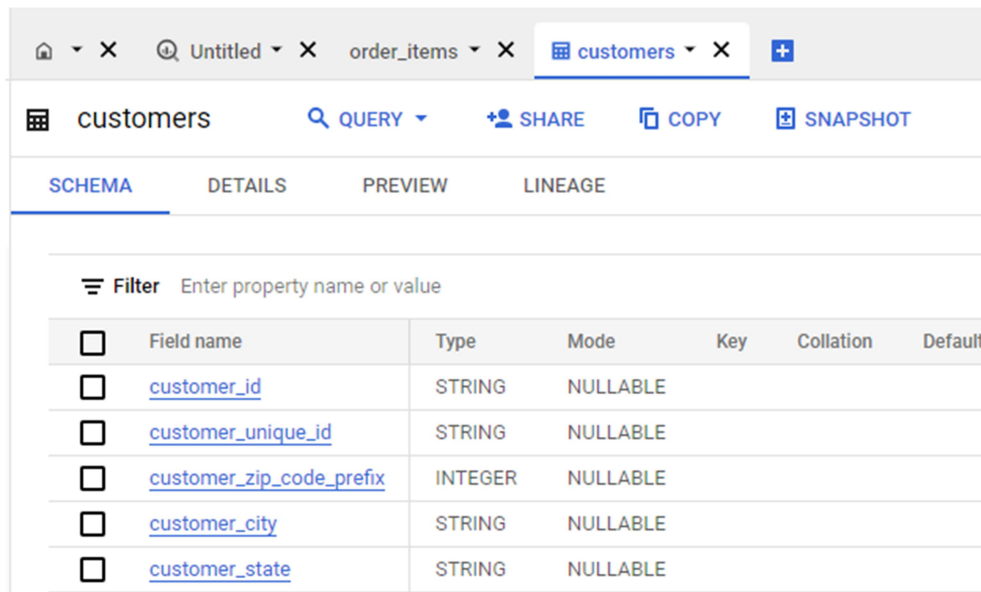


Business Case: Target SQL

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

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

Answer :



The screenshot shows a database interface with a tab for 'customers'. Below the tab, there are options for 'QUERY', 'SHARE', 'COPY', and 'SNAPSHOT'. The 'SCHEMA' tab is selected, displaying a table with columns: Field name, Type, Mode, Key, Collation, and Default. The table lists five columns: customer_id (STRING, NULLABLE), customer_unique_id (STRING, NULLABLE), customer_zip_code_prefix (INTEGER, NULLABLE), customer_city (STRING, NULLABLE), and customer_state (STRING, NULLABLE).

Field name	Type	Mode	Key	Collation	Default
customer_id	STRING	NULLABLE			
customer_unique_id	STRING	NULLABLE			
customer_zip_code_prefix	INTEGER	NULLABLE			
customer_city	STRING	NULLABLE			
customer_state	STRING	NULLABLE			

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

Answer :

```
select min(order_purchase_timestamp) as start_date,  
max(order_purchase_timestamp) as end_date  
from `Target_project.orders`;
```

Row	start_date	end_date
1	2016-09-04 21:15:19 UTC	2018-10-17 17:30:18 UTC

So, the order placed between **2016-09-04 21:15:19 UTC** and **2018-10-17 17:30:18 UTC**.

3. Count the Cities & States of customers who ordered during the given period.

Answer :

```
select
count(distinct customer_city) as customer_city,
count(distinct customer_state) as customer_state
from `Target_project.orders` o
join `Target_project.customers` c on o.customer_id = c.customer_id;
```

Row	customer_city	customer_state
1	4119	27

So, 4119 are customers cities and 27 are customers states.

2. In-depth Exploration:

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

Answer :

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

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

Monthly Order Peaks: Highest Orders: November 2017 (7544 orders)
Lowest Orders: December 2016 (1 order)

```
select extract(year from order_purchase_timestamp) as year,  
count(order_id) as no_of_orders  
from `Target_project.orders`  
group by year  
order by year;
```

Yearly Comparison:

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



From the yearly comparison, we can observe that there was a significant increase in orders from 2016 to 2017, and there was another increase from 2017 to early 2018. However, there is a drastic drop in orders for September and October 2018.

