SQL 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 covered in the dataset

Insights: We haven't made any changes in the data type of columns, we are only filtering the data from the table. As in exploratory analysis , we are able to find that this is the data related to ecommerce orders , sales ,customer orders . As we see in date , date are given in timestamp data type

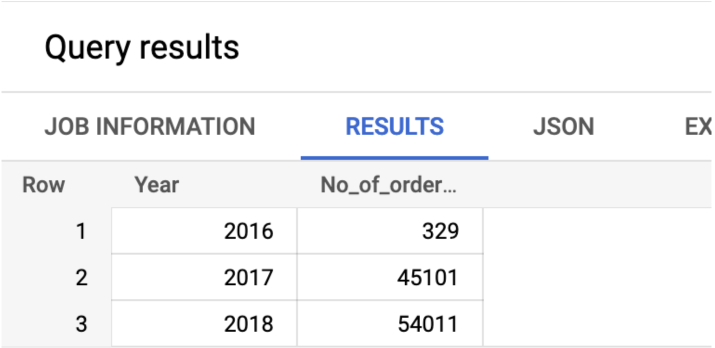
We found different in-depth exploration like

- Growing trend on E-commerce
- Location wise orders
- State wise orders
- Month wise orders and year wise (using Pivot table)
- Mean of Freight Value , Orders_cost - statewise ,Product Category wise
- Product Category wise orders
- Product orders in different daytime(Dawn, Morning , Afternoon , Evening, Night)
- Percentage increase in cost of orders (Category wise)
- Delivered , estimated, purchased date - day difference
- Payment type analysis
- Payment installment analysis
- Customers distribution in Brazil citywise

2. In-depth Exploration:

    1. Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

```
select  extract (year from order_purchase_timestamp) as Year, count(order_id) as
No_of_orders_placed  from `ecommerce.orders`  group by Year order by Year
```



Query results

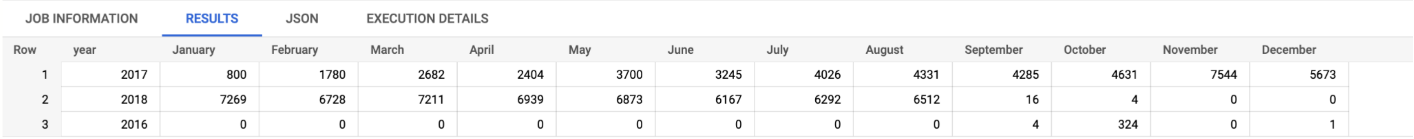
Row	Year	No_of_order...
1	2016	329
2	2017	45101
3	2018	54011

YEAR WISE ORDERS RECEIVED

QUERY:

```
select year, IFNULL(_1, 0 ) as January , IFNULL(_2, 0 ) as February , IFNULL(_3, 0 ) as
March, IFNULL(_4, 0 ) as April, IFNULL(_5, 0 ) as May ,

IFNULL(_6, 0 ) as June , IFNULL(_7, 0 ) as July , IFNULL(_8, 0 ) as August , IFNULL(_9, 0 )
as September , IFNULL(_10, 0 ) as October , IFNULL(_11, 0 ) as November , IFNULL(_12, 0 )
as December from

(SELECT *

FROM (SELECT extract (year from order_purchase_timestamp) as year,

    extract (month from order_purchase_timestamp) as month,

    COUNT(1) as order_count

   FROM `ecommerce.orders`

   GROUP BY extract (year from order_purchase_timestamp),

   extract (month from order_purchase_timestamp))  AS MontlySalesData

PIVOT( SUM(order_count)
```

FOR Month IN (1,2,3,4,5,6,7,8,9,10,11,12)) AS MNamePivot)



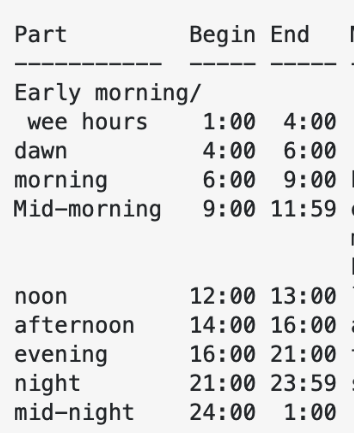
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	year	January	February	March	April	May	June	July	August	September	October	November	December
1	2017	800	1780	2682	2404	3700	3245	4026	4331	4285	4631	7544	5673
2	2018	7269	6728	7211	6939	6873	6167	6292	6512	16	4	0	0
3	2016	0	0	0	0	0	0	0	0	4	324	0	1

Insights: We can clearly see 2017-November has the highest orders received.

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

Precise names for parts of a day represented in analysis



```
Part         Begin End
-----------  ----- -----
Early morning/
 wee hours    1:00  4:00
dawn          4:00  6:00
morning       6:00  9:00
Mid-morning   9:00 11:59


noon         12:00 13:00
afternoon    14:00 16:00
evening      16:00 21:00
night        21:00 23:59
mid-night    24:00  1:00
```

Source: https://english.stackexchange.com/questions/28498/precise-names-for-parts-of-a-day

Query:

(SELECT *

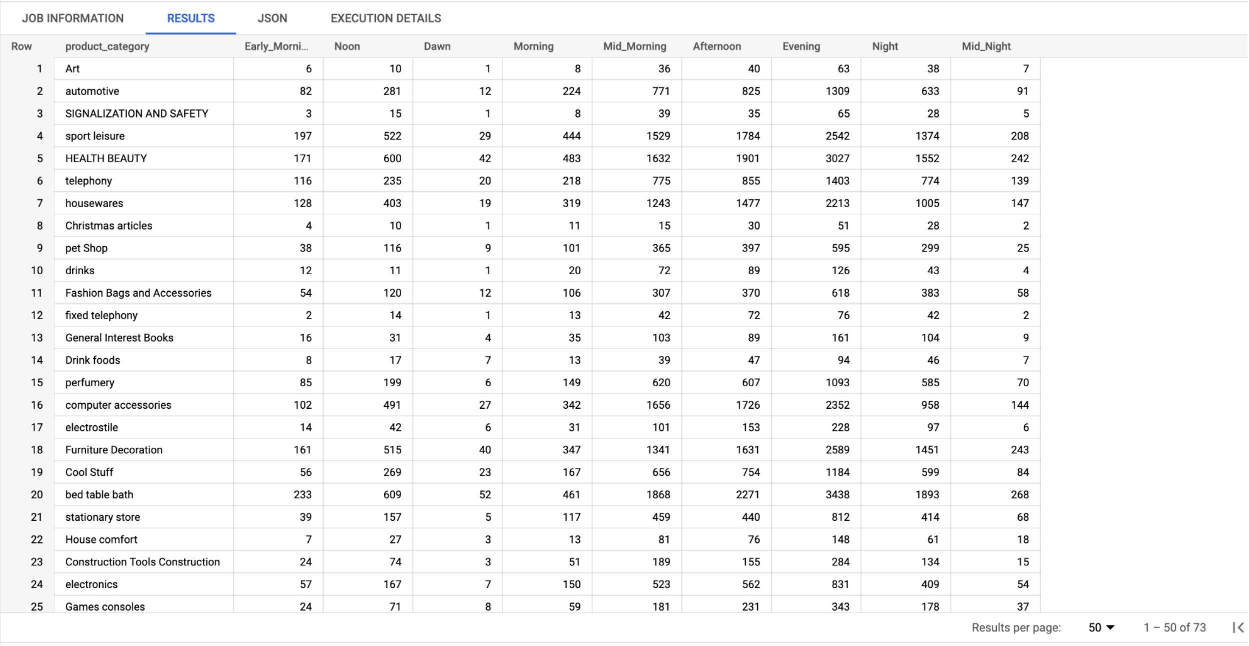
FROM (SELECT a.product_category,

CASE

when TIME(order_purchase_timestamp) BETWEEN '01:00:00' AND '04:00:00' then 'Early_Morning'

