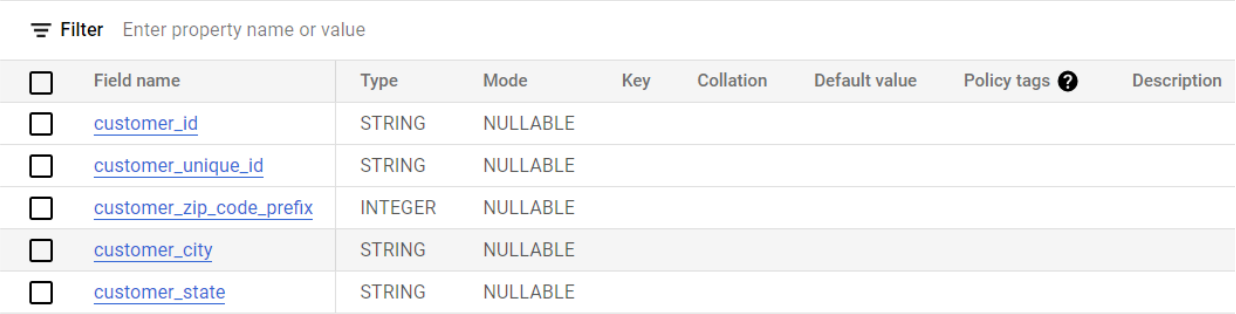
1.Data type of all columns in the "customers" table.

Sol:



Filter Enter property name or value

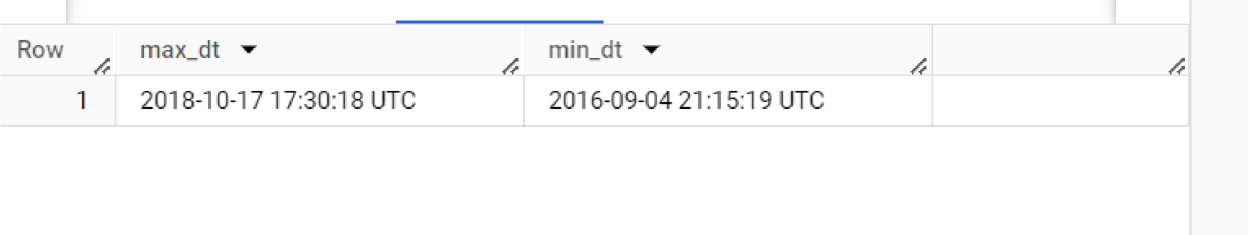
□	Field name	Type	Mode	Key	Collation	Default value	Policy tags	Description
□	customer_id	STRING	NULLABLE					
□	customer_unique_id	STRING	NULLABLE					
□	customer_zip_code_prefix	INTEGER	NULLABLE					
□	customer_city	STRING	NULLABLE					
□	customer_state	STRING	NULLABLE					

There are only 2 data types I,e STRING and INTEGER

2.Get the time range between which the orders were placed.

Sol:

```
select max(order_purchase_timestamp) as max_dt,
 min(order_purchase_timestamp) as min_dt FROM case_study.orders;
```



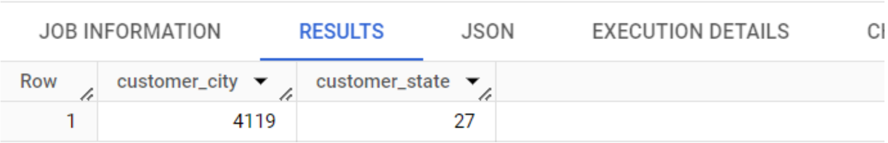
Row	max_dt	min_dt
1	2018-10-17 17:30:18 UTC	2016-09-04 21:15:19 UTC

We can see that , orders has been placed from 04th Sep 2016 to 17th oct 2018 . so, customer engagement was almost for 2 years.

3.Count the Cities & States of customers who ordered during the given period.

Sol:

```
select count(distinct customer_city) as customer_city, count(distinct customer_state ) as
customer_state from case_study.customers
where customer_id IN (select customer_id from case_study.orders );
```



Row	customer_city	customer_state
1	4119	27

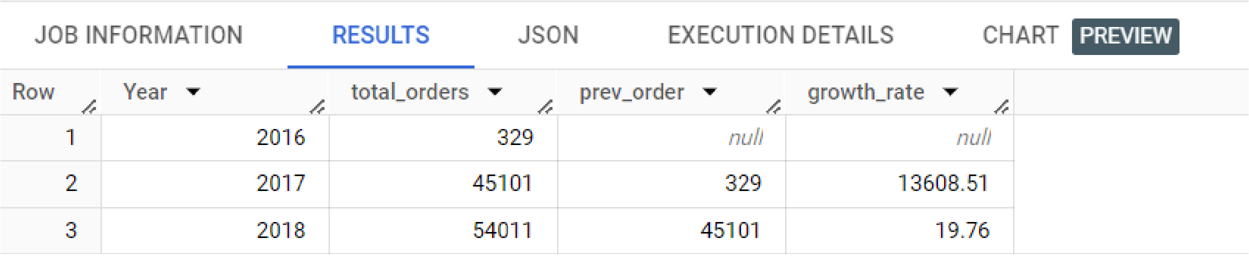
Here , there are total 4119 cities and 27 countries from where customer orders during given time period.

