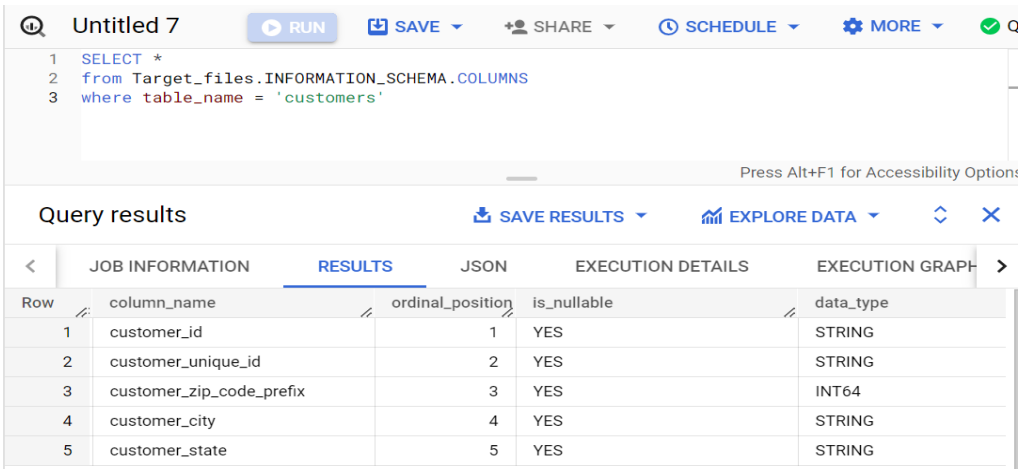
Business Case Study: Target

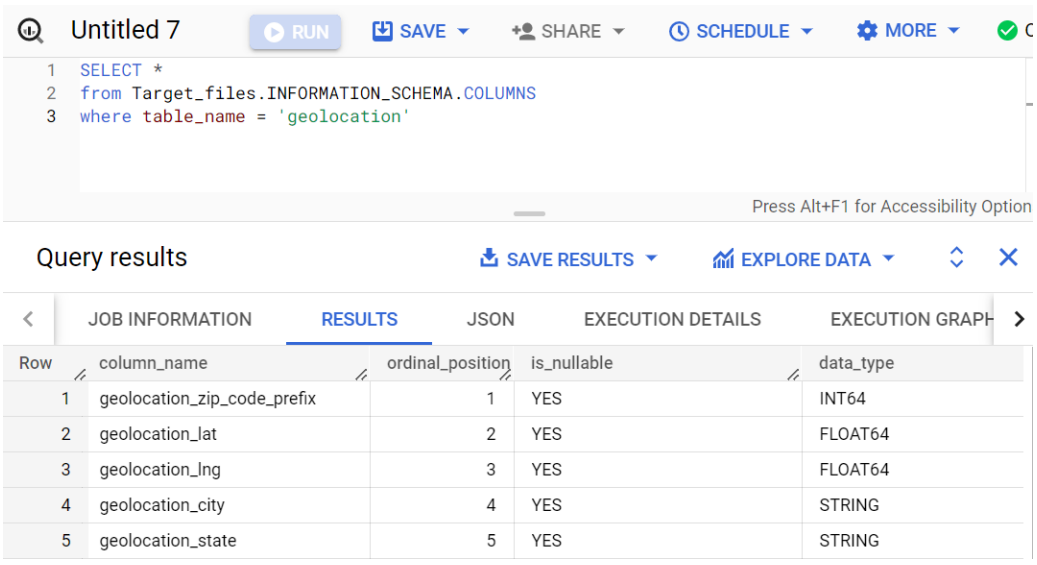
1. Import the dataset and do usual exploratory analysis steps like checking the structure & characteristics of the dataset:

1.1 Data type of columns in a table

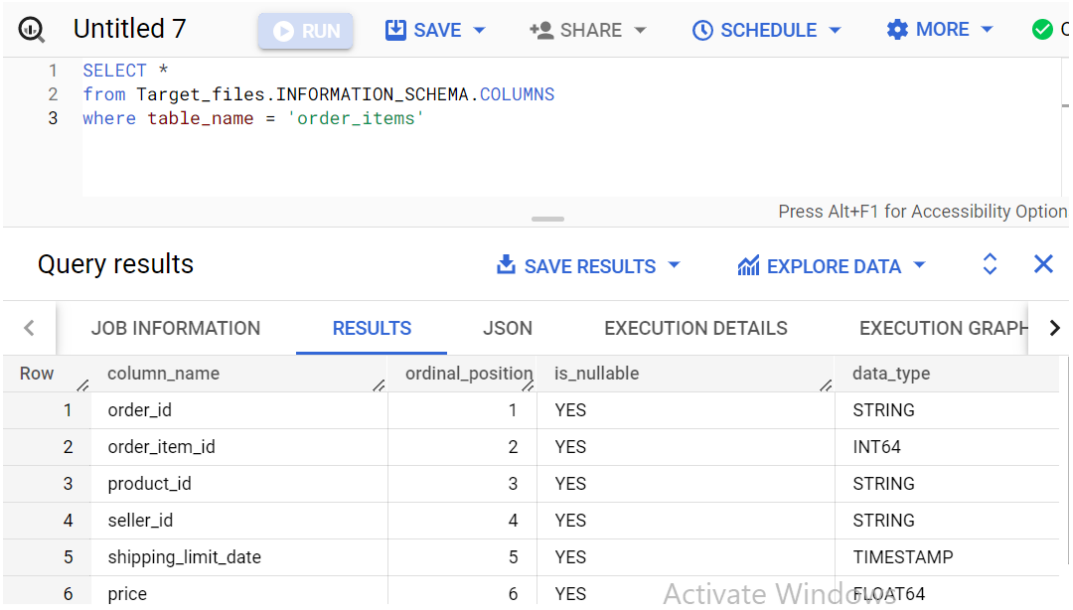
- Columns: ‘customers’