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

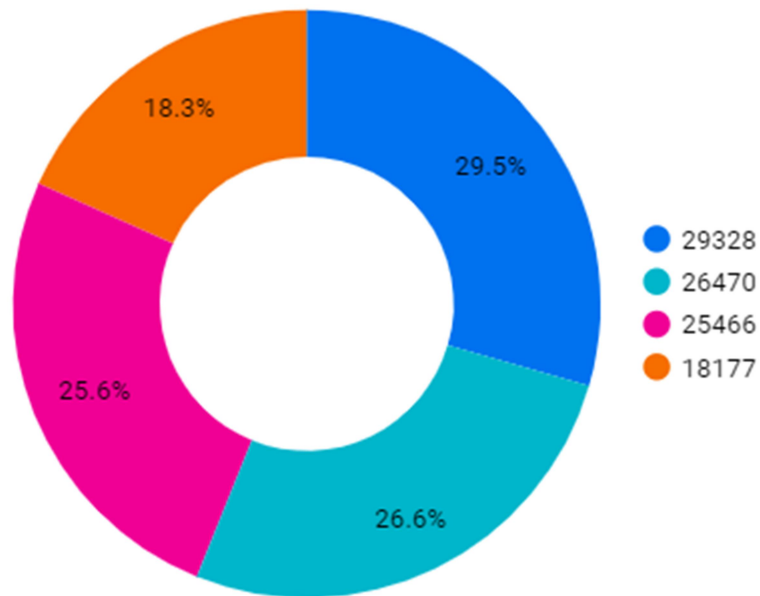
Answer :

```
select
(case when a.qrtr= 1 then 'January-March' when a.qrtr= 2 then 'April-
June' when a.qrtr= 3 then 'July-September' when a.qrtr= 4 then
'October-December' end) quarter,
sum(a.orders) as no_of_orders
from
(
select
ntile(4) over(order by t.month) as qrtr, t.orders
from
(
select
extract(month from order_purchase_timestamp) as month,
count(order_id) as orders,
from `Target_project.orders`
group by month
) t
) a
group by 1
order by 2 desc;
```

Row	quarter ▼	no_of_orders ▼
1	April-June	29328
2	January-March	26470
3	July-September	25466
4	October-December	18177

Monthly Order Peaks: Highest Orders: April-June (29328 orders)

Lowest Orders: October-December (18177 order)



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

- a. 0-6 hrs : Dawn
- b. 7-12 hrs : Mornings
- c. 13-18 hrs : Afternoon
- d. 19-23 hrs : Night

Answer :

```
select sum(a.order_cnt) as total_orders,
Sum(Case When a.hour between 0 and 6
      Then a.order_cnt Else 0 End) Dawn,
Sum(Case When a.hour between 7 and 12
      Then a.order_cnt Else 0 End) Morning,
Sum(Case When a.hour between 13 and 18
      Then a.order_cnt Else 0 End) Afternoon,
Sum(Case When a.hour between 19 and 23
      Then a.order_cnt Else 0 End) Night
from
(
select t.hour, count(*) as order_cnt,
from
(
select extract(hour FROM order_purchase_timestamp) as hour
from `Target_project.orders`
) t
group by t.hour
```

```
order by t.hour
) a;
```

Row	total_orders	Dawn	Morning	Afternoon	Night
1	99441	5242	27733	38135	28331

So, the Brazilian customers mostly place their orders in **Afternoon(13-18 hrs)**

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

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

Answer :

```
select a.year,a.month,a.customer_state,a.order_cnt
from
(
select  t.customer_state,t.year, t.month,
count(t.order_purchase_timestamp) over(partition by t.customer_state)
as order_cnt,
from
(
select order_purchase_timestamp,customer_state,
extract(year FROM order_purchase_timestamp) as year,
extract(month FROM order_purchase_timestamp) as month
from `Target_project.orders` o left join `Target_project.customers` c
on o.customer_id = c.customer_id
) t
group by t.customer_state,t.month,t.order_purchase_timestamp,t.year
) a
group by a.year,a.month,a.customer_state,a.order_cnt
order by a.year, a.month;
```