2.1 Is there a growing trend in the no. of orders placed over the past years?
Sol:

```
with cte as (

   select extract(Year from order_purchase_timestamp) as Year, count(*) as total_orders from
case_study.orders
  group by Year order by 1
)

select * , round((total_orders - prev_order) / prev_order * 100,2) as growth_rate
from ( select *,
lag(total_orders) over(order by Year) as prev_order from cte ) a
order by Year
```



Row	Year	total_orders	prev_order	growth_rate
1	2016	329	null	null
2	2017	45101	329	13608.51
3	2018	54011	45101	19.76

Here we can see that , In 2017 13608% growth from 2016 .

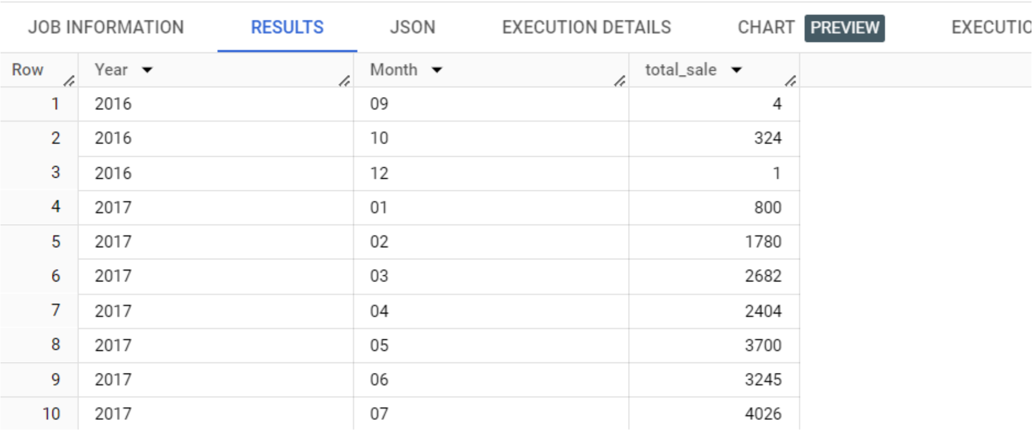
Similarly , In 2018, around 20% growth from 2017.

Overall it's indicate increasing/growing trend on YoY.

2.2 Can we see some kind of monthly seasonality in terms of the no. of orders being placed?

Sol:

```
select FORMAT_TIMESTAMP('%Y',order_purchase_timestamp) as Year,
FORMAT_TIMESTAMP('%m',order_purchase_timestamp) as Month , count(*) as total_sale from
case_study.orders
group by Year,Month
order by 1,2
```



Row	Year	Month	total_sale
1	2016	09	4
2	2016	10	324
3	2016	12	1
4	2017	01	800
5	2017	02	1780
6	2017	03	2682
7	2017	04	2404
8	2017	05	3700
9	2017	06	3245
10	2017	07	4026

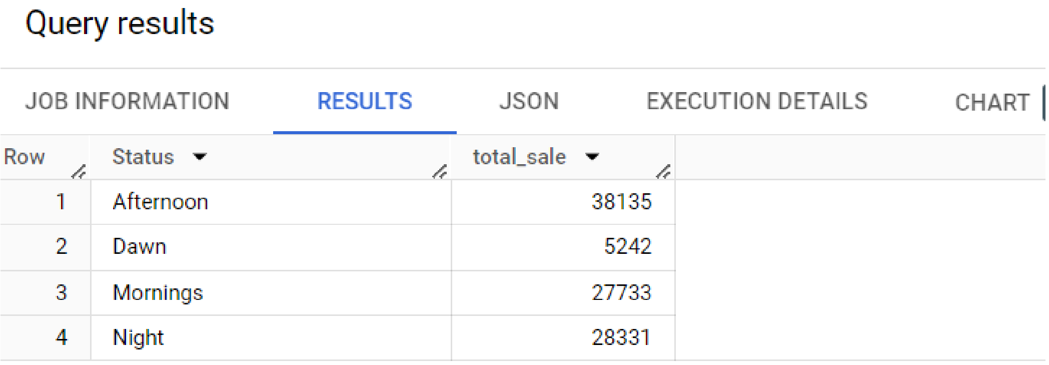
We can see that , there is uptrend in month-wise till 2017 and then again , In 2018 , there is drop may be because of natural calamities , they may have shut the shop.

2.2 During what time of the day, do the Brazilian customers mostly place their orders? (Dawn, Morning, Afternoon or Night)

Sol:

```
with cte as (
select *, EXTRACT(HOUR FROM order_purchase_timestamp) as Hr from case_study.orders
order by Hr
)

select
case when Hr between 0 and 6 then 'Dawn'
     when  Hr between 7 and 12 then 'Mornings'
     when Hr between 13 and 18 then 'Afternoon'
     else 'Night' END Status,
count(*) as total_sale from cte
group by Status
order by 1
```



