Target case study

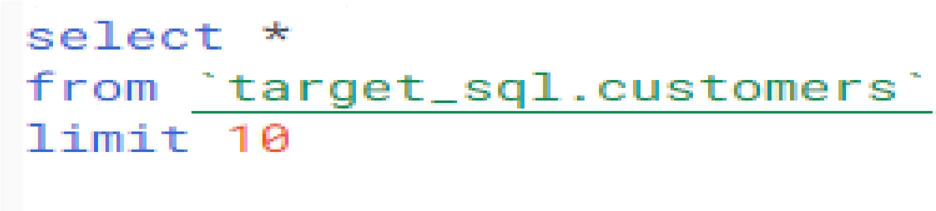
Target is one of the world’s most recognized brands and one of America’s leading retailers. Target makes itself a preferred shopping destination by offering outstanding value, inspiration, innovation and an exceptional guest experience that no other retailer can deliver.

This business case has information of 100k orders from 2016 to 2018 made at Target in Brazil. Its features allow viewing an order from multiple dimensions: from order status, price, payment and freight performance to customer location, product attributes and finally reviews written by customers.

1. Exploratory Analysis

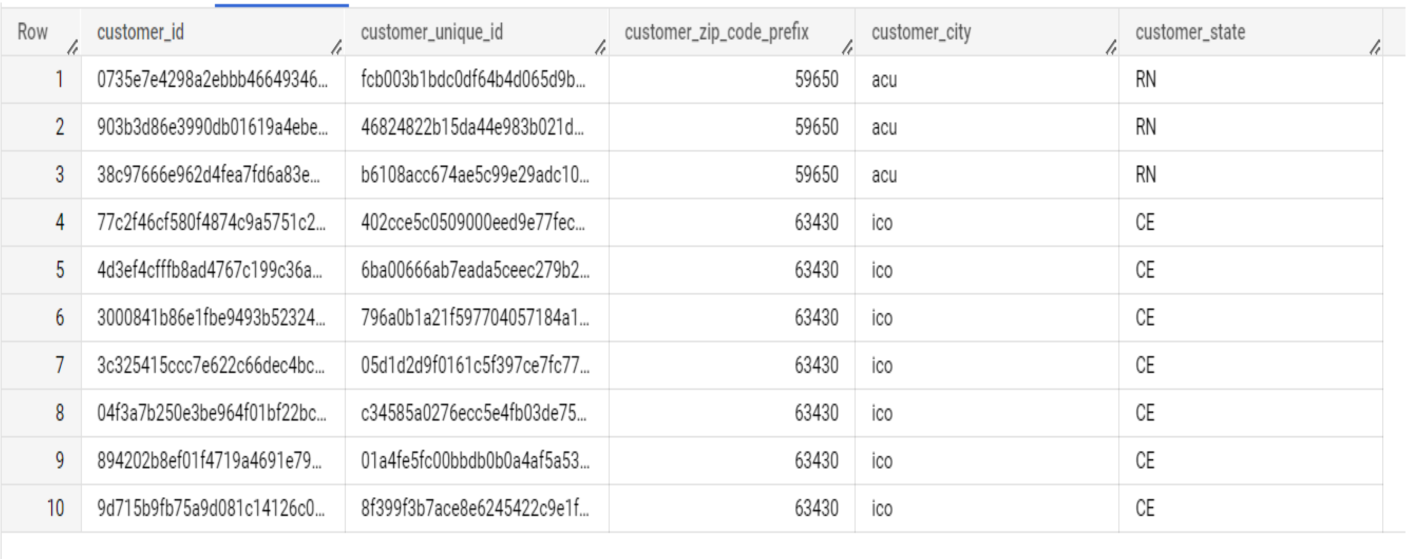
(a) . Customer Data Analysis

Query:



```
select *
from `target_sql.customers`
limit 10
```

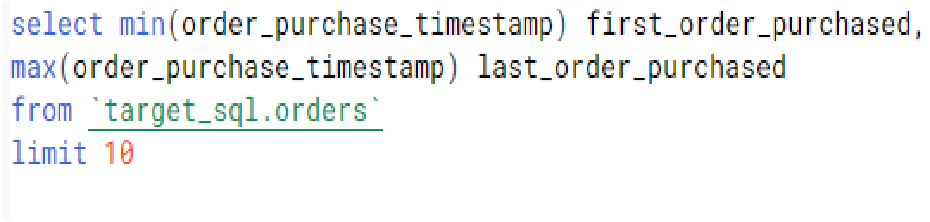
Query Results:



Row	customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state
1	0735e7e4298a2ebbb46649346...	fcb003b1bdc0df64b4d065d9b...	59650	acu	RN
2	903b3d86e3990db01619a4ebe...	46824822b15da44e983b021d...	59650	acu	RN
3	38c97666e962d4fea7fd6a83e...	b6108acc674ae5c99e29adc10...	59650	acu	RN
4	77c2f46cf580f4874c9a5751c2...	402cce5c0509000eed9e77fec...	63430	ico	CE
5	4d3ef4cfffb8ad4767c199c36a...	6ba00666ab7eada5ceec279b2...	63430	ico	CE
6	3000841b86e1fbe9493b52324...	796a0b1a21f597704057184a1...	63430	ico	CE
7	3c325415ccc7e622c66dec4bc...	05d1d2d9f0161c5f397ce7fc77...	63430	ico	CE
8	04f3a7b250e3be964f01bf22bc...	c34585a0276ecc5e4fb03de75...	63430	ico	CE
9	894202b8ef01f4719a4691e79...	01a4fe5fc00bbdb0b0a4af5a53...	63430	ico	CE
10	9d715b9fb75a9d081c14126c0...	8f399f3b7ace8e6245422c9e1f...	63430	ico	CE

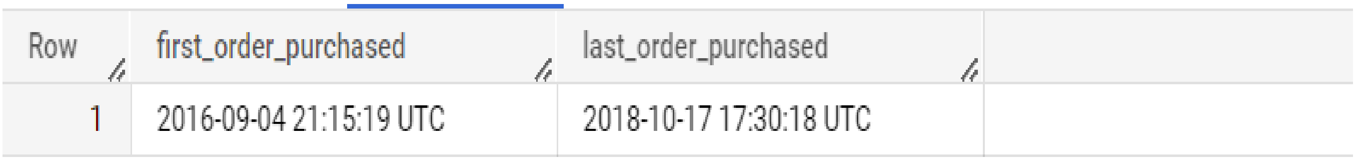
(b) Time period of Data

Query:



```
select min(order_purchase_timestamp) first_order_purchased,
max(order_purchase_timestamp) last_order_purchased
from `target_sql.orders`
limit 10
```

Query Results:



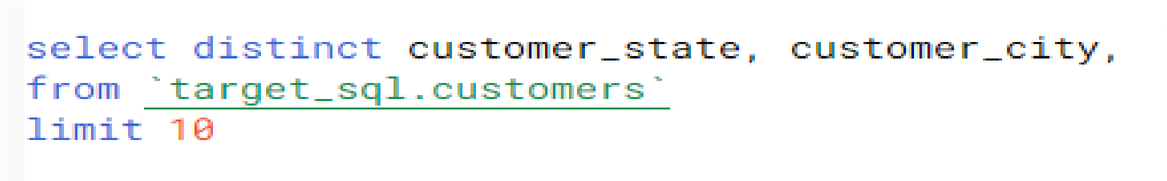
Row	first_order_purchased	last_order_purchased
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

INSIGHTS

From above, it is clear that data is collected from year September-2016 to October-2018

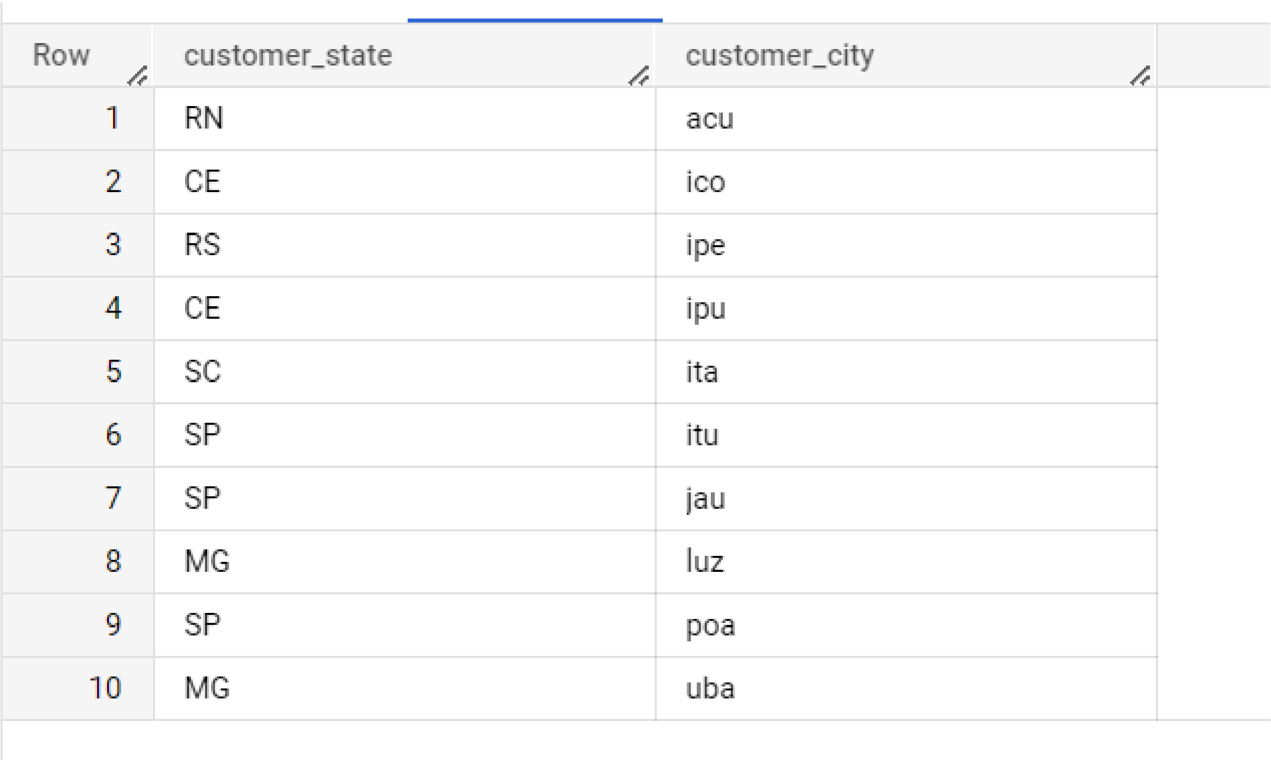
(c) Cities and States of customers ordered during the given period.

