1121
sao paulo	SP	12	840
sao paulo	SP	10	660
sao paulo	SP	9	602

Rio de Janerio, RJ:

—> In the city of Rio de Janerio, RJ highest orders are placed in the month of August.

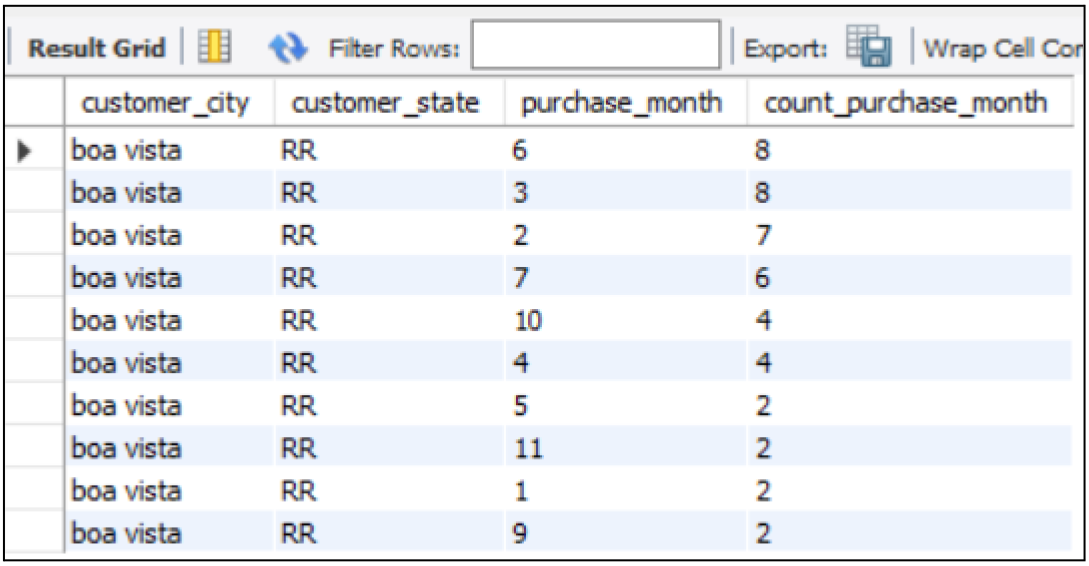
—> Least orders are placed in the month of September



customer_city	customer_state	purchase_month	count_purchase_month
rio de janeiro	RJ	8	783
rio de janeiro	RJ	7	725
rio de janeiro	RJ	5	715
rio de janeiro	RJ	2	673
rio de janeiro	RJ	3	667
rio de janeiro	RJ	4	613
rio de janeiro	RJ	11	597
rio de janeiro	RJ	6	587
rio de janeiro	RJ	1	554
rio de janeiro	RJ	12	418
rio de janeiro	RJ	10	406
rio de janeiro	RJ	9	318

Boa Vista, RR:

—> In the city of Boa Vista, RR highest orders are placed in the month of June.
—> Least orders are placed in the month of September.



customer_city	customer_state	purchase_month	count_purchase_month
boa vista	RR	6	8
boa vista	RR	3	8
boa vista	RR	2	7
boa vista	RR	7	6
boa vista	RR	10	4
boa vista	RR	4	4
boa vista	RR	5	2
boa vista	RR	11	2
boa vista	RR	1	2
boa vista	RR	9	2

Query:

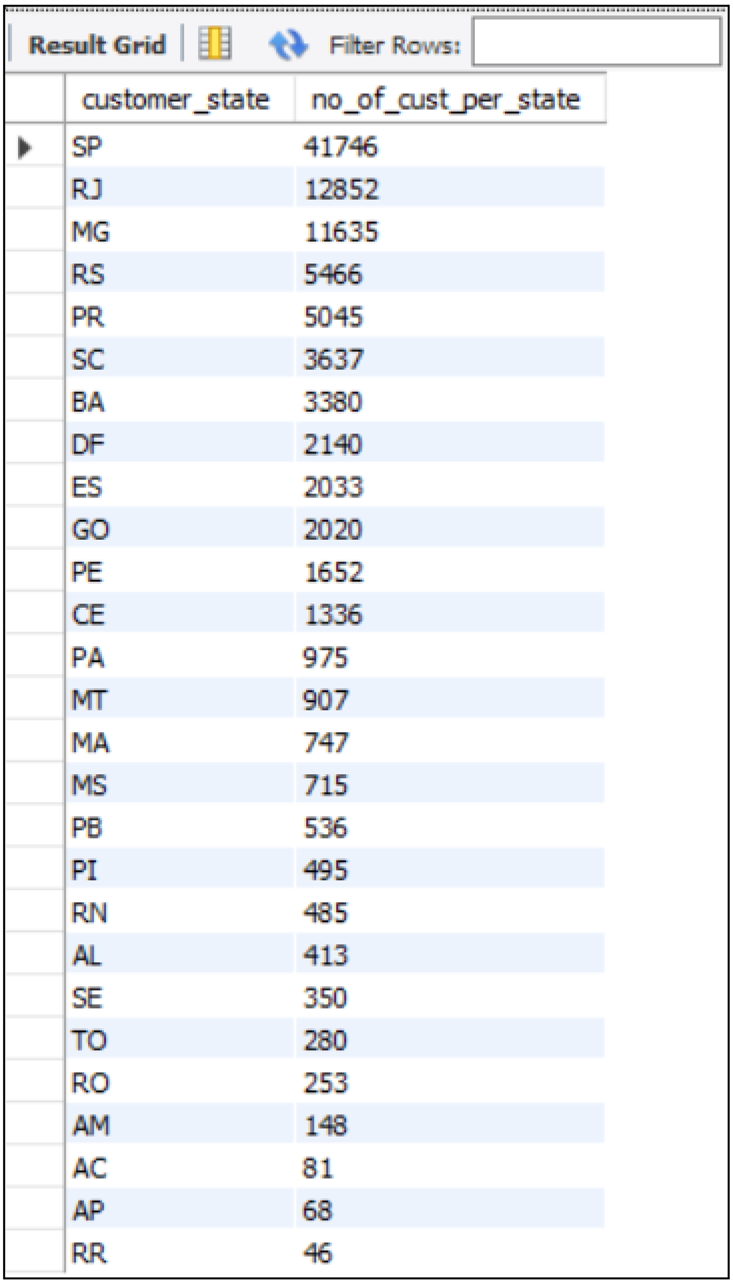
```
SELECT
        x.customer_city,
        x.customer_state,
        x.purchase_month,
        COUNT(x.purchase_month) AS count_purchase_month
FROM (SELECT
        c.customer_id,
        order_purchase_timestamp,
        customer_city,
        customer_state,
        MONTH(DATE_FORMAT(order_purchase_timestamp, '%Y-%m-%d')) "purchase_month"
        FROM target.customers AS c
                LEFT JOIN target.orders AS o
                ON c.customer_id = o.customer_id) AS x
WHERE x.customer_state = "RR" AND x.customer_city LIKE "boa%"
GROUP BY x.purchase_month
ORDER BY count_purchase_month DESC
```

3.2. How are customers distributed in Brazil:

→ Major customers are from Sau Paulo,Rio De Janerio and Minas Gerais(67% of total customers)

Query:

```
SELECT customer_state,
        COUNT(customer_state) AS no_of_cust_per_state
FROM target.customers
GROUP BY customer_state
ORDER BY no_of_cust_per_state DESC
```

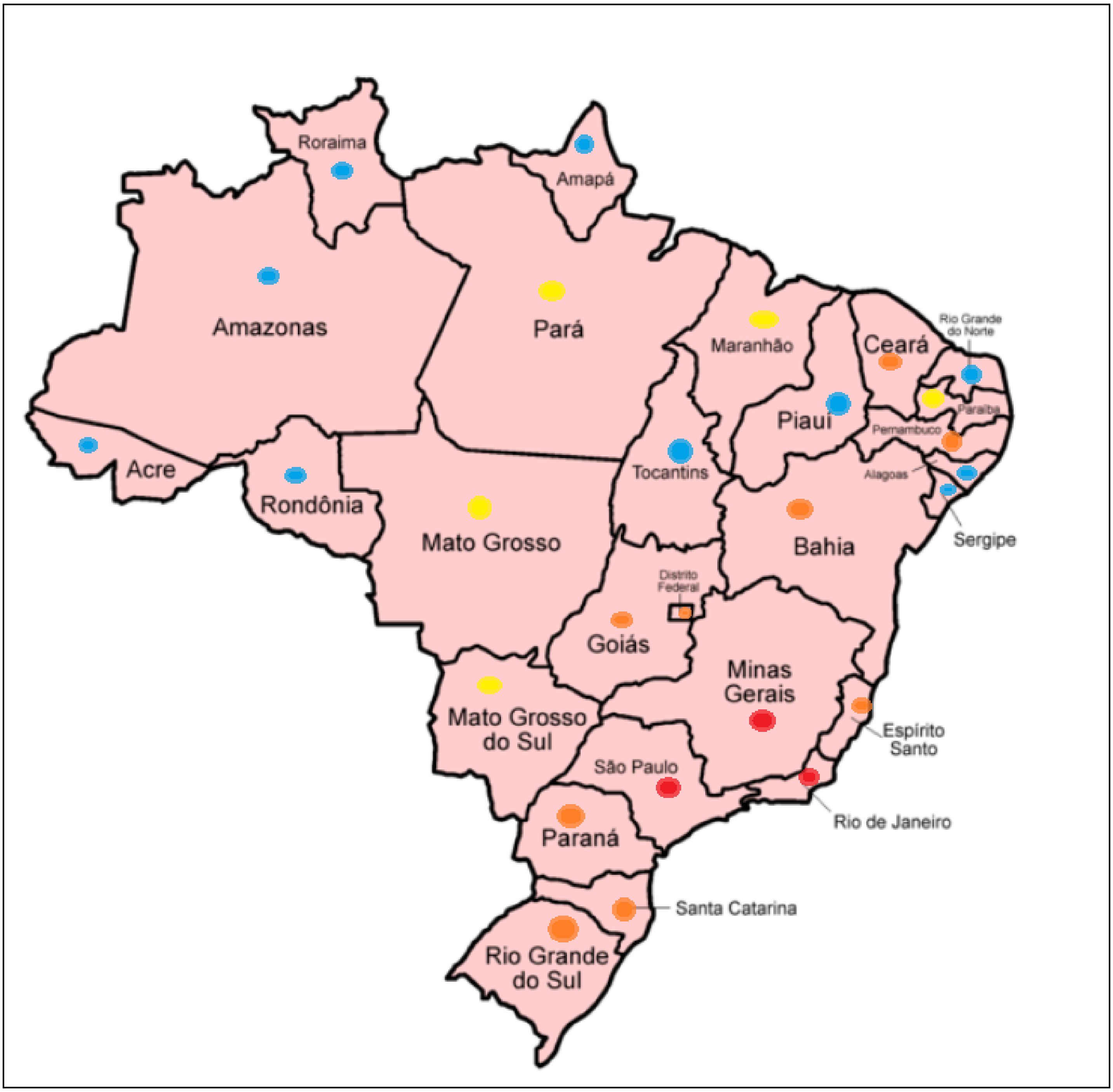


customer_state	no_of_cust_per_state
SP	41746
RJ	12852
MG	11635
RS	5466
PR	5045
SC	3637
BA	3380
DF	2140
ES	2033
GO	2020
PE	1652
CE	1336
PA	975
MT	907
MA	747
MS	715
PB	536
PI	495
RN	485
AL	413
SE	350
TO	280
RO	253
AM	148
AC	81
AP	68
RR	46