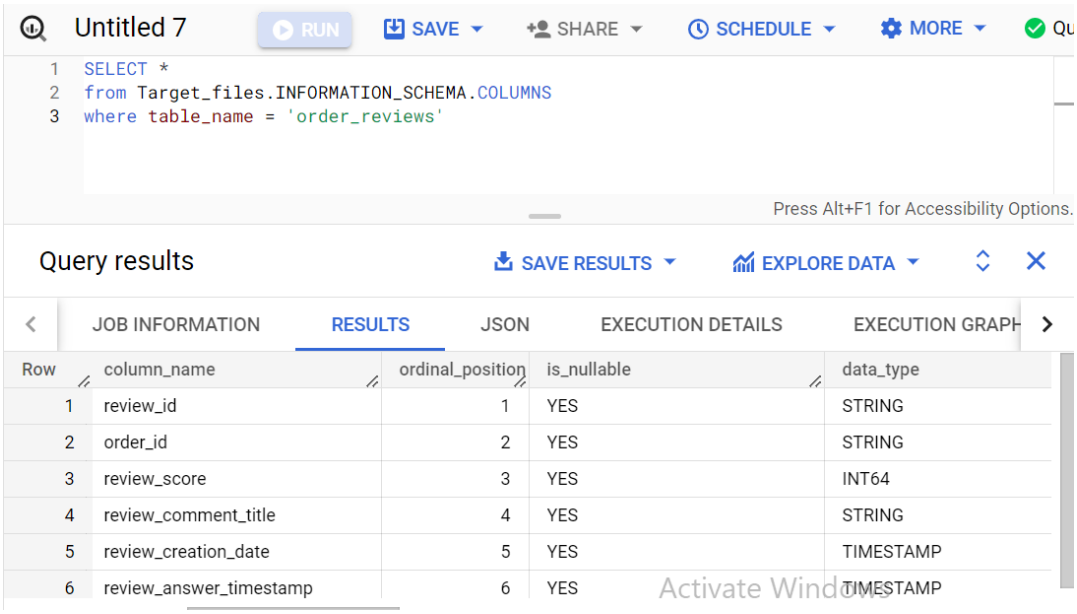
- Column: ‘geolocation’



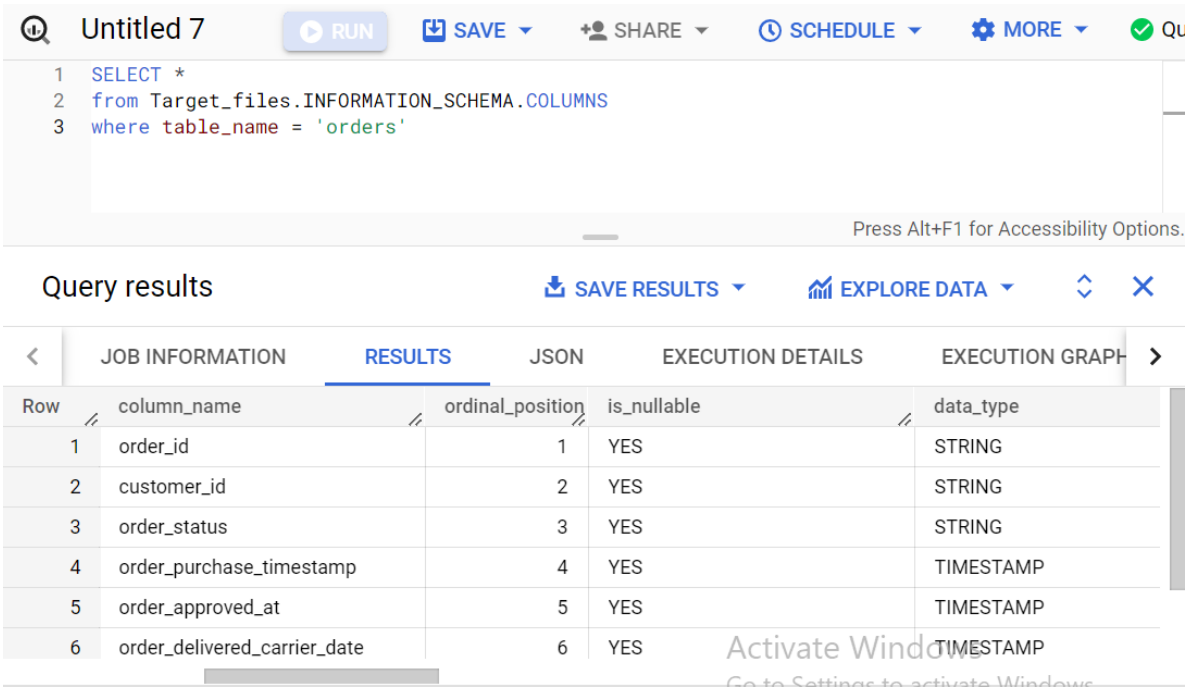
- Columns: ‘order_items’