Query:



```
select distinct customer_state, customer_city,
from `target_sql.customers`
limit 10
```

Query Result:

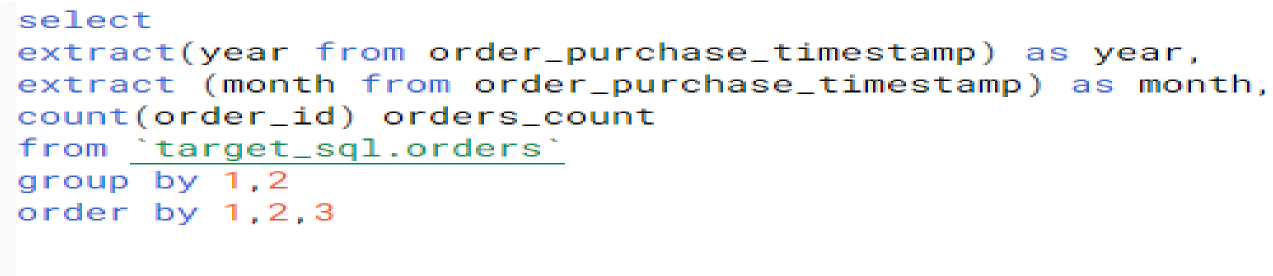


Row	customer_state	customer_city
1	RN	acu
2	CE	ico
3	RS	ipe
4	CE	ipu
5	SC	ita
6	SP	itu
7	SP	jau
8	MG	luz
9	SP	poa
10	MG	uba

2. In-depth Exploration

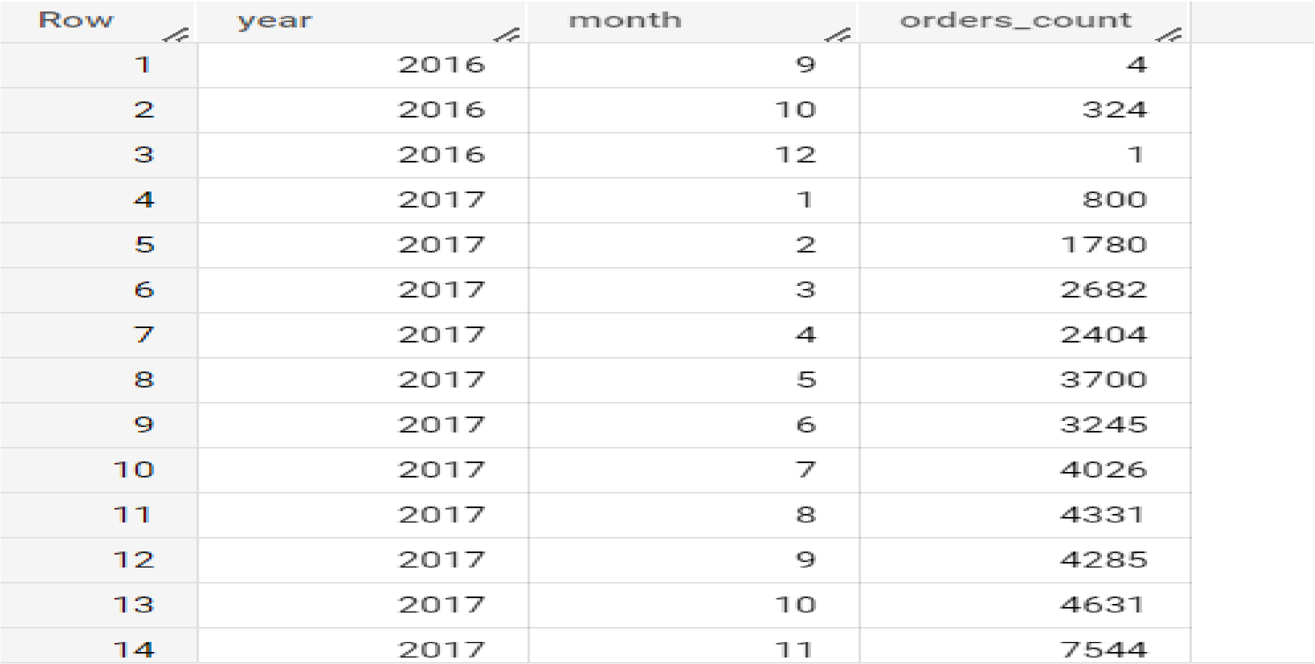
(a) Trend on e-commerce in Brazil

Query:



```
select
extract(year from order_purchase_timestamp) as year,
extract (month from order_purchase_timestamp) as month,
count(order_id) orders_count
from `target_sql.orders`
group by 1,2
order by 1,2,3
```

Query Result:



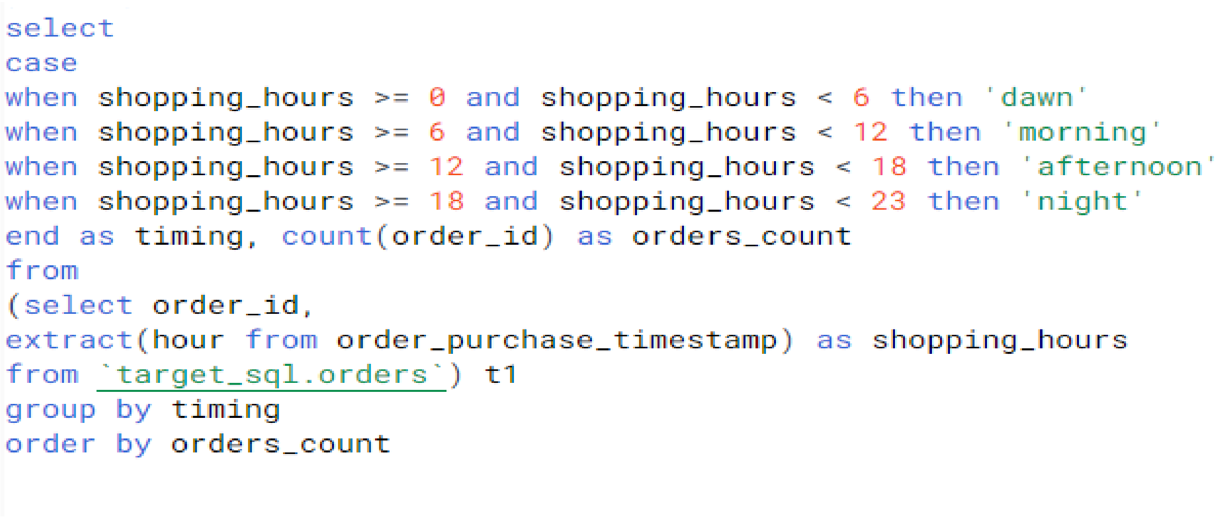
Row	year	month	orders_count
1	2016	9	4
2	2016	10	324
3	2016	12	1
4	2017	1	800
5	2017	2	1780
6	2017	3	2682
7	2017	4	2404
8	2017	5	3700
9	2017	6	3245
10	2017	7	4026
11	2017	8	4331
12	2017	9	4285
13	2017	10	4631
14	2017	11	7544

INSIGHTS

It is noticed that there is significant increase in orders count from year 2016 to 2018, which shows that there is a growth in trend of e-commerce in Brazil.

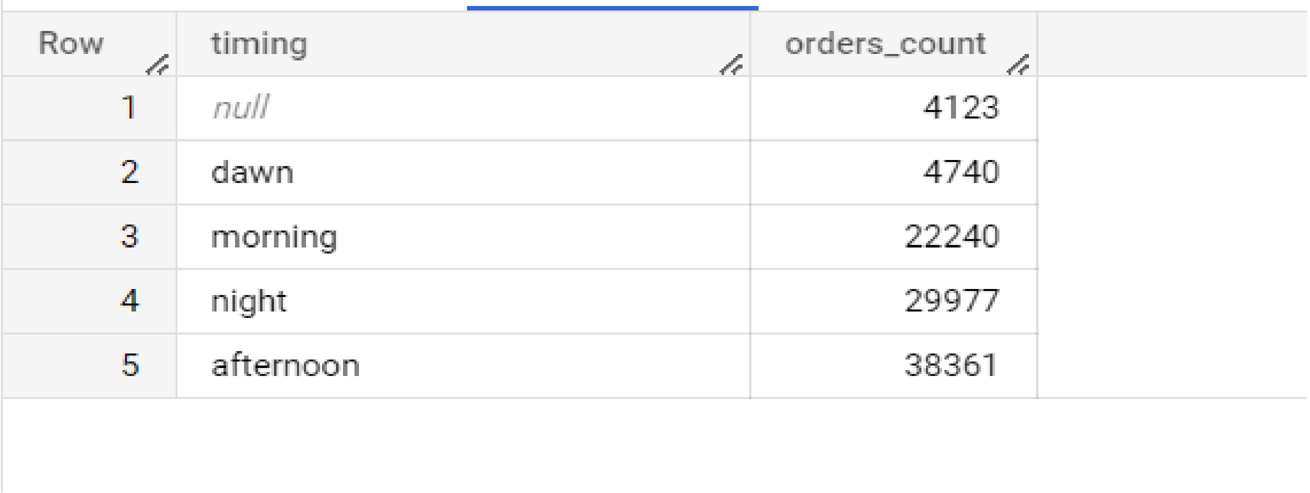
(b) What time do Brazilian customers tend to buy (Dawn, Morning, Afternoon or Night)?

Query:



```
select
case
when shopping_hours >= 0 and shopping_hours < 6 then 'dawn'
when shopping_hours >= 6 and shopping_hours < 12 then 'morning'
when shopping_hours >= 12 and shopping_hours < 18 then 'afternoon'
when shopping_hours >= 18 and shopping_hours < 23 then 'night'
end as timing, count(order_id) as orders_count
from
(select order_id,
extract(hour from order_purchase_timestamp) as shopping_hours
from `target_sql.orders`) t1
group by timing
order by orders_count
```

Query Result:



Row	timing	orders_count
1	*null*	4123
2	dawn	4740
3	morning	22240
4	night	29977
5	afternoon	38361

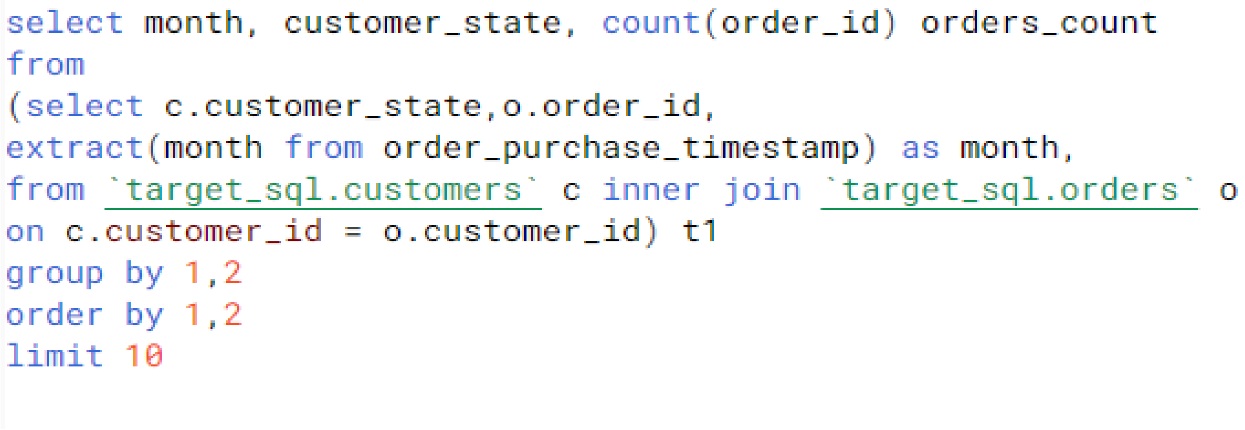
INSIGHTS

It is clear from above that people in Brazil tend to shop more frequently in the afternoon, as the highest count of orders are in the afternoon with 38,361 orders.

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

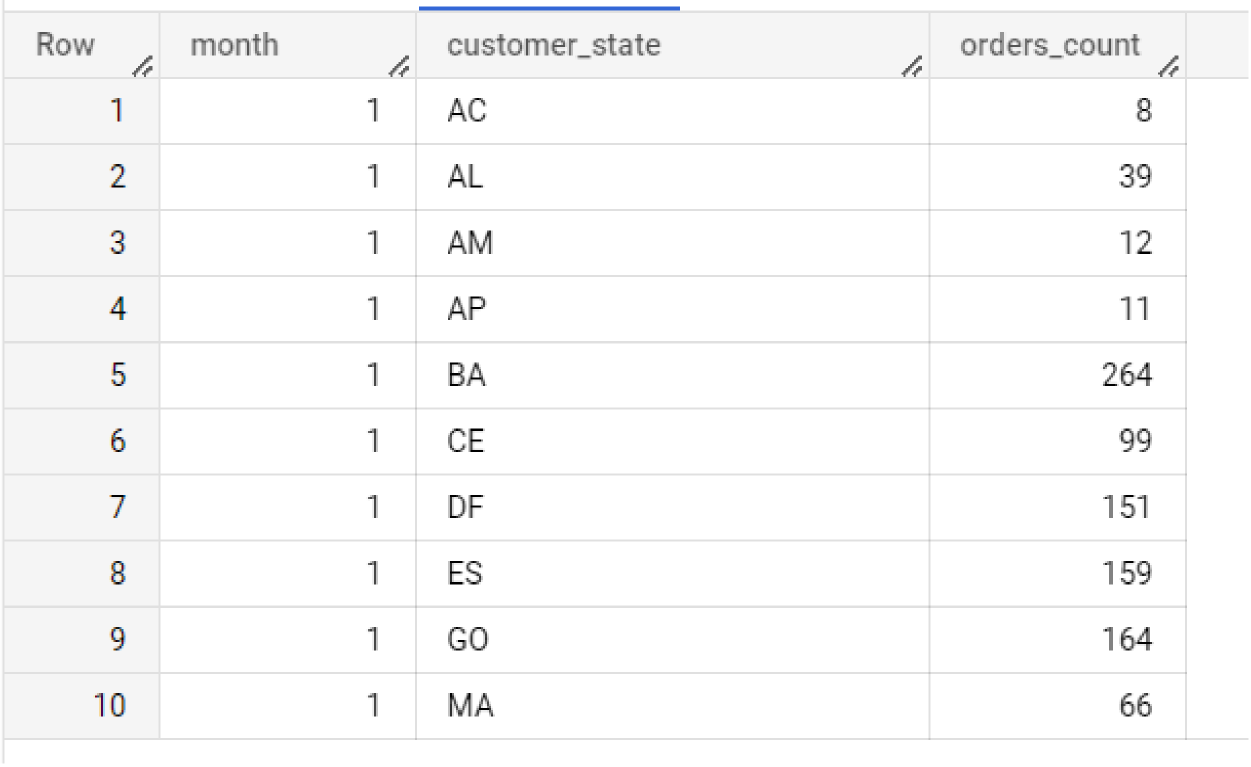
(i) Month on Month orders by States

Query:



```
select month, customer_state, count(order_id) orders_count
from
(select c.customer_state,o.order_id,
extract(month from order_purchase_timestamp) as month,
from `target_sql.customers` c inner join `target_sql.orders` o
on c.customer_id = o.customer_id) t1
group by 1,2
order by 1,2
limit 10
```