State-wise distribution of Customers in Brazil:

Red dot - Represents the state with the highest no. of customers, which is **66%** of the total customers
Orange dot - Represents the state which has **27%** of the total customers
Yellow dot - Represents the state which has **4%** of the total customers
Blue Dot - Represents the state which has **3%** of the total customers

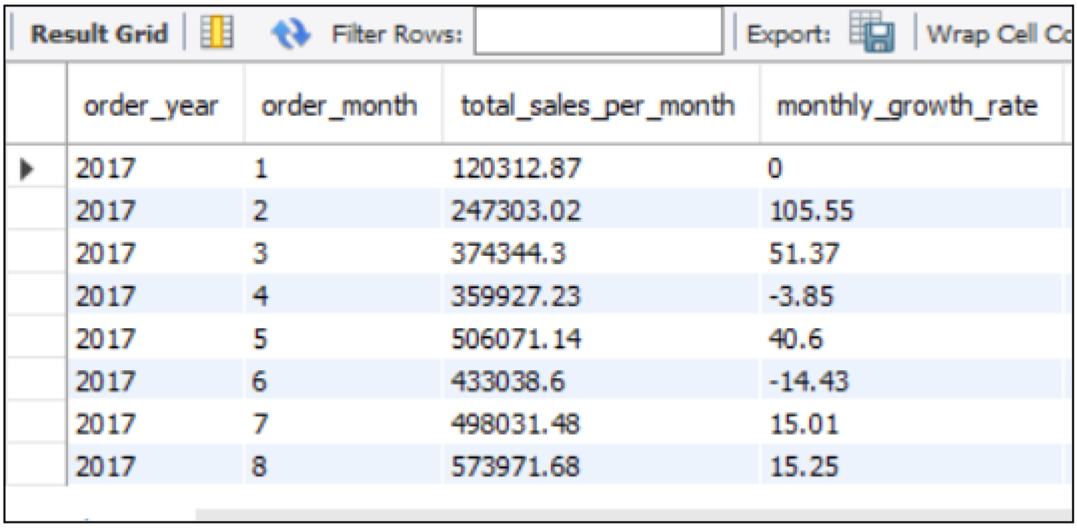
Customer distribution Map



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

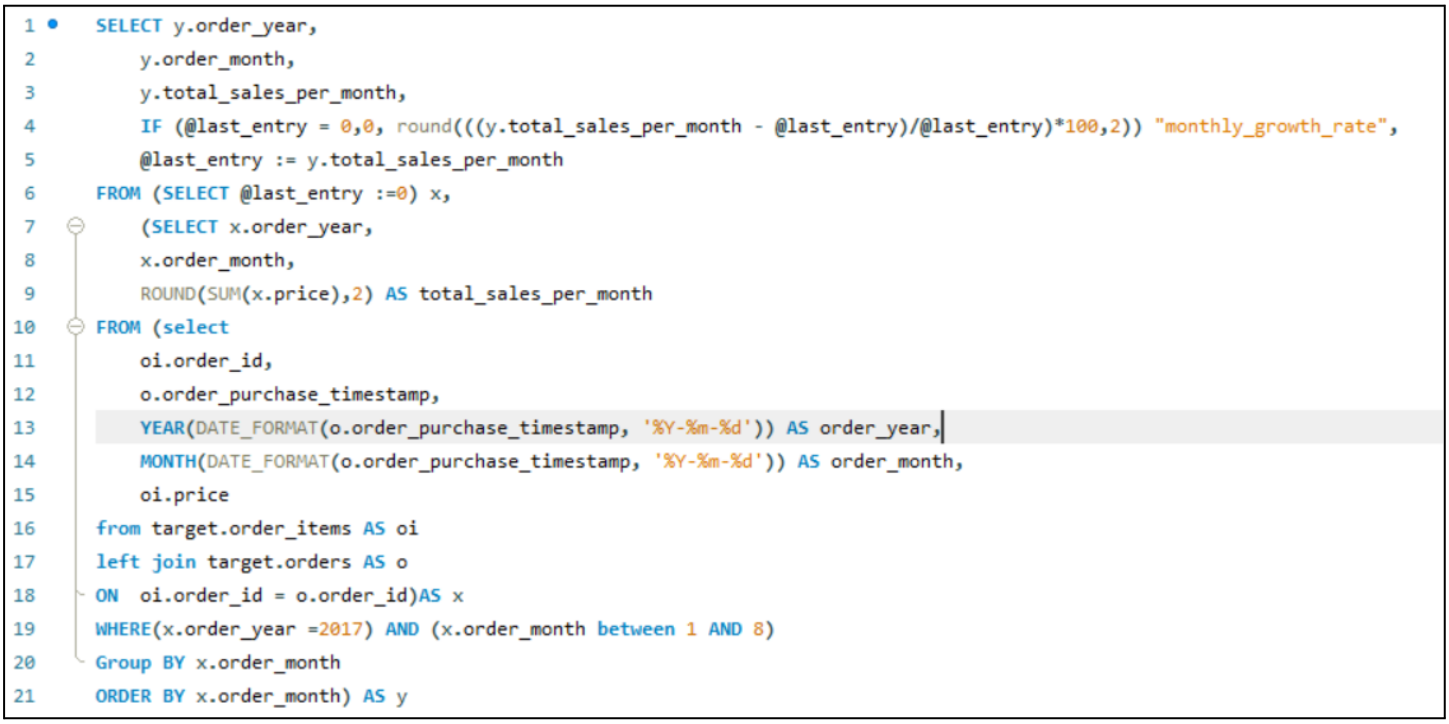
4.1. Percentage increase in the cost of orders from 2017 to 2018 (between Jan to Aug)

The year 2017, total sales per month and monthly growth rate (month between Jan and Aug):
→ The highest growth in total sales is seen in the month of February.
→ In June, the sales went down by 14.43%, which is the biggest fall in the 8 months.



order_year	order_month	total_sales_per_month	monthly_growth_rate
2017	1	120312.87	0
2017	2	247303.02	105.55
2017	3	374344.3	51.37
2017	4	359927.23	-3.85
2017	5	506071.14	40.6
2017	6	433038.6	-14.43
2017	7	498031.48	15.01
2017	8	573971.68	15.25

Query:

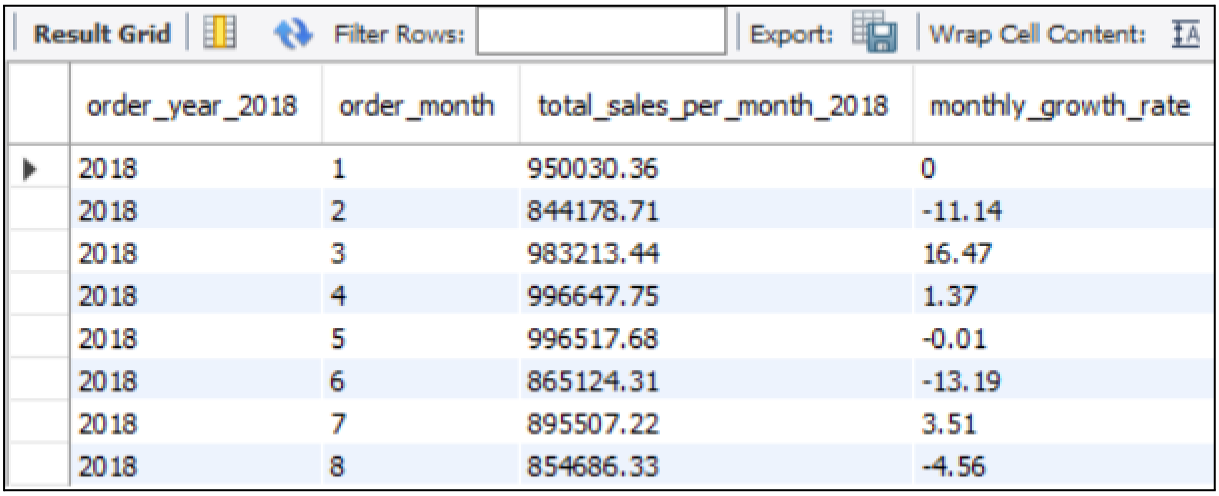


```
SELECT y.order_year,
    y.order_month,
    y.total_sales_per_month,
    IF (@last_entry = 0,0, round(((y.total_sales_per_month - @last_entry)/@last_entry)*100,2)) "monthly_growth_rate",
    @last_entry := y.total_sales_per_month
FROM (SELECT @last_entry :=0) x,
    (SELECT x.order_year,
    x.order_month,
    ROUND(SUM(x.price),2) AS total_sales_per_month
FROM (select
    oi.order_id,
    o.order_purchase_timestamp,
    YEAR(DATE_FORMAT(o.order_purchase_timestamp, '%Y-%m-%d')) AS order_year,
    MONTH(DATE_FORMAT(o.order_purchase_timestamp, '%Y-%m-%d')) AS order_month,
    oi.price
from target.order_items AS oi
left join target.orders AS o
ON  oi.order_id = o.order_id)AS x
WHERE(x.order_year =2017) AND (x.order_month between 1 AND 8)
Group BY x.order_month
ORDER BY x.order_month) AS y
```

The year 2018, total sales per month and monthly growth rate (month between Jan and Aug):