when TIME(order_purchase_timestamp) BETWEEN '04:00:00' AND '06:00:00' then 'Dawn'

```
          when TIME(order_purchase_timestamp)  BETWEEN '06:00:00' AND '09:00:00' then
'Morning'
          when TIME(order_purchase_timestamp)  BETWEEN '09:00:00' AND '11:59:00' then
'Mid_Morning'
          when TIME(order_purchase_timestamp)  BETWEEN '12:00:00' AND '13:00:00' then 'Noon'
          when TIME(order_purchase_timestamp)  BETWEEN '13:00:00' AND '16:00:00' then
'Afternoon'
          when TIME(order_purchase_timestamp)  BETWEEN '16:00:00' AND '21:00:00' then
'Evening'
          when TIME(order_purchase_timestamp)  BETWEEN '21:00:00' AND '23:59:00' then 'Night'
          when TIME(order_purchase_timestamp)  BETWEEN '00:00:00' AND '01:00:00' then
'Mid_Night'
          else 'None'
          end as Period,
              COUNT(1) as order_count
             from `ecommerce.products` a ,`ecommerce.orders` b , `ecommerce.order_items` c
             where a.product_category is not null
          and a.product_id= c.product_id and b.order_id = c.order_id
          group by a.product_category , Period)  AS MonthlyOrderData
          PIVOT( SUM(order_count)
            FOR Period IN ('Early_Morning','Noon','Dawn','Morning','Mid_Morning', 'Afternoon',
'Evening','Night','Mid_Night')) AS orderpivot)
```



Row	product_category	Early_Morni...	Noon	Dawn	Morning	Mid_Morning	Afternoon	Evening	Night	Mid_Night
1	Art	6	10	1	8	36	40	63	38	7
2	automotive	82	281	12	224	771	825	1309	633	91
3	SIGNALIZATION AND SAFETY	3	15	1	8	39	35	65	28	5
4	sport leisure	197	522	29	444	1529	1784	2542	1374	208
5	HEALTH BEAUTY	171	600	42	483	1632	1901	3027	1552	242
6	telephony	116	235	20	218	775	855	1403	774	139
7	housewares	128	403	19	319	1243	1477	2213	1005	147
8	Christmas articles	4	10	1	11	15	30	51	28	2
9	pet Shop	38	116	9	101	365	397	595	299	25
10	drinks	12	11	1	20	72	89	126	43	4
11	Fashion Bags and Accessories	54	120	12	106	307	370	618	383	58
12	fixed telephony	2	14	1	13	42	72	76	42	2
13	General Interest Books	16	31	4	35	103	89	161	104	9
14	Drink foods	8	17	7	13	39	47	94	46	7
15	perfumery	85	199	6	149	620	607	1093	585	70
16	computer accessories	102	491	27	342	1656	1726	2352	958	144
17	electrostile	14	42	6	31	101	153	228	97	6
18	Furniture Decoration	161	515	40	347	1341	1631	2589	1451	243
19	Cool Stuff	56	269	23	167	656	754	1184	599	84
20	bed table bath	233	609	52	461	1868	2271	3438	1893	268
21	stationary store	39	157	5	117	459	440	812	414	68
22	House comfort	7	27	3	13	81	76	148	61	18
23	Construction Tools Construction	24	74	3	51	189	155	284	134	15
24	electronics	57	167	7	150	523	562	831	409	54
25	Games consoles	24	71	8	59	181	231	343	178	37

Insights: We can see there are more number of orders placed in evening and some of category which has more orders are health beauty, Computer accessories , bed table bath.

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

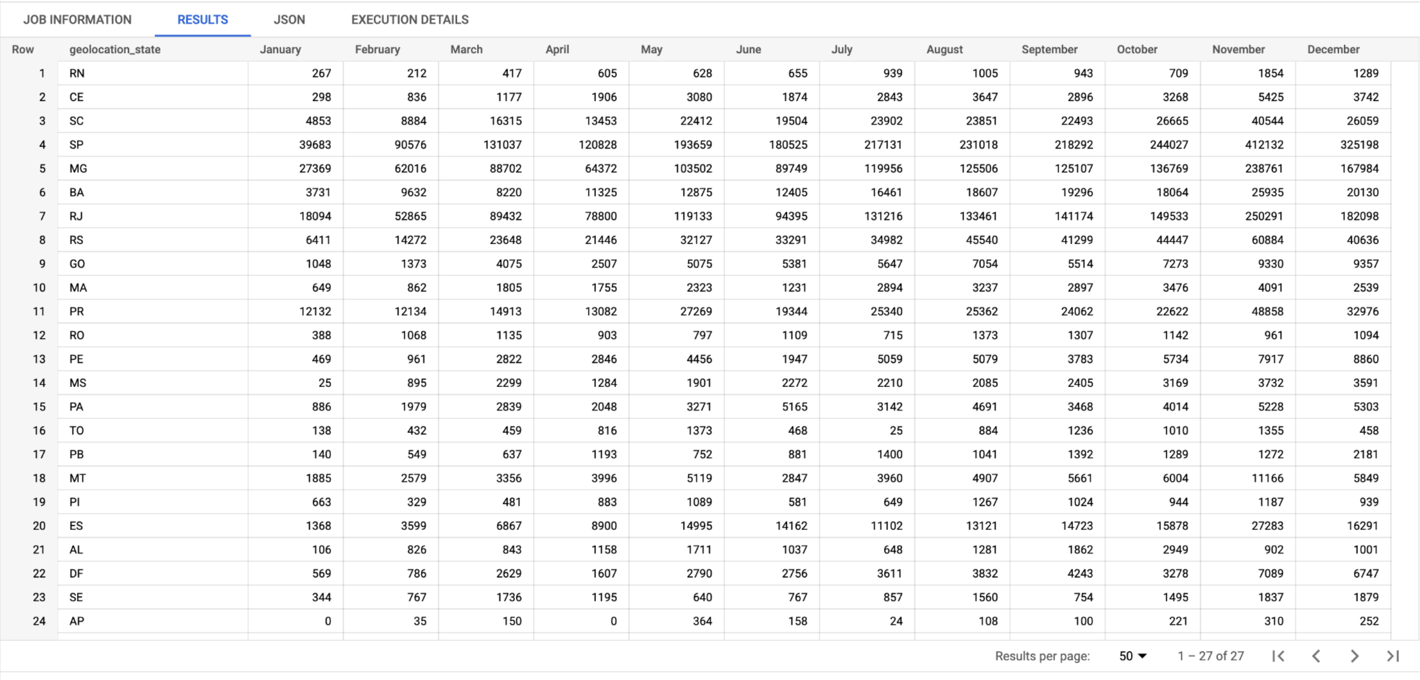
    1. Get month on month orders by region, states

Month on Month Orders by Region, Sales

In this analysis , we are considering 2017 year orders statewise ,and no order_status included in query , because , as we are considering the increasing number of orders month on month.