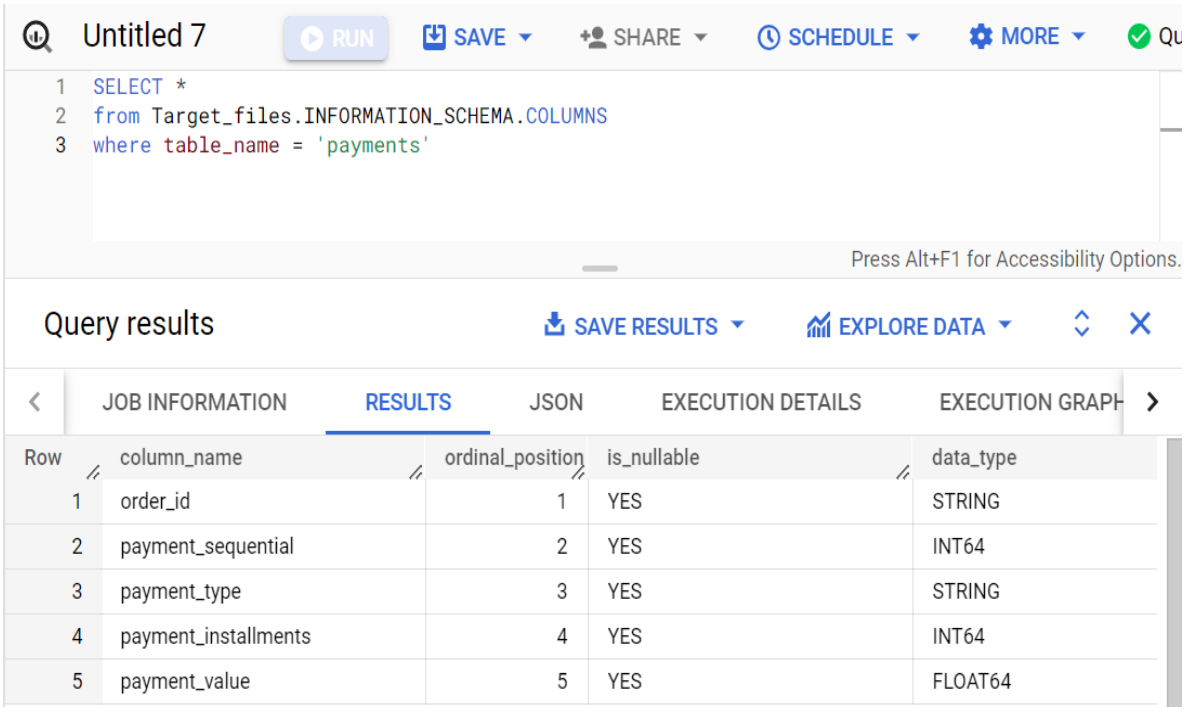
- Columns: ‘order_reviews’



- Columns: ‘orders’



- Columns: ‘payments’



- Columns: ‘products’

Untitled 7 RUN SAVE SHARE SCHEDULE MORE Qu

```
1 SELECT *
2 from Target_files.INFORMATION_SCHEMA.COLUMNS
3 where table_name = 'products'
```

Press Alt+F1 for Accessibility Options.

Query results SAVE RESULTS EXPLORE DATA

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

Row	column_name	ordinal_position	is_nullable	data_type
1	product_id	1	YES	STRING
2	product_category	2	YES	STRING
3	product_name_length	3	YES	INT64
4	product_description_length	4	YES	INT64
5	product_photos_qty	5	YES	INT64
6	product_weight_g	6	YES	INT64

- Columns: ‘sellers’

Untitled 7 RUN SAVE SHARE SCHEDULE MORE Qu

```
1 SELECT *
2 from Target_files.INFORMATION_SCHEMA.COLUMNS
3 where table_name = 'sellers'
```

Press Alt+F1 for Accessibility Options.