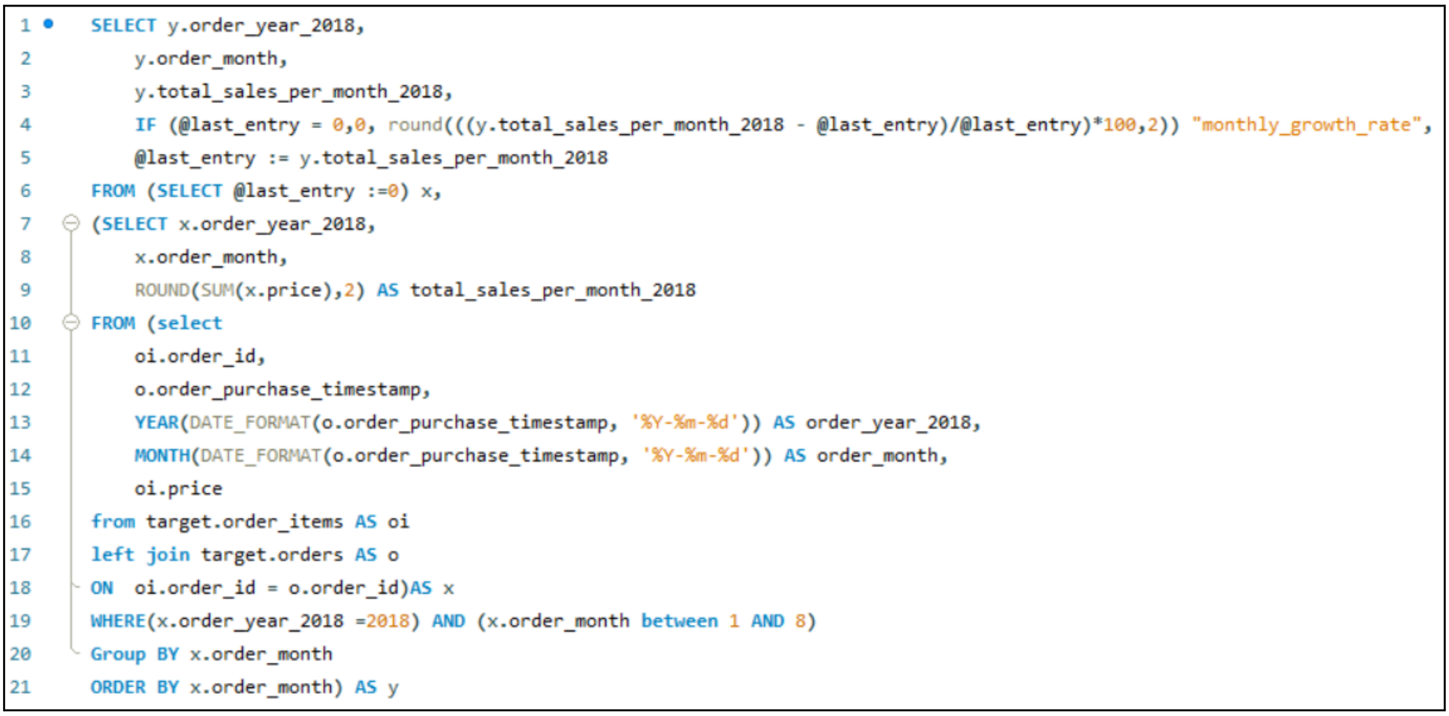
→ The highest growth in total sales is seen in the month of March.

→ In June, the sales went down by 14.43%, which is the biggest fall in the 8 months.



order_year_2018	order_month	total_sales_per_month_2018	monthly_growth_rate
2018	1	950030.36	0
2018	2	844178.71	-11.14
2018	3	983213.44	16.47
2018	4	996647.75	1.37
2018	5	996517.68	-0.01
2018	6	865124.31	-13.19
2018	7	895507.22	3.51
2018	8	854686.33	-4.56

Query:

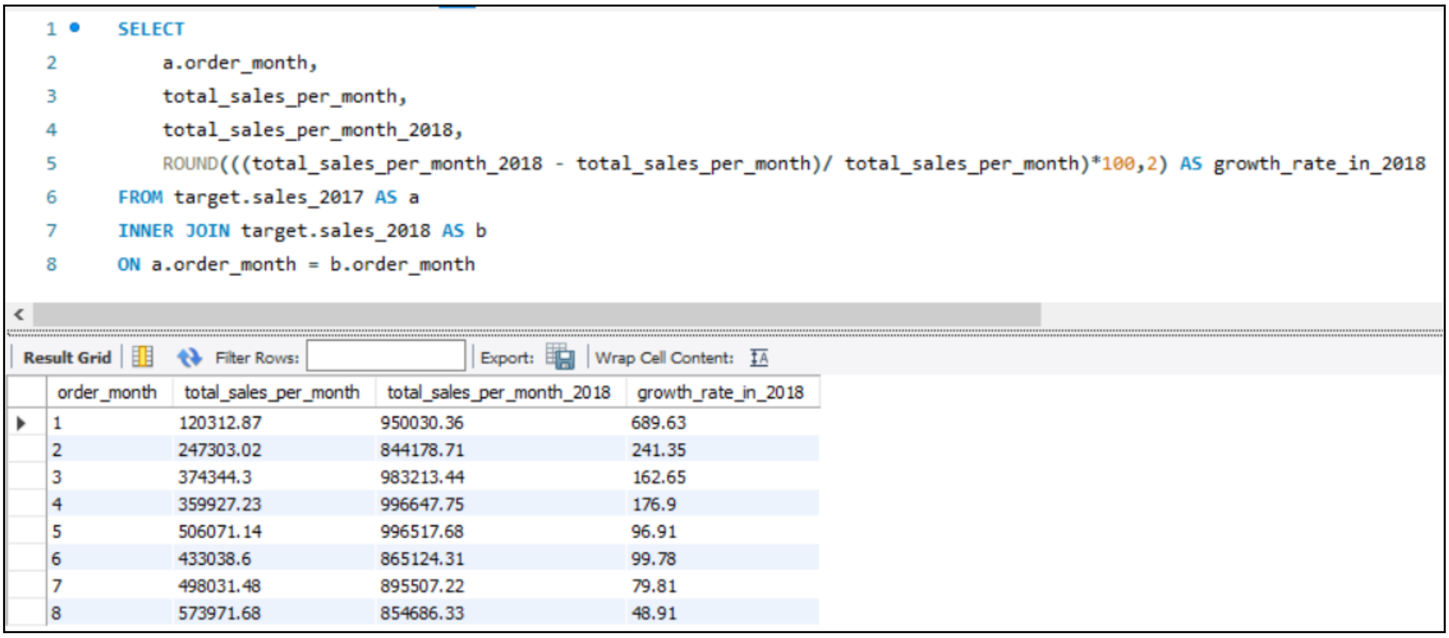


```sql
SELECT y.order_year_2018,
    y.order_month,
    y.total_sales_per_month_2018,
    IF (@last_entry = 0,0, round(((y.total_sales_per_month_2018 - @last_entry)/@last_entry)*100,2)) "monthly_growth_rate",
    @last_entry := y.total_sales_per_month_2018
FROM (SELECT @last_entry :=0) x,
(SELECT x.order_year_2018,
    x.order_month,
    ROUND(SUM(x.price),2) AS total_sales_per_month_2018
FROM (select
    oi.order_id,
    o.order_purchase_timestamp,
    YEAR(DATE_FORMAT(o.order_purchase_timestamp, '%Y-%m-%d')) AS order_year_2018,
    MONTH(DATE_FORMAT(o.order_purchase_timestamp, '%Y-%m-%d')) AS order_month,
    oi.price
from target.order_items AS oi
left join target.orders AS o
ON  oi.order_id = o.order_id)AS x
WHERE(x.order_year_2018 =2018) AND (x.order_month between 1 AND 8)
Group BY x.order_month
ORDER BY x.order_month) AS y
```

Monthly comparison for the years 2017 and 2018 (months between Jan and Aug):

1. Created View for the year 2017 monthly sales AS sales_2017
2. Created View for the year 2018 monthly sales AS sales_2018
3. Joined two Views ON month
4. Calculated the monthly growth in comparison to the previous year i.e growth in a particular month compared to the last year (2017)

Query and Result:



```
SELECT
    a.order_month,
    total_sales_per_month,
    total_sales_per_month_2018,
    ROUND(((total_sales_per_month_2018 - total_sales_per_month)/ total_sales_per_month)*100,2) AS growth_rate_in_2018
FROM target.sales_2017 AS a
INNER JOIN target.sales_2018 AS b
ON a.order_month = b.order_month
```

order_month	total_sales_per_month	total_sales_per_month_2018	growth_rate_in_2018
1	120312.87	950030.36	689.63
2	247303.02	844178.71	241.35
3	374344.3	983213.44	162.65
4	359927.23	996647.75	176.9
5	506071.14	996517.68	96.91
6	433038.6	865124.31	99.78
7	498031.48	895507.22	79.81
8	573971.68	854686.33	48.91

Analysis:

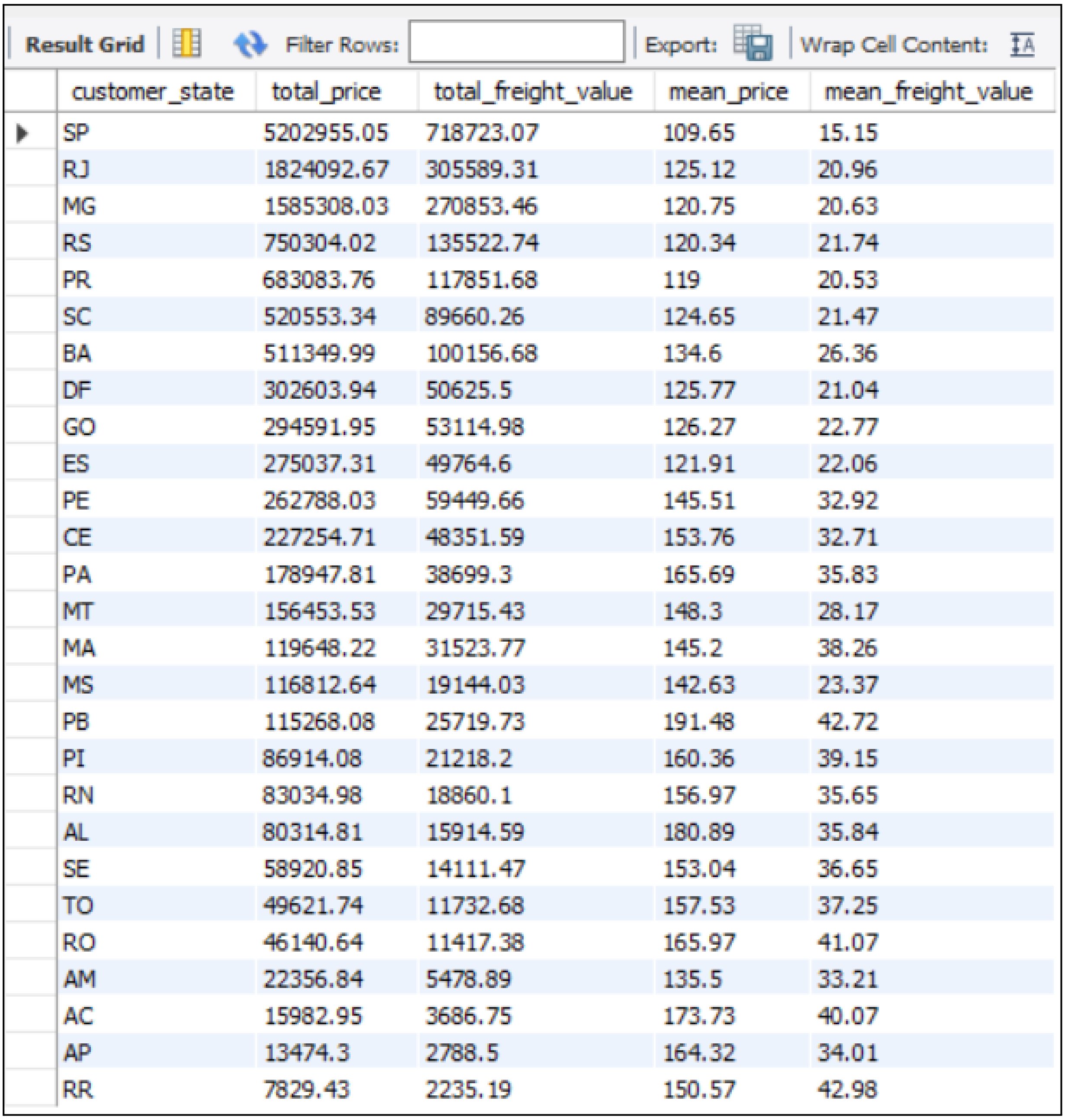
→ The highest growth is seen in January 2018 compared to January 2017.
→ Second highest growth is seen in February 2018 compared to February 2017.
→ The least growth is seen in August 2018 compared to August 2017.

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

Query:

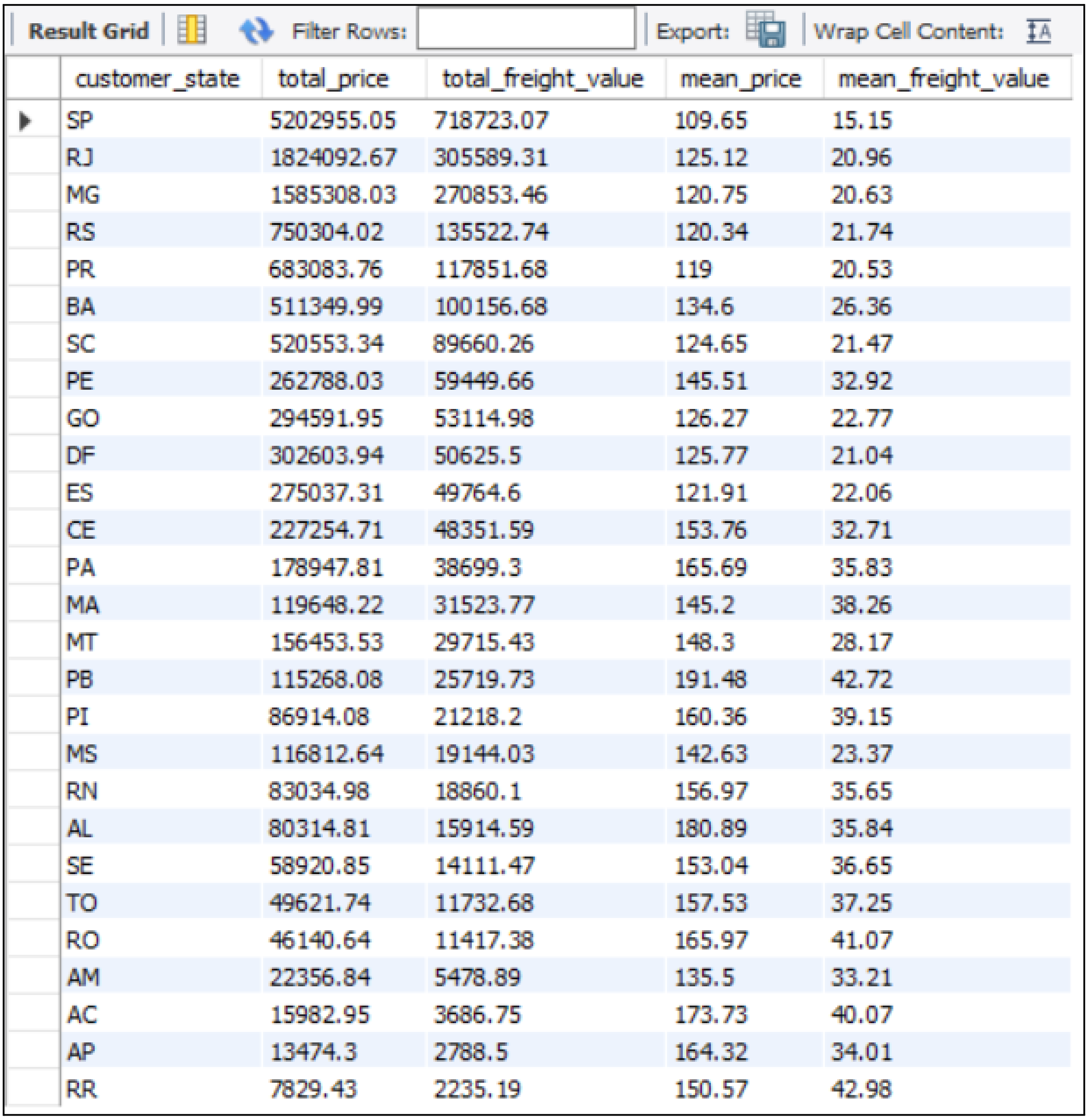
```
SELECT
        customer_state,
        ROUND(SUM(price),2) AS total_price,
        ROUND(SUM(freight_value),2) AS total_freight_value,
        ROUND(AVG(price),2) AS mean_price,
        ROUND(AVG(freight_value),2) AS mean_freight_value
FROM target.customers AS c
        LEFT JOIN target.orders AS o
        ON c.customer_id = o.customer_id
        LEFT JOIN target.order_items AS oi
        ON o.order_id = oi.order_id
GROUP BY customer_state
ORDER BY total_price DESC
```

Sorted by total price (high to low)



customer_state	total_price	total_freight_value	mean_price	mean_freight_value
SP	5202955.05	718723.07	109.65	15.15
RJ	1824092.67	305589.31	125.12	20.96
MG	1585308.03	270853.46	120.75	20.63
RS	750304.02	135522.74	120.34	21.74
PR	683083.76	117851.68	119	20.53
SC	520553.34	89660.26	124.65	21.47
BA	511349.99	100156.68	134.6	26.36
DF	302603.94	50625.5	125.77	21.04
GO	294591.95	53114.98	126.27	22.77
ES	275037.31	49764.6	121.91	22.06
PE	262788.03	59449.66	145.51	32.92
CE	227254.71	48351.59	153.76	32.71
PA	178947.81	38699.3	165.69	35.83
MT	156453.53	29715.43	148.3	28.17
MA	119648.22	31523.77	145.2	38.26
MS	116812.64	19144.03	142.63	23.37
PB	115268.08	25719.73	191.48	42.72
PI	86914.08	21218.2	160.36	39.15
RN	83034.98	18860.1	156.97	35.65
AL	80314.81	15914.59	180.89	35.84
SE	58920.85	14111.47	153.04	36.65
TO	49621.74	11732.68	157.53	37.25
RO	46140.64	11417.38	165.97	41.07
AM	22356.84	5478.89	135.5	33.21
AC	15982.95	3686.75	173.73	40.07
AP	13474.3	2788.5	164.32	34.01
RR	7829.43	2235.19	150.57	42.98

Sorted by total freight value (high to low)



customer_state	total_price	total_freight_value	mean_price	mean_freight_value
SP	5202955.05	718723.07	109.65	15.15
RJ	1824092.67	305589.31	125.12	20.96
MG	1585308.03	270853.46	120.75	20.63
RS	750304.02	135522.74	120.34	21.74
PR	683083.76	117851.68	119	20.53
BA	511349.99	100156.68	134.6	26.36
SC	520553.34	89660.26	124.65	21.47
PE	262788.03	59449.66	145.51	32.92
GO	294591.95	53114.98	126.27	22.77
DF	302603.94	50625.5	125.77	21.04
ES	275037.31	49764.6	121.91	22.06
CE	227254.71	48351.59	153.76	32.71
PA	178947.81	38699.3	165.69	35.83
MA	119648.22	31523.77	145.2	38.26
MT	156453.53	29715.43	148.3	28.17
PB	115268.08	25719.73	191.48	42.72
PI	86914.08	21218.2	160.36	39.15
MS	116812.64	19144.03	142.63	23.37
RN	83034.98	18860.1	156.97	35.65
AL	80314.81	15914.59	180.89	35.84
SE	58920.85	14111.47	153.04	36.65
TO	49621.74	11732.68	157.53	37.25
RO	46140.64	11417.38	165.97	41.07
AM	22356.84	5478.89	135.5	33.21
AC	15982.95	3686.75	173.73	40.07
AP	13474.3	2788.5	164.32	34.01
RR	7829.43	2235.19	150.57	42.98