Query:

```
select geolocation_state, IFNULL(_1, 0 ) as January , IFNULL(_2, 0 ) as February ,
IFNULL(_3, 0 ) as March, IFNULL(_4, 0 ) as April, IFNULL(_5, 0 ) as May ,

IFNULL(_6, 0 ) as June , IFNULL(_7, 0 ) as July , IFNULL(_8, 0 ) as August , IFNULL(_9, 0 )
as September , IFNULL(_10, 0 ) as October , IFNULL(_11, 0 ) as November , IFNULL(_12, 0 )
as December from

(SELECT *

FROM (SELECT b.geolocation_state,

   extract (month from a.order_purchase_timestamp) as month,
     COUNT(1) as order_count

    FROM `ecommerce.orders` a, `ecommerce.geolocation` b , `ecommerce.customers` c

    where b.geolocation_zip_code_prefix = c.customer_zip_code_prefix and a.customer_id=
c.customer_id and extract (year from a.order_purchase_timestamp)= 2017

    GROUP BY b.geolocation_state,

    extract (month from a.order_purchase_timestamp))  AS MonthlyOrderData

PIVOT( SUM(order_count)

  FOR Month IN (1,2,3,4,5,6,7,8,9,10,11,12)) AS orderpivot)
```



Row	geolocation_state	January	February	March	April	May	June	July	August	September	October	November	December
1	RN	267	212	417	605	628	655	939	1005	943	709	1854	1289
2	CE	298	836	1177	1906	3080	1874	2843	3647	2896	3268	5425	3742
3	SC	4853	8884	16315	13453	22412	19504	23902	23851	22493	26665	40544	26059
4	SP	39683	90576	131037	120828	193659	180525	217131	231018	218292	244027	412132	325198
5	MG	27369	62016	88702	64372	103502	89749	119956	125506	125107	136769	238761	167984
6	BA	3731	9632	8220	11325	12875	12405	16461	18607	19296	18064	25935	20130
7	RJ	18094	52865	89432	78800	119133	94395	131216	133461	141174	149533	250291	182098
8	RS	6411	14272	23648	21446	32127	33291	34982	45540	41299	44447	60884	40636
9	GO	1048	1373	4075	2507	5075	5381	5647	7054	5514	7273	9330	9357
10	MA	649	862	1805	1755	2323	1231	2894	3237	2897	3476	4091	2539
11	PR	12132	12134	14913	13082	27269	19344	25340	25362	24062	22622	48858	32976
12	RO	388	1068	1135	903	797	1109	715	1373	1307	1142	961	1094
13	PE	469	961	2822	2846	4456	1947	5059	5079	3783	5734	7917	8860
14	MS	25	895	2299	1284	1901	2272	2210	2085	2405	3169	3732	3591
15	PA	886	1979	2839	2048	3271	5165	3142	4691	3468	4014	5228	5303
16	TO	138	432	459	816	1373	468	25	884	1236	1010	1355	458
17	PB	140	549	637	1193	752	881	1400	1041	1392	1289	1272	2181
18	MT	1885	2579	3356	3996	5119	2847	3960	4907	5661	6004	11166	5849
19	PI	663	329	481	883	1089	581	649	1267	1024	944	1187	939
20	ES	1368	3599	6867	8900	14995	14162	11102	13121	14723	15878	27283	16291
21	AL	106	826	843	1158	1711	1037	648	1281	1862	2949	902	1001
22	DF	569	786	2629	1607	2790	2756	3611	3832	4243	3278	7089	6747
23	SE	344	767	1736	1195	640	767	857	1560	754	1495	1837	1879
24	AP	0	35	150	0	364	158	24	108	100	221	310	252

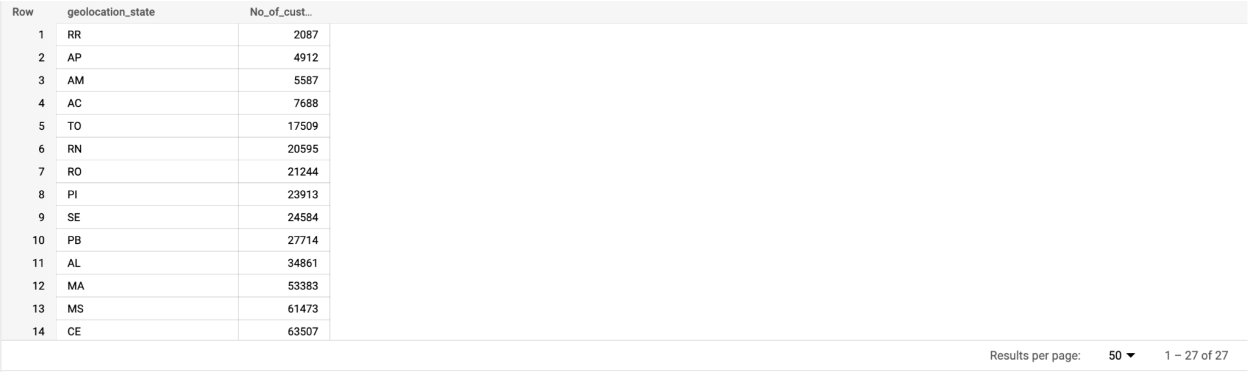
Insights: As we clearly see from the data , more number of orders are from SP State.and less number of orders from AP state. And the months in which more orders placed are November and December

2. How are customers distributed in Brazil

STATE WISE CUSTOMER DISTRIBUTION:

Query:

```
select b.geolocation_state, count(customer_id) as No_of_customers

from `ecommerce.customers` a , `ecommerce.geolocation` b

where b.geolocation_zip_code_prefix = a.customer_zip_code_prefix

group by b.geolocation_state

order by No_of_customers
```



Row	geolocation_state	No_of_cust...
1	RR	2087
2	AP	4912
3	AM	5587
4	AC	7688
5	TO	17509
6	RN	20595
7	RO	21244
8	PI	23913
9	SE	24584
10	PB	27714
11	AL	34861
12	MA	53383
13	MS	61473
14	CE	63507

Insights: As we see from data, less customers are from RR state and more from SP

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

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

      % INCREASE IN COST ORDERS

      Creating view for 2017 year:

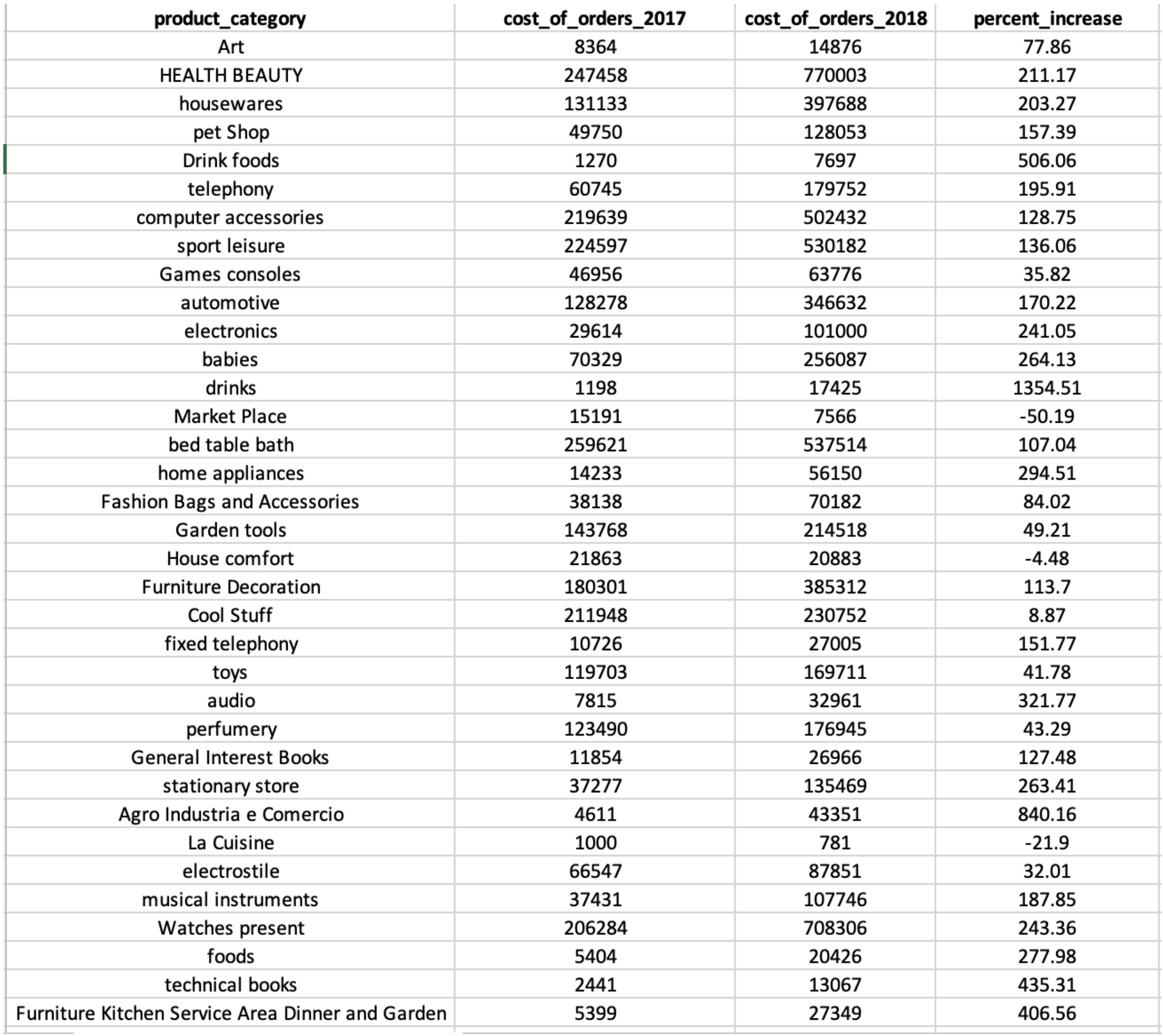
```
create view `mysqlproject-360810.ecommerce.myview` as

(select a.product_category, round(sum(b.price),0) as Price  from

`ecommerce.products` a , `ecommerce.order_items` b , `ecommerce.orders` c
where a.product_id = b.product_id and b.order_id= c.order_id and extract (year from
c.order_purchase_timestamp)=2017 and

(extract (month from c.order_purchase_timestamp) between 1 and 8) and

c.order_status not in ('canceled','unavailable') and a.product_category is not null

group by a.product_category)
```

Creating view for 2018 year

```
create view `mysqlproject-360810.ecommerce.myview1` as

(select a.product_category,  round(sum(b.price),0)  as Price   from

`ecommerce.products` a , `ecommerce.order_items` b , `ecommerce.orders` c

where a.product_id = b.product_id and b.order_id= c.order_id and extract (year from
c.order_purchase_timestamp)=2018 and

(extract (month from c.order_purchase_timestamp) between 1 and 8) and

c.order_status not in ('canceled','unavailable') and a.product_category is not null

group by a.product_category)
```

Query Statement

```
select a.product_category , a.price as cost_of_orders_2017 , b.price as
cost_of_orders_2018 , round((((b.price-a.price)/a.price)*100),2) as percent_increase

from `ecommerce.myview` a, `ecommerce.myview1` b

where a.product_category = b.product_category
```



product_category	cost_of_orders_2017	cost_of_orders_2018	percent_increase
Art	8364	14876	77.86
HEALTH BEAUTY	247458	770003	211.17
housewares	131133	397688	203.27
pet Shop	49750	128053	157.39
Drink foods	1270	7697	506.06
telephony	60745	179752	195.91
computer accessories	219639	502432	128.75
sport leisure	224597	530182	136.06
Games consoles	46956	63776	35.82
automotive	128278	346632	170.22
electronics	29614	101000	241.05
babies	70329	256087	264.13
drinks	1198	17425	1354.51
Market Place	15191	7566	-50.19
bed table bath	259621	537514	107.04
home appliances	14233	56150	294.51
Fashion Bags and Accessories	38138	70182	84.02
Garden tools	143768	214518	49.21
House comfort	21863	20883	-4.48
Furniture Decoration	180301	385312	113.7
Cool Stuff	211948	230752	8.87
fixed telephony	10726	27005	151.77
toys	119703	169711	41.78
audio	7815	32961	321.77
perfumery	123490	176945	43.29
General Interest Books	11854	26966	127.48
stationary store	37277	135469	263.41
Agro Industria e Comercio	4611	43351	840.16
La Cuisine	1000	781	-21.9
electrostile	66547	87851	32.01
musical instruments	37431	107746	187.85
Watches present	206284	708306	243.36
foods	5404	20426	277.98
technical books	2441	13067	435.31
Furniture Kitchen Service Area Dinner and Garden	5399	27349	406.56

Insights: As we see from data, we can see major increase in drinks , Agro industries and decline in Market place ,La Cuisine , House Comfort product categories

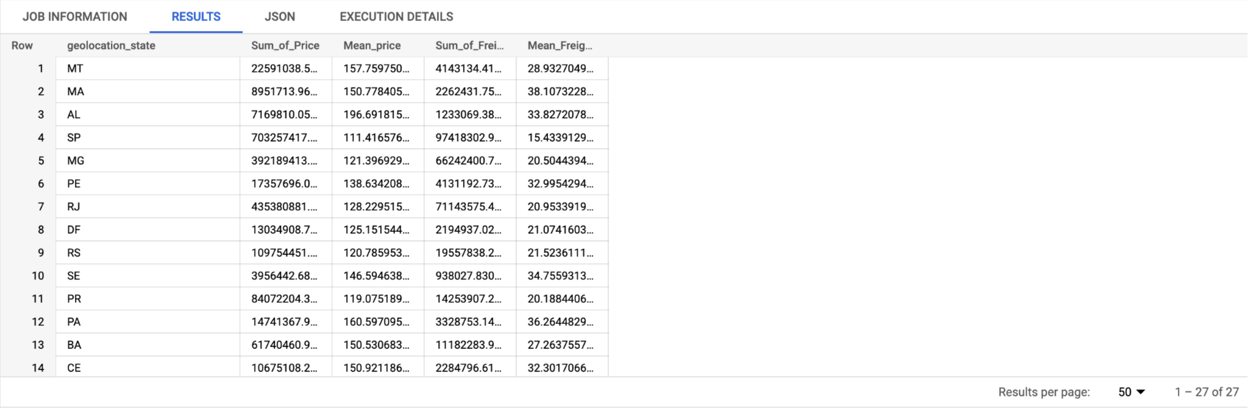
2. Mean & Sum of price and freight value by customer state