Row	year ▼	month ▼	customer_state ▼	order_cnt ▼
1	2016	9	SP	41560
2	2016	9	RR	46
3	2016	9	RS	5450
4	2016	10	RJ	12809
5	2016	10	MT	903
6	2016	10	RN	485
7	2016	10	BA	3367
8	2016	10	RR	46
9	2016	10	SE	350
10	2016	10	AL	411

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

Answer :

```
select t.customer_state, t.customer
from
(
select customer_state, count(customer_id) over(partition by
customer_state) as customer from `Target_project.customers`
) t
group by t.customer_state, t.customer
order by t.customer desc;
```

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

State Customers Peaks: Maximum customers: SP (41746 customers)
Minimum customers: RR (46 customers)

4. 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). You can use the "payment_value" column in the payments table to get the cost of orders.

Answer :

```
select b.year, b.order_cost,
round(((lead(b.order_cost,1) over(order by b.order_cost) -
b.order_cost)/b.order_cost)*100) as percnt_change
from
(
select a.year, a.order_cost
from
(
select t.payment_value, t.year, t.month,
sum(t.payment_value) over(partition by t.year) as order_cost
from
(
select p.payment_value,
extract(year FROM order_purchase_timestamp) as year,
extract(month FROM order_purchase_timestamp) as month
from `Target_project.customers` c
join `Target_project.orders` o on c.customer_id = o.customer_id
join `Target_project.payments` p on o.order_id = p.order_id
) t
where (t.year = 2017 or t.year = 2018) and t.month between 1 and 8
) a
group by a.year, a.order_cost
) b
order by b.year;
```

Row	year	order_cost	percnt_change
1	2017	3669022.12	137.0
2	2018	8694733.84	null

So, the percent increase in the cost of orders from year 2017 to 2018 is 137%

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

Answer :

```
select t.customer_state, t.total_value, t.avg_value
from
(
select c.customer_state, p.payment_value,
sum( p.payment_value) over(partition by c.customer_state) as
total_value,
round(avg( p.payment_value) over(partition by c.customer_state),2) as
avg_value
from `Target_project.customers` c
join `Target_project.orders` o on c.customer_id = o.customer_id
join `Target_project.payments` p on o.order_id = p.order_id
) t
group by t.customer_state, t.total_value, t.avg_value
order by t.total_value desc, t.avg_value desc;
```

Row	customer_state	total_value	avg_value
1	SP	5998226.96	137.5
2	RJ	2144379.69	158.53
3	MG	1872257.26	154.71
4	RS	890898.54	157.18
5	PR	811156.38	154.15
6	SC	623086.43	165.98
7	BA	616645.82	170.82
8	DF	355141.08	161.13
9	GO	350092.31	165.76
10	ES	325967.55	154.71

So, Maximum total value of order price is 5998226.96 for SP state and,
Minimum total value of order price is 10064.62 for RR state and,
Maximum avg value of order price is 248.33 for PB state and,
Minimum avg value of order price is 137.5 for SP state

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

Answer :

```
select t.customer_state, t.total_value, t.avg_value
from
(
select c.customer_state, i.freight_value,
round(sum( i.freight_value) over(partition by c.customer_state),2) as
total_value,
round(avg( i.freight_value) over(partition by c.customer_state),2) as
avg_value
from `Target_project.customers` c
join `Target_project.orders` o on c.customer_id = o.customer_id
join `Target_project.order_items` i on o.order_id = i.order_id
) t
group by t.customer_state, t.total_value, t.avg_value
order by t.total_value desc, t.avg_value desc;
```

Row	customer_state	total_value	avg_value
1	SP	718723.07	15.15
2	RJ	305589.31	20.96
3	MG	270853.46	20.63
4	RS	135522.74	21.74
5	PR	117851.68	20.53
6	BA	100156.68	26.36
7	SC	89660.26	21.47
8	PE	59449.66	32.92
9	GO	53114.98	22.77
10	DF	50625.5	21.04

So, Maximum total value of order freight is 718723.07 for SP state and,
Minimum total value of order freight is 2235.19 for RR state and,
Maximum avg value of order freight is 42.98 for RR state and,
Minimum avg value of order freight is 15.15 for SP state

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.

