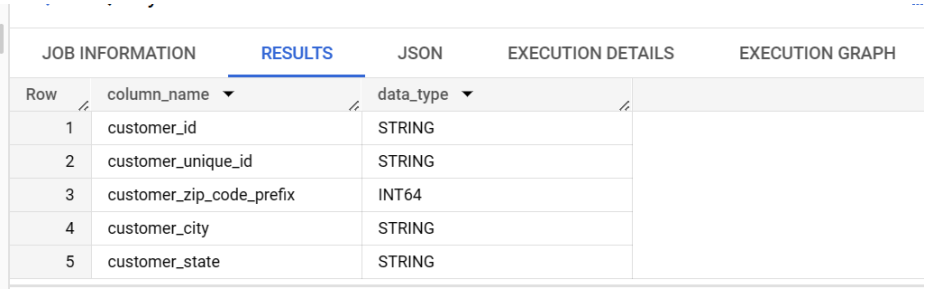
Checking the structure & characteristics of the dataset:

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

```
SELECT column_name, data_type
FROM `businessoperation.Target01.INFORMATION_SCHEMA.COLUMNS`
WHERE table_name = 'customers';
```



JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	column_name	data_type
1	customer_id	STRING
2	customer_unique_id	STRING
3	customer_zip_code_prefix	INT64
4	customer_city	STRING
5	customer_state	STRING

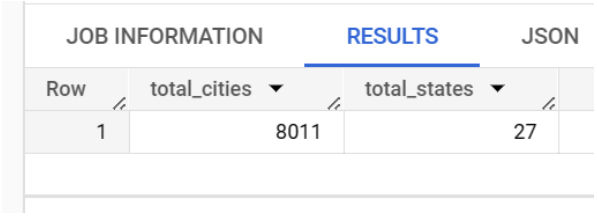
2. Get the time range between which the orders were placed:

```
select order_id, time(order_purchase_timestamp) as time_range
from `Target01.orders`;
```

	order_id	time_range
1		
2	7a4df5d8cff4090	11:10:33
3	35de4050331c6	1:07:58
4	b5359909123fa0	1:07:52
5	dba5062fbda3af	17:21:04
6	90ab3e7d52544	13:12:34
7	fa65dad1b0e818	12:45:34
8	1df2775799eecd	11:03:05
9	6190a94657e10	13:36:30
10	58ce513a55c74(	18:05:07

3. Count the number of Cities and States in our dataset.

```
select count(distinct geolocation_city) total_cities, count(distinct
geolocation_state) total_states
from `businessoperation.Target01.geolocation`;
```



JOB INFORMATION | RESULTS | JSON

Row	total_cities	total_states
1	8011	27

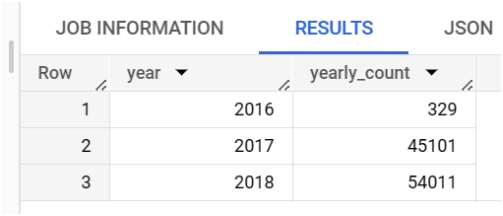
In-depth Exploration

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

```
select * from
(select extract(year from order_purchase_timestamp) as year,
count(order_id) as yearly_count,
from `businessoperation.Target01.orders`
```

```
group by extract(year from order_purchase_timestamp))
order by year;
```

Yes as per the result of the query, there is a upward trend over the year with sharp growth in year 2017 from 2016.



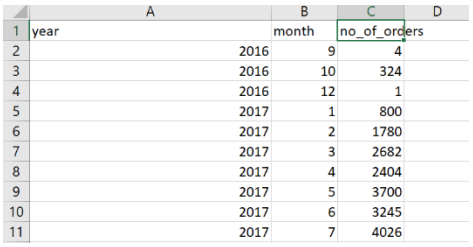
JOB INFORMATION | RESULTS | JSON

Row	year	yearly_count
1	2016	329
2	2017	45101
3	2018	54011

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

```
select year, month, count(order_id) as no_of_orders from
(select order_id, extract(month from order_purchase_timestamp) as month,
extract(year from order_purchase_timestamp) as year
from `businessoperation.Target01.orders`)
group by year, month
order by year, month asc;
```

No, I see no seasonality in the given data as the order purchase data comprises of 3 years data and I find no tends in the given consecutive year.
It is not accurate to find seasonality on the basis of approximately 2 complete year.



	A	B	C	D
1	year	month	no_of_orders	
2	2016	9	4	
3	2016	10	324	
4	2016	12	1	
5	2017	1	800	
6	2017	2	1780	
7	2017	3	2682	
8	2017	4	2404	
9	2017	5	3700	
10	2017	6	3245	
11	2017	7	4026	

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

```
Select time_zone,count(time_zone)
from (
select case
when (time(order_purchase_timestamp) between '00:00:00' and '06:00:00') THEN
'Dawn'
when (time(order_purchase_timestamp) between '07:00:00' and '12:00:00') then
'Mornings'
when (time(order_purchase_timestamp) between '13:00:00' and '18:00:00') then
'Afternoon'
else 'Night'
end as time_zone
from `businessoperation.Target01.orders`
order by time(order_purchase_timestamp))
group by time_zone
order by count(time_zone) desc;
```

Highest order placed during Night time.

