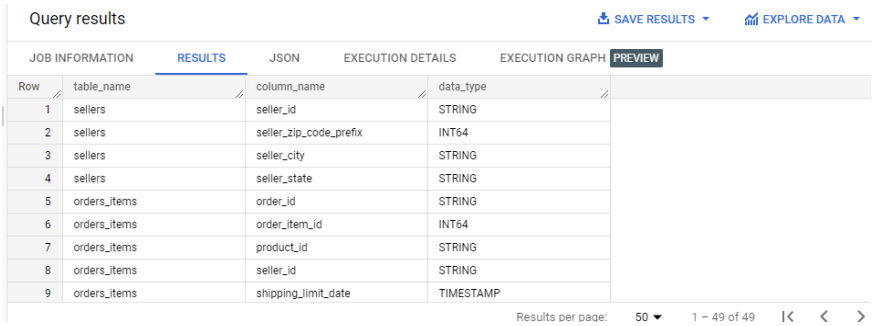
1.1: Data type of columns in a table

```
SELECT table_name,column_name,data_type FROM target.INFORMATION_SCHEMA.COLUMNS ;
```



Query results

SAVE RESULTS | EXPLORE DATA

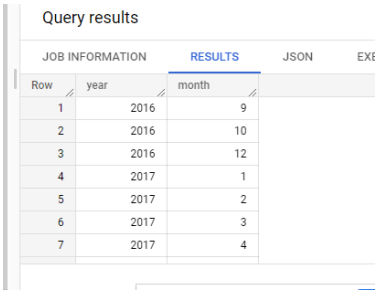
JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRAPH | PREVIEW

Row	table_name	column_name	data_type
1	sellers	seller_id	STRING
2	sellers	seller_zip_code_prefix	INT64
3	sellers	seller_city	STRING
4	sellers	seller_state	STRING
5	orders_items	order_id	STRING
6	orders_items	order_item_id	INT64
7	orders_items	product_id	STRING
8	orders_items	seller_id	STRING
9	orders_items	shipping_limit_date	TIMESTAMP

Results per page: 50 | 1 – 49 of 49

1.2 Time period for which the data is given

```
SELECT distinct extract(year from  order_purchase_timestamp) as year,count(*) as counting FROM
 `target.orders` group by year
order by counting desc
```



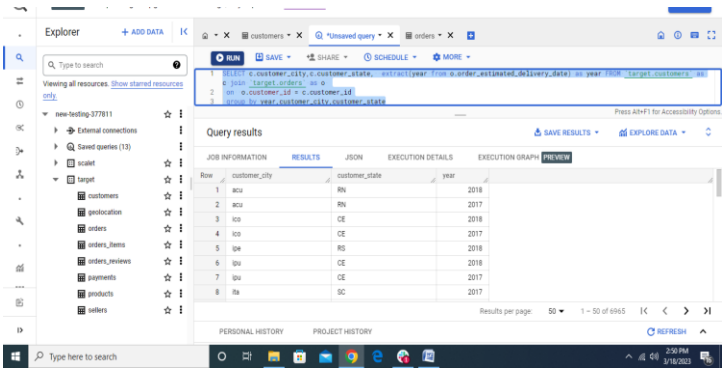
Query results

JOB INFORMATION | RESULTS | JSON | EXE

Row	year	month
1	2016	9
2	2016	10
3	2016	12
4	2017	1
5	2017	2
6	2017	3
7	2017	4

1.3 Cities and States of customers ordered during the given period

```
SELECT c.customer_city,c.customer_state,  extract(year from o.order_estimated_delivery_date) a
s year FROM `target.customers` as c join `target.orders` as o
 on  o.customer_id = c.customer_id
 group by year,customer_city,customer_state
```



Explorer | + ADD DATA

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_city	customer_state	year
1	acu	RN	2018
2	acu	RN	2017
3	ico	CE	2018
4	ico	CE	2017
5	ipe	RS	2018
6	ipu	CE	2018
7	ipu	CE	2017
8	ita	SC	2017

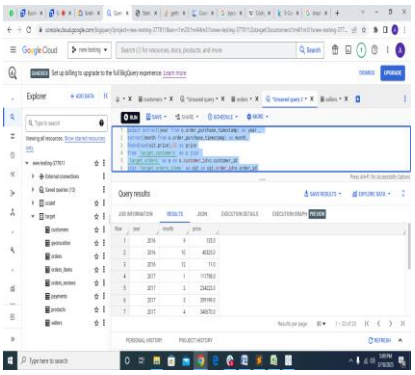
Results per page: 50 | 1 – 50 of 6965

PERSONAL HISTORY | PROJECT HISTORY | REFRESH

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?