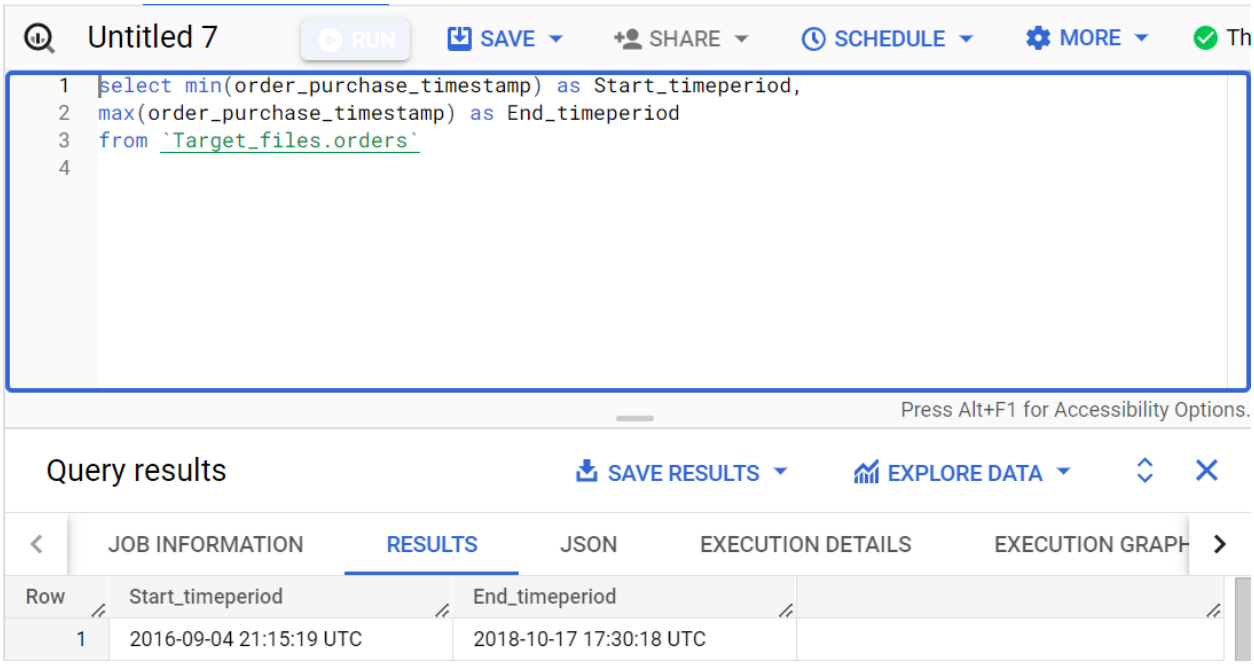
Query results SAVE RESULTS EXPLORE DATA

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

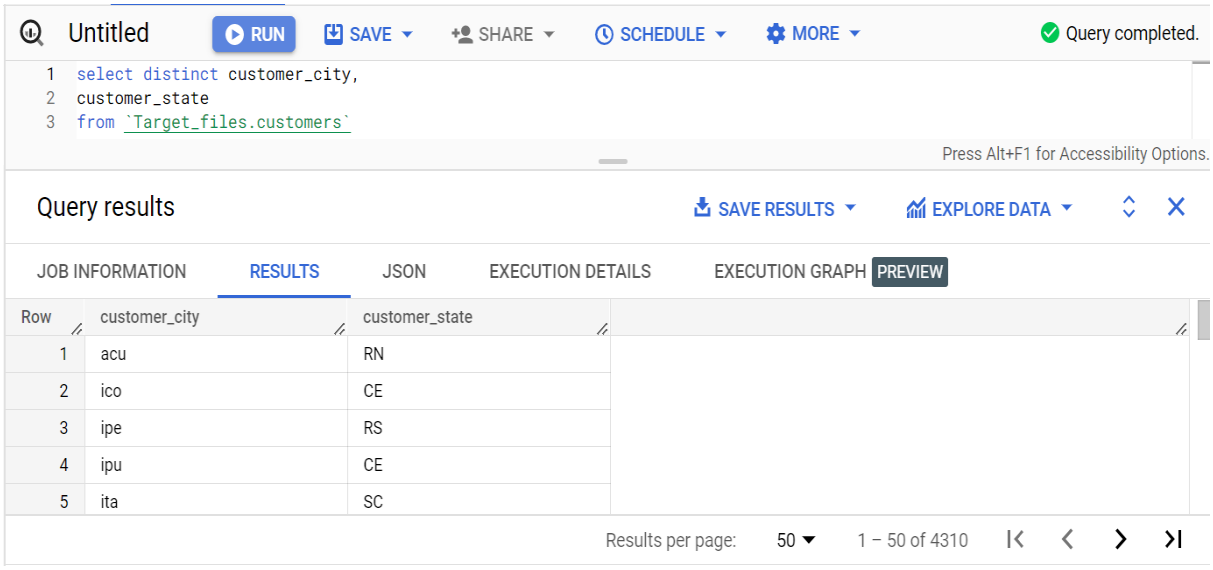
Row	column_name	ordinal_position	is_nullable	data_type
1	seller_id	1	YES	STRING
2	seller_zip_code_prefix	2	YES	INT64
3	seller_city	3	YES	STRING
4	seller_state	4	YES	STRING

