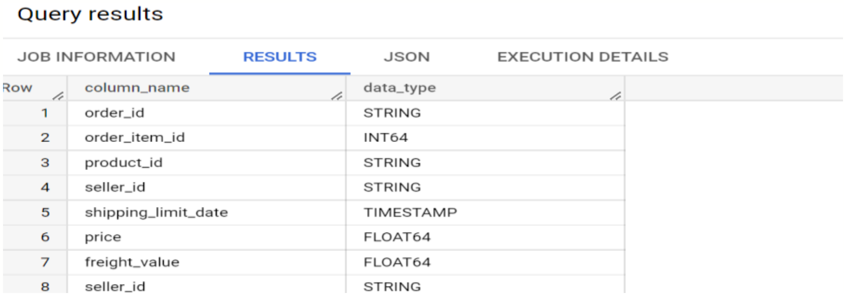
Target Case Study

1.1 Data type of columns in a table?

Answer:

```
Select column_name, data_type from `target-practice-
project.Target.INFORMATION_SCHEMA.COLUMNS`

WHERE table_name in ('customers','geolocation', 'order_items', 'order_reviews', 'payments',
'orders','products','sellers')
```

Output: Below is the given output for the above query, it gives the data type of all 48 columns in a data set.



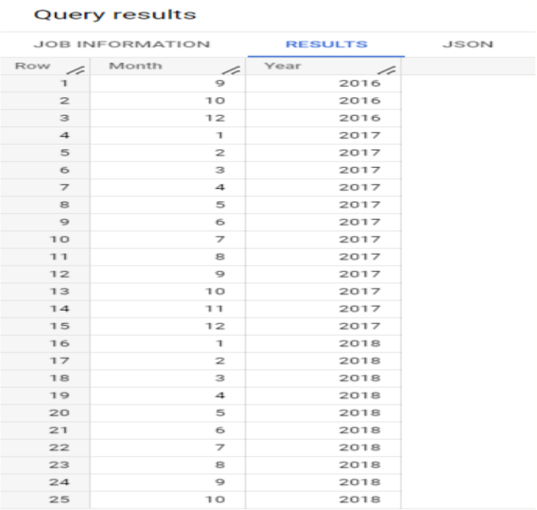
Row	column_name	data_type
1	order_id	STRING
2	order_item_id	INT64
3	product_id	STRING
4	seller_id	STRING
5	shipping_limit_date	TIMESTAMP
6	price	FLOAT64
7	freight_value	FLOAT64
8	seller_id	STRING

1.2 Time period for which data is given?

Answer:

```
select distinct extract(Month from order_purchase_timestamp) as Month, extract(Year from or
der_purchase_timestamp)as Year FROM `target-practice-
project.Target.orders` order by Year, Month;
```

Output: The data given is from 9th month (September) 2016 till 10th month (October) 2018



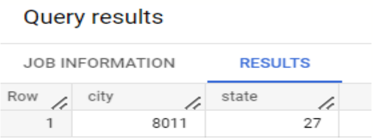
Row	Month	Year
1	9	2016
2	10	2016
3	12	2016
4	1	2017
5	2	2017
6	3	2017
7	4	2017
8	5	2017
9	6	2017
10	7	2017
11	8	2017
12	9	2017
13	10	2017
14	11	2017
15	12	2017
16	1	2018
17	2	2018
18	3	2018
19	4	2018
20	5	2018
21	6	2018
22	7	2018
23	8	2018
24	9	2018
25	10	2018

1.3 Cities and States covered in the dataset?

Answer:

```
select count(distinct geolocation_city) as city, count(distinct geolocation_state) as state
 FROM `target-practice-project.Target.geolocation`;
```

Output: There are 8011 different cities and 27 different states covered in the dataset



Query results

Row	city	state
1	8011	27

In-depth exploration

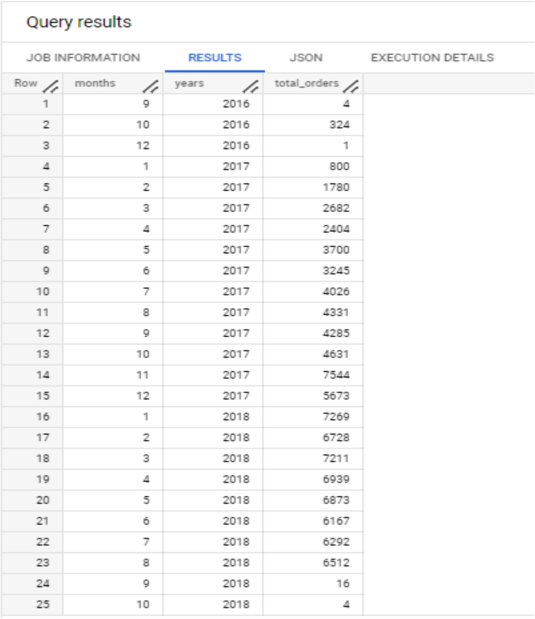
2.1 Is there a growing trend on e-commerce in Brazil? How can we describe a complete scenario? Can we see some seasonality with peaks at specific months?