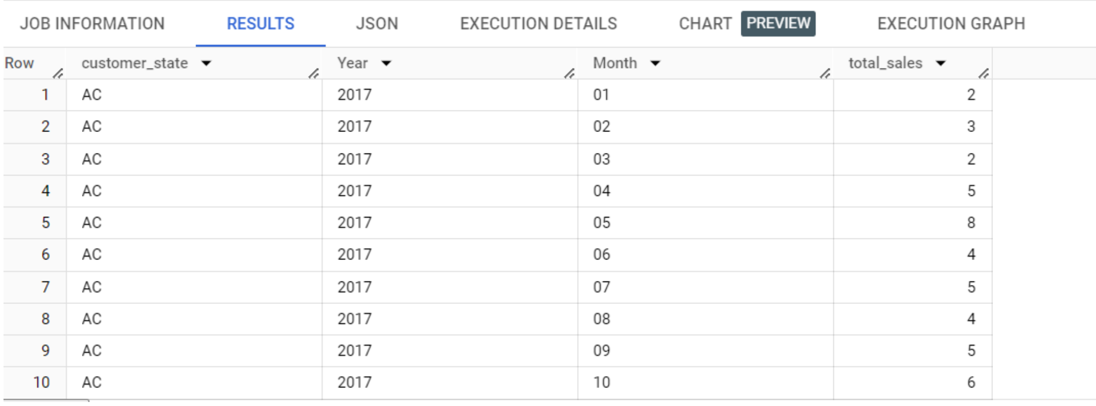
Row	Status	total_sale
1	Afternoon	38135
2	Dawn	5242
3	Mornings	27733
4	Night	28331

We can see that , Brazilian customers mostly place their order in the afternoon . Then after, at 2nd position , order has been placed in Night . Dawn has very poor traffic since very less order has been placed that time.

3.1 Get the month on month no. of orders placed in each state.

Sol:

```
select  c.customer_state, FORMAT_TIMESTAMP('%Y',o.order_purchase_timestamp) as
Year,FORMAT_TIMESTAMP('%m',o.order_purchase_timestamp) as Month,
count(*) as total_sales from case_study.orders o
inner join case_study.customers c on o.customer_id = c.customer_id
group by c.customer_state, Year, Month
order by c.customer_state,Year, Month
```



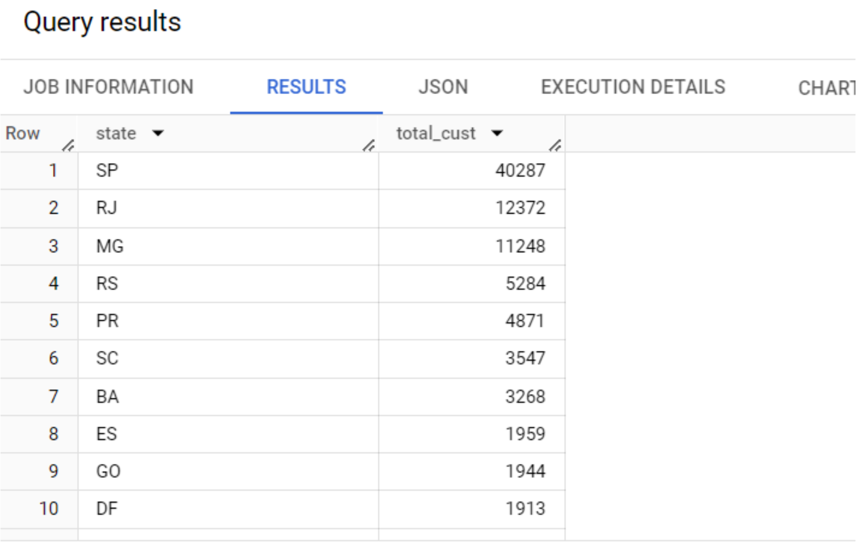
Row	customer_state	Year	Month	total_sales
1	AC	2017	01	2
2	AC	2017	02	3
3	AC	2017	03	2
4	AC	2017	04	5
5	AC	2017	05	8
6	AC	2017	06	4
7	AC	2017	07	5
8	AC	2017	08	4
9	AC	2017	09	5
10	AC	2017	10	6

As we can see here , state AC has avg 5-6 number of orders are placed in each months. Also, we can see there is up / increasing trend in order s placed across year. Year 2016 has lowest number of orders while 2018 has highest number of orders placed.

3.2 How are the customers distributed across all the states?

Sol:

```
select g.geolocation_state as state, count(distinct customer_unique_id) as total_cust from
case_study.customers c INNER JOIN case_study.geolocation g
on c.customer_zip_code_prefix = g.geolocation_zip_code_prefix
group by g.geolocation_state
order by total_cust desc
```



Row	state	total_cust
1	SP	40287
2	RJ	12372
3	MG	11248
4	RS	5284
5	PR	4871
6	SC	3547
7	BA	3268
8	ES	1959
9	GO	1944
10	DF	1913