1.2 Time period for which the data is given

Analysis: The time period for which the data is provided is from 04-09-2016 to 17-10-2018



1.3 Cities and States of customers ordered during the given period

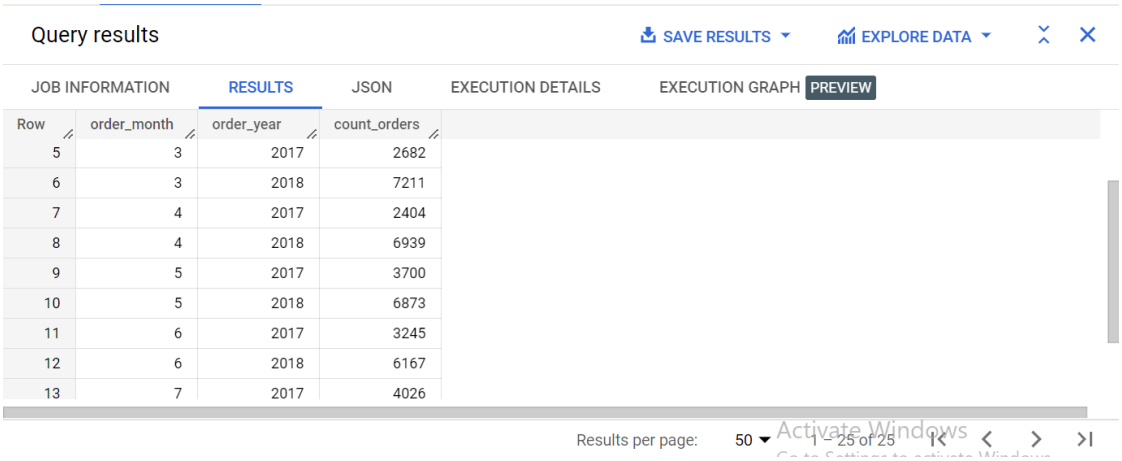


2. In-depth Exploration:

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?

Code :

```
select
extract(month from order_purchase_timestamp) as order_month,
extract(year from order_purchase_timestamp) as order_year,
count(order_id) as count_orders
from `Target_files.orders`
group by order_month, order_year
order by order_month, order_year
```



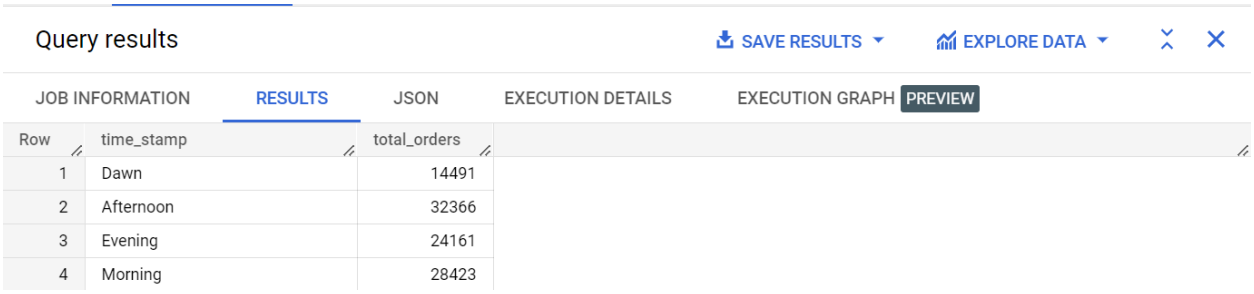
Analysis : Following are the orders given in every month (UP). Following are the orders with order by count_orders desc (DOWN). As we can notice order count is more in November, January, March. These are the peak seasons since most of the Brazilian festivals like Carnival, Bonfim Stairs Washing happens around these months.

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

Analysis : Most sales are done in Afternoon.

Code :

```
select
time_stamp, count(time_stamp) as total_orders
from
(select
extract(hour from order_purchase_timestamp) as purchase_duration,
case
when extract(hour from order_purchase_timestamp) between 5 and 12 then 'Morning'
when extract(hour from order_purchase_timestamp) between 13 and 17 then 'Afternoon'
when extract(hour from order_purchase_timestamp) between 18 and 21 then 'Evening'
when extract(hour from order_purchase_timestamp) between 22 and 4 then 'Night'
else 'Dawn'
end as time_stamp
from `Target_files.customers` c
join `Target_files.orders` o
on o.customer_id = c.customer_id) as tbl1
group by time_stamp
```



Row	time_stamp	total_orders
1	Dawn	14491
2	Afternoon	32366
3	Evening	24161
4	Morning	28423