Answer:

```
select
extract(month from order_purchase_timestamp) as months,
extract(year from order_purchase_timestamp) as years,
count(order_id) as total_orders from `target-practice-project.Target.orders`
group by years, months
order by years, months;
```

Output:

1. Yes, there is an upward trend in the ecommerce sales on the basis of year-on-year orders of data
2. There is no seasonality on the peaks visible as in year 2018 in 3rd month there are highest number of orders, there are very limited orders in year 2017 in 3rd month



Query results

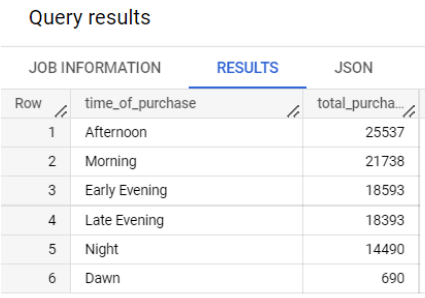
Row	months	years	total_orders
1	9	2016	4
2	10	2016	324
3	12	2016	1
4	1	2017	800
5	2	2017	1780
6	3	2017	2682
7	4	2017	2404
8	5	2017	3700
9	6	2017	3245
10	7	2017	4026
11	8	2017	4331
12	9	2017	4285
13	10	2017	4631
14	11	2017	7544
15	12	2017	5673
16	1	2018	7269
17	2	2018	6728
18	3	2018	7211
19	4	2018	6939
20	5	2018	6873
21	6	2018	6167
22	7	2018	6292
23	8	2018	6512
24	9	2018	16
25	10	2018	4

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

Answer:

```
select
case
when extract(time from order_purchase_timestamp) between '05:00:00' and '7:00:00'
then 'Dawn'
when extract(time from order_purchase_timestamp) between '07:00:01' and '11:59:59'
then 'Morning'
when extract(time from order_purchase_timestamp) between '12:00:00' and '16:00:00'
then 'Afternoon'
when extract(time from order_purchase_timestamp) between '16:00:01' and '19:00:00'
then 'Early Evening'
when extract(time from order_purchase_timestamp) between '19:00:01' and '22:00:00'
then 'Late Evening'
Else 'Night'
END AS time_of_purchase, count(order_id)as total_purchase
from `target-practice-project.Target.orders`
group by time_of_purchase
order by total_purchase desc;
```

Output: As per the given data set Brazilians customers prefers to buy during After noon, i.e.; 12:00:01 PM to 16:00:00 PM, and least purchase is made is during dawn i.e.; 5:00:00 AM to 7:00:00 AM



Row	time_of_purchase	total_purcha...
1	Afternoon	25537
2	Morning	21738
3	Early Evening	18593
4	Late Evening	18393
5	Night	14490
6	Dawn	690

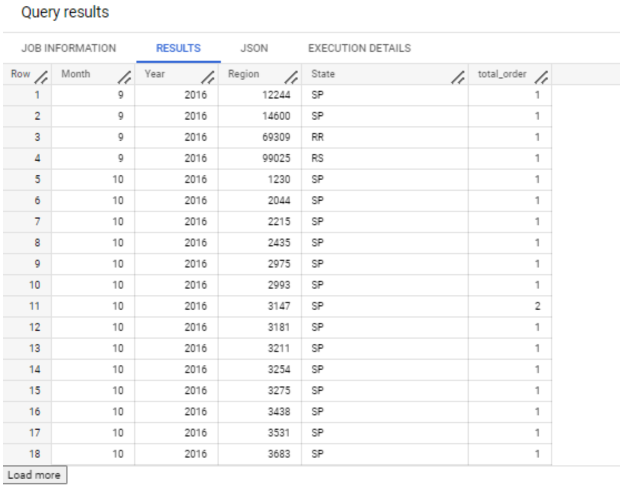
Evolution of E-commerce orders in the Brazilian Region

3.1 Get month on month orders by region, states?

Answer:

```
select extract(month from o.order_purchase_timestamp) as Month, extract(year from o.order_p
urchase_timestamp) as Year,c.customer_zip_code_prefix as Region,c.customer_state as State,
count(o.order_id) as total_order
from `target-practice-project.Target.orders` as o join
`target-practice-project.Target.customers` as c on
o.customer_id = c.customer_id
group by Month, Year, Region, State
order by Year,Month,Region, State;
```

Output:



Query results

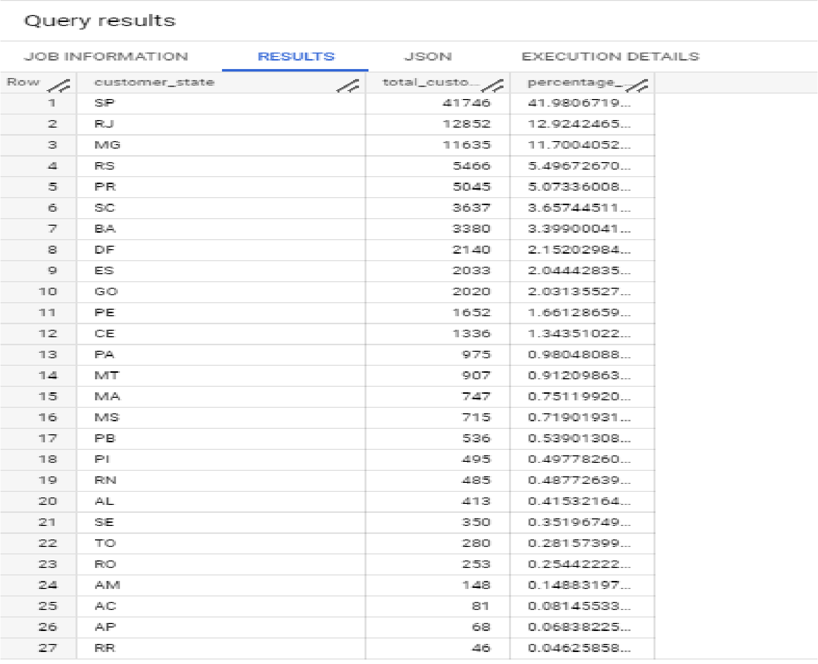
Row	Month	Year	Region	State	total_order
1	9	2016	12244	SP	1
2	9	2016	14600	SP	1
3	9	2016	69309	RR	1
4	9	2016	99025	RS	1
5	10	2016	1230	SP	1
6	10	2016	2044	SP	1
7	10	2016	2215	SP	1
8	10	2016	2435	SP	1
9	10	2016	2975	SP	1
10	10	2016	2993	SP	1
11	10	2016	3147	SP	2
12	10	2016	3181	SP	1
13	10	2016	3211	SP	1
14	10	2016	3254	SP	1
15	10	2016	3275	SP	1
16	10	2016	3438	SP	1
17	10	2016	3531	SP	1
18	10	2016	3683	SP	1

3.2 How are customers distributed in Brazil?

Answer:

```
with total as (
select customer_state, count(customer_id) as total_customers
from`target-practice-project.Target.customers`
group by customer_state
)
select customer_state, total_customers,
total_customers *100.0/(select sum(total_customers) from total) as percentage_of_customers
from total
order by percentage_of_customers DESC;
```

Output: The top 3 states with highest number of customers are: 1. SP (total customers: 41,746 i.e.; 41.98%) 2.RU (total customers: 12,853 i.e.; 12.92%) 3.MG (total customers: 11,635 i.e.; 11.70%)



Query results

Row	customer_state	total_custo...	percentage_...
1	SP	41746	41.9806719...
2	RJ	12852	12.9242465...
3	MG	11635	11.7004052...
4	RS	5466	5.49672670...
5	PR	5045	5.07336008...
6	SC	3637	3.65744511...
7	BA	3380	3.39900041...
8	DF	2140	2.15202984...
9	ES	2033	2.04442835...
10	GO	2020	2.03135527...
11	PE	1652	1.66128659...
12	CE	1336	1.34351022...
13	PA	975	0.98048088...
14	MT	907	0.91209863...
15	MA	747	0.75119920...
16	MS	715	0.71901931...
17	PB	536	0.53901308...
18	PI	495	0.49778260...
19	RN	485	0.48772639...
20	AL	413	0.41532164...
21	SE	350	0.35196749...
22	TO	280	0.28157399...
23	RO	253	0.25442222...
24	AM	148	0.14883197...
25	AC	81	0.08145533...
26	AP	68	0.06838225...
27	RR	46	0.04625858...

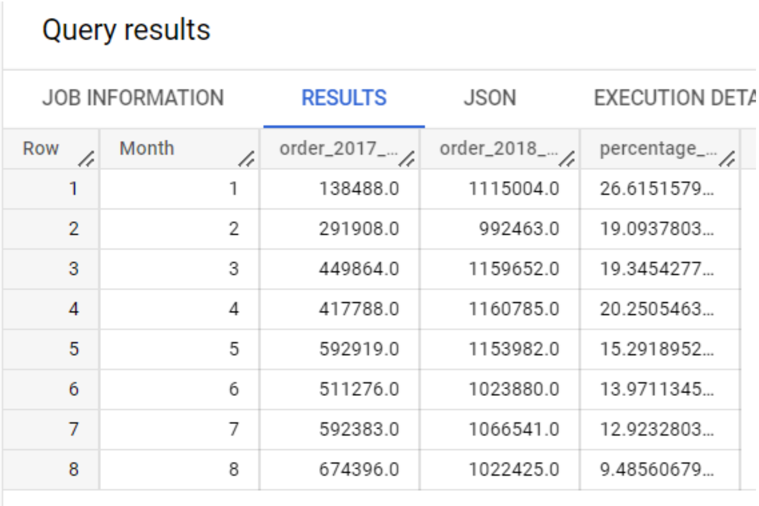
4. Impact on Economy: Analyse the money movement by e-commerce by looking at order prices, freight and others.