5. Analysis on sales, freight and delivery time

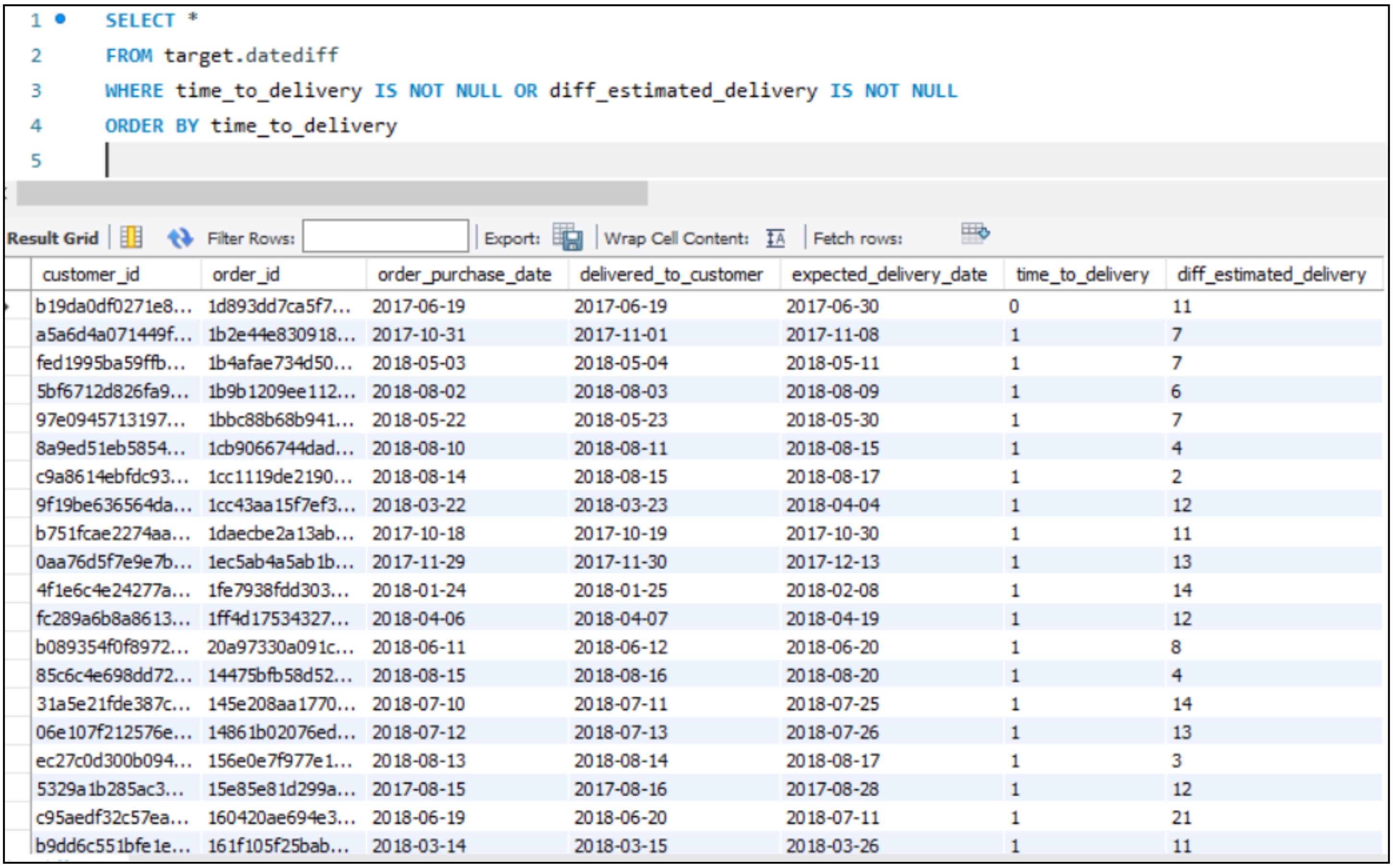
5.1. Calculate days between purchasing, delivering and estimated delivery

Created a view:

Query:

```
CREATE VIEW target.datediff AS
SELECT
        customer_id,
        order_id,
        order_purchase_date,
        delivered_to_customer,
        expected_delivery_date,
        DATEDIFF(delivered_to_customer, order_purchase_date ) AS time_to_delivery,
        DATEDIFF(expected_delivery_date, delivered_to_customer ) AS diff_estimated_delivery
FROM target.order_schedule
```

5.2. Query:



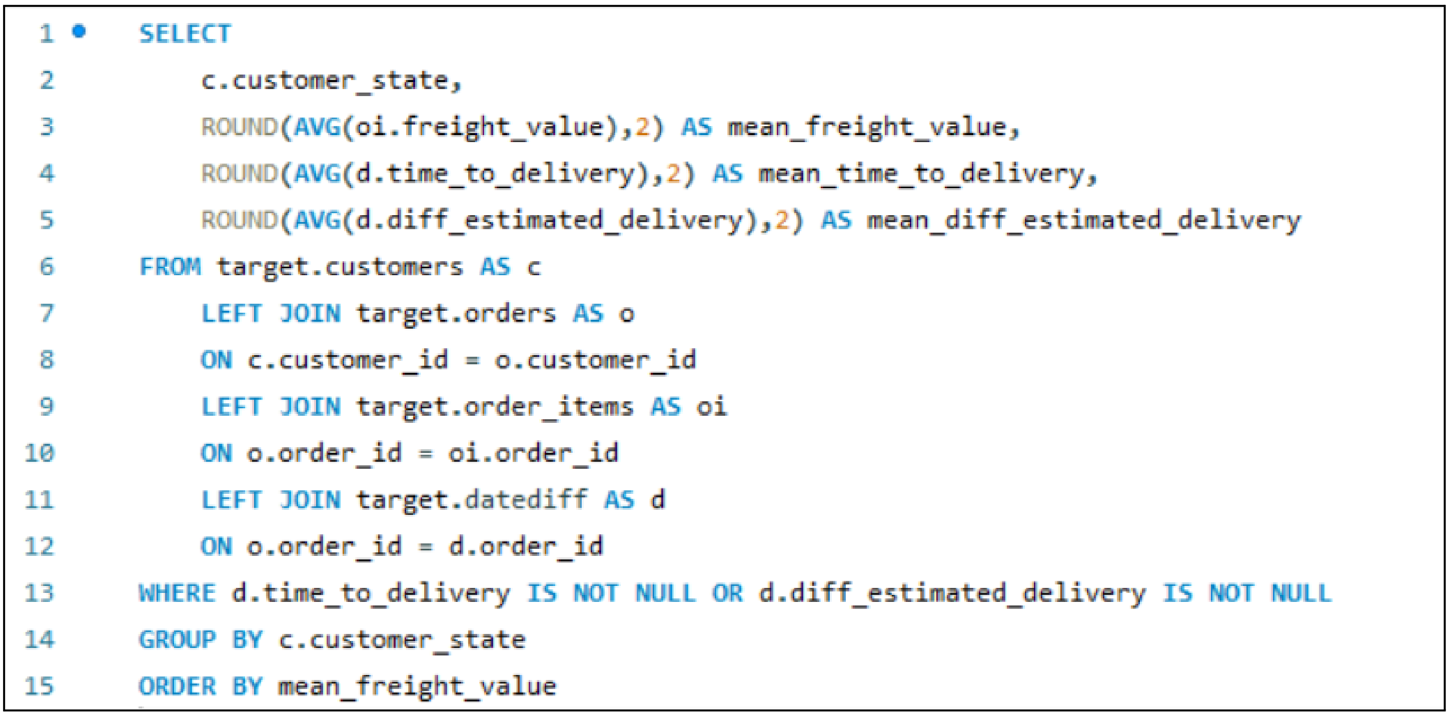
```
SELECT *
FROM target.datediff
WHERE time_to_delivery IS NOT NULL OR diff_estimated_delivery IS NOT NULL
ORDER BY time_to_delivery
```

customer_id	order_id	order_purchase_date	delivered_to_customer	expected_delivery_date	time_to_delivery	diff_estimated_delivery
b19da0df0271e8...	1d893dd7ca5f7...	2017-06-19	2017-06-19	2017-06-30	0	11
a5a6d4a071449f...	1b2e44e830918...	2017-10-31	2017-11-01	2017-11-08	1	7
fed1995ba59ffb...	1b4afae734d50...	2018-05-03	2018-05-04	2018-05-11	1	7
5bf6712d826fa9...	1b9b1209ee112...	2018-08-02	2018-08-03	2018-08-09	1	6
97e0945713197...	1bbc88b68b941...	2018-05-22	2018-05-23	2018-05-30	1	7
8a9ed51eb5854...	1cb9066744dad...	2018-08-10	2018-08-11	2018-08-15	1	4
c9a8614ebfdc93...	1cc1119de2190...	2018-08-14	2018-08-15	2018-08-17	1	2
9f19be636564da...	1cc43aa15f7ef3...	2018-03-22	2018-03-23	2018-04-04	1	12
b751fcae2274aa...	1daecbe2a13ab...	2017-10-18	2017-10-19	2017-10-30	1	11
0aa76d5f7e9e7b...	1ec5ab4a5ab1b...	2017-11-29	2017-11-30	2017-12-13	1	13
4f1e6c4e24277a...	1fe7938fdd303...	2018-01-24	2018-01-25	2018-02-08	1	14
fc289a6b8a8613...	1ff4d17534327...	2018-04-06	2018-04-07	2018-04-19	1	12
b089354f0f8972...	20a97330a091c...	2018-06-11	2018-06-12	2018-06-20	1	8
85c6c4e698dd72...	14475bfb58d52...	2018-08-15	2018-08-16	2018-08-20	1	4
31a5e21fde387c...	145e208aa1770...	2018-07-10	2018-07-11	2018-07-25	1	14
06e107f212576e...	14861b02076ed...	2018-07-12	2018-07-13	2018-07-26	1	13
ec27c0d300b094...	156e0e7f977e1...	2018-08-13	2018-08-14	2018-08-17	1	3
5329a1b285ac3...	15e85e81d299a...	2017-08-15	2017-08-16	2017-08-28	1	12
c95aedf32c57ea...	160420ae694e3...	2018-06-19	2018-06-20	2018-07-11	1	21
b9dd6c551bfe1e...	161f105f25bab...	2018-03-14	2018-03-15	2018-03-26	1	11