MEAN, SUM OF PRICE AND FREIGHT BY STATE WISE:

QUERY:

```
select e.geolocation_state,  sum(b.price)  as Sum_of_Price , Avg(b.price) as Mean_price ,
sum(b.freight_value) as Sum_of_Freight_value , avg(b.freight_value) as Mean_Freight_value  from

`ecommerce.products` a , `ecommerce.order_items` b ,`ecommerce.orders` c ,
`ecommerce.customers` d , `ecommerce.geolocation` e
```

```
where a.product_id = b.product_id and b.order_id = c.order_id and a.product_category is not null and
d.customer_id = c.customer_id

and e.geolocation_zip_code_prefix = d.customer_zip_code_prefix

group by e.geolocation_state
```



Row	geolocation_state	Sum_of_Price	Mean_price	Sum_of_Frei...	Mean_Freig...
1	MT	22591038.5...	157.759750...	4143134.41...	28.9327049...
2	MA	8951713.96...	150.778405...	2262431.75...	38.1073228...
3	AL	7169810.05...	196.691815...	1233069.38...	33.8272078...
4	SP	703257417....	111.416576...	97418302.9...	15.4339129...
5	MG	392189413....	121.396929...	66242400.7...	20.5044394...
6	PE	17357696.0...	138.634208...	4131192.73...	32.9954294...
7	RJ	435380881....	128.229515...	71143575.4...	20.9533919...
8	DF	13034908.7...	125.151544...	2194937.02...	21.0741603...
9	RS	109754451....	120.785953...	19557838.2...	21.5236111...
10	SE	3956442.68...	146.594638...	938027.830...	34.7559313...
11	PR	84072204.3...	119.075189...	14253907.2...	20.1884406...
12	PA	14741367.9...	160.597095...	3328753.14...	36.2644829...
13	BA	61740460.9...	150.530683...	11182283.9...	27.2637557...
14	CE	10675108.2...	150.921186...	2284796.61...	32.3017066...

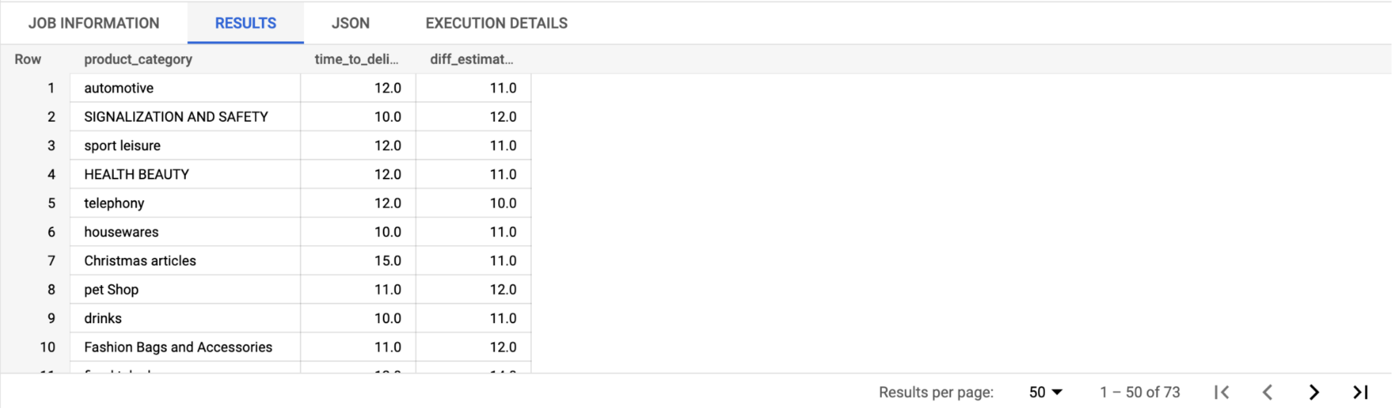
Insights: We can see, Mean price and Freight value of orders placed in various stated from the above data. And AL state has highest mean price and RR State has highest Freight mean price

5. Analysis on sales, freight and delivery time

    1. Calculate days between purchasing, delivering and estimated delivery

Days between Purchasing, Delivering and Estimated delivery

```
select b.product_category, ROUND(

avg(DATE_DIFF(a.order_delivered_customer_date,a.order_purchase_timestamp , DAY)),0)
as time_to_delivery,

ROUND(avg(DATE_DIFF(a.order_estimated_delivery_date,a.order_delivered_customer_dat
e , DAY)),0) as diff_estimated_delivery

from `ecommerce.orders` a , `ecommerce.products` b , `ecommerce.order_items` c

where order_status = 'delivered' and c.product_id = b.product_id and a.order_id = c.order_id
and b.product_category is not null

group by b.product_category
```



Row	product_category	time_to_deli...	diff_estimat...
1	automotive	12.0	11.0
2	SIGNALIZATION AND SAFETY	10.0	12.0
3	sport leisure	12.0	11.0
4	HEALTH BEAUTY	12.0	11.0
5	telephony	12.0	10.0
6	housewares	10.0	11.0
7	Christmas articles	15.0	11.0
8	pet Shop	11.0	12.0
9	drinks	10.0	11.0
10	Fashion Bags and Accessories	11.0	12.0

Results per page: 50 1 – 50 of 73

Insights: We can clearly see Musical instruments, Christmas articles took more days for delivery, and La Cuisine product category delivered earlier than estimated time.

2. Create columns:

    1. time_to_delivery = order_purchase_timestamp-order_delivered_customer_date

    2. diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

—- Added Columns and displayed the screenshot above

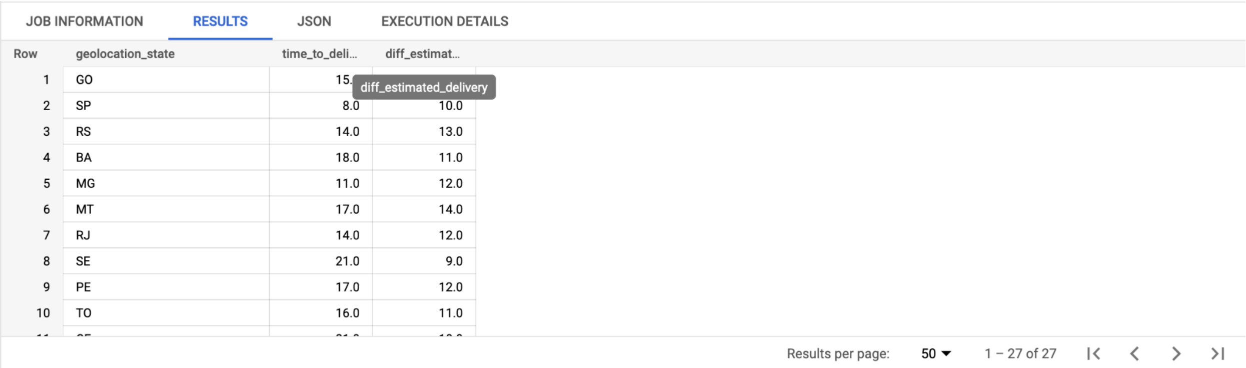
3. Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