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

Answer:

```
with year_2017 as (
select extract(month from o.order_purchase_timestamp) as Month, extract(year from o.order_p
urchase_timestamp) as Year, round(sum(p.payment_value),0) as order_2017_cost
from `target-practice-project.Target.orders`as o join
`target-practice-project.Target.payments` as p on
o.order_id = p.order_id
where extract(year from o.order_purchase_timestamp) = 2017 and extract(month from o.order_p
urchase_timestamp) between 1 and 8
group by Year, Month
order by Year, Month),

year_2018 as(
select extract(month from o.order_purchase_timestamp) as Month, extract(year from o.order_p
urchase_timestamp) as Year, round(sum(p.payment_value),0) as order_2018_cost
from `target-practice-project.Target.orders`as o join
`target-practice-project.Target.payments` as p on
o.order_id = p.order_id
where extract(year from o.order_purchase_timestamp) = 2018 and extract(month from o.order_p
urchase_timestamp) between 1 and 8
group by Year, Month
order by Year, Month
)

select Y17.Month, Y17.order_2017_cost, Y18.order_2018_cost,
((Y18.order_2018_cost - Y17.order_2017_cost)/(select sum(order_2017_cost) from year_2017))*
100 as percentage_change
from year_2017 as Y17
join year_2018 as Y18 on Y17.Month = Y18.month
order by Month;
```

Output:



Query results

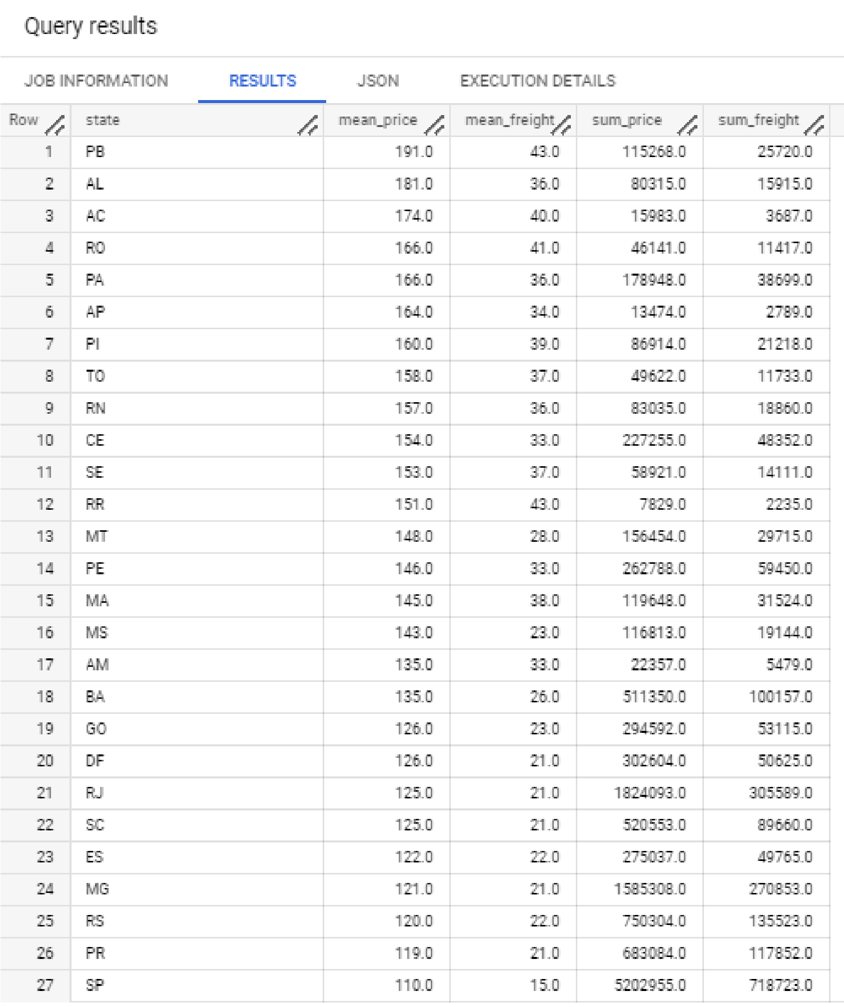
Row	Month	order_2017_...	order_2018_...	percentage_...
1	1	138488.0	1115004.0	26.6151579...
2	2	291908.0	992463.0	19.0937803...
3	3	449864.0	1159652.0	19.3454277...
4	4	417788.0	1160785.0	20.2505463...
5	5	592919.0	1153982.0	15.2918952...
6	6	511276.0	1023880.0	13.9711345...
7	7	592383.0	1066541.0	12.9232803...
8	8	674396.0	1022425.0	9.48560679...

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

Answer:

```
with test as (select o.order_id as order_id, c.customer_state as state from `target-
practice-project.Target.orders` as o
join `target-practice-project.Target.customers` as c on
o.customer_id = c.customer_id)
select t.state as state, round(avg(o.price),0) as mean_price,round(avg(o.freight_value),0)a
s mean_freight,
round(sum(o.price),0) as sum_price, round(sum(o.freight_value),0) as sum_freight from
`target-practice-project.Target.order_items`as o
join test as t on o.order_id =t.order_id
group by state
order by mean_price DESC, mean_freight DESC,sum_price DESC,sum_freIght DESC ;
```

Output: PB is the state with the highest mean and freight value.