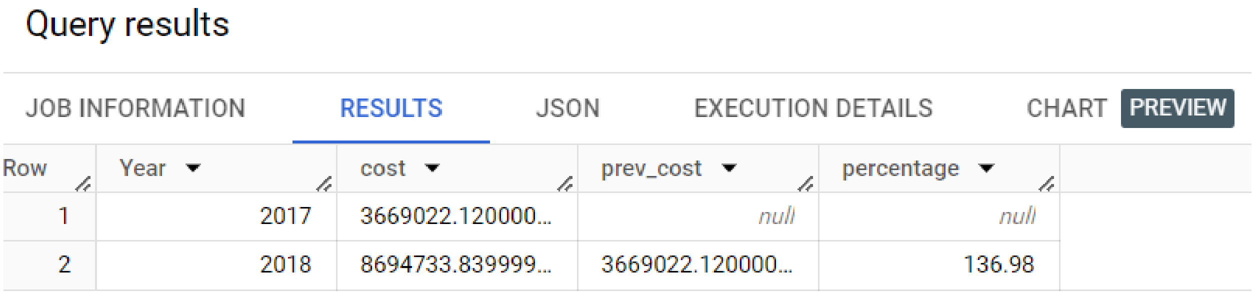
Here , SP state has highest number of customer ie 40k. RJ has the second most number of customers. DF has 1913 numbers of customers are distributed. We can see that , SP state may generate more revenue than other states since more number of customer are distributed there.

4.1 Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).

Sol:

```
with cte as (
select extract(Year from order_purchase_timestamp) as Year, sum(payment_value) as cost from
case_study.orders o
Inner join case_study.payments p on o.order_id = p.order_id
where extract(Year from order_purchase_timestamp) between 2017 and 2018
and extract(Month from order_purchase_timestamp) between 01 and 08
group by Year
order by Year )

select *,
lag(cost) over(order by cost ) as prev_cost,
round((cost-lag(cost) over(order by cost )) / lag(cost) over(order by cost ) * 100,2) as
percentage from cte
order by Year
```



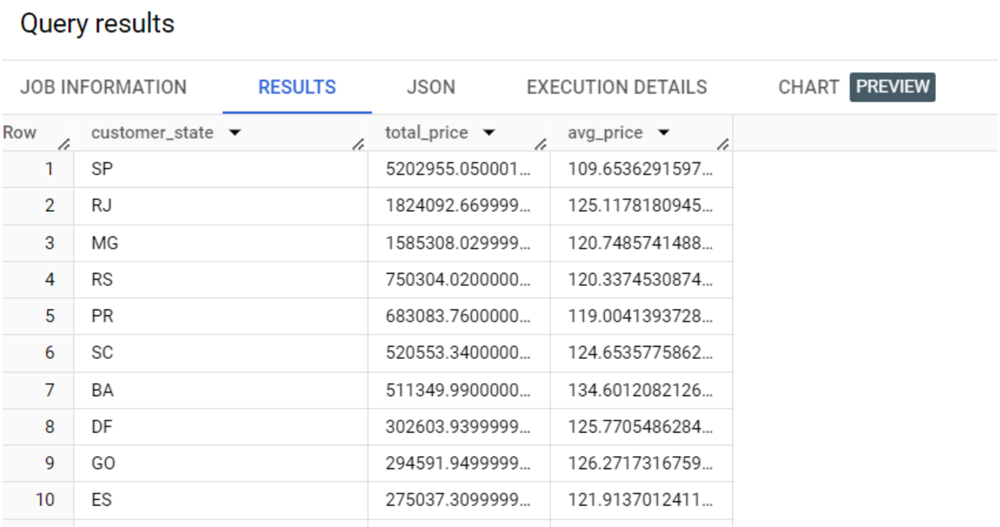
Row	Year	cost	prev_cost	percentage
1	2017	3669022.120000...	*null*	*null*
2	2018	8694733.839999...	3669022.120000...	136.98

Here we can see that , in 2017 , total cost was 3669022 while in 2018, it's increased and reached to 8694733 which is 136% hike from 2017.

4.2 Calculate the Total & Average value of order price for each state.

Sol:

```
select c.customer_state, sum(i.price) as total_price, avg(i.price) as avg_price from
case_study.order_items i
inner join case_study.orders o on i.order_id = o.order_id
Inner join case_study.customers c on o.customer_id = c.customer_id
group by c.customer_state
order by total_price desc ,avg_price desc
```



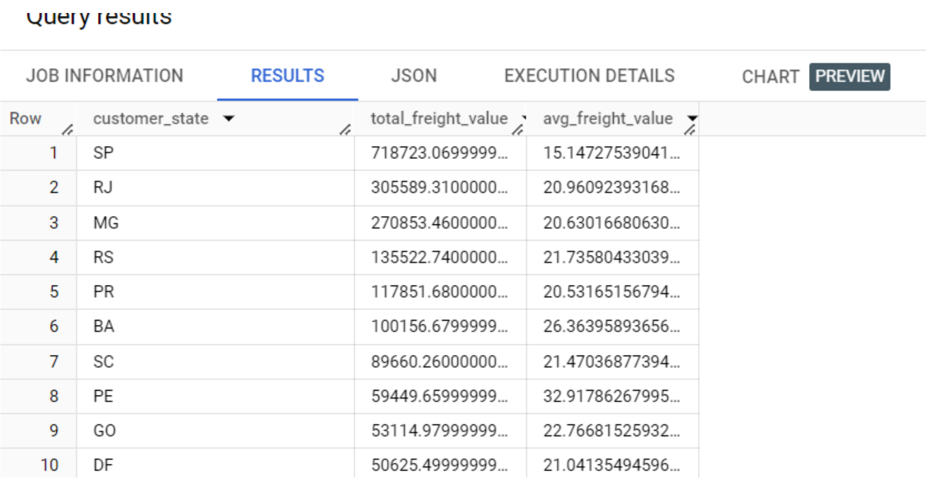
Row	customer_state	total_price	avg_price
1	SP	5202955.050001...	109.6536291597...
2	RJ	1824092.669999...	125.1178180945...
3	MG	1585308.029999...	120.7485741488...
4	RS	750304.0200000...	120.3374530874...
5	PR	683083.7600000...	119.0041393728...
6	SC	520553.3400000...	124.6535775862...
7	BA	511349.9900000...	134.6012082126...
8	DF	302603.9399999...	125.7705486284...
9	GO	294591.9499999...	126.2717316759...
10	ES	275037.3099999...	121.9137012411...