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

3.1 Get month on month orders by states

Code :

```
select
FORMAT_DATETIME("%B", DATETIME(order_purchase_timestamp)) as month_name,
customer_city,
customer_state
from
`Target_files.orders` o
join `Target_files.customers` c
on o.customer_id = c.customer_id
```

```
group by customer_city,customer_state, order_purchase_timestamp
order by customer_city, customer_state
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	month_name	customer_city	customer_state
1	September	abadia dos dourados	MG
2	July	abadia dos dourados	MG
3	March	abadia dos dourados	MG
4	January	abadiania	GO
5	March	abaete	MG
6	May	abaete	MG
7	November	abaete	MG
8	July	abaete	MG
9	November	abaete	MG

Results per page: 50 1 – 50 of 99140

3.2 Distribution of customers across the states in Brazil

Code :

```
select
c.customer_id, c.customer_city, c.customer_state,
o.order_id, o.order_status,
oi.seller_id, oi.product_id
from `Target_files.customers` c
join `Target_files.orders` o
on c.customer_id = o.customer_id
join `Target_files.order_items` oi
on o.order_id = oi.order_id
where order_status = 'delivered'
```

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

Row	customer_id	customer_city	customer_state	order_id	order_status	seller_id	product_id
101	44eb235a2b...	parnamirim	RN	7c971e77...	delivered	d98eec89...	179d54e1466a...
102	7d8140efc8...	parnamirim	RN	6b2f7dba...	delivered	d98eec89...	ce582a7aa032...
103	6d7caf3c6d...	parnamirim	RN	41932b5...	delivered	2138ccb8...	eba9c1e37886...
104	6d7caf3c6d...	parnamirim	RN	41932b5...	delivered	7d76b645...	41db6d8062fc...
105	63e7a08af4...	boa vista	RR	16a73b5...	delivered	ffeee66ac...	1347d4320dcd...
106	1141b746bb...	maceio	AL	811ecf2d...	delivered	897060da...	b40ec43bdfc6...
107	d28fb373c7...	aracaju	SE	d4c2efa6...	delivered	5f1dc280...	7346d3bfa9ffb...
108	53d0d57a83...	pirambu	SE	4316bb5...	delivered	53243585...	431d674f9a4fb...
109	b846152f6c...	natal	RN	0794929f...	delivered	25c5c91f6...	6987398dff454...

Results per page: 50 101 – 150 of 110197

4. Impact on Economy: Analyze 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) - You can use “payment_value” column in payments table

Code :

```
select
round(((ty_sales - ly_sales)/(ly_sales))*100.0) as Percentage_Increase
from(
select
sum(case when year = 2018 and month between 1 and 8 then payment_value end) as ty_sales,
sum(case when year = 2017 and month between 1 and 8 then payment_value end) as ly_sales
from
(
  select
  extract(month from o.order_purchase_timestamp) as Month,
  extract(year from o.order_purchase_timestamp) as Year,
  p.payment_value
  from `Target_files.orders` o
  join `Target_files.payments` p
  on o.order_id = p.order_id))
```

Query results

JOB INFORMATION | RESULTS | JSON | EXECU

Row	Percentage_Increase
1	137.0

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

Code :

```
select
customer_state,
sum(oi.price) as price_sum,
avg(oi.freight_value) as freight_avg,
sum(oi.freight_value) as freight_sum,
avg(oi.price) as price_avg
from `Target_files.customers` c
join `Target_files.orders` o
on o.customer_id = c.customer_id
```

```
join `Target_files.order_items` oi
on o.order_id = oi.order_id
group by customer_state
```

Query results

SAVE RESULTS | EXPLORE DATA

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

Row	customer_state	price_sum	freight_avg	freight_sum	price_avg
1	MT	156453.529...	28.1662843...	29715.4300...	148.297184...
2	MA	119648.219...	38.2570024...	31523.7700...	145.204150...
3	AL	80314.81	35.8436711...	15914.5899...	180.889211...
4	SP	5202955.05...	15.1472753...	718723.069...	109.653629...
5	MG	1585308.02...	20.6301668...	270853.460...	120.748574...
6	PE	262788.029...	32.9178626...	59449.6599...	145.508322...
7	RJ	1824092.66...	20.9609239...	305589.310...	125.117818...
8	DF	302603.939...	21.0413549...	50625.4999...	125.770548...
9	RS	750304.020...	21.7358043...	135522.740...	120.337453...

Results per page: 50 1 – 27 of 27

5. Analysis on sales, freight and delivery time

5.1 Calculate days between purchasing, delivering and estimated delivery

Code :