Also, calculate the difference (in days) between the estimated & actual

delivery date of an order.

Do this in a single query.

You can calculate the delivery time and the difference between the estimated & actual delivery date using the given formula:

- **time_to_deliver = order_delivered_customer_date - order_purchase_timestamp**
- **diff_estimated_delivery = order_estimated_delivery_date - order_delivered_customer_date**

Answer :

```
select order_id, t.time_to_deliver, t.diff_estimated_delivery
from
(
select order_id, order_purchase_timestamp, order_status,
order_delivered_customer_date, order_estimated_delivery_date,
datetime_diff(order_delivered_customer_date, order_purchase_timestamp,
day) as time_to_deliver,
datetime_diff(order_delivered_customer_date,
order_estimated_delivery_date, day) as diff_estimated_delivery
from `Target_project.orders`
where order_status = 'delivered'
and order_delivered_customer_date is not null
) t
order by t.time_to_deliver desc , t.diff_estimated_delivery desc;
```

Row	order_id	time_to_deliver	diff_estimated_delivery
1	ca07593549f1816d26a572e06...	209	181
2	1b3190b2dfa9d789e1f14c05b...	208	188
3	440d0d17af552815d15a9e41a...	195	165
4	285ab9426d6982034523a855f...	194	166
5	0f4519c5f1c541ddec9f21b3bd...	194	161
6	2fb597c2f772eca01b1f5c561b...	194	155
7	47b40429ed8cce3aee9199792...	191	175
8	2fe324feb907e3ea3f2aa9650...	189	167
9	2d7561026d542c8dbd8f0daea...	188	159
10	c27815f7e3dd0b926b5855262...	187	162

Here, difference in estimated delivery column (-) value refers early delivery of the order than expected and (+) value refers delay in delivery of the order than expected in days.

In time to deliver column, '0' means order delivered within 24 hours so considered as 0 day.

So we can clearly see that in row 1, order is delivered in 209 days to the customer instead of 28 days but unfortunately it takes 181 more days to reach the customer.

"So, Seller has to tell the approximate ETA of the order to the customers so that customers will order accordingly and also goodwill of the seller is maintained."

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

Answer :

```
select t.customer_state, t.avg_freight_value
from
(
select c.customer_state, i.freight_value,
round(avg( i.freight_value) over(partition by c.customer_state),2) as
avg_freight_value
from `Target_project.customers` c
join `Target_project.orders` o on c.customer_id = o.customer_id
join `Target_project.order_items` i on o.order_id = i.order_id
) t
group by t.customer_state, t.avg_freight_value
order by t.avg_freight_value desc
limit 5;
```

States with Highest average freight value

Row	customer_state	avg_freight_value
1	RR	42.98
2	PB	42.72
3	RO	41.07
4	AC	40.07
5	PI	39.15

```
select t.customer_state, t.avg_freight_value
from
(
select c.customer_state, i.freight_value,
round(avg( i.freight_value) over(partition by c.customer_state),2) as
avg_freight_value
from `Target_project.customers` c
```

```

join `Target_project.orders` o on c.customer_id = o.customer_id
join `Target_project.order_items` i on o.order_id = i.order_id
) t
group by t.customer_state, t.avg_freight_value
order by t.avg_freight_value
limit 5;

```

States with Lowest average freight value

Row	customer_state	avg_freight_value
1	SP	15.15
2	PR	20.53
3	MG	20.63
4	RJ	20.96
5	DF	21.04

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

Answer :

```

select t.customer_state, round(avg(t.time_to_deliver),2) as
avg_dlvry_time
from
(
select customer_state, order_purchase_timestamp, order_status,
order_delivered_customer_date,
datetime_diff(order_delivered_customer_date, order_purchase_timestamp,
day) as time_to_deliver
from `Target_project.customers` c
join `Target_project.orders` o on c.customer_id = o.customer_id
where order_status = 'delivered'
and order_delivered_customer_date is not null
) t
group by t.customer_state
order by 2 desc
limit 5;