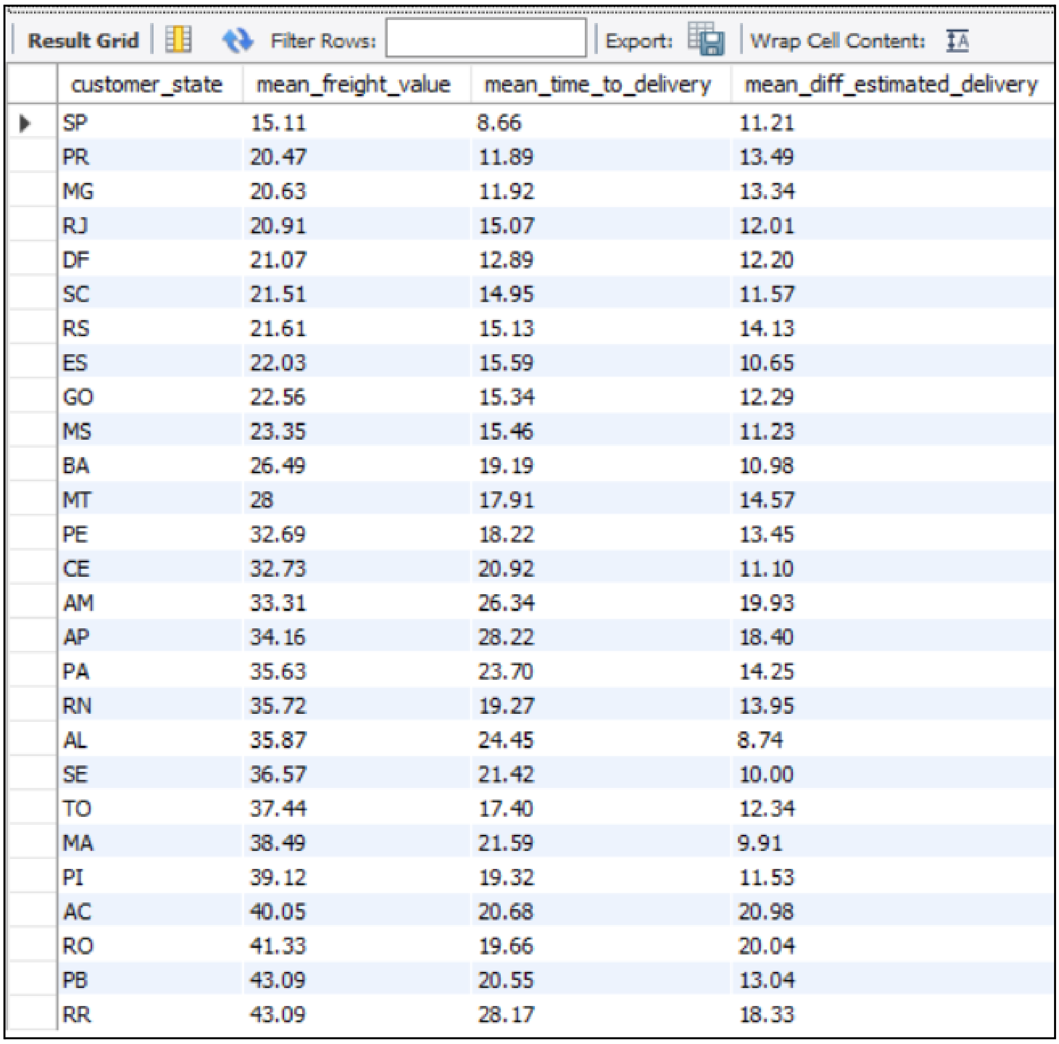
Total null values: 2965

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

Query & Result:

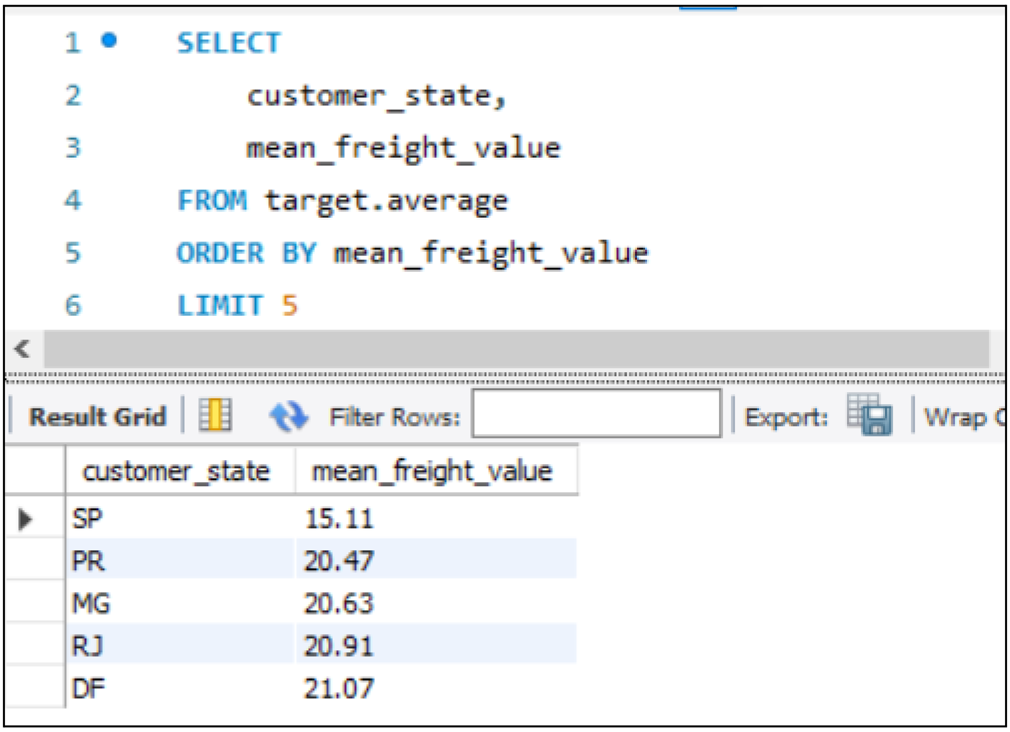


```
SELECT
    c.customer_state,
    ROUND(AVG(oi.freight_value),2) AS mean_freight_value,
    ROUND(AVG(d.time_to_delivery),2) AS mean_time_to_delivery,
    ROUND(AVG(d.diff_estimated_delivery),2) AS mean_diff_estimated_delivery
FROM target.customers AS c
    LEFT JOIN target.orders AS o
    ON c.customer_id = o.customer_id
    LEFT JOIN target.order_items AS oi
    ON o.order_id = oi.order_id
    LEFT JOIN target.datediff AS d
    ON o.order_id = d.order_id
WHERE d.time_to_delivery IS NOT NULL OR d.diff_estimated_delivery IS NOT NULL
GROUP BY c.customer_state
ORDER BY mean_freight_value
```



customer_state	mean_freight_value	mean_time_to_delivery	mean_diff_estimated_delivery
SP	15.11	8.66	11.21
PR	20.47	11.89	13.49
MG	20.63	11.92	13.34
RJ	20.91	15.07	12.01
DF	21.07	12.89	12.20
SC	21.51	14.95	11.57
RS	21.61	15.13	14.13
ES	22.03	15.59	10.65
GO	22.56	15.34	12.29
MS	23.35	15.46	11.23
BA	26.49	19.19	10.98
MT	28	17.91	14.57
PE	32.69	18.22	13.45
CE	32.73	20.92	11.10
AM	33.31	26.34	19.93
AP	34.16	28.22	18.40
PA	35.63	23.70	14.25
RN	35.72	19.27	13.95
AL	35.87	24.45	8.74
SE	36.57	21.42	10.00
TO	37.44	17.40	12.34
MA	38.49	21.59	9.91
PI	39.12	19.32	11.53
AC	40.05	20.68	20.98
RO	41.33	19.66	20.04
PB	43.09	20.55	13.04
RR	43.09	28.17	18.33

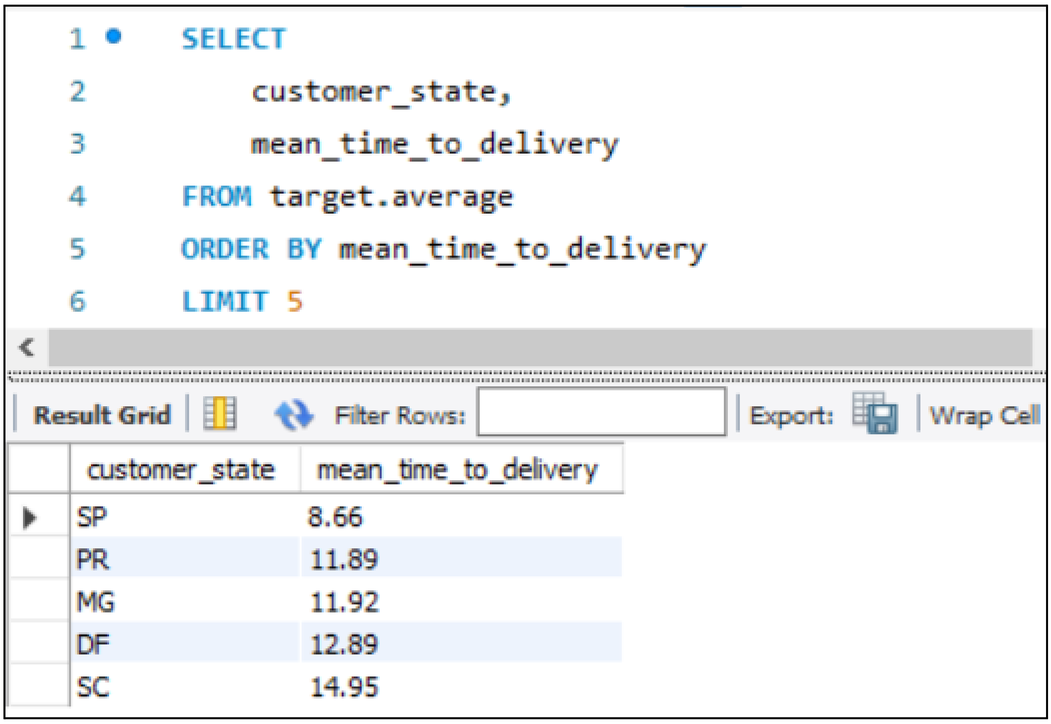
5.4.1. Top 5 states with lowest average freight value



```sql
SELECT
    customer_state,
    mean_freight_value
FROM target.average
ORDER BY mean_freight_value
LIMIT 5
```