STATE WISE DELIVERY TIME AND FREIGHT VALUE

Query

```
select e.geolocation_state, ROUND(

avg(DATE_DIFF(a.order_delivered_customer_date,a.order_purchase_timestamp , DAY)),0)

as time_to_delivery,

ROUND(avg(DATE_DIFF(a.order_estimated_delivery_date,a.order_delivered_customer_dat

e , DAY)),0) as diff_estimated_delivery

from `ecommerce.orders` a , `ecommerce.products` b , `ecommerce.order_items` c ,

`ecommerce.customers` d , `ecommerce.geolocation` e
```

```
where order_status = 'delivered' and e.geolocation_zip_code_prefix =
d.customer_zip_code_prefix and a.customer_id=d.customer_id and c.product_id =
b.product_id and a.order_id = c.order_id and b.product_category is not null

group by e.geolocation_state
```



Row	geolocation_state	time_to_deli...	diff_estimat...
1	GO	15.	
2	SP	8.0	10.0
3	RS	14.0	13.0
4	BA	18.0	11.0
5	MG	11.0	12.0
6	MT	17.0	14.0
7	RJ	14.0	12.0
8	SE	21.0	9.0
9	PE	17.0	12.0
10	TO	16.0	11.0

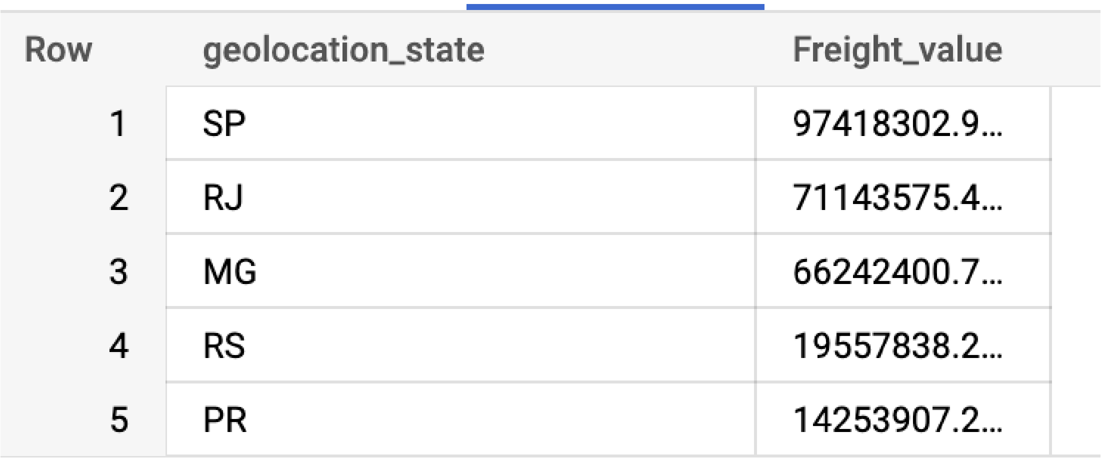
Results per page: 50 ▾ 1 – 27 of 27

4. Sort the data to get the following:

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

      ### TOP 5 HIGHEST AVERAGE

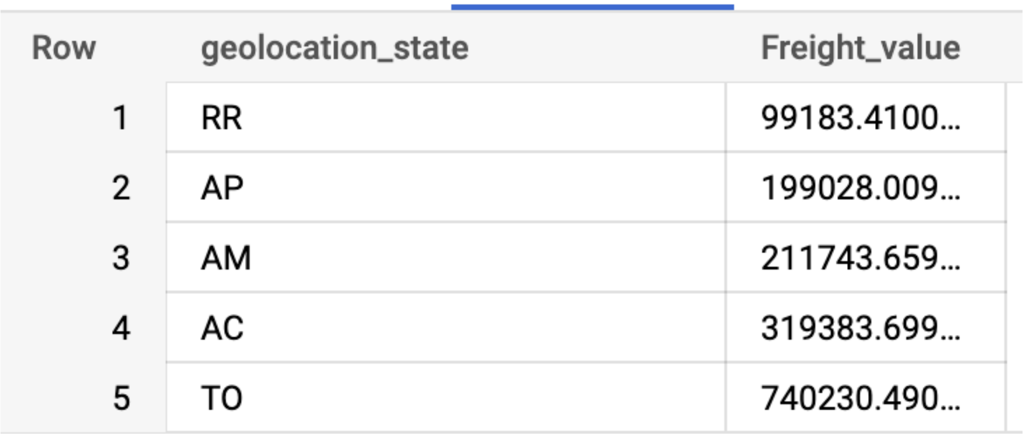
```
select geolocation_state , Freight_value from (select e.geolocation_state,
sum(b.price)  as Price , Avg(b.price) as Mean_price , sum(b.freight_value) as
Freight_value  from

`ecommerce.products` a ,`ecommerce.order_items` b ,`ecommerce.orders` c ,
`ecommerce.customers` d ,`ecommerce.geolocation` e

where a.product_id = b.product_id and b.order_id = c.order_id and
a.product_category is not null and d.customer_id = c.customer_id

and e.geolocation_zip_code_prefix = d.customer_zip_code_prefix

group by e.geolocation_state)

order by Freight_value desc limit 5
```



Row	geolocation_state	Freight_value
1	SP	97418302.9...
2	RJ	71143575.4...
3	MG	66242400.7...
4	RS	19557838.2...
5	PR	14253907.2...

TOP 5 LOWEST AVERAGE

```
select geolocation_state , Freight_value from (select e.geolocation_state,
sum(b.price)  as Price , Avg(b.price) as Mean_price , sum(b.freight_value) as
Freight_value  from

`ecommerce.products` a , `ecommerce.order_items` b ,`ecommerce.orders` c ,
`ecommerce.customers` d , `ecommerce.geolocation` e

where a.product_id = b.product_id and b.order_id = c.order_id and
a.product_category is not null and d.customer_id = c.customer_id

and e.geolocation_zip_code_prefix = d.customer_zip_code_prefix

group by e.geolocation_state)

order by Freight_value asc limit 5
```

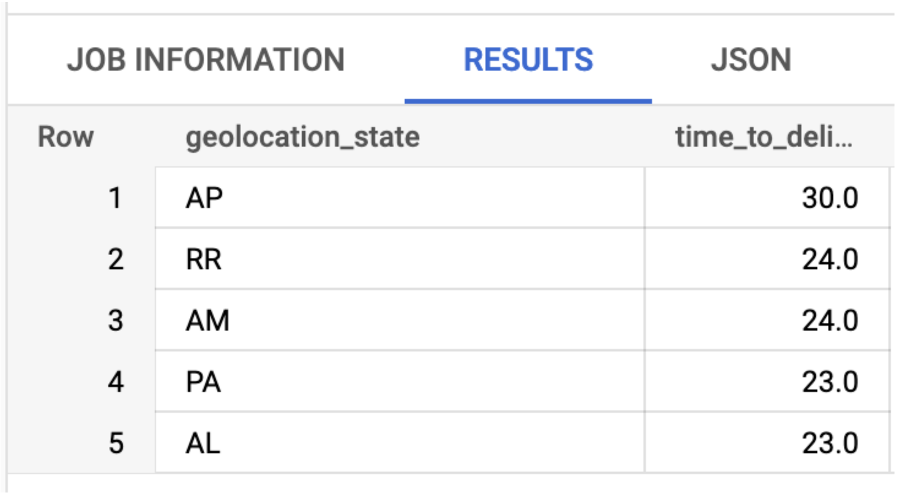


Row	geolocation_state	Freight_value
1	RR	99183.4100...
2	AP	199028.009...
3	AM	211743.659...
4	AC	319383.699...
5	TO	740230.490...