Row	state	mean_price	mean_freight	sum_price	sum_freight
1	PB	191.0	43.0	115268.0	25720.0
2	AL	181.0	36.0	80315.0	15915.0
3	AC	174.0	40.0	15983.0	3687.0
4	RO	166.0	41.0	46141.0	11417.0
5	PA	166.0	36.0	178948.0	38699.0
6	AP	164.0	34.0	13474.0	2789.0
7	PI	160.0	39.0	86914.0	21218.0
8	TO	158.0	37.0	49622.0	11733.0
9	RN	157.0	36.0	83035.0	18860.0
10	CE	154.0	33.0	227255.0	48352.0
11	SE	153.0	37.0	58921.0	14111.0
12	RR	151.0	43.0	7829.0	2235.0
13	MT	148.0	28.0	156454.0	29715.0
14	PE	146.0	33.0	262788.0	59450.0
15	MA	145.0	38.0	119648.0	31524.0
16	MS	143.0	23.0	116813.0	19144.0
17	AM	135.0	33.0	22357.0	5479.0
18	BA	135.0	26.0	511350.0	100157.0
19	GO	126.0	23.0	294592.0	53115.0
20	DF	126.0	21.0	302604.0	50625.0
21	RJ	125.0	21.0	1824093.0	305589.0
22	SC	125.0	21.0	520553.0	89660.0
23	ES	122.0	22.0	275037.0	49765.0
24	MG	121.0	21.0	1585308.0	270853.0
25	RS	120.0	22.0	750304.0	135523.0
26	PR	119.0	21.0	683084.0	117852.0
27	SP	110.0	15.0	5202955.0	718723.0

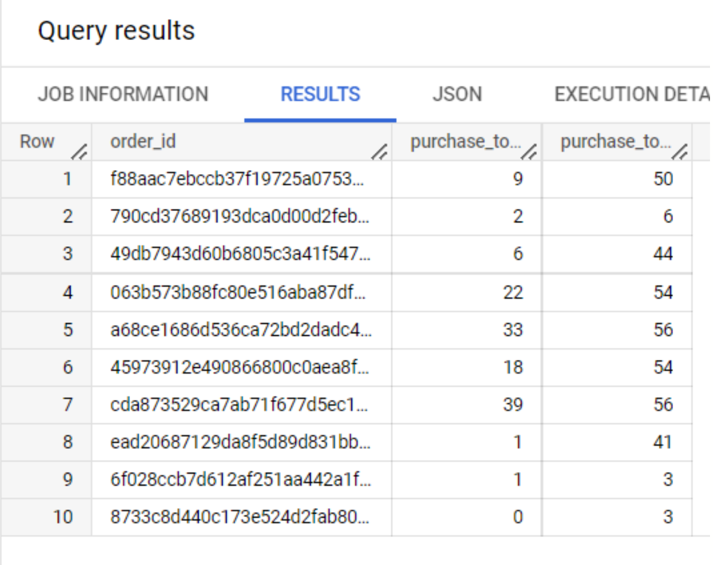
5.Analysis on sales, freight and delivery time

5.1 Calculate days between purchasing, delivering and estimated delivery?

Answer:

```
select order_id, datetime_diff(order_delivered_carrier_date,order_purchase_timestamp,day)
as purchase_to_carrier_delievery,
datetime_diff(order_estimated_delivery_date,order_purchase_timestamp,day) as purchase_to_or
der_estimated_delievery
from `target-practice-project.Target.orders`;
```

Output:



Query results

Row	order_id	purchase_to...	purchase_to...
1	f88aac7ebccb37f19725a0753...	9	50
2	790cd37689193dca0d00d2feb...	2	6
3	49db7943d60b6805c3a41f547...	6	44
4	063b573b88fc80e516aba87df...	22	54
5	a68ce1686d536ca72bd2dadc4...	33	56
6	45973912e490866800c0aea8f...	18	54
7	cda873529ca7ab71f677d5ec1...	39	56
8	ead20687129da8f5d89d831bb...	1	41
9	6f028ccb7d612af251aa442a1f...	1	3
10	8733c8d440c173e524d2fab80...	0	3

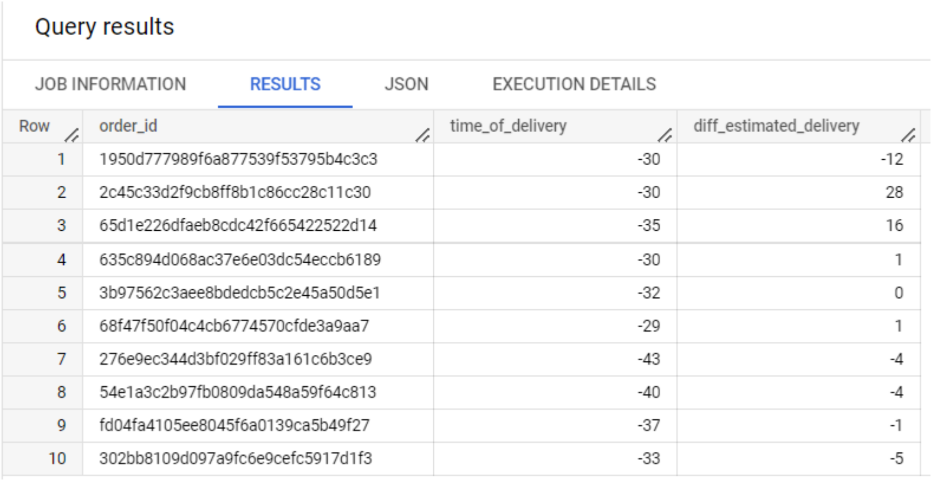
5.2 Create columns:

- ○ time_to_delivery = order_purchase_timestamp-order_delivered_customer_date
- ○ diff_estimated_delivery = order_estimated_delivery_date-order_delivered_customer_date

Answer:

```
select order_id, datetime_diff(order_purchase_timestamp,order_delivered_customer_date,day)
as time_of_delivery,
datetime_diff(order_estimated_delivery_date,order_delivered_customer_date,day) as diff_esti
mated_delivery
from `target-practice-project.Target.orders`;
```

Output:



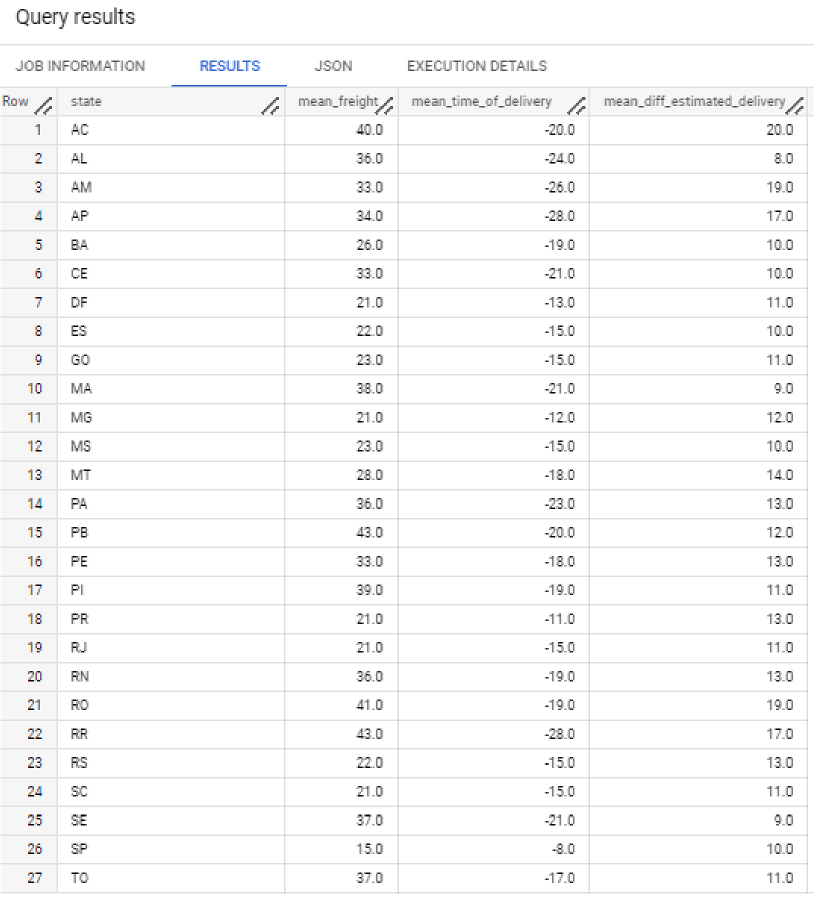
Row	order_id	time_of_delivery	diff_estimated_delivery
1	1950d777989f6a877539f53795b4c3c3	-30	-12
2	2c45c33d2f9cb8ff8b1c86cc28c11c30	-30	28
3	65d1e226dfaeb8cdc42f665422522d14	-35	16
4	635c894d068ac37e6e03dc54eccb6189	-30	1
5	3b97562c3aee8bdedcb5c2e45a50d5e1	-32	0
6	68f47f50f04c4cb6774570cfde3a9aa7	-29	1
7	276e9ec344d3bf029ff83a161c6b3ce9	-43	-4
8	54e1a3c2b97fb0809da548a59f64c813	-40	-4
9	fd04fa4105ee8045f6a0139ca5b49f27	-37	-1
10	302bb8109d097a9fc6e9cefc5917d1f3	-33	-5

5.3 Group data by state, take mean of freight value, time_to_delivery, diff_estimated_delivery?

Answer:

```
with test as (select o.order_id as order_id, c.customer_state as state from `target-
practice-project.Target.orders` as o
join `target-practice-project.Target.customers` as c on
o.customer_id = c.customer_id)

select t.state as state,round(avg(o.freight_value),0)as mean_freight,
ROUND(AVG(datetime_diff(o1.order_purchase_timestamp,o1.order_delivered_customer_date,day)),
0) as mean_time_of_delivery,
ROUND(AVG(datetime_diff(o1.order_estimated_delivery_date,o1.order_delivered_customer_date,d
ay)),0) as mean_diff_estimated_delivery

from `target-practice-project.Target.order_items`as o
join test as t on o.order_id =t.order_id

join `target-practice-project.Target.orders` as o1 on
t.order_id = o1.order_id
group by state

order by state;
```