customer_state	mean_freight_value
SP	15.11
PR	20.47
MG	20.63
RJ	20.91
DF	21.07

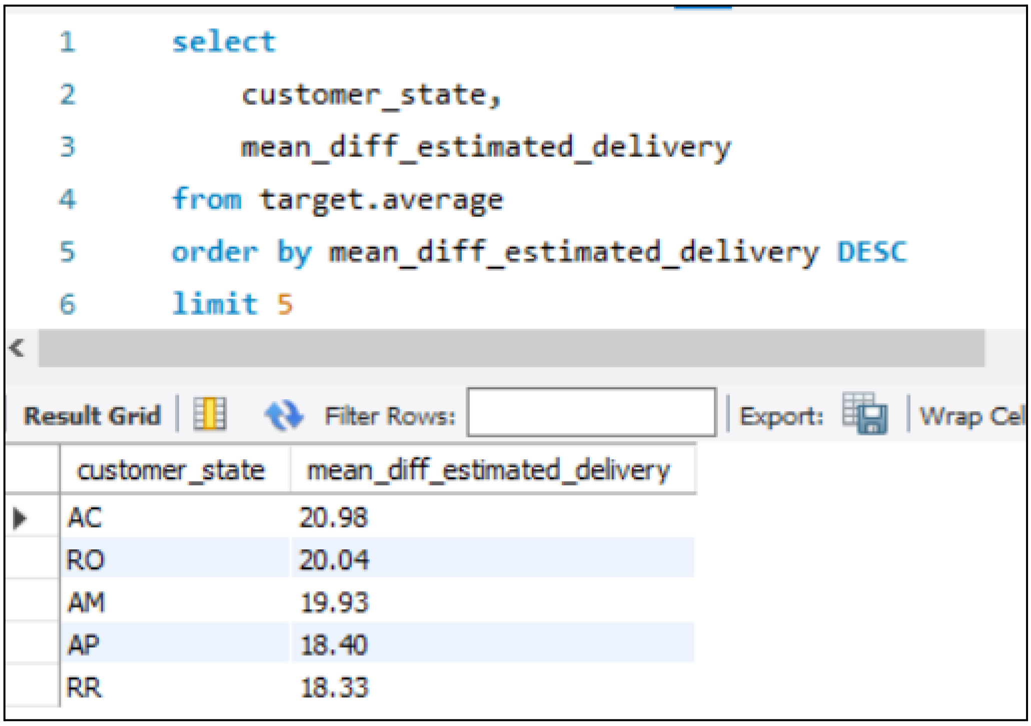
5.4.2. Top 5 states with the lowest average time to delivery



```sql
SELECT
    customer_state,
    mean_time_to_delivery
FROM target.average
ORDER BY mean_time_to_delivery
LIMIT 5
```

customer_state	mean_time_to_delivery
SP	8.66
PR	11.89
MG	11.92
DF	12.89
SC	14.95

5.4.3. Top 5 states where delivery is really fast compared to estimated date



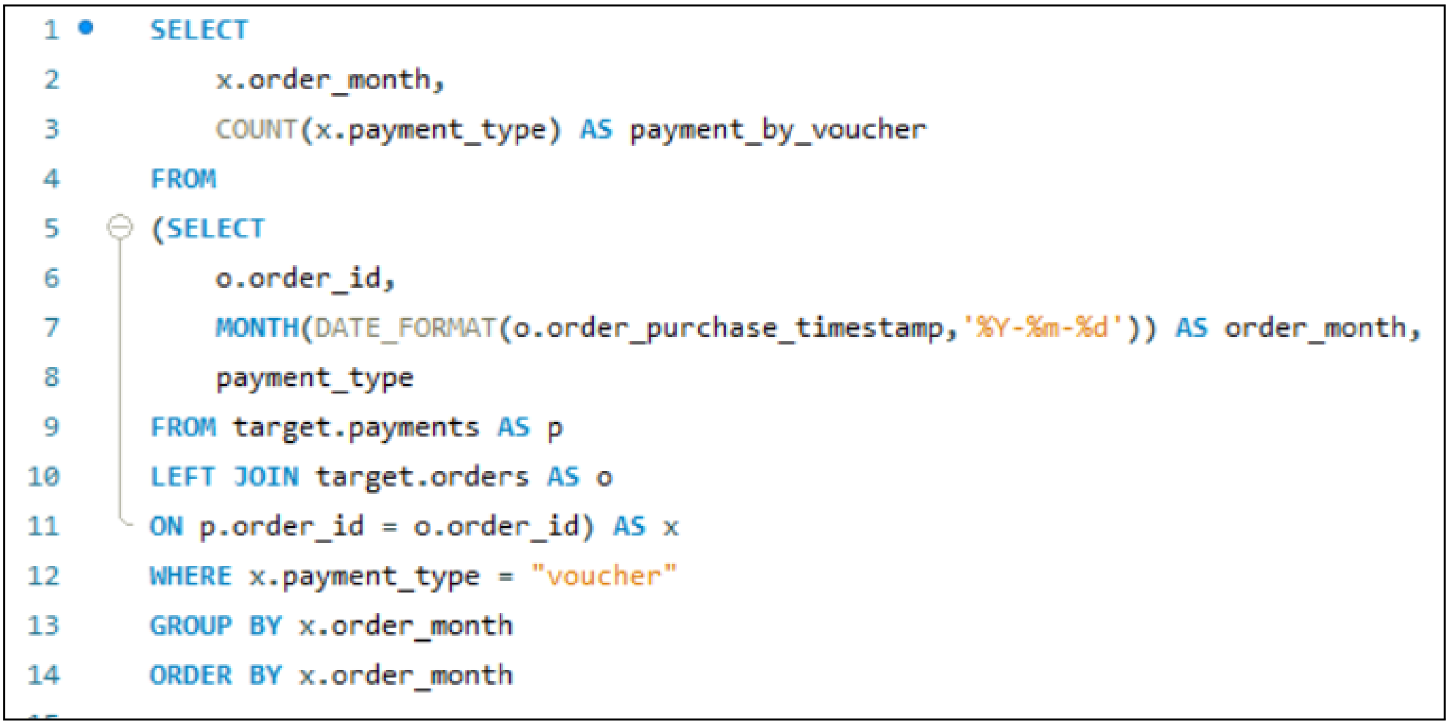
```
select
    customer_state,
    mean_diff_estimated_delivery
from target.average
order by mean_diff_estimated_delivery DESC
limit 5
```

customer_state	mean_diff_estimated_delivery
AC	20.98
RO	20.04
AM	19.93
AP	18.40
RR	18.33

6. **Payment Analysis:**

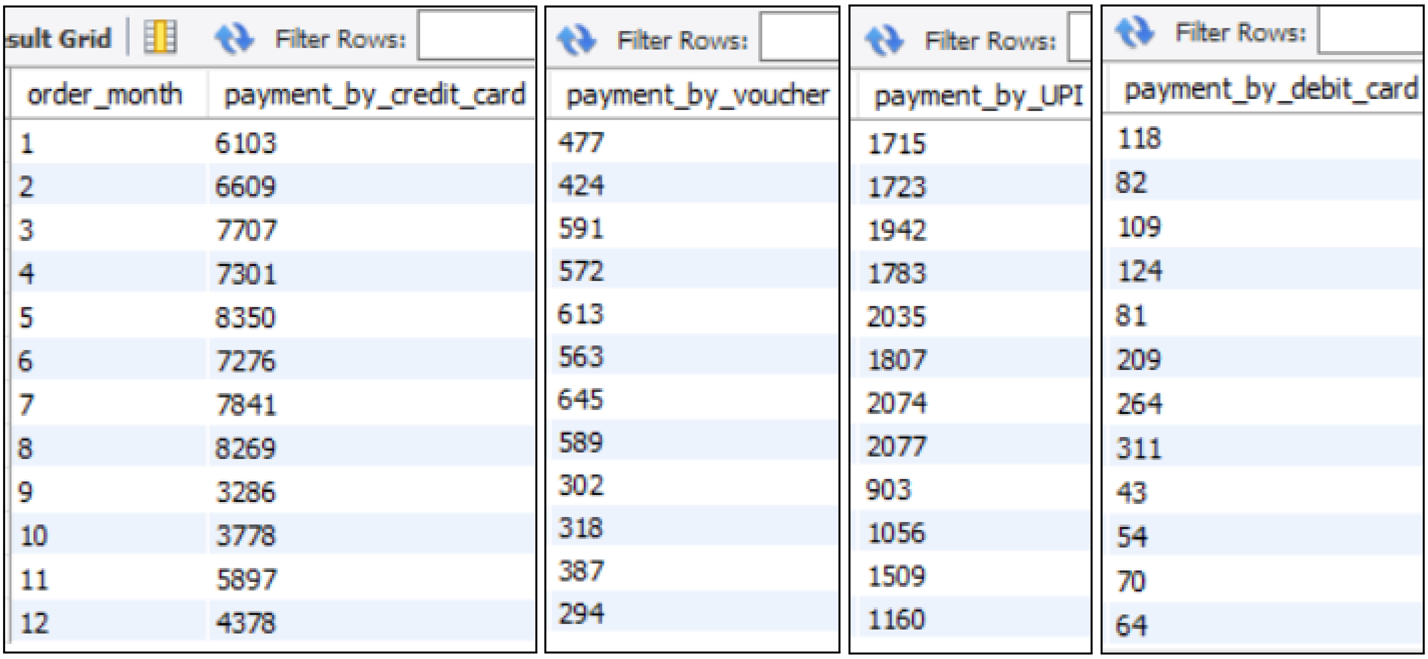
6.1. Month over Month count of orders for different payment types

Query for different payment types



```
SELECT
    x.order_month,
    COUNT(x.payment_type) AS payment_by_voucher
FROM
(SELECT
    o.order_id,
    MONTH(DATE_FORMAT(o.order_purchase_timestamp,'%Y-%m-%d')) AS order_month,
    payment_type
FROM target.payments AS p
LEFT JOIN target.orders AS o
ON p.order_id = o.order_id) AS x
WHERE x.payment_type = "voucher"
GROUP BY x.order_month
ORDER BY x.order_month
```

Result:

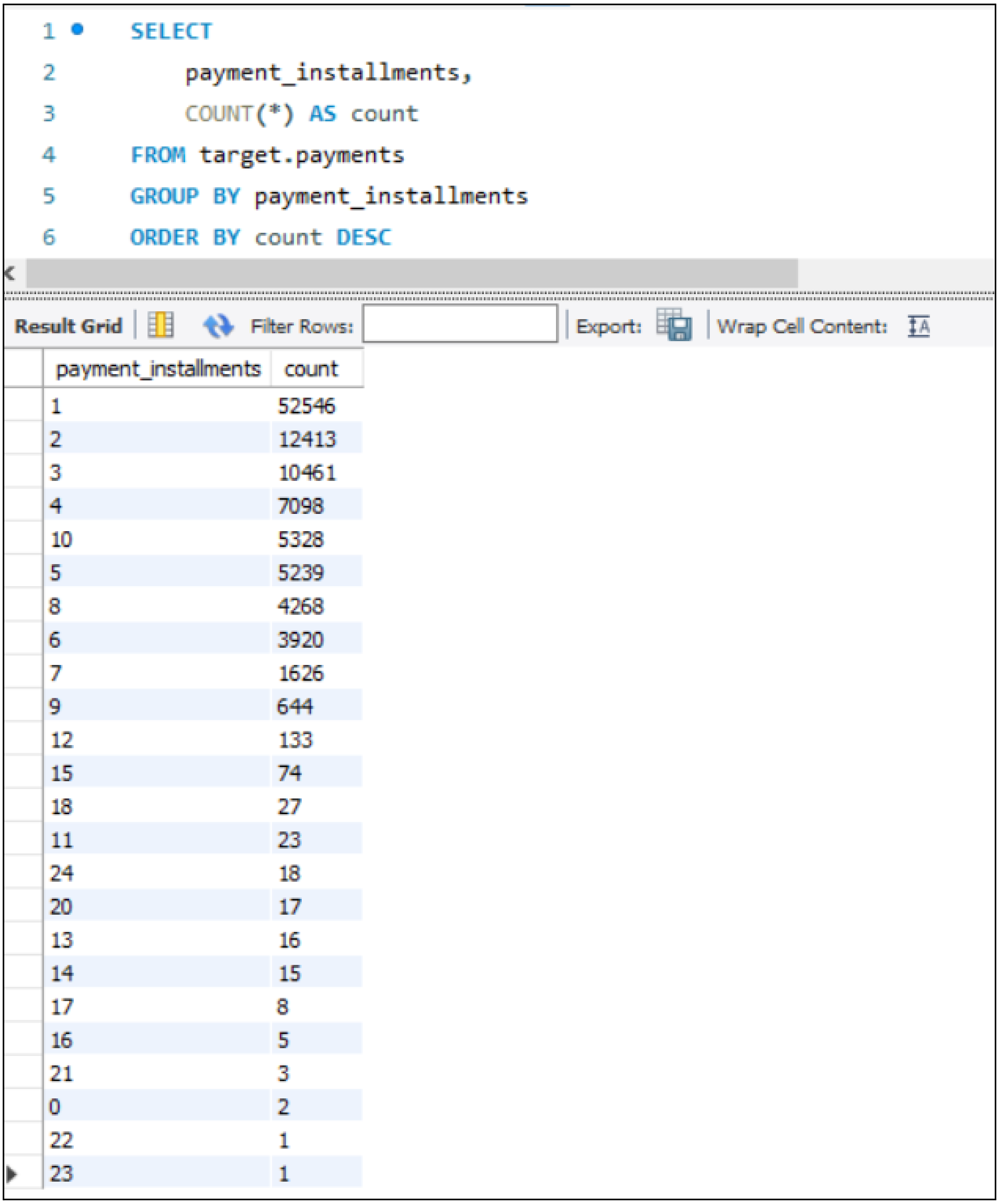


order_month	payment_by_credit_card	payment_by_voucher	payment_by_UPI	payment_by_debit_card
1	6103	477	1715	118
2	6609	424	1723	82
3	7707	591	1942	109
4	7301	572	1783	124
5	8350	613	2035	81
6	7276	563	1807	209
7	7841	645	2074	264
8	8269	589	2077	311
9	3286	302	903	43
10	3778	318	1056	54
11	5897	387	1509	70
12	4378	294	1160	64

—> 3 payment type is not defined (payment type - “not_defined”)

6.2. Distribution of payment installments and count of orders

Query and Result:



```
SELECT
    payment_installments,
    COUNT(*) AS count
FROM target.payments
GROUP BY payment_installments
ORDER BY count DESC
```

payment_installments	count
1	52546
2	12413
3	10461
4	7098
10	5328
5	5239
8	4268
6	3920
7	1626
9	644
12	133
15	74
18	27
11	23
24	18
20	17
13	16
14	15
17	8
16	5
21	3
0	2
22	1
23	1

Analysis and Recommendation based on the above insights:

Question no.	**Description on Analysis**	**Recommendation based on Analysis**
1.1	Target data has total 8 columns	-->Sau Paulo has the highest no. of orders and customers. Hence build warehouse near Sau Paulo which will reduce the delivery time also it will reduce the freight charges. --> 6-10% customers are from North-west of Brazil. Run some campaings, discounts fro this region in order to boost the sales. A warehouse in Northwest will speed up the delivery time and reduce the freight cost.
1.2	The data is for the year 2016,2017,2018 & 2020.	
	No records for the year 2019	
	Highest no.of orders where made in the year 2018 & Lowest in 2020	
1.3	Total no.of states - **27**	
	Most of the customers are from **Sau Paulo** state and **least from Roraima**	
	The customers are from **4119 distinct** cities of Brazil	
	Highest no. of the customers are from **Sao Paulo city - 15540 (count)**	
2.1	**2nd Quarter** has the highest purchase i.e **April, May & June, irrespective of the year**	1. Irrespective of the year September stands at the lowest in sales/orders --> As good sales is seen in August each year, a gift voucher with a validity of month would attract the customer to purchase in September. 2. January has the highest growth in orders. --> If some offers are offered in January then similar can be applied for December. Which will increase the sales in December as fall is seen December sale each year.
	Most of the customers tend to buy in the month of **August** and least in **September irrespective of the year.**	
	As per the trend, **the best year is 2017**, the growth is **more than 120x** of the previous year.	
	And in **2018 the growth is only 20%**, which is comparatively too low.	
	In the year 2016: → The highest purchase in 2016 was made in the month of **October**, then it dropped in **December.**	
	In the year 2017: → Purchases dropped in the month of **April, June, September, and December** compared to the previous month's purchases. → **In December**, the purchase's dropped the most compared to other months in comparison to the previous month's	

	purchases. ⟶ Highest growth rate is seen in **January 2017** **⟶ Highest growth was seen in the month of November purchases made compared to the previous month.**	
	In the year 2018: ⟶ The biggest fall in purchases is seen in **September**. ⟶ Highest purchases were done in **January**.	
2.2	—> Most of the orders are placed at **Night (Between 19:00 - 05:59 hrs)** —> Second highest orders are placed in the **Afternoon (Between 12:00 - 15:59 hrs)** —> Least orders are placed @ **Dawn (Between 05:00 - 06:59 hrs)**	1. Customers tend to purchase between between 1pm to 5pm. --> This 4hrs can we utilised for running online campaigns, advertisements on different platform to attract more customers. 2. People tend to spend more time online during this time of the day, which can be most target time for the company.
	Further deep-diving into Night hours, below are the analysis: ⟶ Most of the orders placed in the nighttime are placed between **19:00 - 21:59 hrs**.	
	The hourly analysis says that most of the orders are placed between **13:00 - 16:59 hrs.**	
3.1	**Sao Paulo, SP:** —> In the city Sao Paulo,SP highest orders are placed in the month of **August.** —> **Least orders are placed in the month of September**	1. Major Customers are from Sau Paulo, Rio De Janerio and the least orders are placed by them in September. --> Highly recommend to give vouchers with validity of 30 days, because most purchases are made in August. 2. These 3 states has the major chunk of customers, hence focus on bringing the same customers back in September 3. Most of customers are from the east of Brazil, and least from west. ---> Retain the existing customers in East and explore how to capture the west market.
	Rio de Janerio, RJ: —> In the city of Rio de Janerio, RJ highest orders are placed in the month of **August**. —> **Least orders are placed in the month of September**	
	Boa Vista, RR: —> In the city of Boa Vista, RR highest orders are placed in the month of **June**. —> **Least orders are placed in the month of September.**	
3.2	**How are customers distributed in Brazil:** ⟶ Major customers are from **Sau Paulo,Rio De Janerio and Minas Gerais(67% of total customers)**	

	Most customers are from **South-east of Brazil**, then from North-east and least from the west of Brazil.	
4.1	**The year 2017,** total sales per month and monthly growth rate (month between Jan and Aug): → The highest growth in total sales is seen in the month of **February**. → **In June, the sales went down by 14.43%**, which is the biggest fall in the 8 months.	Apply similar strategy of 1st quarter sales to 3rd quarter, as the 1st quarter has very good growth and downfall in 3rd quarter.
	The year 2018, total sales per month and monthly growth rate (month between Jan and Aug): → The highest growth in total sales is seen in the month of **March**. → **In June, the sales went down by 13.19%**, which is the biggest fall in the 8 months.	
	Monthly comparison for the years 2017 and 2018 (months between Jan and Aug): → The highest growth is seen in **January 2018** compared to January 2017. → Second highest growth is seen in February 2018 compared to February 2017. **→ The least growth is seen in August 2018 compared to August 2017.**	
4.2	**4.2. Mean & Sum of price and freight value by customer state** → Lowest average freight value is for **Sau Paulo**, as the number of orders is more. → Highest average freight value is for **Roraima**, as the number of orders is less.	
5.3	**Best average delivery time** is between **8 to 9 days** post the order is placed.	1. Strategise to keep 9-10 days as the deilevery time to customers post the order is made. 2. Decrease the estimated delivery time in west of Brazil.
	Worst average deilvery time is **28 days** and more post the order is placed	
	Maximum 20-21 days before on an average of the estimated delivery time is done	
	Minimum 8-9 days before on an average of the estimated delivery time is done	

5.4.1	**Top 5 states with lowest average freight value** ⟶ Sau Paulo ⟶ Parana ⟶ Minas Gerais ⟶ Rio de Janerio ⟶ Distrito Federal	
	Top 5 states with the lowest average time to delivery ⟶ Sau Paulo ⟶ Parana ⟶ Minas Gerais ⟶ Distrito Federal ⟶ Santa Catrina	
	Top 5 states where delivery is really fast compared to estimated date ⟶ Acre ⟶ Rondonia ⟶ Amazonas ⟶ Amapa ⟶ Roraima	
6.1	Highest no.of payments are done through **Credit Card**	1. Continue with Credit card offers to attract customers. 2. Provide EMI 3. Give attractive offers on CRedit Card 4. Offers Vouchers in August and July with validity of 30 days to attract customers for September purchase.
	Least are done through **Debit Card**	
	Highest payment done by **Credit Card is in May month**	
	Highest payment done by **Voucher is in July month**	
	Highest payment done by **UPI is in August month**	
	Highest paymeny done by **Debit Card is in August month**	
	Payment type for 3 transactions is "not_defined"	
6.2	Maximum installments payments are for **1 installment**	
	Minimum installments payments are for **22 and 23 installments**	