2. Top 5 states with highest/lowest average time to delivery

TOP 5 HIGHEST AVERAGE DELIVERY TIME:

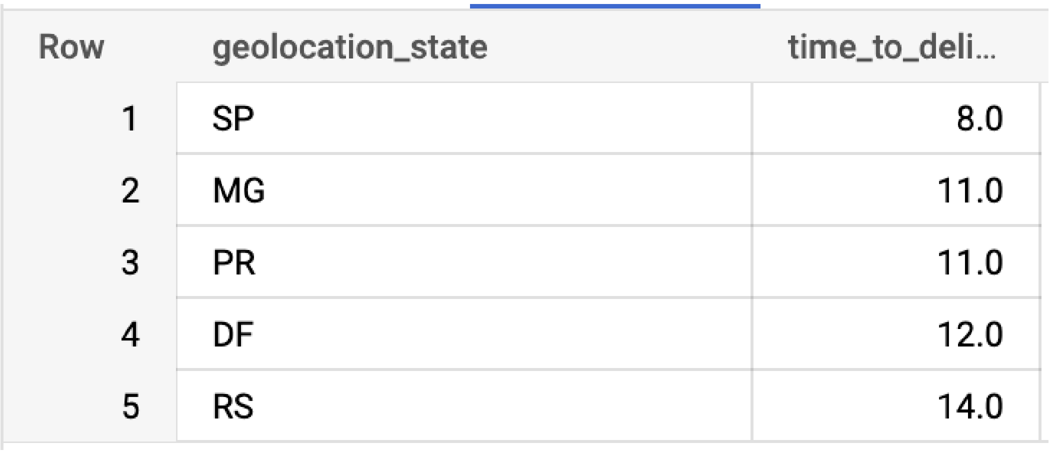
```
select geolocation_state , time_to_delivery from(select e.geolocation_state,
ROUND(

avg(DATE_DIFF(a.order_delivered_customer_date,a.order_purchase_timestamp ,
DAY)),0) as time_to_delivery,

ROUND(avg(DATE_DIFF(a.order_estimated_delivery_date,a.order_delivered_customer_date , DAY)),0) as diff_estimated_delivery

from `ecommerce.orders` a , `ecommerce.products` b , `ecommerce.order_items` c
, `ecommerce.customers` d , `ecommerce.geolocation` e

where order_status = 'delivered' and e.geolocation_zip_code_prefix =
d.customer_zip_code_prefix and a.customer_id=d.customer_id and c.product_id =
b.product_id and a.order_id = c.order_id and b.product_category is not null

group by e.geolocation_state)

order by time_to_delivery desc limit 5
```



Row	geolocation_state	time_to_deli...
1	AP	30.0
2	RR	24.0
3	AM	24.0
4	PA	23.0
5	AL	23.0

TOP 5 LOWEST AVERAGE DELIVERY TIME:

```
select geolocation_state , time_to_delivery from(select e.geolocation_state,
ROUND(

avg(DATE_DIFF(a.order_delivered_customer_date,a.order_purchase_timestamp ,
DAY)),0) as time_to_delivery,

ROUND(avg(DATE_DIFF(a.order_estimated_delivery_date,a.order_delivered_customer_date , DAY)),0) as diff_estimated_delivery

from `ecommerce.orders` a , `ecommerce.products` b , `ecommerce.order_items` c
, `ecommerce.customers` d , `ecommerce.geolocation` e

where order_status = 'delivered' and e.geolocation_zip_code_prefix =
d.customer_zip_code_prefix and a.customer_id=d.customer_id and c.product_id =
b.product_id and a.order_id = c.order_id and b.product_category is not null

group by e.geolocation_state)

order by time_to_delivery asc limit 5
```

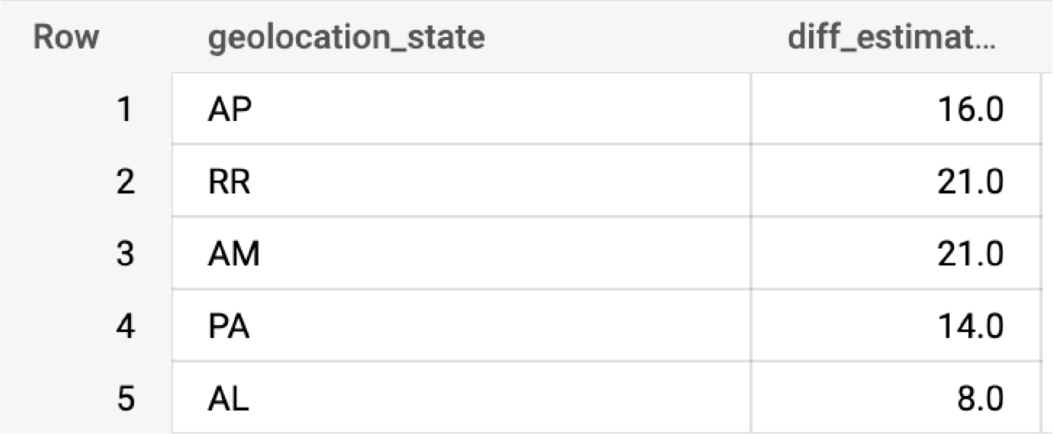


Row	geolocation_state	time_to_deli...
1	SP	8.0
2	MG	11.0
3	PR	11.0
4	DF	12.0
5	RS	14.0

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

TOP 5 STATES WITH FASTEST DELIVERY:

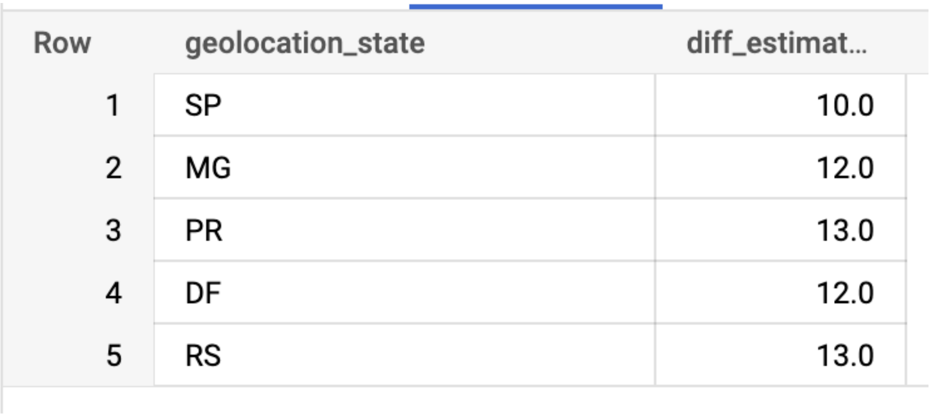
```
select geolocation_state , diff_estimated_delivery from(select e.geolocation_state,
ROUND(

avg(DATE_DIFF(a.order_delivered_customer_date,a.order_purchase_timestamp ,
DAY)),0) as time_to_delivery,

ROUND(avg(DATE_DIFF(a.order_estimated_delivery_date,a.order_delivered_custo
mer_date , DAY)),0) as diff_estimated_delivery

from `ecommerce.orders` a , `ecommerce.products` b , `ecommerce.order_items` c
, `ecommerce.customers` d , `ecommerce.geolocation` e

where order_status = 'delivered' and e.geolocation_zip_code_prefix =
d.customer_zip_code_prefix and a.customer_id=d.customer_id and c.product_id =
b.product_id and a.order_id = c.order_id and b.product_category is not null

group by e.geolocation_state)

order by time_to_delivery desc limit 5
```