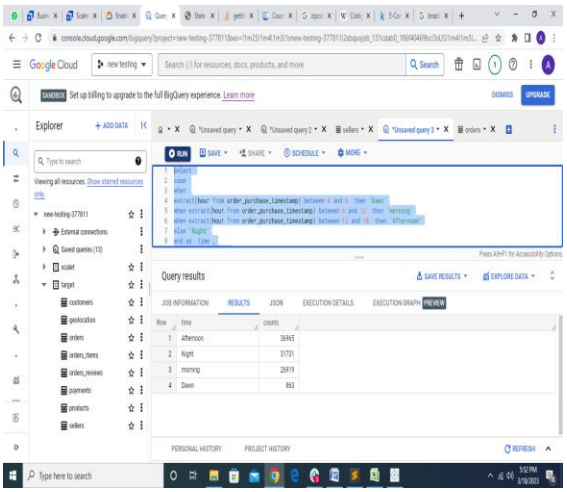
Peaks at 5th month month during the 1st half and peaks in 11 month during the second half of the year

```
select extract(year from o.order_purchase_timestamp) as year ,
extract(month from o.order_purchase_timestamp) as month,
Round(sum(oit.price),0) as price
from `target.customers` as c join
`target.orders` as o on o.customer_id=c.customer_id
join `target.orders_items` as oit on oit.order_id=o.order_id
where o.order_status='delivered'
group by year,month
order by year,month
```



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

```
select
case
when
extract(hour from order_purchase_timestamp) between 4 and 6  then 'Dawn'
when extract(hour from order_purchase_timestamp) between 6 and 12  then 'morning'
when extract(hour from order_purchase_timestamp) between 12 and 18  then 'Afternoon'
else 'Night'
end as  time ,
count(*) as counts
 from `target.orders` where order_status='delivered'
 group by time
```



3.1 Get month on month orders by states

```
select
c.customer_state as state,
extract(month from o.order_purchase_timestamp) as month,
count(c.customer_id)
from `target.customers` as c
join  `target.orders` as o on
c.customer_id = o.customer_id
group by state, month
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS

Row	state	month	f0_
1	RN	1	51
2	RN	12	30
3	RN	5	39
4	CE	2	101
5	CE	3	126
6	CE	5	136

3.2: Distribution of customers across the states in Brazil

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

JOB INFORMATION | RESULTS | JSON | EXECUTI

Row	customer_state	count
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380

4.1 Get % increase in cost of orders from 2017 to 2018 (include months between Jan to Aug only) - You can use “payment_value” column in payments table

```
with A as (
select
extract(year from o.order_purchase_timestamp) as yr,
sum(py.payment_value) as cost_of_order
 from `target.payments` as py join `target.orders` as o on
o.order_id=py.order_id
where extract(month from o.order_purchase_timestamp) between 1 and 8
group by 1
having yr in (2017,2018))

select Round(((A1.cost_of_order / A2.cost_of_order ) -1)*100,2) as prc_increase
from A as A1, A as A2
where A1.yr=2018 and A2.yr=2017
```

Query results

JOB INFORMATION | RESULTS | JS

Row	prc_increase
1	136.98

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

```
select c.customer_state,avg(oi.price) as avg_price
,avg(oi.freight_value) as avg_freight
```

```
,Round(sum(oi.price),0) as total_price
,Round(sum(oi.freight_value),0) as total_freight
 from `target.customers` as c
join `target.orders`  as o on o.customer_id=c.customer_id
join `target.orders_items` as oi on oi.order_id=o.order_id
group by c.customer_state
order by total_price,total_freight
limit 5
```

Query results

SAV

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

Row	customer_state	avg_price	avg_freight	total_price	total_freight
1	RR	150.565961...	42.9844230...	7829.0	2235.0
2	AP	164.320731...	34.0060975...	13474.0	2789.0
3	AC	173.727717...	40.0733695...	15983.0	3687.0
4	AM	135.495999...	33.2053939...	22357.0	5479.0
5	RO	165.973525...	41.0697122...	46141.0	11417.0

5.1 Calculate days between purchasing, delivering and estimated delivery

5.2 Find time_to_delivery & diff_estimated_delivery. Formula for the same given below:

```
select
o.order_id,
o.order_purchase_timestamp,o.order_delivered_customer_date,order_estimated_delivery_date,
date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day) as time_to_delivery,
date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day) as diff_estimat
ed_delivery
from `target.orders` as o
left join `target.customers` as c
on c.customer_id=o.customer_id
left join `target.orders_items` as oi on
oi.order_id=o.order_id
order by o.order_id
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row		order_purchase_timestamp	order_delivered_customer_date	order_estimated_delivery_date	time_to_delivery	diff_estimated_c
1	e8c5a6d1ba2dd792...	2017-09-13 08:59:02 UTC	2017-09-20 23:43:48 UTC	2017-09-29 00:00:00 UTC	7	8
2	f0320c557190d7a1...	2017-04-26 10:53:06 UTC	2017-05-12 16:04:24 UTC	2017-05-15 00:00:00 UTC	16	2
3	98224ef6ca0657da...	2018-01-14 14:33:31 UTC	2018-01-22 13:19:16 UTC	2018-02-05 00:00:00 UTC	7	13
4	df0a6daa1e931b03...	2018-08-08 10:00:35 UTC	2018-08-14 13:32:39 UTC	2018-08-20 00:00:00 UTC	6	5
5	f59d7ce69dfabb4e...	2017-02-04 13:57:51 UTC	2017-03-01 16:42:31 UTC	2017-03-17 00:00:00 UTC	25	15
6	e777c65dbb7d2a06...	2017-05-15 21:42:34 UTC	2017-05-22 13:44:35 UTC	2017-06-06 00:00:00 UTC	6	14
7	1b9d7675808bcb8...	2017-12-10 11:53:48 UTC	2017-12-18 22:03:38 UTC	2018-01-04 00:00:00 UTC	8	16
8	9319847cbb9d288c...	2018-07-04 12:08:27 UTC	2018-07-09 14:04:07 UTC	2018-07-25 00:00:00 UTC	5	15
9	28c9d785b8e2b08...	2018-03-19 18:40:33 UTC	2018-03-29 18:17:31 UTC	2018-03-29 00:00:00 UTC	9	0

5.3 Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery

```
select
c.customer_state ,
oi.freight_value,
avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)) as avg_time_to_
delivery,
avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)) as avg_dif
f_estimated_delivery
from `target.orders` as o
left join `target.customers` as c
on c.customer_id=o.customer_id
left join `target.orders_items` as oi on
oi.order_id=o.order_id
group by c.customer_state,oi.freight_value
order by c.customer_state
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXEC

Row	customer_state	freight_value	avg_time_to_del	avg_diff_estimat
1	AC	42.39	*null*	*null*
2	AC	41.26	18.0	33.0
3	AC	14.86	18.0	33.0
4	AC	26.04	16.0	21.0
5	AC	27.75	15.0	32.0

5.4 Sort the data to get the following

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

Ascending order

```
select
c.customer_state ,
avg(oi.freight_value) as avg_freight,
Round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),0) as avg
_time_to_delivery,
Round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)),0) a
s avg_diff_estimated_delivery
from `target.orders` as o
left join `target.customers` as c
on c.customer_id=o.customer_id
left join `target.orders_items` as oi on
oi.order_id=o.order_id
group by c.customer_state
order by avg_freight  limit 5
```

Highest time

```
select
c.customer_state ,
avg(oi.freight_value) as avg_freight,
-- count(c.customer_state) as count_state,
Round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),0) as avg
_time_to_delivery,
Round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)),0) a
s avg_diff_estimated_delivery
from `target.orders` as o
left join `target.customers` as c
on c.customer_id=o.customer_id
left join `target.orders_items` as oi on
oi.order_id=o.order_id
group by c.customer_state
order by avg_freight desc limit 5
```

NFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GR

customer_state	avg_freight	avg_time_to_del	avg_diff_estimat
RR	42.9844230...	28.0	17.0
PB	42.7238039...	20.0	12.0
RO	41.0697122...	19.0	19.0
AC	40.0733695...	20.0	20.0
PI	39.1479704...	19.0	11.0

5.6 Top 5 states with highest/lowest average time to delivery

Hightest