JOB INFORMATION | RESULTS | JSON | EXEC

Row	time_zone	f0_
1	Night	40593
2	Afternoon	32370
3	Mornings	21738
4	Dawn	4740

Evolution of E-commerce orders in the Brazil region:

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

```
select state, year, month, count(month) from(
select extract(year from ord.order_purchase_timestamp) as year,
extract(month from ord.order_purchase_timestamp) as month,
cus.customer_state as state from `Target01.orders` ord
left join
`Target01.customers` cus
on ord.customer_id = cus.customer_id)
group by state,year,month
order by year,month;
```

Since, the data fetching is not for any particular year, I have showed the year column to be more articulate with the data.

	state	year	month	f0_
2	RR	2016	9	1
3	RS	2016	9	1
4	SP	2016	9	2
5	SP	2016	10	113
6	RS	2016	10	24
7	RJ	2016	10	56
8	MT	2016	10	3
9	GO	2016	10	9
10	MG	2016	10	40
11	CE	2016	10	8

2. How are the customers distributed across all the states?

```
select customer_state, count(*) as customer_count
from `Target01.customers`
group by customer_state
order by customer_count desc;
```

	customer_state	customer_count
2	SP	41746
3	RJ	12852
4	MG	11635
5	RS	5466
6	PR	5045
7	SC	3637
8	BA	3380
9	DF	2140
10	ES	2033
11	GO	2020

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

1. Get the % increase in the cost of orders from year 2017 to 2018 (include months between Jan to Aug only).
   Join 2 table order and payment to get month and count of year group by (year)
   You can use the "payment_value" column in the payments table to get the cost of orders.

```
select
```

```
((abs((sum_17 - sum_18))/sum_17)* 100) as percentage_increase
from(
select sum(pay.payment_value) as sum_17 from
`Target01.payments` pay join `Target01.orders` o
on pay.order_id = o.order_id
where extract(year from o.order_purchase_timestamp) = 2017
and extract(month from o.order_purchase_timestamp) not in (9,10,11,12)),
(select sum(pay.payment_value) as sum_18 from `Target01.payments` pay join
`Target01.orders` o
on pay.order_id = o.order_id
where extract(year from o.order_purchase_timestamp) = 2018
and extract(month from o.order_purchase_timestamp) not in (9,10,11,12));
```

OUTPUT (%INCREASE) : 136.97687164.... (136.98 approx)

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

```
select state, sum(amount) as Total, sum(amount)/count(state) as Average_value
from(
select cus.customer_state as state, pay.payment_value as amount from
`Target01.payments` pay
join `Target01.orders` ord on pay.order_id = ord.order_id
join `Target01.customers` cus on ord.customer_id = cus.customer_id)
group by state;
```

	state	Total	Average_value
2	BA	616645.8	170.816
3	SP	5998227	137.5046
4	RJ	2144380	158.5259
5	MT	187029.3	195.2289
6	GO	350092.3	165.7634
7	ES	325967.6	154.707
8	RS	890898.5	157.1804
9	MG	1872257	154.7064
10	MA	152523	198.8566
11	SC	623086.4	165.9793

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

```
select cus.customer_state, sum(items.freight_value) as Total,
sum(items.freight_value)/count(distinct cus.customer_state) Average_value from
`Target01.order_items` items
join `Target01.orders` ord
on items.order_id = ord.order_id
join `Target01.customers` cus
on ord.customer_id = cus.customer_id
group by cus.customer_state
order by cus.customer_state;
```

	customer_state	Total	Average_value
2	AC	3686.75	3686.75
3	AL	15914.59	15914.59
4	AM	5478.89	5478.89
5	AP	2788.5	2788.5
6	BA	100156.7	100156.7
7	CE	48351.59	48351.59
8	DF	50625.5	50625.5
9	ES	49764.6	49764.6
10	GO	53114.98	53114.98
11	MA	31523.77	31523.77
12	MG	270853.5	270853.5

Analysis based on sales, freight and delivery time.

1. Find the no. of days taken to deliver each order from the order's purchase date as delivery time.

```
select order_id,date_diff(order_delivered_customer_date,
```

```
order_purchase_timestamp,day) delivery_time
from `Target01.orders`
order by date_diff(order_delivered_customer_date,
order_purchase_timestamp,day) desc;
```

	order_id	delivery_time
1	ca07593549f181	209
2	1b3190b2dfa9d7	208
3	440d0d17af5528	195
4	0f4519c5f1c541c	194
5	285ab9426d698	194
6	2fb597c2f772eca	194
7	47b40429ed8cce	191
8	2fe324febf907e3	189
9	2d7561026d542	188
10	437222e3fd1b07	187
11	c27815f7e3dd0b	187