Output:



Query results

JOB INFORMATION RESULTS JSON EXECUTION DETAILS

Row	state	mean_freight	mean_time_of_delivery	mean_diff_estimated_delivery
1	AC	40.0	-20.0	20.0
2	AL	36.0	-24.0	8.0
3	AM	33.0	-26.0	19.0
4	AP	34.0	-28.0	17.0
5	BA	26.0	-19.0	10.0
6	CE	33.0	-21.0	10.0
7	DF	21.0	-13.0	11.0
8	ES	22.0	-15.0	10.0
9	GO	23.0	-15.0	11.0
10	MA	38.0	-21.0	9.0
11	MG	21.0	-12.0	12.0
12	MS	23.0	-15.0	10.0
13	MT	28.0	-18.0	14.0
14	PA	36.0	-23.0	13.0
15	PB	43.0	-20.0	12.0
16	PE	33.0	-18.0	13.0
17	PI	39.0	-19.0	11.0
18	PR	21.0	-11.0	13.0
19	RJ	21.0	-15.0	11.0
20	RN	36.0	-19.0	13.0
21	RO	41.0	-19.0	19.0
22	RR	43.0	-28.0	17.0
23	RS	22.0	-15.0	13.0
24	SC	21.0	-15.0	11.0
25	SE	37.0	-21.0	9.0
26	SP	15.0	-8.0	10.0
27	TO	37.0	-17.0	11.0

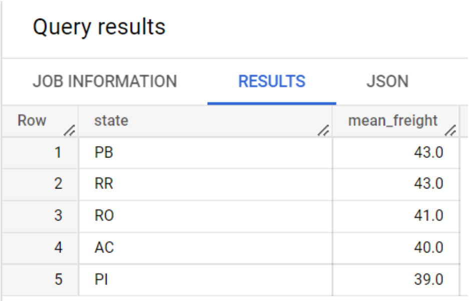
5.4 Sort the data to get the following

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

Answer:

```
with test as (select o.order_id as order_id, c.customer_state as state from `target-practice-project.Target.orders` as o
join `target-practice-project.Target.customers` as c on
o.customer_id = c.customer_id)
select t.state as state,round(avg(o.freight_value),0)as mean_freight,
from `target-practice-project.Target.order_items`as o
join test as t on o.order_id =t.order_id
group by state
order by mean_freight DESC
Limit 5;
```

Output: Top 5 States with highest average freight value



Query results

JOB INFORMATION RESULTS JSON

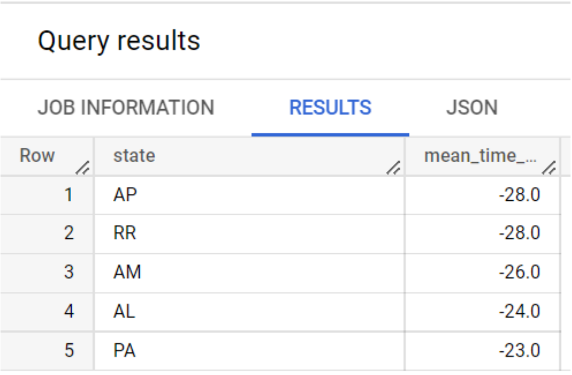
Row	state	mean_freight
1	PB	43.0
2	RR	43.0
3	RO	41.0
4	AC	40.0
5	PI	39.0

5.4.2 Top 5 states with highest/lowest average time to delivery?

Answer:

```
with test as (select o.order_id as order_id, c.customer_state as state from `target-
practice-project.Target.orders` as o
join `target-practice-project.Target.customers` as c on
o.customer_id = c.customer_id)

select t.state as state,
ROUND(AVG(datetime_diff(o1.order_purchase_timestamp,o1.order_delivered_customer_date,day)),
0) as mean_time_of_delivery

from `target-practice-project.Target.order_items`as o
join test as t on o.order_id =t.order_id

join `target-practice-project.Target.orders` as o1 on
t.order_id = o1.order_id
group by state

order by mean_time_of_delivery

limit 5;
```

Output:



Query results

Row	state	mean_time_...
1	AP	-28.0
2	RR	-28.0
3	AM	-26.0
4	AL	-24.0
5	PA	-23.0

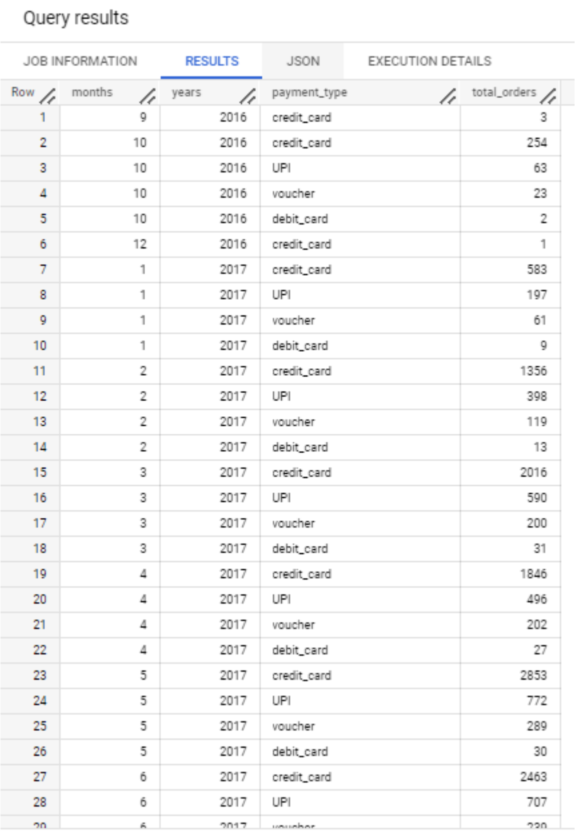
6. Payment type analysis

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

Answer:

```
select
extract(month from o.order_purchase_timestamp) as months,
extract(year from o.order_purchase_timestamp) as years,
p.payment_type as payment_type,
count(p.payment_type) as total_orders from `target-practice-project.Target.orders` as o
join `target-practice-project.Target.payments` as p
on o.order_id = p.order_id
group by years, months,payment_type
order by years, months,total_orders DESC;
```

Output:



Query results

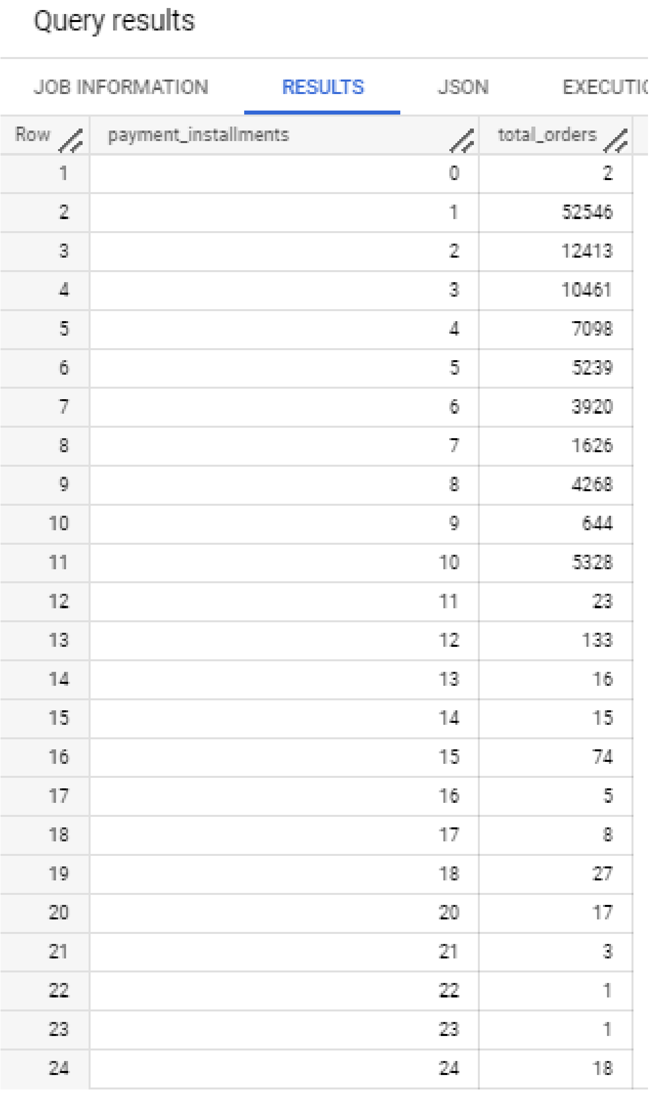
Row	months	years	payment_type	total_orders
1	9	2016	credit_card	3
2	10	2016	credit_card	254
3	10	2016	UPI	63
4	10	2016	voucher	23
5	10	2016	debit_card	2
6	12	2016	credit_card	1
7	1	2017	credit_card	583
8	1	2017	UPI	197
9	1	2017	voucher	61
10	1	2017	debit_card	9
11	2	2017	credit_card	1356
12	2	2017	UPI	398
13	2	2017	voucher	119
14	2	2017	debit_card	13
15	3	2017	credit_card	2016
16	3	2017	UPI	590
17	3	2017	voucher	200
18	3	2017	debit_card	31
19	4	2017	credit_card	1846
20	4	2017	UPI	496
21	4	2017	voucher	202
22	4	2017	debit_card	27
23	5	2017	credit_card	2853
24	5	2017	UPI	772
25	5	2017	voucher	289
26	5	2017	debit_card	30
27	6	2017	credit_card	2463
28	6	2017	UPI	707
29	6	2017	voucher	230

6.2 Distribution of payment instalments and count of orders?

Answer:

```
SELECT payment_installments, count(order_id) as total_orders
from `target-practice-project.Target.payments`
group by payment_installments;
```

Output:



Row	payment_installments	total_orders
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
11	10	5328
12	11	23
13	12	133
14	13	16
15	14	15
16	15	74
17	16	5
18	17	8
19	18	27
20	20	17
21	21	3
22	22	1
23	23	1
24	24	18