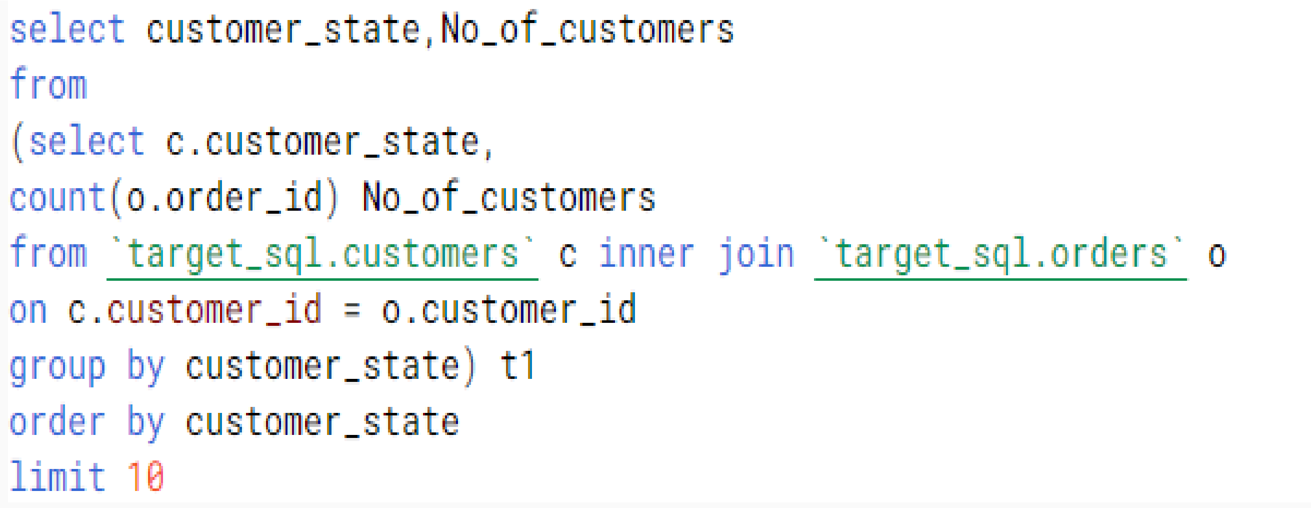
Query Results:



Row	month	customer_state	orders_count
1	1	AC	8
2	1	AL	39
3	1	AM	12
4	1	AP	11
5	1	BA	264
6	1	CE	99
7	1	DF	151
8	1	ES	159
9	1	GO	164
10	1	MA	66

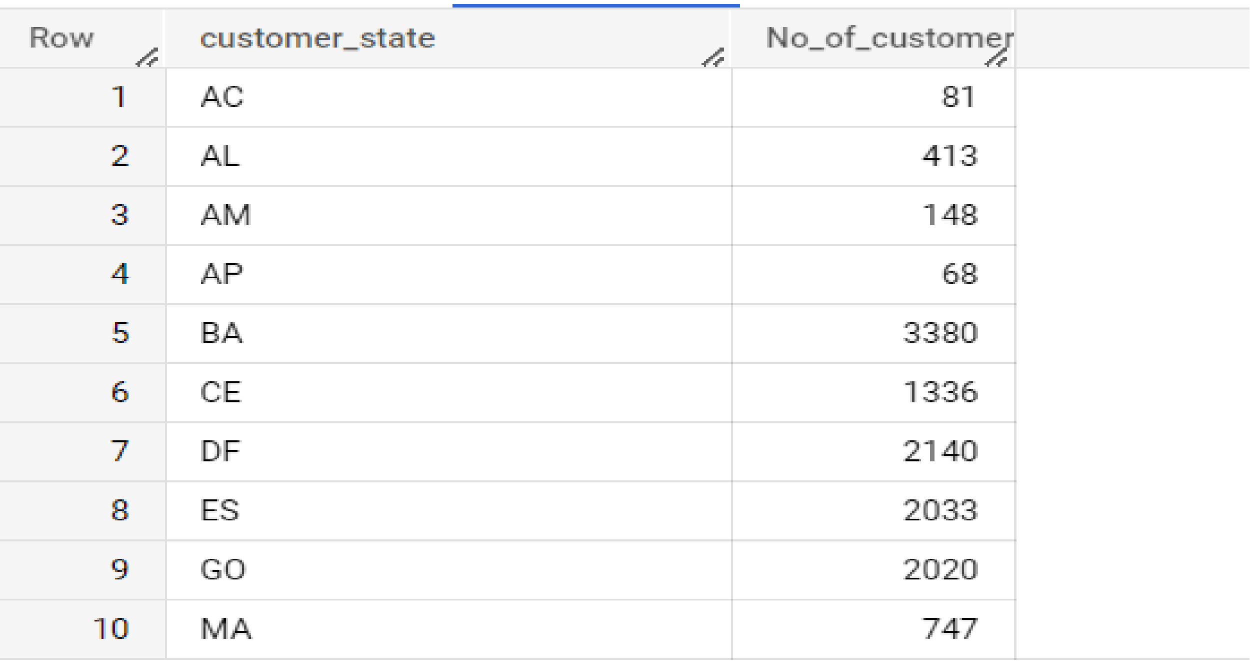
(ii) Distribution of customers across the states in Brazil

Query:



```
select customer_state,No_of_customers
from
(select c.customer_state,
count(o.order_id) No_of_customers
from `target_sql.customers` c inner join `target_sql.orders` o
on c.customer_id = o.customer_id
group by customer_state) t1
order by customer_state
limit 10
```

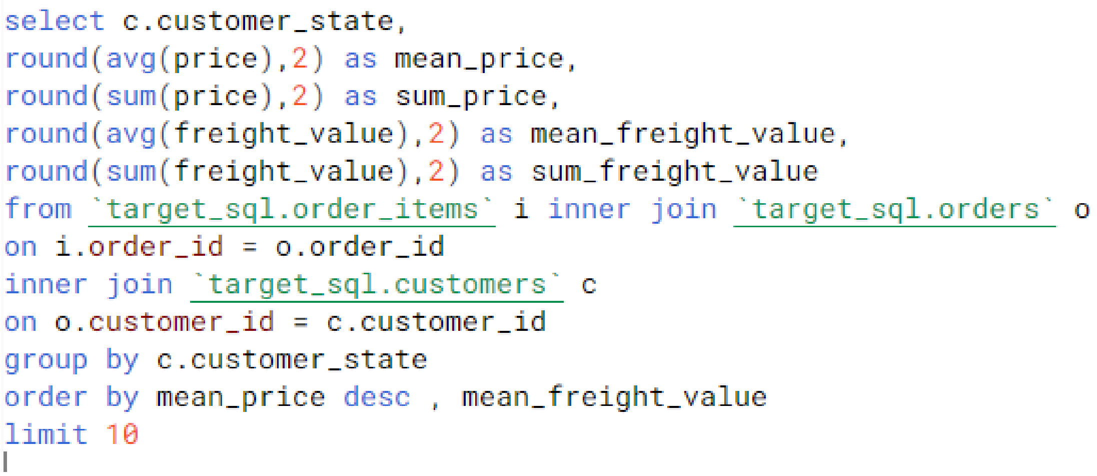
Query Results:



Row	customer_state	No_of_customer
1	AC	81
2	AL	413
3	AM	148
4	AP	68
5	BA	3380
6	CE	1336
7	DF	2140
8	ES	2033
9	GO	2020
10	MA	747

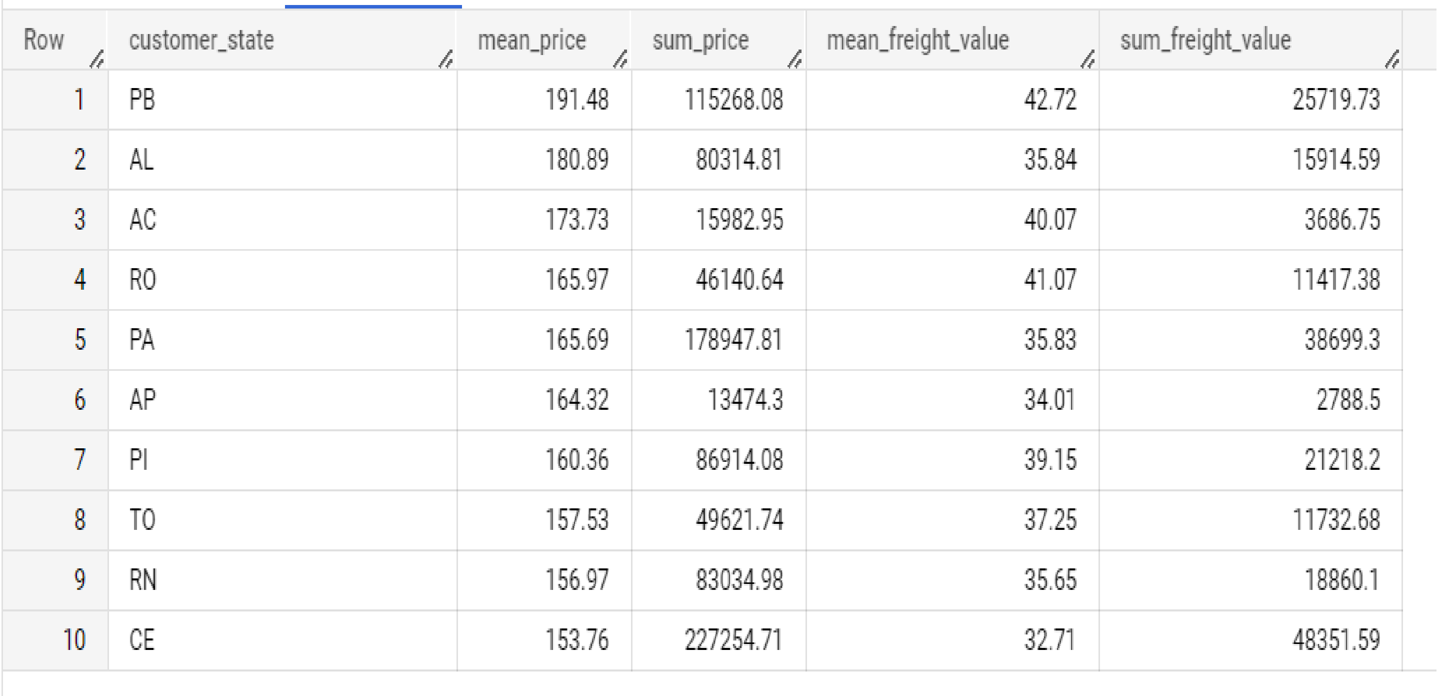
4. Mean & Sum of price and freight value by customer state

Query:



```
select c.customer_state,
round(avg(price),2) as mean_price,
round(sum(price),2) as sum_price,
round(avg(freight_value),2) as mean_freight_value,
round(sum(freight_value),2) as sum_freight_value
from `target_sql.order_items` i inner join `target_sql.orders` o
on i.order_id = o.order_id
inner join `target_sql.customers` c
on o.customer_id = c.customer_id
group by c.customer_state
order by mean_price desc , mean_freight_value
limit 10
```