2. Calculate the difference (in days) between the estimated & actual delivery date of an order.

```
select
order_id,date_diff(order_estimated_delivery_date,order_delivered_customer_date,d
ay) diff_estimated_delivery
from `Target01.orders`
order by order_id;
```

	order_id	diff_estimated_delivery
1	00010242fe8c5a6d1ba2dd792cb16214	8
2	00018f77f2f0320c557190d7a144bdd3	2
3	000229ec398224ef6ca0657da4fc703e	13
4	00024acbcdf0a6daa1e931b038114c75	5
5	00042b26cf59d7ce69dfabb4e55b4fd9	15
6	00048cc3ae777c65dbb7d2a0634bc1ea	14
7	00054e8431b9d7675808bcb819fb4a32	16
8	000576fe39319847cbb9d288c5617fa6	15
9	0005a1a1728c9d785b8e2b08b904576c	0
10	0005f50442cb953dcd1d21e1fb923495	18

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

-- highest

```
with cal as (select cus.customer_state  as cus1, (sum(ord_it.freight_value)/5)
avg1 from
`Target01.order_items` ord_it join `Target01.orders` ord
on ord_it.order_id = ord.order_id
join `Target01.customers` cus
on ord.customer_id=cus.customer_id
group by cus.customer_state
order by sum(ord_it.freight_value)/5)
select cus1, avg1
from cal
order by cal.avg1 desc
limit 5;
```

Row	cus1	avg1
1	SP	143744.6139999...
2	RJ	61117.86200000...
3	MG	54170.69200000...
4	RS	27104.54800000...
5	PR	23570.33600000...

-- lowest

```
with cal1 as (select cus.customer_state as cus1, (sum(ord_it.freight_value)/5)
avg1 from
`Target01.order_items` ord_it join `Target01.orders` ord
```

```
on ord_it.order_id = ord.order_id
join `Target01.customers` cus
on ord.customer_id=cus.customer_id
group by cus.customer_state
order by sum(ord_it.freight_value)/5)
select cus1, avg1
from cal1
order by cal1.avg1
limit 5;
```

Row	cus1	avg1
1	RR	447.038
2	AP	557.7000000000...
3	AC	737.3499999999...
4	AM	1095.777999999...
5	RO	2283.475999999...

4. Find out the top 5 states with the highest & lowest average delivery time.

```
select cus.customer_state,
round(avg(TIMESTAMP_DIFF(TIMESTAMP(ord.order_delivered_customer_date),TIMESTAMP(
ord.order_purchase_timestamp),hour)),2) as delivery_time
from `Target01.orders` ord
left join `Target01.customers` cus
on ord.customer_id = cus.customer_id
group by cus.customer_state
order by delivery_time desc
limit 5;
```

Row	customer_state	delivery_time
1	RR	704.73
2	AP	651.97
3	AM	633.7
4	AL	588.54
5	PA	570.05

```
select cus.customer_state,
round(avg(TIMESTAMP_DIFF(TIMESTAMP(ord.order_delivered_customer_date),TIMESTAMP(
ord.order_purchase_timestamp),hour)),2) as delivery_time
from `Target01.orders` ord
left join `Target01.customers` cus
on ord.customer_id = cus.customer_id
group by cus.customer_state
order by delivery_time
limit 5;
```

Row	customer_state	delivery_time
1	SP	209.77
2	PR	287.3
3	MG	287.75
4	DF	310.72
5	SC	358.52

5. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

```
(select cus.customer_state,
round(avg(date_diff(ord.order_estimated_delivery_date,ord.order_delivered_custom
er_date,day)))as delivery
from `Target01.orders` ord
```

```
left join `Target01.customers` cus
on ord.customer_id = cus.customer_id
group by cus.customer_state)
order by delivery desc
limit 5;
```

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

```
(select cus.customer_state,
round(avg(date_diff(ord.order_estimated_delivery_date,ord.order_delivered_custom
er_date,day)))as delivery
from `Target01.orders` ord
left join `Target01.customers` cus
on ord.customer_id = cus.customer_id
group by cus.customer_state)
order by delivery
limit 5;
```

Row	customer_state	delivery
1	AL	8.0
2	MA	9.0
3	SE	9.0
4	SP	10.0
5	BA	10.0

Analysis based on the payments:

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

```
select year,month,payment_type, sum(month_count) over(order by month)
from(
  select (extract(year from ord.order_purchase_timestamp))year,
  (extract(month from order_purchase_timestamp))month,
  pay.payment_type, count(ord.order_id) as month_count
  from `Target01.payments` pay
  join `Target01.orders` ord
  on pay.order_id = ord.order_id
  group by extract(year from ord.order_purchase_timestamp),
  extract(month from ord.order_purchase_timestamp),
  pay.payment_type)
order by payment_type,year,month;
```

Row	year	month	payment_type	f0_
1	2016	10	UPI	90127
2	2017	1	UPI	8413
3	2017	2	UPI	17251
4	2017	3	UPI	27600
5	2017	4	UPI	37380
6	2017	5	UPI	48459
7	2017	6	UPI	58314
8	2017	7	UPI	69138
9	2017	8	UPI	80386

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

```
select payment_installments, count(order_id) as no_of_orders
from `Target01.payments`
group by payment_installments
```

Row	payment_installment	no_of_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