We can see that , SP state has highest number of total sale followed by RJ , MG etc while , lowest sale has RR state which is only 8k approximately. Similarly , Avg price of SP states is 109 .

4.3 Calculate the Total & Average value of order freight for each state.

Sol:

```
select c.customer_state, sum(i.freight_value) as total_price, avg(i.freight_value) as
avg_price from case_study.order_items i
inner join case_study.orders o on i.order_id = o.order_id
Inner join case_study.customers c on o.customer_id = c.customer_id
group by c.customer_state
order by total_price  desc
```



Row	customer_state	total_freight_value	avg_freight_value
1	SP	718723.0699999...	15.14727539041...
2	RJ	305589.3100000...	20.96092393168...
3	MG	270853.4600000...	20.63016680630...
4	RS	135522.7400000...	21.73580433039...
5	PR	117851.6800000...	20.53165156794...
6	BA	100156.6799999...	26.36395893656...
7	SC	89660.26000000...	21.47036877394...
8	PE	59449.65999999...	32.91786267995...
9	GO	53114.97999999...	22.76681525932...
10	DF	50625.49999999...	21.04135494596...

Here , we can see that state SP has highest freight value , followed by RJ , MG , RS and so on .

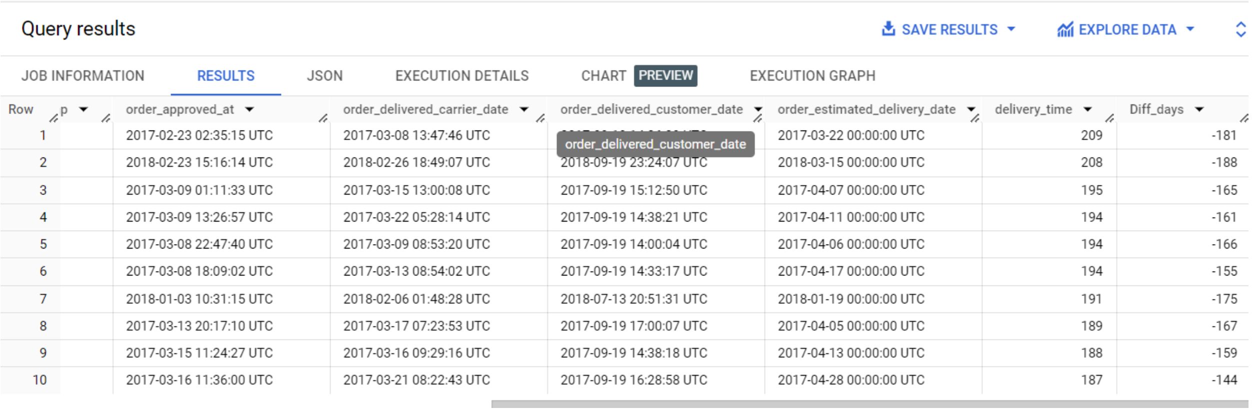
Highest freight value is calculated as 718723 in SP state. Also , Average fright value of SP has 15 .

It’s observe that avg freight value comes to 20 to 21 across country.

5.1 Find the no. of days taken to deliver each order from the order’s purchase date as delivery time.
Also, calculate the difference (in days) between the estimated & actual delivery date of an order.
Do this in a single query.

Sol:

```
select *, date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY) as
delivery_time,
date_diff(order_estimated_delivery_date,order_delivered_customer_date,DAY) as Diff_days from
case_study.orders
order by delivery_time desc
```



Row	p	order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date	delivery_time	Diff_days
1		2017-02-23 02:35:15 UTC	2017-03-08 13:47:46 UTC	[illegible]	2017-03-22 00:00:00 UTC	209	-181
2		2018-02-23 15:16:14 UTC	2018-02-26 18:49:07 UTC	2018-09-19 23:24:07 UTC	2018-03-15 00:00:00 UTC	208	-188
3		2017-03-09 01:11:33 UTC	2017-03-15 13:00:08 UTC	2017-09-19 15:12:50 UTC	2017-04-07 00:00:00 UTC	195	-165
4		2017-03-09 13:26:57 UTC	2017-03-22 05:28:14 UTC	2017-09-19 14:38:21 UTC	2017-04-11 00:00:00 UTC	194	-161
5		2017-03-08 22:47:40 UTC	2017-03-09 08:53:20 UTC	2017-09-19 14:00:04 UTC	2017-04-06 00:00:00 UTC	194	-166
6		2017-03-08 18:09:02 UTC	2017-03-13 08:54:02 UTC	2017-09-19 14:33:17 UTC	2017-04-17 00:00:00 UTC	194	-155
7		2018-01-03 10:31:15 UTC	2018-02-06 01:48:28 UTC	2018-07-13 20:51:31 UTC	2018-01-19 00:00:00 UTC	191	-175
8		2017-03-13 20:17:10 UTC	2017-03-17 07:23:53 UTC	2017-09-19 17:00:07 UTC	2017-04-05 00:00:00 UTC	189	-167
9		2017-03-15 11:24:27 UTC	2017-03-16 09:29:16 UTC	2017-09-19 14:38:18 UTC	2017-04-13 00:00:00 UTC	188	-159
10		2017-03-16 11:36:00 UTC	2017-03-21 08:22:43 UTC	2017-09-19 16:28:58 UTC	2017-04-28 00:00:00 UTC	187	-144