```

States with Highest average delivery time in days

Row	customer_state	avg_dlvry_time
1	RR	28.98
2	AP	26.73
3	AM	25.99
4	AL	24.04
5	PA	23.32

```
select t.customer_state, round(avg(t.time_to_deliver),2) as
avg_dlvry_time
from
(
select customer_state, order_purchase_timestamp, order_status,
order_delivered_customer_date,
datetime_diff(order_delivered_customer_date, order_purchase_timestamp,
day) as time_to_deliver
from `Target_project.customers` c
join `Target_project.orders` o on c.customer_id = o.customer_id
where order_status = 'delivered'
and order_delivered_customer_date is not null
) t
group by t.customer_state
order by 2
limit 5;
```

States with Lowest average delivery time in days

Row	customer_state	avg_dlvry_time
1	SP	8.3
2	PR	11.53
3	MG	11.54
4	DF	12.51
5	SC	14.48

- 4. Find out the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.**
You can use the difference between the averages of actual & estimated delivery date to figure out how fast the delivery was for each state.

Answer :

```
select a.customer_state, a.avg_actual_time, a.avg_estmt_time,
a.delivery_early_days,
round((a.delivery_early_days/a.avg_actual_time)*100) as
delivery_early_prnt
from
(
select t.customer_state, round(avg(t.time_to_deliver),2) as
avg_actual_time, round(avg(t.estimated_delivery_time),2) as
avg_estmt_time, round(avg(t.time_to_deliver) -
avg(t.estimated_delivery_time ),2) as delivery_early_days
from
(
select customer_state, order_purchase_timestamp,
order_status,order_estimated_delivery_date,
order_delivered_customer_date,
datetime_diff(order_delivered_customer_date, order_purchase_timestamp,
day) as time_to_deliver,
datetime_diff(order_estimated_delivery_date,order_purchase_timestamp,
day) as estimated_delivery_time
from `Target_project.customers` c
join `Target_project.orders` o on c.customer_id = o.customer_id
where order_status = 'delivered'
and order_delivered_customer_date is not null
) t
group by t.customer_state
) a
order by 5
limit 5;
```

Row	customer_state	avg_actual_time	avg_estmt_time	delivery_early_days	delivery_early_prnt
1	SP	8.3	18.78	-10.48	-126.0
2	MG	11.54	24.19	-12.65	-110.0
3	PR	11.53	24.25	-12.73	-110.0
4	RO	18.91	38.39	-19.47	-103.0
5	AC	20.64	40.72	-20.09	-97.0

So, these are the top 5 states where the order delivery is really fast as compared to the estimated date of delivery.

Here delivery early percent signifies the change is percent to get how fast delivery is as compared to actual delivery time and (-) value refers early delivery than expected.

6. Analysis based on the payments:

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

Answer :

```
select extract(month from o.order_purchase_timestamp) as month,
count(distinct o.order_id) as orders, p.payment_type
from `Target_project.orders` o
join `Target_project.payments` p on o.order_id = p.order_id
group by month, payment_type
order by month;
```

Row	month	orders	payment_type
1	1	6093	credit_card
2	1	1715	UPI
3	1	337	voucher
4	1	118	debit_card
5	2	1723	UPI
6	2	6582	credit_card
7	2	288	voucher
8	2	82	debit_card
9	3	7682	credit_card
10	3	1942	UPI

Here, different types are credit card, voucher, debit card and UPI.

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

Answer :

```
select payment_installments, count(distinct order_id) as no_of_orders
from `Target_project.payments`
where payment_installments > 0
group by payment_installments
order by 2 desc;
```


Row	payment_installment	no_of_orders ▼
1	1	49060
2	2	12389
3	3	10443
4	4	7088
5	5	5234
6	6	3916
7	7	1623
8	8	4253
9	9	644
10	10	5315

So 50% of the orders, where customers like to purchase the orders on the single payment installments.

Prepared By-
Shubhaditya Kumar
(DSML June 23'batch)