Query Results:

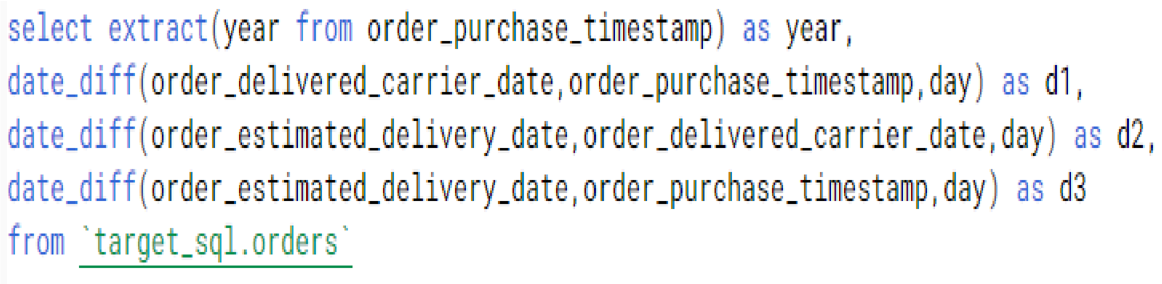


Row	customer_state	mean_price	sum_price	mean_freight_value	sum_freight_value
1	PB	191.48	115268.08	42.72	25719.73
2	AL	180.89	80314.81	35.84	15914.59
3	AC	173.73	15982.95	40.07	3686.75
4	RO	165.97	46140.64	41.07	11417.38
5	PA	165.69	178947.81	35.83	38699.3
6	AP	164.32	13474.3	34.01	2788.5
7	PI	160.36	86914.08	39.15	21218.2
8	TO	157.53	49621.74	37.25	11732.68
9	RN	156.97	83034.98	35.65	18860.1
10	CE	153.76	227254.71	32.71	48351.59

5. Analysis on sales, freight and delivery time

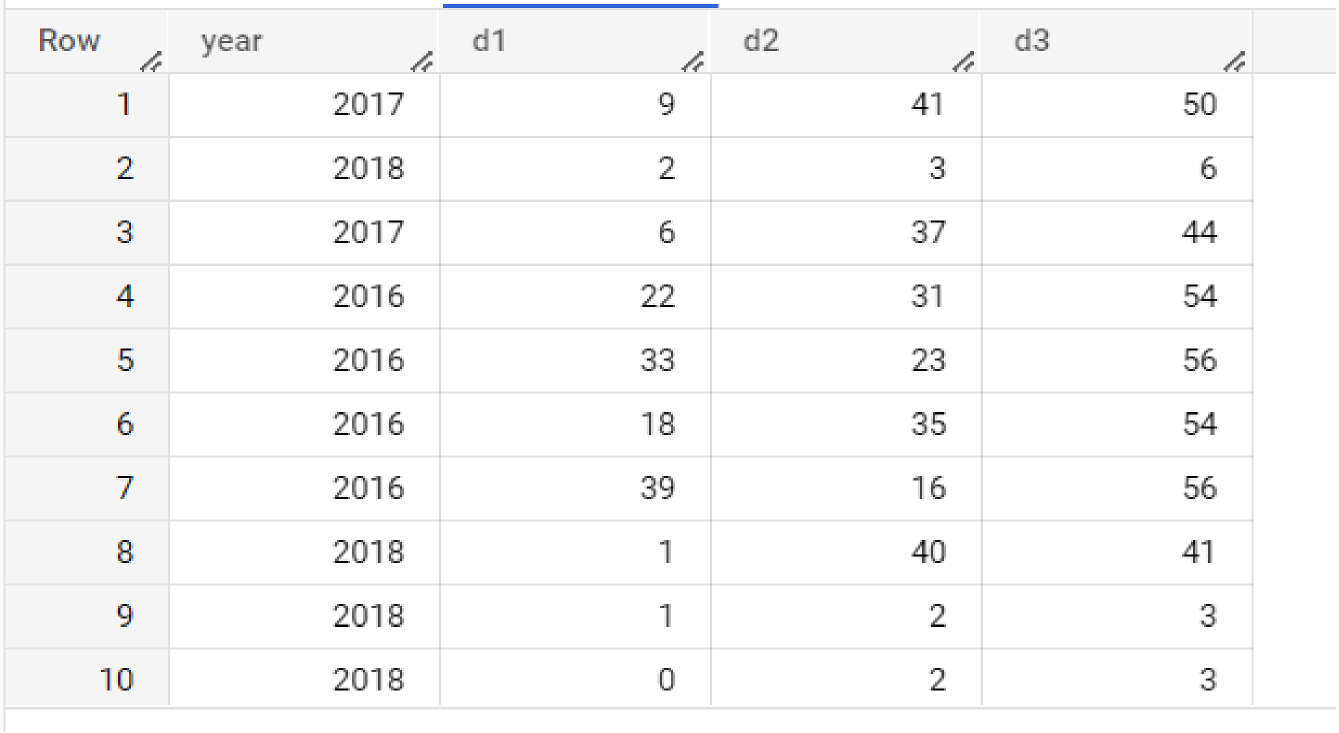
(a) Days between purchasing, delivering and estimated delivery.

Query:



```
select extract(year from order_purchase_timestamp) as year,
date_diff(order_delivered_carrier_date,order_purchase_timestamp,day) as d1,
date_diff(order_estimated_delivery_date,order_delivered_carrier_date,day) as d2,
date_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as d3
from `target_sql.orders`
```

Query Results:



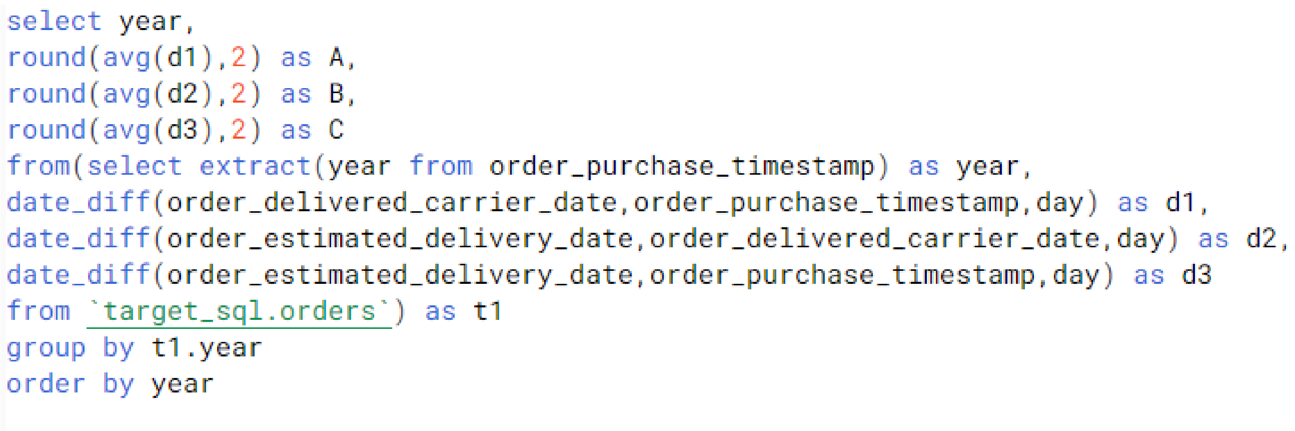
Row	year	d1	d2	d3
1	2017	9	41	50
2	2018	2	3	6
3	2017	6	37	44
4	2016	22	31	54
5	2016	33	23	56
6	2016	18	35	54
7	2016	39	16	56
8	2018	1	40	41
9	2018	1	2	3
10	2018	0	2	3

INSIGHTS

- **d1 - Days between order delivered carrier date and order purchase time stamp.**
- **d2 - Days between order estimated delivery date and order delivered carrier.**
- **d3 - Days between order estimated delivery date and order purchase time stamp.**

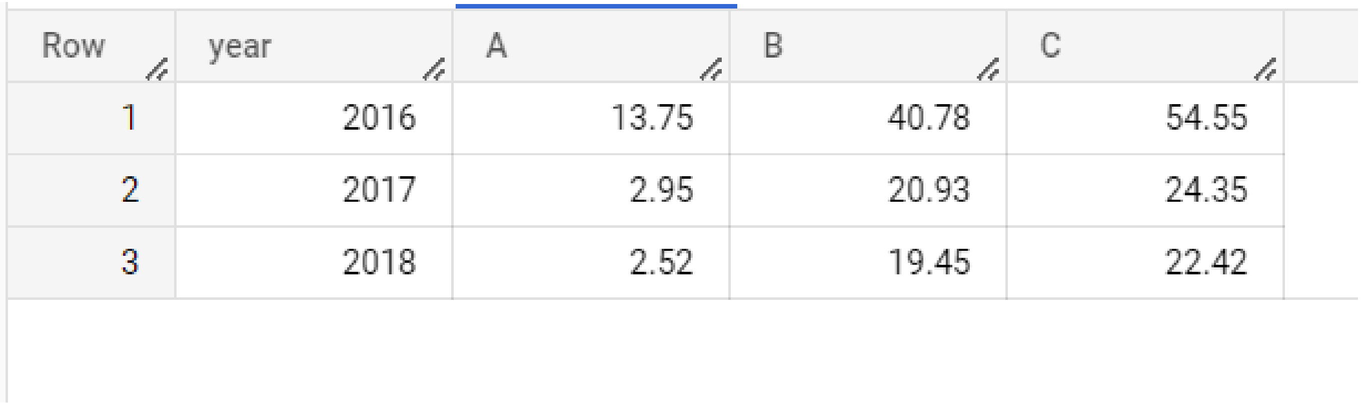
(b) Average days between purchasing, delivering and estimated delivery.

Query:



```
select year,
round(avg(d1),2) as A,
round(avg(d2),2) as B,
round(avg(d3),2) as C
from(select extract(year from order_purchase_timestamp) as year,
date_diff(order_delivered_carrier_date,order_purchase_timestamp,day) as d1,
date_diff(order_estimated_delivery_date,order_delivered_carrier_date,day) as d2,
date_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as d3
from `target_sql.orders`) as t1
group by t1.year
order by year
```

Query Results:



Row	year	A	B	C
1	2016	13.75	40.78	54.55
2	2017	2.95	20.93	24.35
3	2018	2.52	19.45	22.42

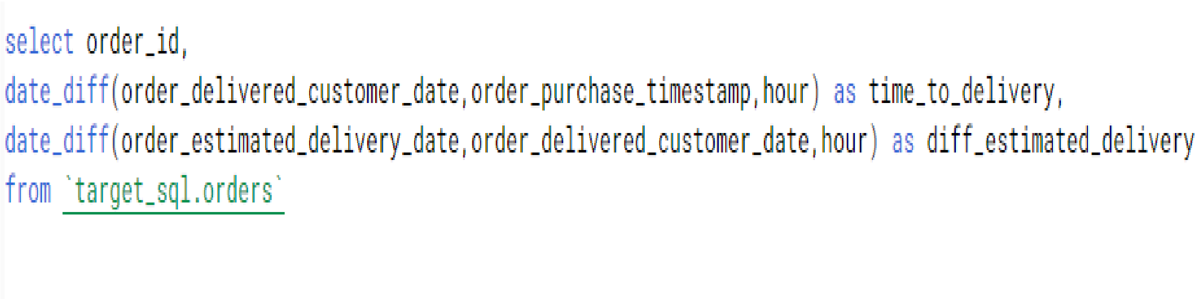
INSIGHTS

Here we have the average no. of days in given years.