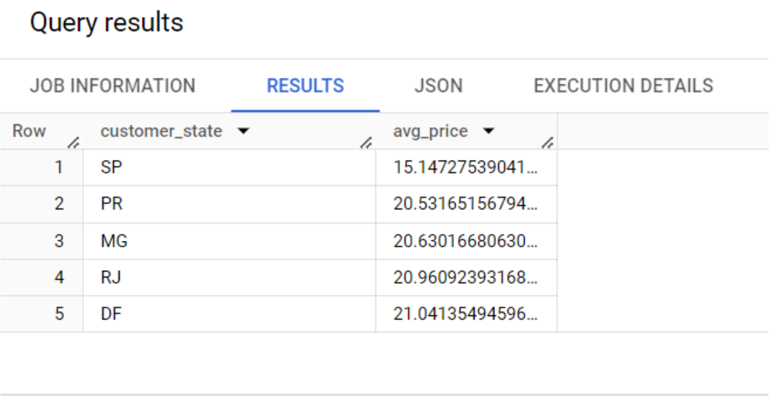
Here , as you can see , max delivery time has taken 209 days too from the date of purchase. So, for almost all order delivery time has taken more than 100 days atleast so seems they need to improve in logistic area. Also, maximum day taken from estimated to time from purchase is -181 .

5.2 Find out the top 5 states with the highest & lowest average freight value.

Sol:

--FOR LOWEST STATE

```
select c.customer_state, avg(i.freight_value) as lowest_avg_price from case_study.order_items
i
inner join case_study.orders o on i.order_id = o.order_id
Inner join case_study.customers c on o.customer_id = c.customer_id
group by c.customer_state
order by lowest_avg_price
limit 5;
```

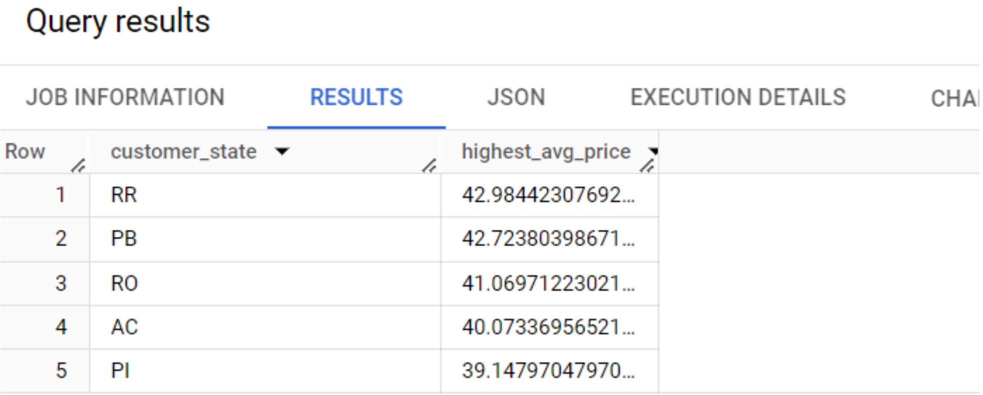


Row	customer_state	avg_price
1	SP	15.14727539041...
2	PR	20.53165156794...
3	MG	20.63016680630...
4	RJ	20.96092393168...
5	DF	21.04135494596...

Here we can see that , first lowest avg freight value state is SP which is 15.14 , followed by PR, then MG etc. DF has 21.04 avg freight value.

- FOR HIGHEST STATE

```
select c.customer_state, avg(i.freight_value) as highest_avg_price from case_study.order_items
i
inner join case_study.orders o on i.order_id = o.order_id
Inner join case_study.customers c on o.customer_id = c.customer_id
group by c.customer_state
order by highest_avg_price desc
limit 5;
```

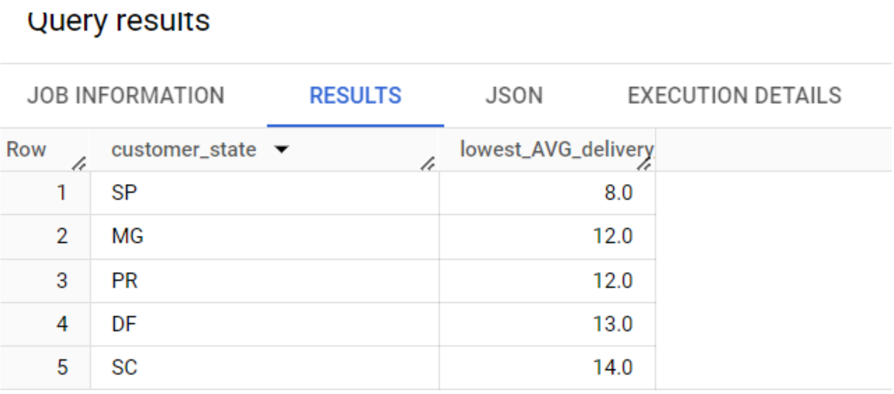


Row	customer_state	highest_avg_price
1	RR	42.98442307692...
2	PB	42.72380398671...
3	RO	41.06971223021...
4	AC	40.07336956521...
5	PI	39.14797047970...