```
select
customer_id, order_id, order_status,
date(order_purchase_timestamp) as purchased_order,
date(order_delivered_customer_date) as delivered_date,
date(order_estimated_delivery_date) as estimated_date
from `Target_files.orders`
```

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

Row	customer_id	order_id	order_status	purchased_orde	delivered_date	estimated_date
1	25456ee3b0cf846...	c158e9806f85...	delivered	2017-04-14	2017-05-08	2017-05-18
2	2f9902d85fcd930...	b60b53ad0bb...	delivered	2017-05-10	2017-05-23	2017-05-18
3	af626bcc9c27c08...	c830f223aae0...	delivered	2017-04-22	2017-05-05	2017-05-18
4	2c5519c36277c3f...	a8aa2cd070ee...	delivered	2017-05-09	2017-05-16	2017-05-18
5	33ff667cdb878cb...	813c55ce9b6b...	delivered	2017-04-26	2017-05-08	2017-05-18
6	40e2a5bab2a3629...	44558a1547e4...	delivered	2017-05-10	2017-05-12	2017-05-18
7	04719f10ea55e8d...	036b7918978...	delivered	2017-05-10	2017-05-17	2017-05-18
8	6be28898a686e86...	1aba60c04110...	delivered	2017-04-18	2017-05-10	2017-05-18
9	45cce495588388d...	0312ecf90786...	delivered	2017-05-10	2017-05-18	2017-05-18

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

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

Code :

```
select
order_id, customer_id, order_status,
abs(date_diff(order_delivered_customer_date, order_purchase_timestamp, day)) as time_to_delivery,
abs(date_diff(order_delivered_customer_date, order_estimated_delivery_date, day)) as diff_estimated_delivery
from `Target_files.orders`
```

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

Row	order_id	customer_id	order_status	time_to_delivery	diff_estimated_d
1	770d331c84e5b214bd9dc70a1...	6c57e6119369185e575b36712...	canceled	7	45
2	dabf2b0e35b423f94618bf965f...	5cdec0bb8cbdf53ffc8fdc212c...	canceled	7	44
3	8beb59392e21af5eb9547ae1a...	bf609b5741f71697f65ce3852c...	canceled	10	41
4	1a0b31f08d0d7e87935b819ed...	7e769bb9acb55403ebd7ea57a...	delivered	6	29
5	cec8f5f7a13e5ab934a486ec9e...	6be61d704faaff9b97ccf47289...	delivered	20	40
6	58527ee4726911bee84a0f42c...	b7d68eb92ede54186f0385024...	delivered	10	48
7	10ed5499d1623638ee810eff1...	2bf569d940353f09136cab77b...	delivered	28	29
8	818996ea247803ddc123789f2...	19b1122a589ca4893fcce3cb2...	delivered	9	35
9	d195cac9ccaa1394ede717d38...	a3a156d272fd0b0e302d20203...	delivered	10	41

Results per page: 50 | 1 – 50 of 99441

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

Code:

```
select customer_state,
avg(freight_value) as avg_freight_value,
avg(tbl1.time_to_delivery) as avg_time_to_delivery,
avg(tbl1.diff_estimated_delivery) as avg_diff_estimated_delivery
from
(select c.customer_state, freight_value,
abs(date_diff(order_delivered_customer_date, order_purchase_timestamp, day)) as time_to_delivery,
abs(date_diff(order_estimated_delivery_date,order_delivered_customer_date, day)) as diff_estimated_delivery
from `Target_files.orders` o
```

```
join `Target_files.customers` c
on o.customer_id = c.customer_id
join `Target_files.order_items` oi
on oi.order_id = o.order_id) tbl1
group by customer_state
```

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

Row	customer_state	avg_freight_valu	avg_time_to_deli	avg_diff_estimat
1	MT	28.1662843...	17.5081967...	14.8871745...
2	MA	38.2570024...	21.2037500...	12.6624999...
3	AL	35.8436711...	23.9929742...	12.0608899...
4	SP	15.1472753...	8.25960855...	10.9855737...
5	MG	20.6301668...	11.5155221...	13.1347836...
6	PE	32.9178626...	17.7920962...	14.6884306...
7	RJ	20.9609239...	14.6893821...	14.2255054...
8	DF	21.0413549...	12.5014861...	12.2199575...
9	RS	21.7358043...	14.7082993...	14.4588292...

Results per page: 50

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 5

Code (asc):

```
select customer_state, avg(freight_value) as avg_freight_value
from `Target_files.customers` c
join `Target_files.orders` o
on c.customer_id = o.customer_id
join `Target_files.order_items` oi
on oi.order_id = o.order_id
group by customer_state
order by avg_freight_value
limit 5
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	avg_freight_valu
1	SP	15.1472753...
2	PR	20.5316515...
3	MG	20.6301668...
4	RJ	20.9609239...
5	DF	21.0413549...