- **A – Average days between order delivered carrier date and order purchase time stamp.**
- **B - Average days between order estimated delivery date and order delivered carrier.**
- **C - Average days between order estimated delivery date and order purchase time stamp.**

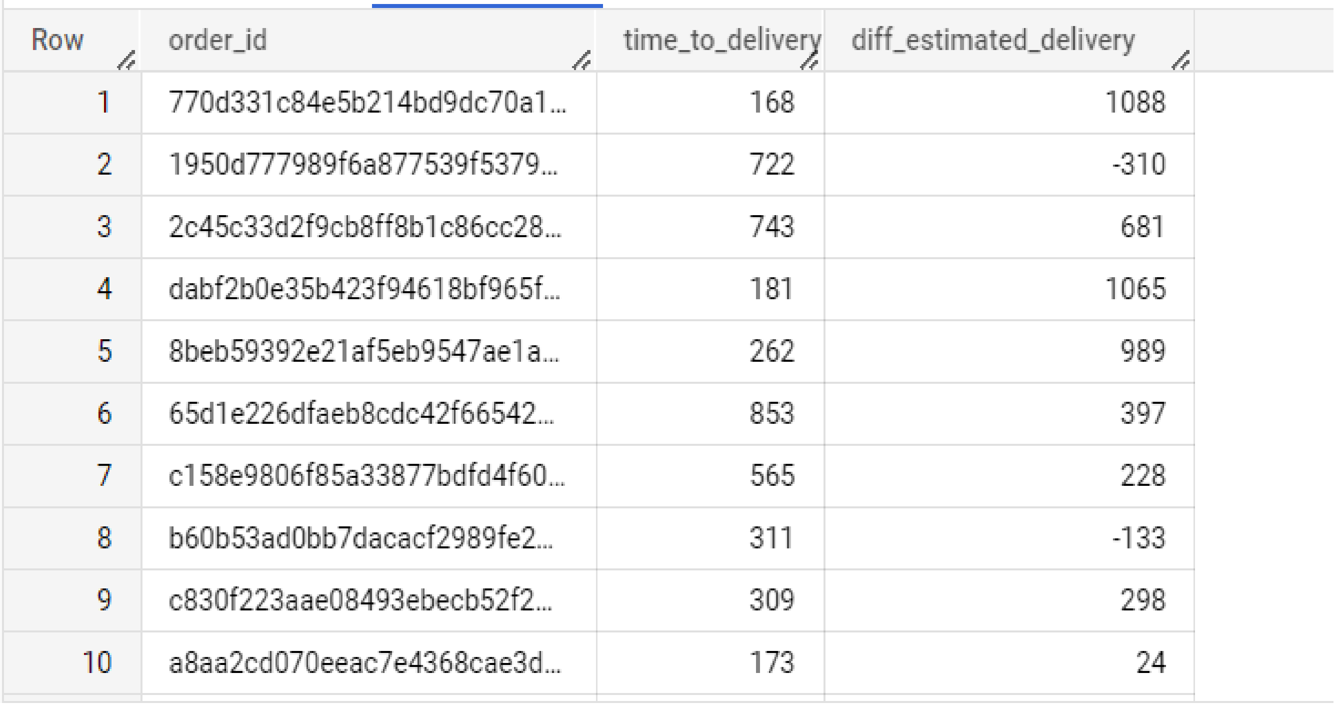
(c) Time to delivery & diff estimated delivery.

Query:



```
select order_id,
date_diff(order_delivered_customer_date,order_purchase_timestamp,hour) as time_to_delivery,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,hour) as diff_estimated_delivery
from `target_sql.orders`
```

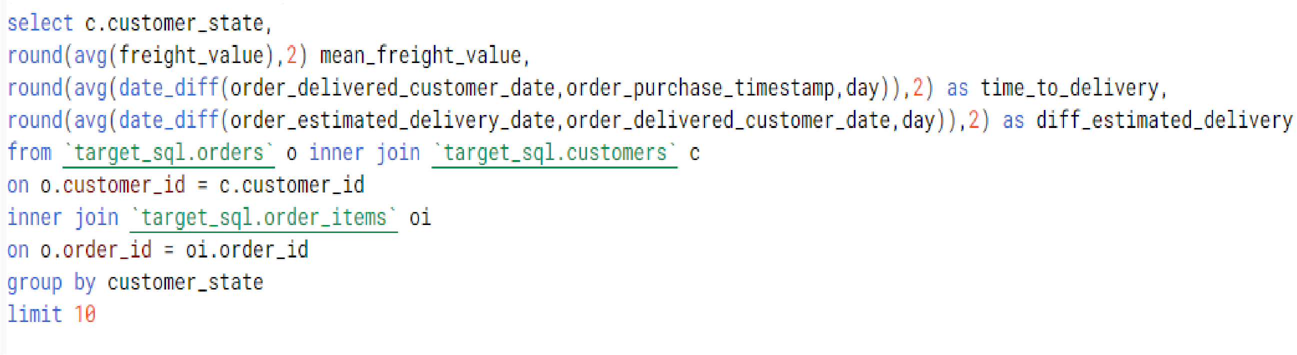
Query Results:



Row	order_id	time_to_delivery	diff_estimated_delivery
1	770d331c84e5b214bd9dc70a1...	168	1088
2	1950d777989f6a877539f5379...	722	-310
3	2c45c33d2f9cb8ff8b1c86cc28...	743	681
4	dabf2b0e35b423f94618bf965f...	181	1065
5	8beb59392e21af5eb9547ae1a...	262	989
6	65d1e226dfaeb8cdc42f66542...	853	397
7	c158e9806f85a33877bdfd4f60...	565	228
8	b60b53ad0bb7dacacf2989fe2...	311	-133
9	c830f223aae08493ebecb52f2...	309	298
10	a8aa2cd070eeac7e4368cae3d...	173	24

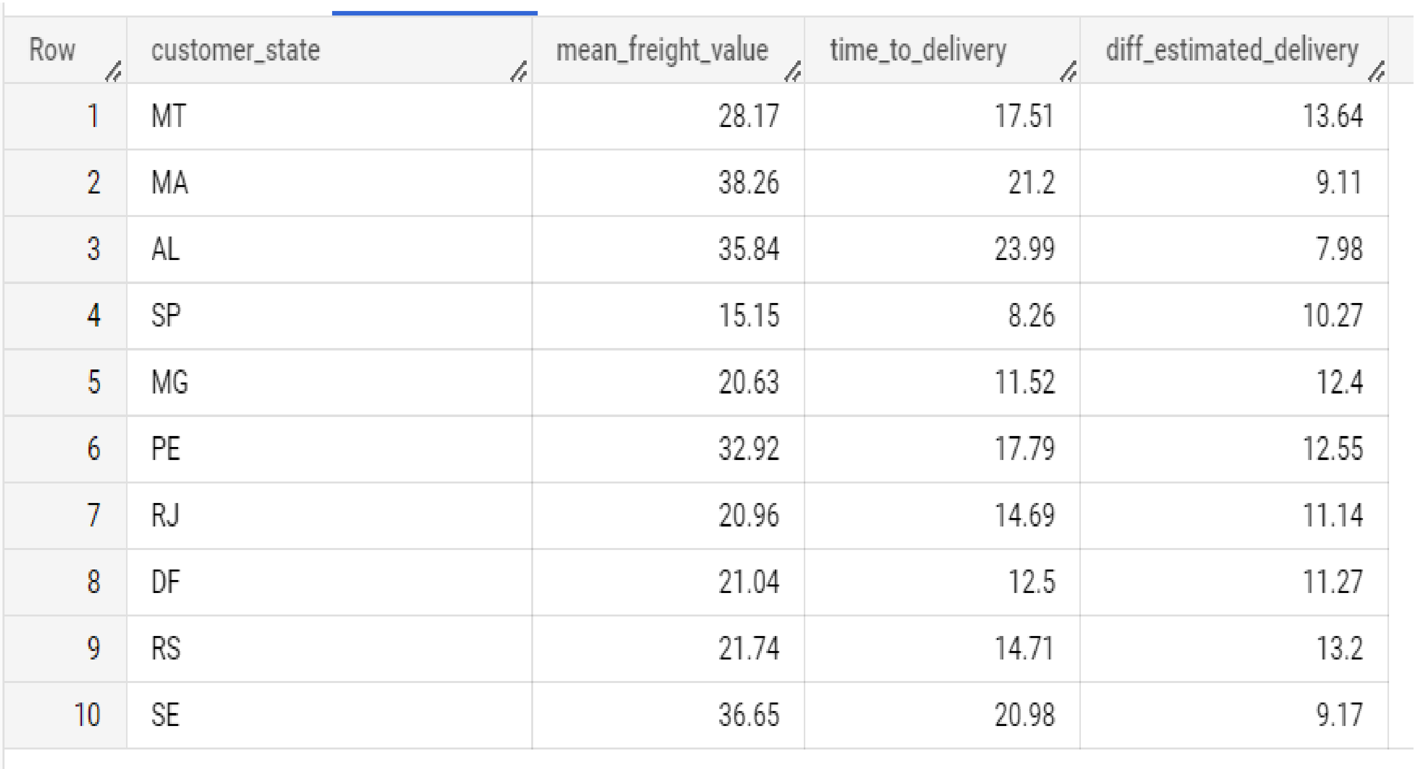
(d) Group data by state, take mean of freight value, time to delivery, diff estimated delivery.

Query:



```
select c.customer_state,
round(avg(freight_value),2) mean_freight_value,
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
from `target_sql.orders` o inner join `target_sql.customers` c
on o.customer_id = c.customer_id
inner join `target_sql.order_items` oi
on o.order_id = oi.order_id
group by customer_state
limit 10
```

Query Results:

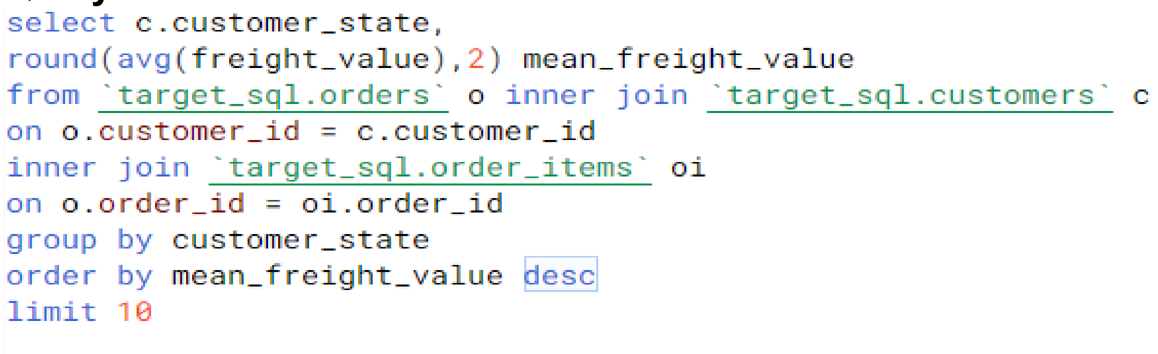


Row	customer_state	mean_freight_value	time_to_delivery	diff_estimated_delivery
1	MT	28.17	17.51	13.64
2	MA	38.26	21.2	9.11
3	AL	35.84	23.99	7.98
4	SP	15.15	8.26	10.27
5	MG	20.63	11.52	12.4
6	PE	32.92	17.79	12.55
7	RJ	20.96	14.69	11.14
8	DF	21.04	12.5	11.27
9	RS	21.74	14.71	13.2
10	SE	36.65	20.98	9.17

(e) Sorting Data

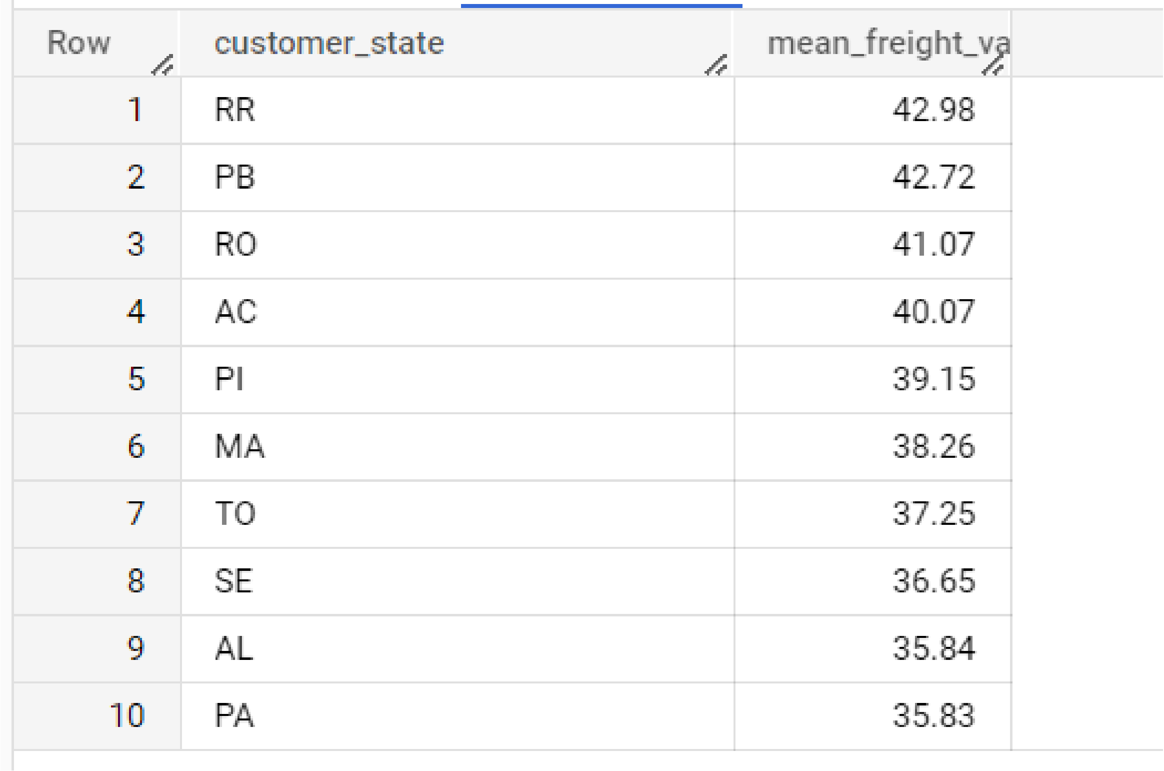
➢ States with highest freight value

Query:



```
select c.customer_state,
round(avg(freight_value),2) mean_freight_value
from `target_sql.orders` o inner join `target_sql.customers` c
on o.customer_id = c.customer_id
inner join `target_sql.order_items` oi
on o.order_id = oi.order_id
group by customer_state
order by mean_freight_value desc
limit 10
```