```
select
c.customer_state ,
avg(oi.freight_value) as avg_freight,
-- count(c.customer_state) as count_state,
Round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),0) as avg
_time_to_delivery,
Round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)),0) a
s avg_diff_estimated_delivery
from `target.orders` as o
left join `target.customers` as c
on c.customer_id=o.customer_id
left join `target.orders_items` as oi on
oi.order_id=o.order_id
group by c.customer_state
order by avg_time_to_delivery desc
limit 5
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION G

ow	customer_state	avg_freight	avg_time_to_del	avg_diff_estimat
1	AP	34.0060975...	28.0	17.0
2	RR	42.9844230...	28.0	17.0
3	AM	33.2053939...	26.0	19.0
4	AL	35.8436711...	24.0	8.0
5	PA	35.8326851...	23.0	13.0

lowest

```
select
c.customer_state ,
avg(oi.freight_value) as avg_freight,
-- count(c.customer_state) as count_state,
Round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),0) as avg
_time_to_delivery,
Round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)),0) a
s avg_diff_estimated_delivery
from `target.orders` as o
left join `target.customers` as c
on c.customer_id=o.customer_id
left join `target.orders_items` as oi on
oi.order_id=o.order_id
group by c.customer_state
order by avg_time_to_delivery
limit 5
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GRA

Row	customer_state	avg_freight	avg_time_to_del	avg_diff_estimat
1	SP	15.1472753...	8.0	10.0
2	PR	20.5316515...	11.0	13.0
3	MG	20.6301668...	12.0	12.0
4	DF	21.0413549...	13.0	11.0
5	RS	21.7358043...	15.0	13.0

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

really fast

```
select

    c.customer_state ,
    avg(oi.freight_value) as avg_freight,
    Round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),0) as
     avg_time_to_delivery,
    Round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)),
    0) as avg_diff_estimated_delivery
     from `target.orders` as o
    left join `target.customers` as c
    on c.customer_id=o.customer_id
    left join `target.orders_items` as oi on
     oi.order_id=o.order_id
    group by c.customer_state
    order by avg_diff_estimated_delivery
    limit 5
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION

Row	customer_state	avg_freight	avg_time_to_del	avg_diff_estimat
1	AL	35.8436711...	24.0	8.0
2	MA	38.2570024...	21.0	9.0
3	SE	36.6531688...	21.0	9.0
4	SP	15.1472753...	8.0	10.0
5	BA	26.3639589...	19.0	10.0

not so fast

```
select
c.customer_state ,
avg(oi.freight_value) as avg_freight,
```

```
Round(avg(date_diff(o.order_delivered_customer_date,o.order_purchase_timestamp,day)),0) as avg
_time_to_delivery,
Round(avg(date_diff(o.order_estimated_delivery_date,o.order_delivered_customer_date,day)),0) a
s avg_diff_estimated_delivery
from `target.orders` as o
left join `target.customers` as c
on c.customer_id=o.customer_id
left join `target.orders_items` as oi on
oi.order_id=o.order_id
group by c.customer_state
order by avg_diff_estimated_delivery desc
limit 5
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETAILS | EXECUTION GR

Row	customer_state	avg_freight	avg_time_to_del	avg_diff_estimat
1	AC	40.0733695...	20.0	20.0
2	RO	41.0697122...	19.0	19.0
3	AM	33.2053939...	26.0	19.0
4	RR	42.9844230...	28.0	17.0
5	AP	34.0060975...	28.0	17.0

6.1Month over Month count of orders for different payment types

```
Select p.payment_type,

    extract(month from o.order_purchase_timestamp) as month,
    count(o.order_id) as order_count
    from `target.orders` as o join
    `target.payments` as p on
    p.order_id=o.order_id
    group by month ,p.payment_type
```

JOB INFORMATION | RESULTS | JSON | EXECUTION DETA

Row	payment_type	month	order_count
1	credit_card	5	8350
2	credit_card	4	7301
3	voucher	1	477
4	voucher	4	572
5	voucher	10	318
6	not_defined	9	1
7	not_defined	8	2
8	voucher	6	563
9	voucher	5	613

6.2 Count of orders based on the no. of payment installments

```
select
p.payment_installments,
count(*) as counting
 from `target.orders` as o
join `target.payments` as p
on p.order_id = o.order_id
group by p.payment_installments
order by counting desc
```

Query results

JOB INFORMATION | RESULTS | JSC

Row	payment_installr	counting
1	1	52546
2	2	12413
3	3	10461
4	4	7098
5	10	5328