Row	geolocation_state	diff_estimat...
1	AP	16.0
2	RR	21.0
3	AM	21.0
4	PA	14.0
5	AL	8.0

TOP 5 STATES WITH SLOWEST DELIVERY:

```
select geolocation_state , diff_estimated_delivery from(select e.geolocation_state,
ROUND(

avg(DATE_DIFF(a.order_delivered_customer_date,a.order_purchase_timestamp ,
DAY)),0) as time_to_delivery,

ROUND(avg(DATE_DIFF(a.order_estimated_delivery_date,a.order_delivered_custo
mer_date , DAY)),0) as diff_estimated_delivery

from `ecommerce.orders` a , `ecommerce.products` b , `ecommerce.order_items` c
, `ecommerce.customers` d , `ecommerce.geolocation` e

where order_status = 'delivered' and e.geolocation_zip_code_prefix =
d.customer_zip_code_prefix and a.customer_id=d.customer_id and c.product_id =
b.product_id and a.order_id = c.order_id and b.product_category is not null

group by e.geolocation_state)

order by time_to_delivery asc limit 5
```



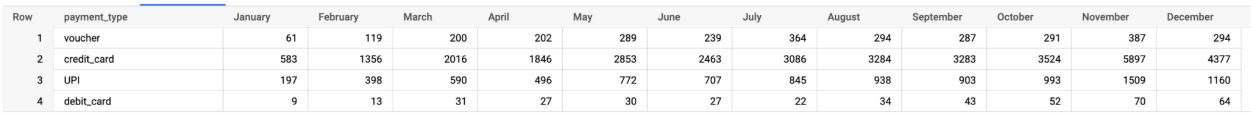
Row	geolocation_state	diff_estimat...
1	SP	10.0
2	MG	12.0
3	PR	13.0
4	DF	12.0
5	RS	13.0

6. Payment type analysis:

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

MONTH ON MONTH PAYMENT TYPES COUNT

```
select payment_type, IFNULL(_1, 0 ) as January , IFNULL(_2, 0 ) as February , IFNULL(_3,
0 ) as March, IFNULL(_4, 0 ) as April, IFNULL(_5, 0 ) as May ,

IFNULL(_6, 0 ) as June , IFNULL(_7, 0 ) as July , IFNULL(_8, 0 ) as August , IFNULL(_9, 0 )
as September , IFNULL(_10, 0 ) as October , IFNULL(_11, 0 ) as November , IFNULL(_12, 0 )
as December from

(select * from (SELECT *

FROM (SELECT a.payment_type,

extract (month from b.order_purchase_timestamp) as Month,

COUNT(1) AS order_count

  FROM `ecommerce.payments` a, `ecommerce.orders` b

  where a.order_id= b.order_id and a.payment_type <> 'not_defined' and extract (year from
b.order_purchase_timestamp)= 2017

  GROUP BY a.payment_type,

  extract (month from b.order_purchase_timestamp))  AS MonthlyOrderData

PIVOT( SUM(order_count)

 FOR Month IN (1,2,3,4,5,6,7,8,9,10,11,12)) AS monthlypivot))
```



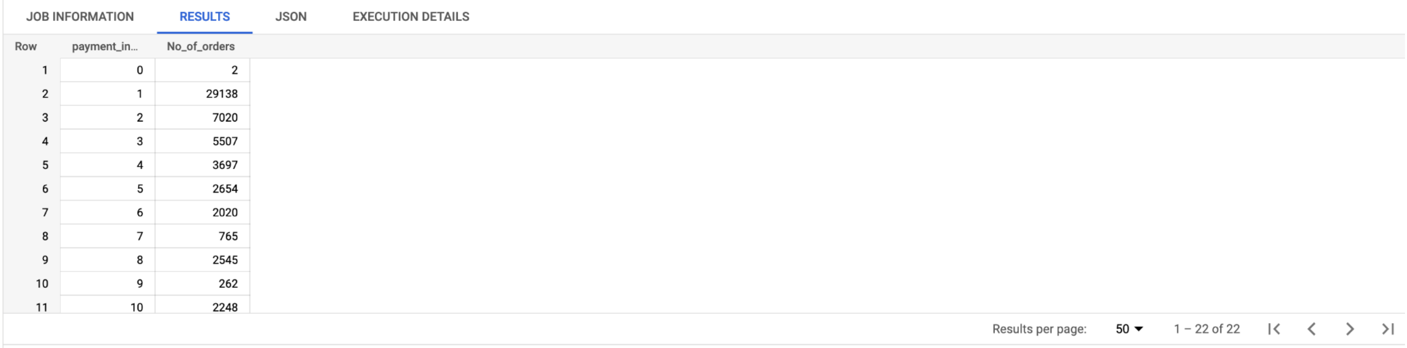
Row	payment_type	January	February	March	April	May	June	July	August	September	October	November	December
1	voucher	61	119	200	202	289	239	364	294	287	291	387	294
2	credit_card	583	1356	2016	1846	2853	2463	3086	3284	3283	3524	5897	4377
3	UPI	197	398	590	496	772	707	845	938	903	993	1509	1160
4	debit_card	9	13	31	27	30	27	22	34	43	52	70	64

Insights: As we can see from the data, More Credit payments are made in 2017 and more orders are placed in november.

2. Distribution of payment installments and count of orders

Payment installment and order count:

```
select a.payment_installments as payment_installments,count(a.order_id) as No_of_orders

from `ecommerce.payments` a, `ecommerce.orders` b

where a.order_id = b.order_id and extract(year from b.order_purchase_timestamp)=2018

group by payment_installments

order by a.payment_installments
```



Row	payment_in...	No_of_orders
1	0	2
2	1	29138
3	2	7020
4	3	5507
5	4	3697
6	5	2654
7	6	2020
8	7	765
9	8	2545
10	9	262
11	10	2248

Insights: We can see , 1 month payment installments is preferred by most customers and full payment for orders are not preferred by customers

Actionable insights

- We can see November month and december has more orders placed
- Also , Credit card payment method is most preferable by customers
- Full payment paid for the orders placed is not preferred by customers
- SP state has more orders and AP state has less orders recorded
- Some of the orders placed might get cancelled, not available
- There were major increase in drinks, Agro industries in sales order
- And Decline in Market place, La Cuisine , House comfort product categories

- We also see , Musical instruments , christmas articles took more data for delivery
- And La Cuisine product categories delivered earlier than estimated time
- We can see there are more number of orders placed in evening
- Product Categories like health beauty, Computer accessories , bed table bath has more orders in evening

Recommendations:

- More credit card offers can be given , to attract more customer orders
- Can concentrate on states like AP , to increase sales order
- Can make customers have short term goals savings plan to purchase high price goods and can implement this in Application/ Website. This can decrease the payment installments and motivate for savings
- Attractive offers can be given on October ,november, December months to further attract more orders
- Market place , La Cuisine product category offers and attractive products can be released to increase the sales
- La Cuisine product category -early delivery (as it is delivered earlier than estimated already) offers can be shown on websites to attract more sales