Here , it's observe that highest avg freight value is 42 which is for RR value , followed by PB which is 42.72 then RP etc. Average value of PI is 39.

5.3 Find out the top 5 states with the highest & lowest average delivery time.
Sol:

```
select c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY))) as
lowest_AVG_delivery_time
from case_study.orders o
Inner join case_study.customers c on o.customer_id = c.customer_id
group by c.customer_state
order by lowest_AVG_delivery_time
limit 5;
```

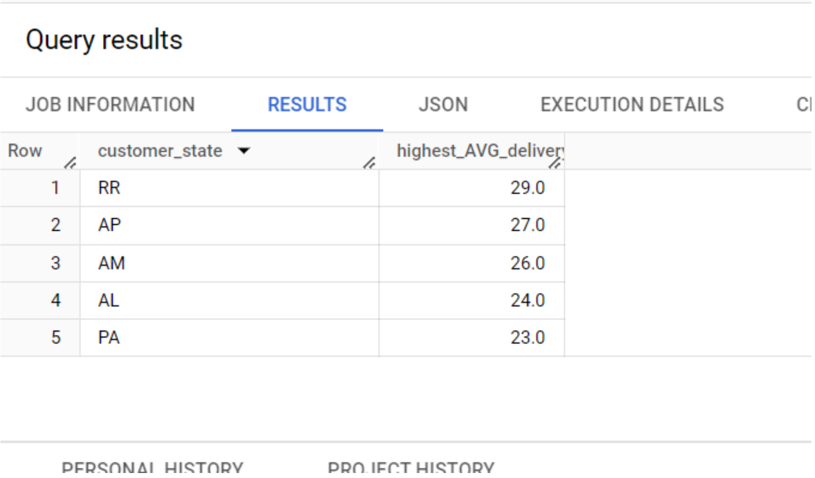


Row	customer_state	lowest_AVG_delivery
1	SP	8.0
2	MG	12.0
3	PR	12.0
4	DF	13.0
5	SC	14.0

It’s observe that , SP state has lowest avg delivey time that is 8 days , followed by MG which is 12 days etc. SC state has 14 days delivery time.

-- for highest state

```
select c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_purchase_timestamp,DAY))) as
highest_AVG_delivery_time
from case_study.orders o
Inner join case_study.customers c on o.customer_id = c.customer_id
group by c.customer_state
order by highest_AVG_delivery_time desc
limit 5;
```



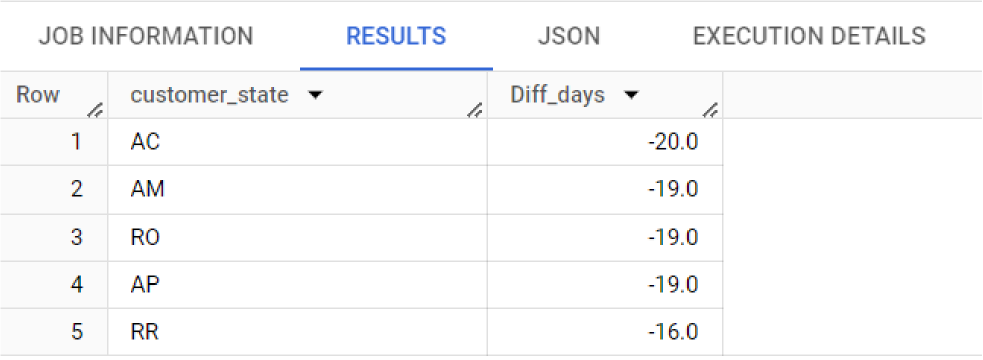
Row	customer_state	highest_AVG_deliver
1	RR	29.0
2	AP	27.0
3	AM	26.0
4	AL	24.0
5	PA	23.0

Here , we can see that , RR state has highest avg delivert time which is 29 , followed by AP whch is 27 day etc. PA has 23 days avg delivery time , which is the least highest.