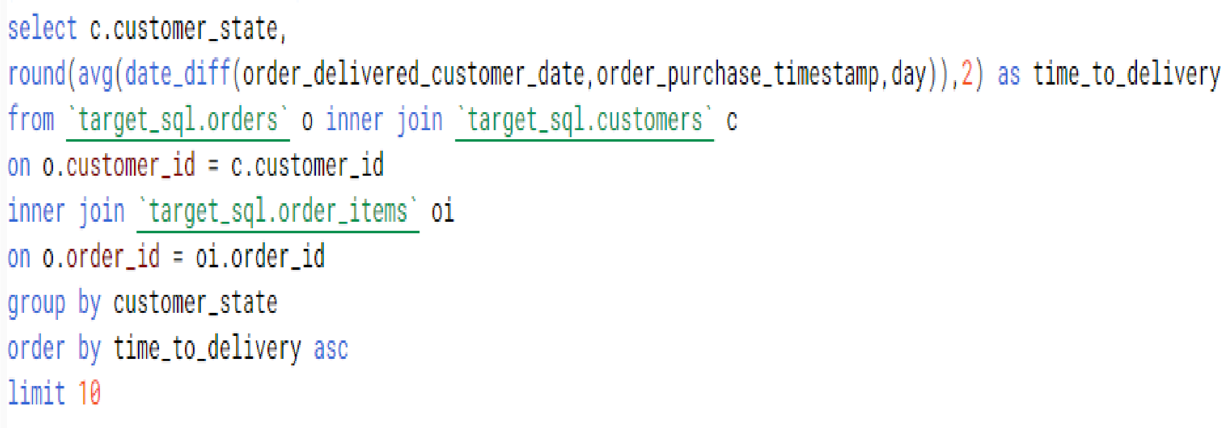
Query Result:



Row	customer_state	mean_freight_va
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15
6	MA	38.26
7	TO	37.25
8	SE	36.65
9	AL	35.84
10	PA	35.83

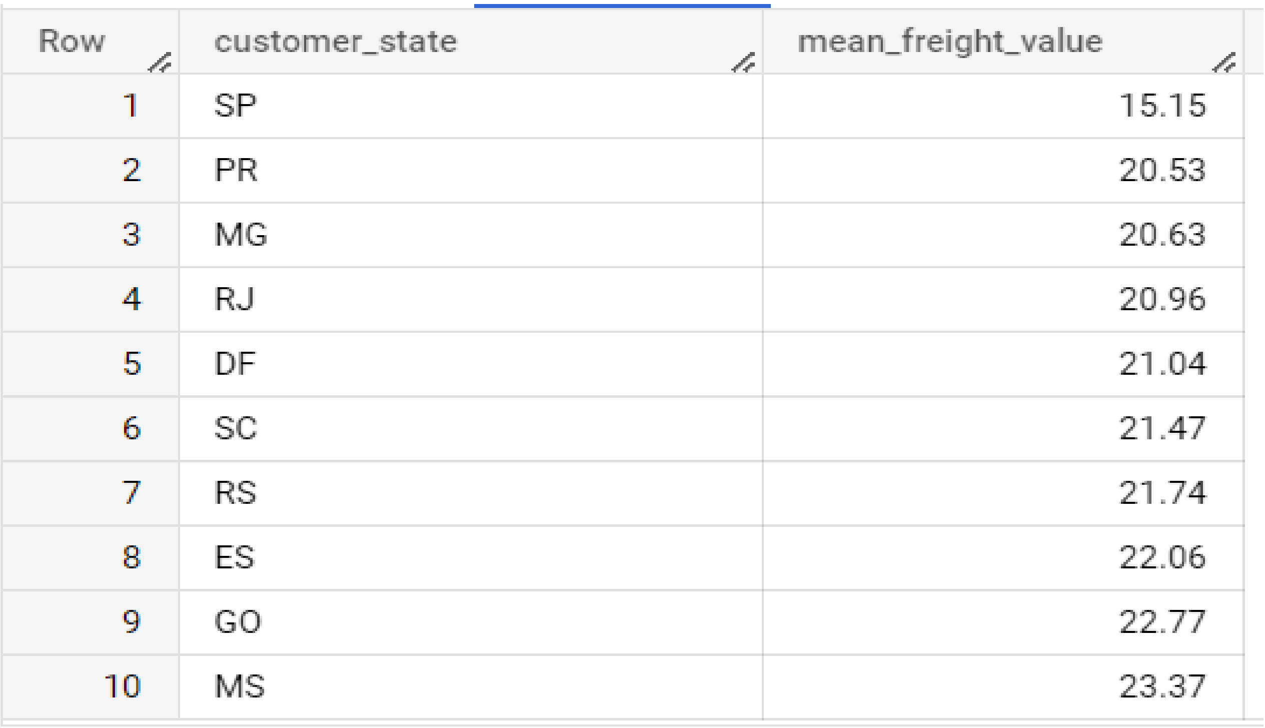
- **States with lowest freight values**

Query:



```
select c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2) as time_to_delivery
from `target_sql.orders` o inner join `target_sql.customers` c
on o.customer_id = c.customer_id
inner join `target_sql.order_items` oi
on o.order_id = oi.order_id
group by customer_state
order by time_to_delivery asc
limit 10
```

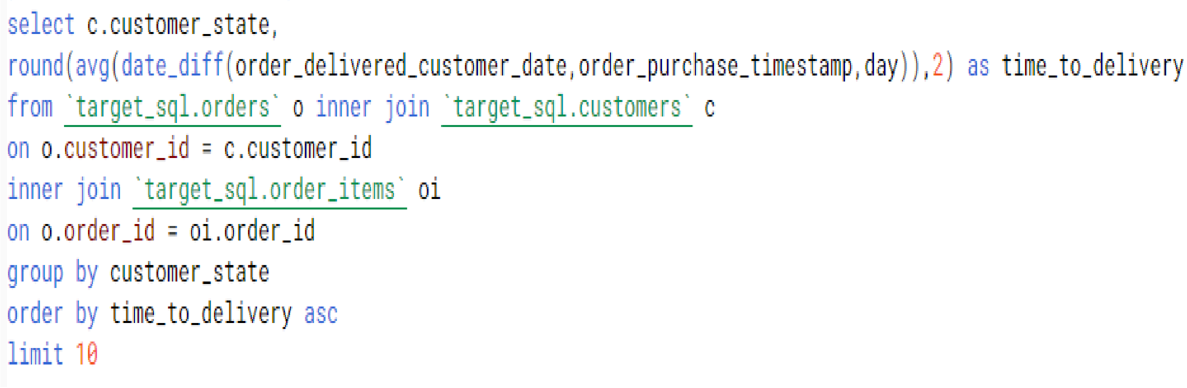
Query Results:



Row	customer_state	mean_freight_value
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04
6	SC	21.47
7	RS	21.74
8	ES	22.06
9	GO	22.77
10	MS	23.37

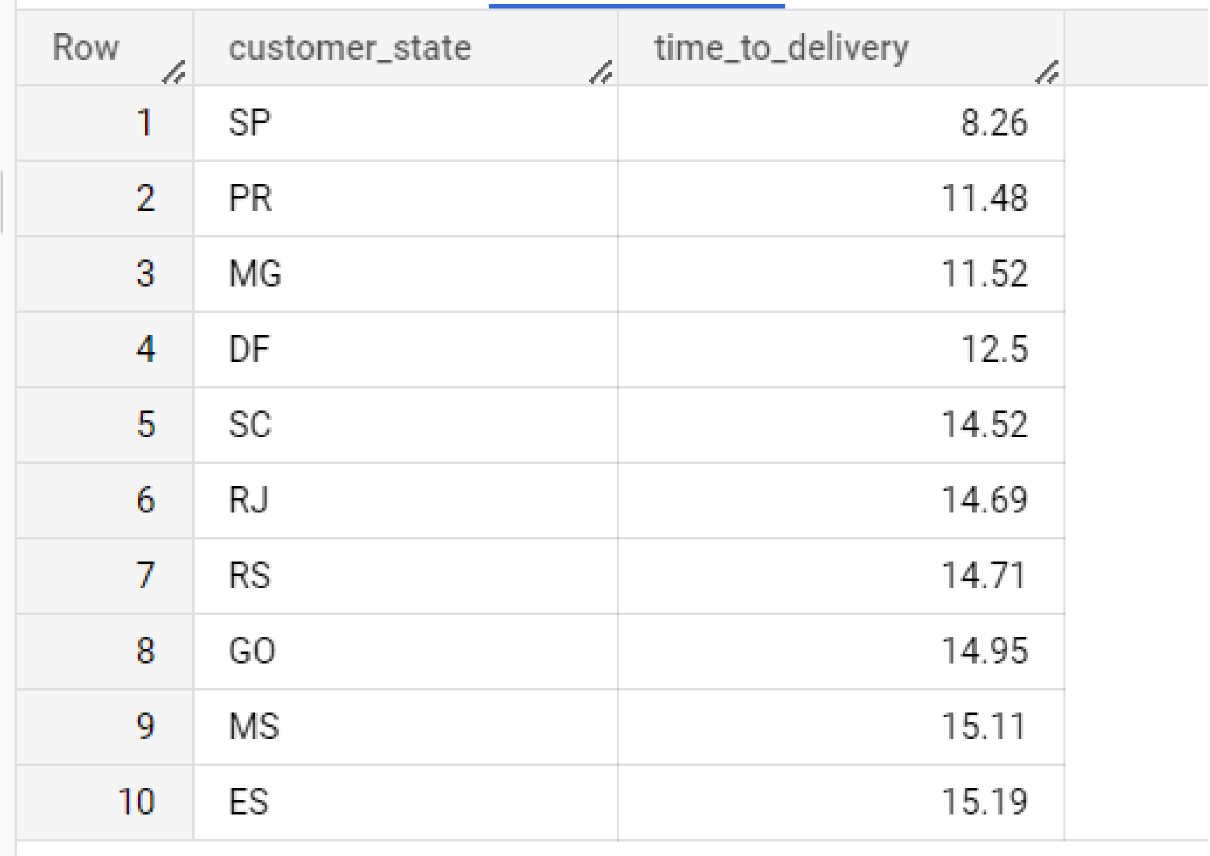
- **States with the lowest average time to delivery.**

Query:



```
select c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2) as time_to_delivery
from `target_sql.orders` o inner join `target_sql.customers` c
on o.customer_id = c.customer_id
inner join `target_sql.order_items` oi
on o.order_id = oi.order_id
group by customer_state
order by time_to_delivery asc
limit 10
```