Code (desc):

```
select customer_state, avg(freight_value) as avg_freight_value
from `Target_files.customers` c
```

```
join `Target_files.orders` o
on c.customer_id = o.customer_id
join `Target_files.order_items` oi
on oi.order_id = o.order_id
group by customer_state
order by avg_freight_value desc
limit 5
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	avg_freight_valu
1	RR	42.9844230...
2	PB	42.7238039...
3	RO	41.0697122...
4	AC	40.0733695...
5	PI	39.1479704...

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

Code(Lowest):

```
select customer_state,
avg(tbl1.time_to_delivery) as avg_time_to_delivery
from
(select c.customer_state,
abs(date_diff(order_delivered_customer_date, order_purchase_timestamp, day)) as time_to_delive
ry,
from `Target_files.orders` o
join `Target_files.customers` c
on o.customer_id = c.customer_id
join `Target_files.order_items` oi
on oi.order_id = o.order_id) tbl1
group by customer_state
order by avg_time_to_delivery
limit 5
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	avg_time_to_del
1	SP	8.25960855...
2	PR	11.4807930...
3	MG	11.5155221...
4	DF	12.5014861...
5	SC	14.5209858...

Code(Highest):

```
select customer_state,
avg(tbl1.time_to_delivery) as avg_time_to_delivery
from
(select c.customer_state,
abs(date_diff(order_delivered_customer_date, order_purchase_timestamp, day)) as time_to_delive
ry,
from `Target_files.orders` o
join `Target_files.customers` c
on o.customer_id = c.customer_id
join `Target_files.order_items` oi
on oi.order_id = o.order_id) tbl1
group by customer_state
order by avg_time_to_delivery desc
limit 5
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	avg_time_to_deli
1	RR	27.8260869...
2	AP	27.7530864...
3	AM	25.9631901...
4	AL	23.9929742...
5	PA	23.3017077...

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

Code(Fast):

```
select customer_state,
avg(tbl1.diff_estimated_delivery) as avg_diff_estimated_delivery
from
(select c.customer_state, freight_value,
abs(date_diff(order_estimated_delivery_date,order_delivered_customer_date, day)) as diff_estim
ated_delivery
from `Target_files.orders` o
join `Target_files.customers` c
on o.customer_id = c.customer_id
join `Target_files.order_items` oi
on oi.order_id = o.order_id) tbl1
group by customer_state
order by avg_diff_estimated_delivery desc
limit 5
```

JOB INFORMATION | RESULTS | JSON | EXECUTION

Row	customer_state	avg_diff_estimated_delivery
1	RR	25.347826086956527
2	AP	24.58024691358025
3	AC	21.241758241758237
4	AM	20.472392638036805
5	RO	19.600732600732595

Code(Not so Fast):

```
select customer_state,
avg(tbl1.diff_estimated_delivery) as avg_diff_estimated_delivery
from
(select c.customer_state, freight_value,
abs(date_diff(order_estimated_delivery_date,order_delivered_customer_date, day)) as diff_estim
ated_delivery
from `Target_files.orders` o
join `Target_files.customers` c
on o.customer_id = c.customer_id
join `Target_files.order_items` oi
on oi.order_id = o.order_id) tbl1
group by customer_state
order by avg_diff_estimated_delivery
limit 5
```

JOB INFORMATION | RESULTS | JSON

Row	customer_state	avg_diff_estimat
1	SP	10.9855737...
2	MS	11.7731196...
3	AL	12.0608899...
4	ES	12.0885393...
5	SC	12.1178623...

6. Payment type analysis:

6.1 Month over Month count of orders for different payment types

Code:

```
select payment_type, count(tbl1.month_name) as count_month
from
(select FORMAT_DATETIME("%B", DATETIME(order_purchase_timestamp)) as month_name,
payment_type
from `Target_files.payments` p
join `Target_files.orders` o
on p.order_id = o.order_id) tbl1
group by payment_type
```

JOB INFORMATION | RESULTS | JSON

Row	payment_type	count_month
1	credit_card	76795
2	voucher	5775
3	not_defined	3
4	debit_card	1529
5	UPI	19784

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

Code:

```
select payment_installments, count(o.order_id) as no_of_order
from `Target_files.payments` p
join `Target_files.orders`  o
on p.order_id = o.order_id
group by payment_installments
order by payment_installments
```

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

Row	payment_install	no_of_order
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

Results per page: 100

Key Points:

1. Credit Card is used by most customers in terms of payment methods.

2. Highest no. of customers order in Afternoon

3. 'RR' Customer State has the highest average freight value in comparison to their delivery rate

4. Customer give maximum count of orders with less payment installments

5. Order count is more in November, January, March. These are the peak seasons since most of the Brazilian festivals like Carnival, Bonfim Stairs Washing happens around these months.

6. States like 'RJ', 'DF' and 'RS' have an average freight value but the time to delivery and the difference between estimated delivery time are the same. (Order is getting delivered on time)

Row	customer_state	avg_freight_valu	avg_time_to_deli	avg_diff_estimat
7	RJ	20.9609239...	14.6893821...	14.2255054...
8	DF	21.0413549...	12.5014861...	12.2199575...
9	RS	21.7358043...	14.7082993...	14.4588292...

7. Sales drastically increased from 2018 to 2017 if calculated for the data between January to August for every year

Row	Percentage_Increase
1	137.0

8. More than 90% of products were getting delivered before the estimated date since on comparison their estimated date is more than the delivery date.