5.4 Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

Sol:

```
select c.customer_state,
round(avg(date_diff(order_delivered_customer_date,order_estimated_delivery_date,DAY))) as
Diff_days from
case_study.orders o
Inner join case_study.customers c on o.customer_id = c.customer_id
where date_diff(order_estimated_delivery_date,o.order_delivered_customer_date,DAY) IS NOT NULL
group by c.customer_state
order by Diff_days
limit 5
```



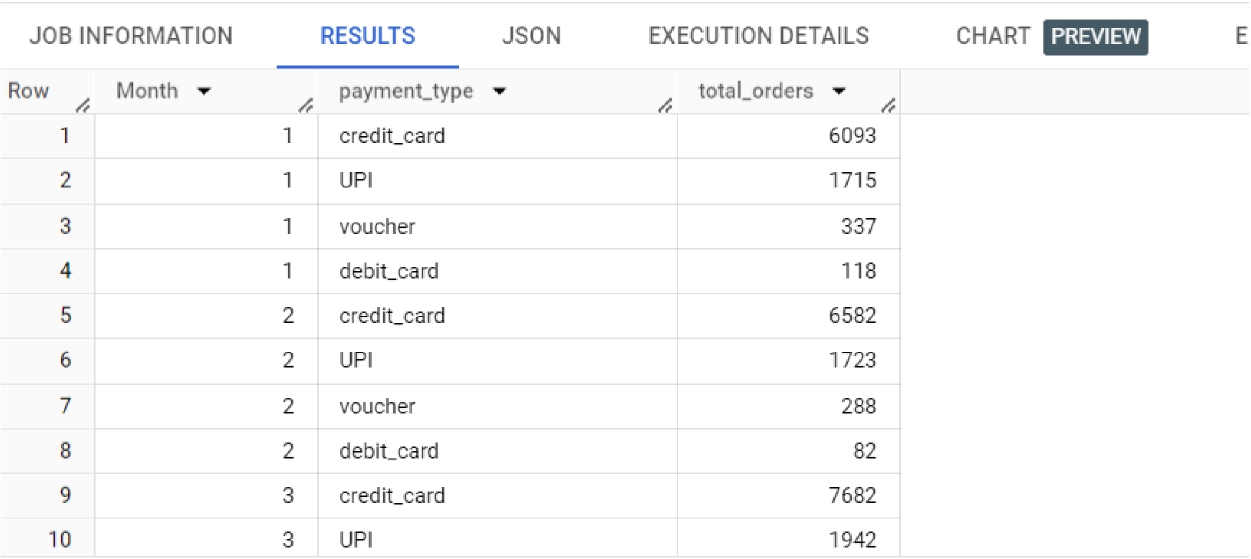
Row	customer_state	Diff_days
1	AC	-20.0
2	AM	-19.0
3	RO	-19.0
4	AP	-19.0
5	RR	-16.0

Here, it’s notice that state AC has reached 20 days earlier which is fastest delivery days, followed by AM which is 19 days, and so on. RR state has 16 days fastest delivery than estimated delivery time.

6.1 Find the month on month no. of orders placed using different payment types.

Sol:

```
select extract(Month from o.order_purchase_timestamp) as Month,p.payment_type, count(distinct
p.order_id) as total_orders from case_study.payments p
inner join case_study.orders o on
p.order_id = o.order_id
group by Month, p.payment_type
order by Month,total_orders desc
```



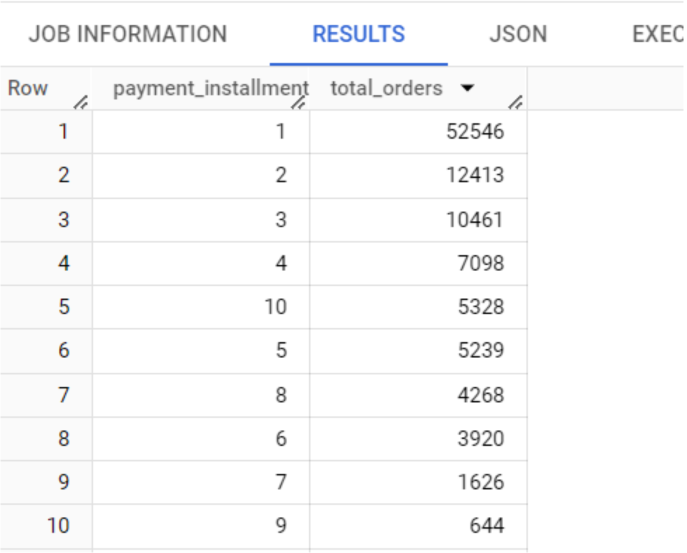
Row	Month	payment_type	total_orders
1	1	credit_card	6093
2	1	UPI	1715
3	1	voucher	337
4	1	debit_card	118
5	2	credit_card	6582
6	2	UPI	1723
7	2	voucher	288
8	2	debit_card	82
9	3	credit_card	7682
10	3	UPI	1942

It is observe that , we have CC , UPI , voucher, DC as various payment method. In these , Credit card ares widely used by customer which is 6093. second highest method is UPI while debit cards are least widely used across country.

6.2 Find the no. of orders placed on the basis of the payment installments that have been paid.

Sol:

```
select payment_installments, count(*) as total_orders from case_study.payments
group by 1
order by total_orders desc
```



Row	payment_installment	total_orders
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328
6	5	5239
7	8	4268
8	6	3920
9	7	1626
10	9	644

Here it’s observe that , payment installment one has highest total orders which is 52546 and as payment installments increasing , total orders placed are decreasing.

Conclusion:

After analyzing all data, it’s observe that this is the 3 year data from 2016 to 2018. There is up trend in sales across year from 2016 to 2018. Also , in terms of monthly seasonality , we can see increasing sales till 2017 and there is a bit downtrend in 2018. Also , customers are so active in afternoon so it would be better if target can send some sorta discount in that period.

Also , from payment point of view , it’s observe that , customer are using credit card most. So they should give more discounts on credit cards. Second highest method people using is UPI.