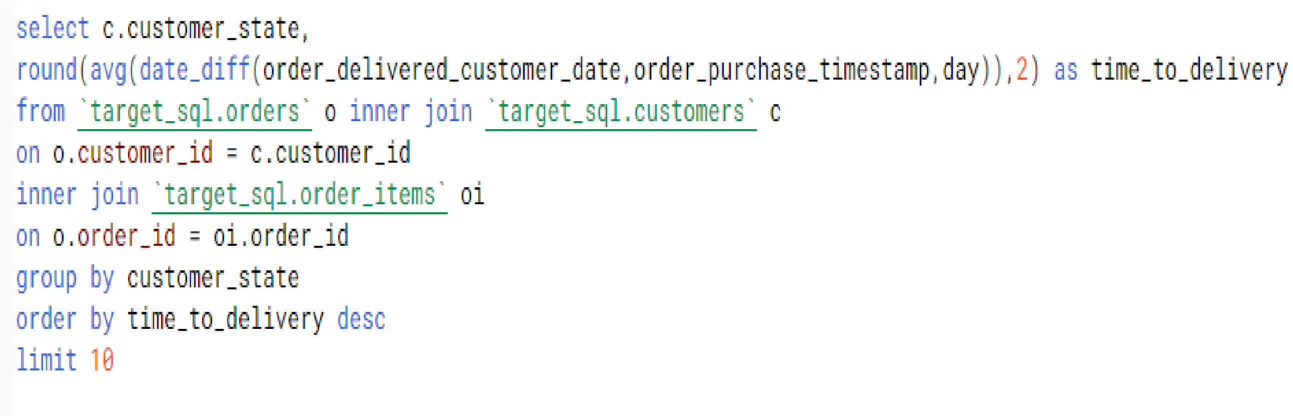
Query Result:



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
6	RJ	14.69
7	RS	14.71
8	GO	14.95
9	MS	15.11
10	ES	15.19

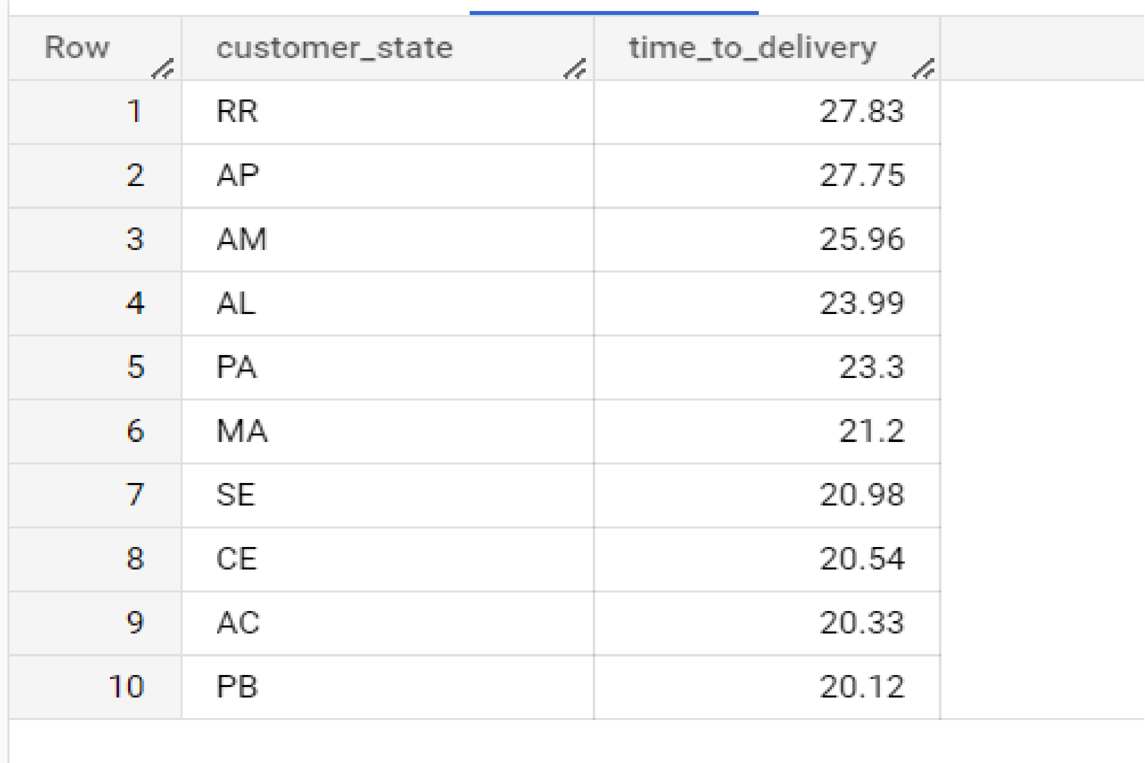
- **States with the highest time to delivery.**

Query:



```
select c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,day)),2) as time_to_delivery
from `target_sql.orders` o inner join `target_sql.customers` c
on o.customer_id = c.customer_id
inner join `target_sql.order_items` oi
on o.order_id = oi.order_id
group by customer_state
order by time_to_delivery desc
limit 10
```

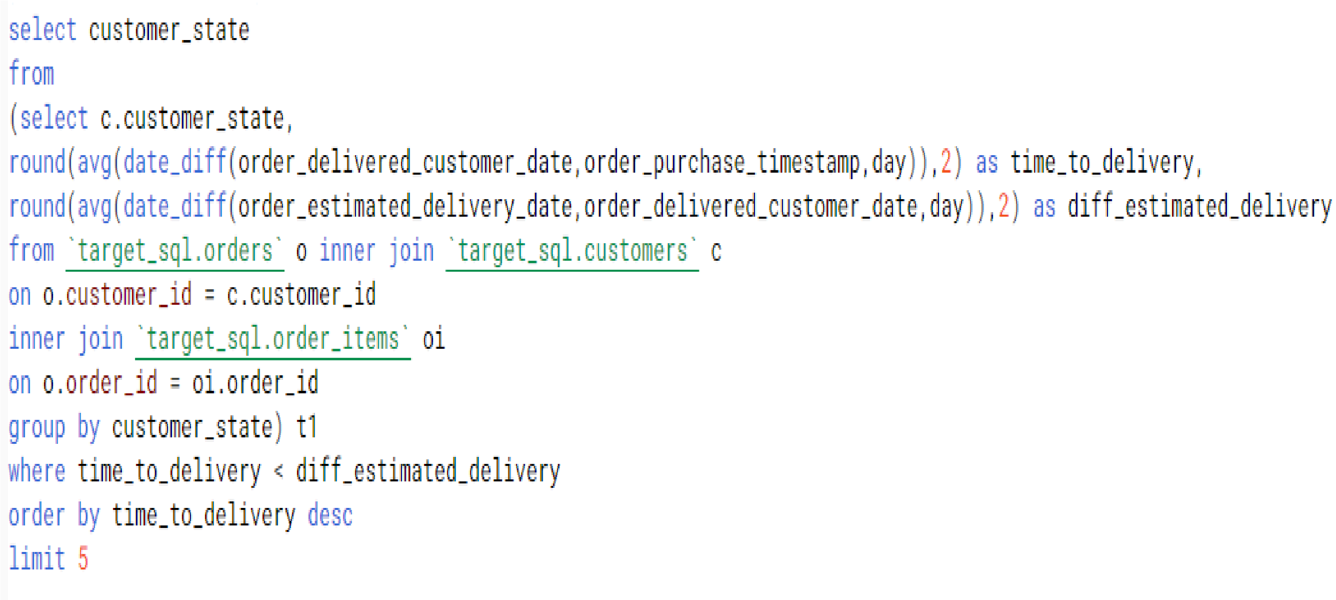
Query Results:



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
6	MA	21.2
7	SE	20.98
8	CE	20.54
9	AC	20.33
10	PB	20.12

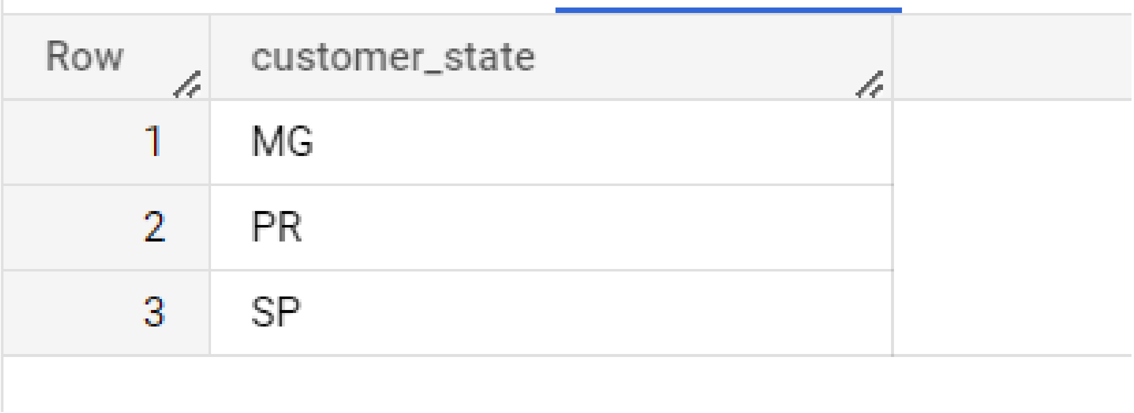
- **States with faster delivery as compared to estimated date.**

Query:



```
select customer_state
from
(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
from `target_sql.orders` o inner join `target_sql.customers` c
on o.customer_id = c.customer_id
inner join `target_sql.order_items` oi
on o.order_id = oi.order_id
group by customer_state) t1
where time_to_delivery < diff_estimated_delivery
order by time_to_delivery desc
limit 5
```

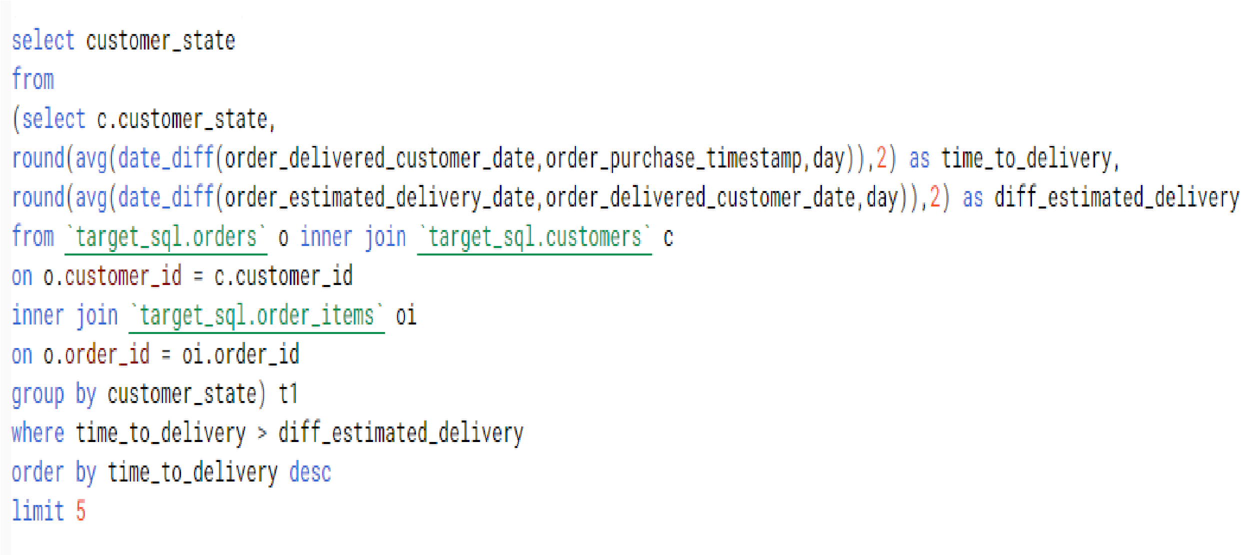
Query Results:



Row	customer_state
1	MG
2	PR
3	SP

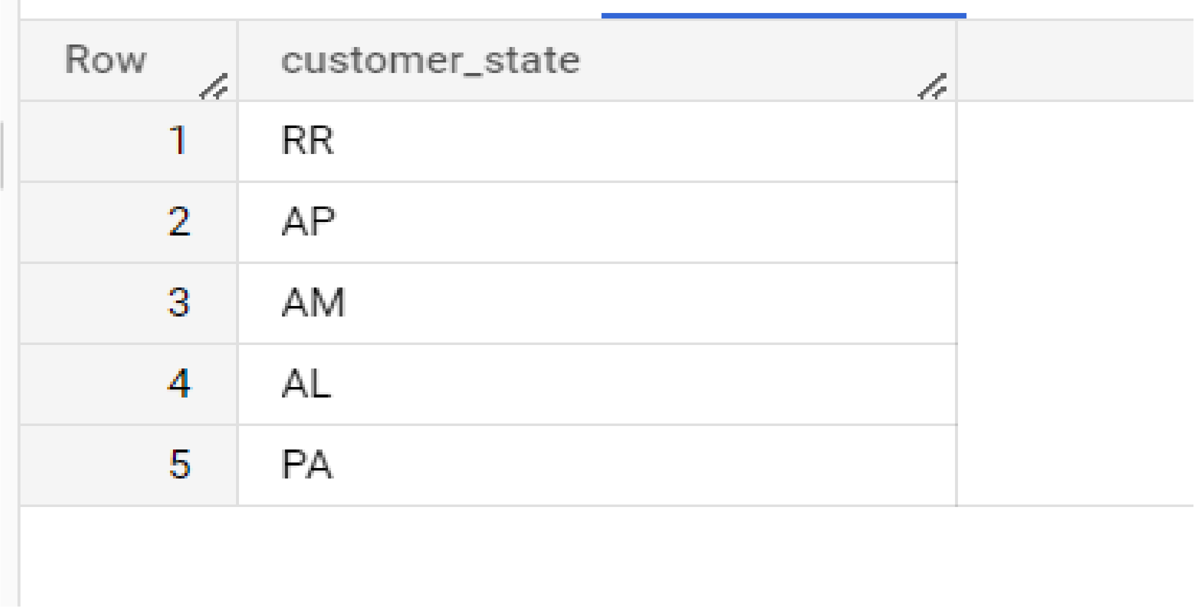
- **States with slower delivery as compared to estimated date.**

Query:



```
select customer_state
from
(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
from `target_sql.orders` o inner join `target_sql.customers` c
on o.customer_id = c.customer_id
inner join `target_sql.order_items` oi
on o.order_id = oi.order_id
group by customer_state) t1
where time_to_delivery > diff_estimated_delivery
order by time_to_delivery desc
limit 5
```

Query Results:

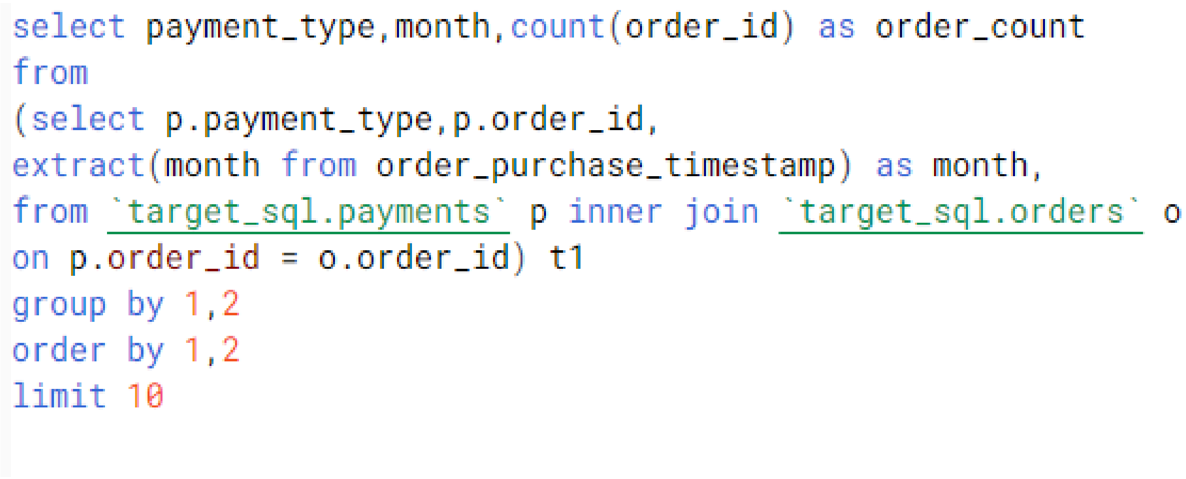


Row	customer_state
1	RR
2	AP
3	AM
4	AL
5	PA

6. Payment type analysis.

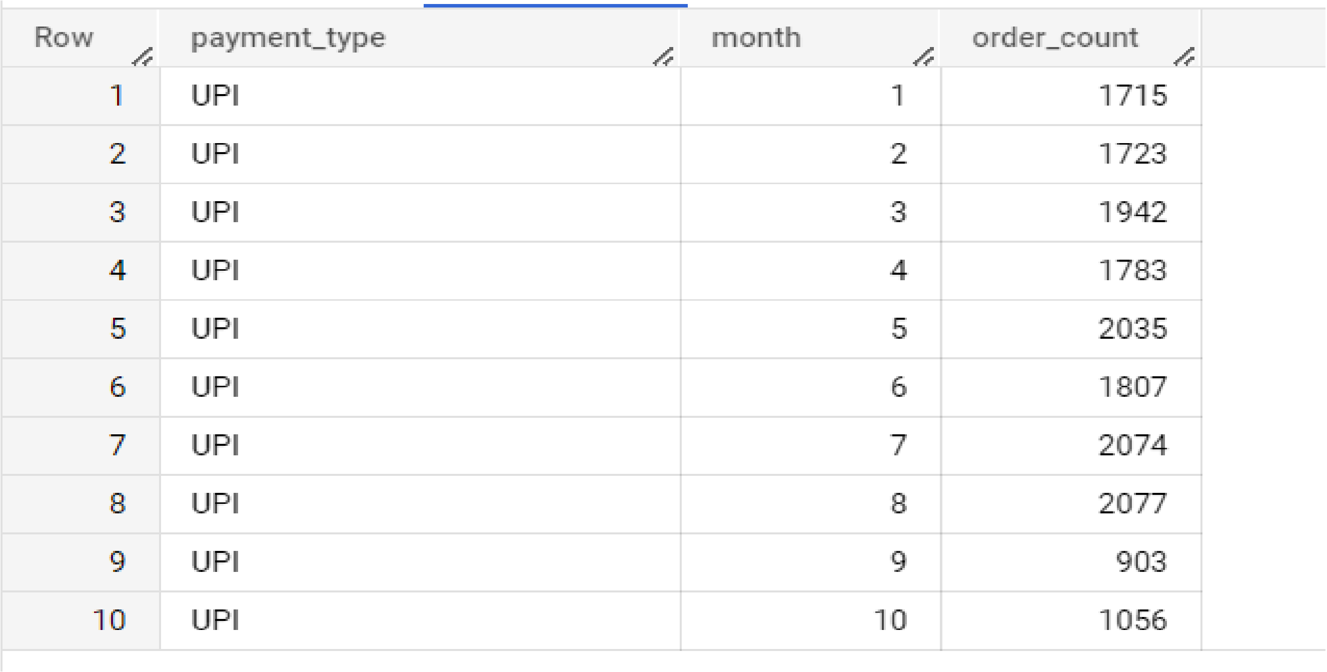
(a) Month over Month count of orders for different payment types.

Query:



```
select payment_type,month,count(order_id) as order_count
from
(select p.payment_type,p.order_id,
extract(month from order_purchase_timestamp) as month,
from `target_sql.payments` p inner join `target_sql.orders` o
on p.order_id = o.order_id) t1
group by 1,2
order by 1,2
limit 10
```

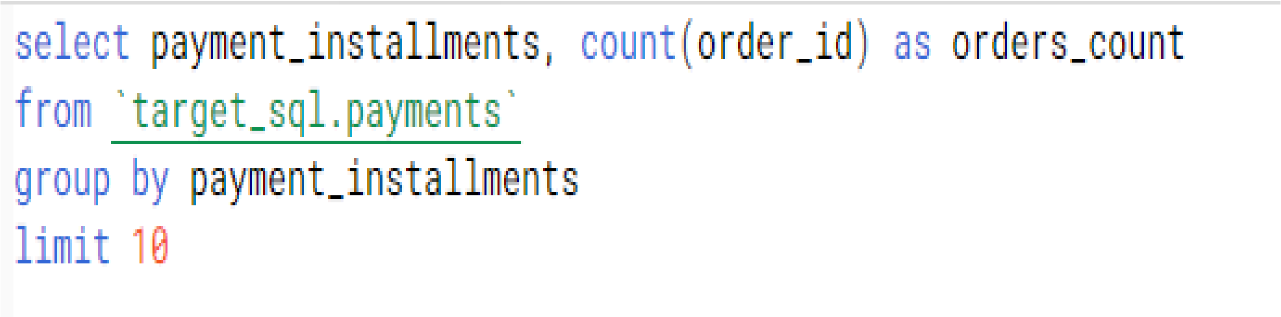
Query Results:



Row	payment_type	month	order_count
1	UPI	1	1715
2	UPI	2	1723
3	UPI	3	1942
4	UPI	4	1783
5	UPI	5	2035
6	UPI	6	1807
7	UPI	7	2074
8	UPI	8	2077
9	UPI	9	903
10	UPI	10	1056

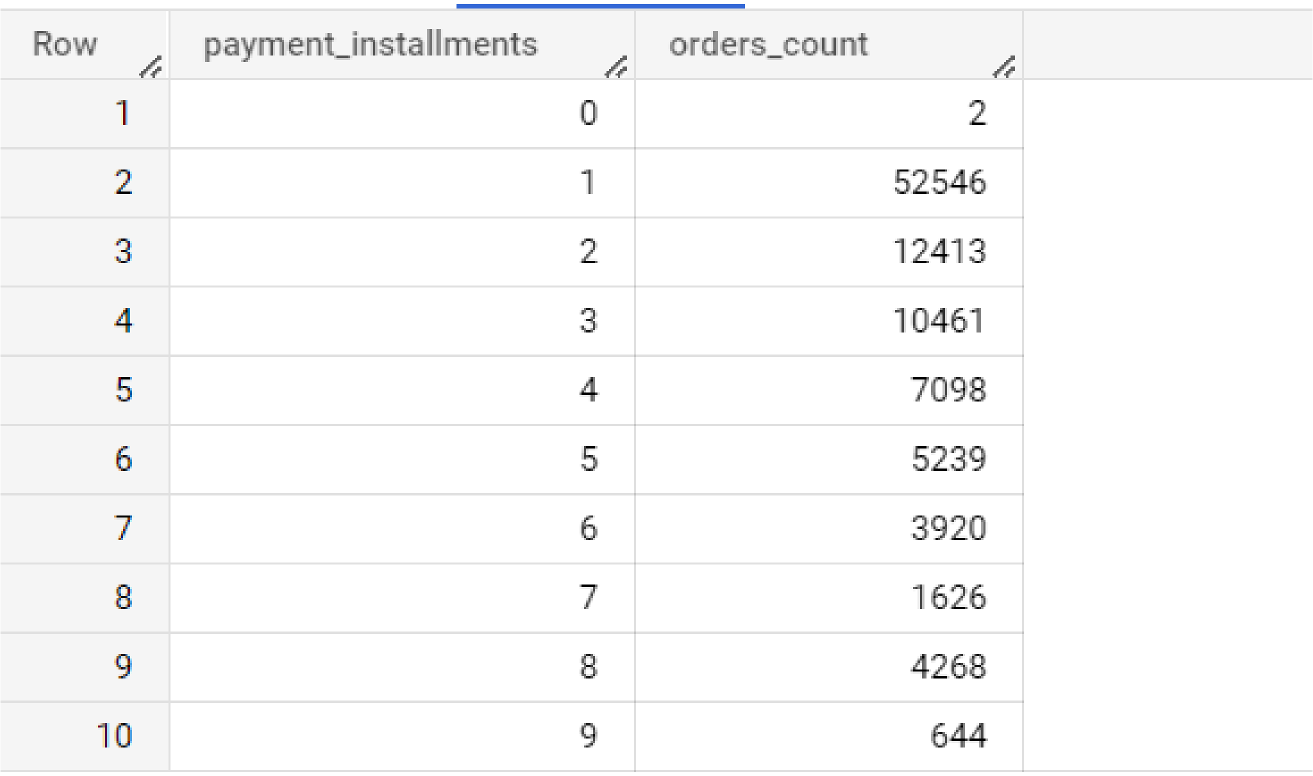
(b) Count of orders based on no. of payment installments.

Query:



```sql
select payment_installments, count(order_id) as orders_count
from `target_sql.payments`
group by payment_installments
limit 10
```

Query Results:



Row	payment_installments	orders_count
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