1. **Data type of columns in a table**
2. Customers.csv-

Column Name	Data type
Customer_id	TEXT
customer_unique_id	TEXT
customer_zip_code_prefix	TEXT
customer_city	TEXT
customer_state	TEXT

3. Sellers.csv

Column Name	Data type
seller_id	TEXT
seller_zip_code_prefix	TEXT
seller_city	TEXT
seller_state	TEXT

4. Order_items.csv

Column Name	Data type
order_id	TEXT
order_item_id	INT
product_id	TEXT
seller_id	TEXT
shipping_limit_date	TEXT
price	DOUBLE
freight_value	DOUBLE

5. Geolocations.csv

Column Name	Data Type
geolocation_zip_code_prefix	TEXT
geolocation_lat	DOUBLE
geolocation_lng	DOUBLE
geolocation_city	TEXT
geolocation_state	TEXT

6. Payments.csv

Column Name	Data Type
order_id	TEXT
payment_sequential	INT
payment_type	TEXT
payment_installments	INT
payment_value	DOUBLE

7. Orders.csv

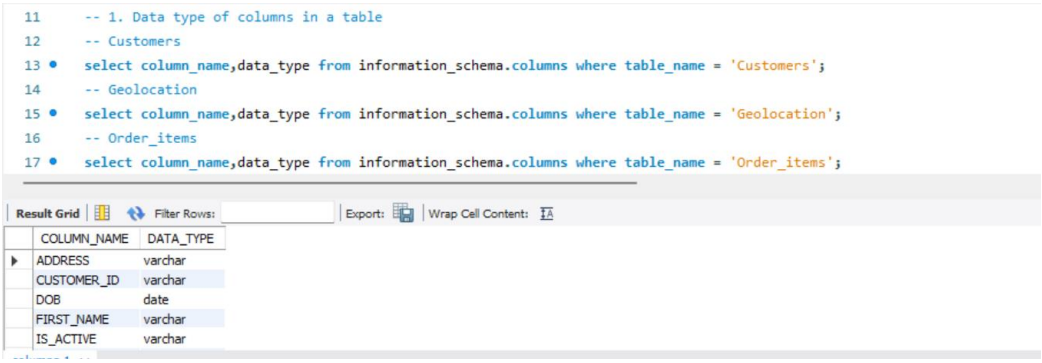
Column Name	Data Type
order_id	TEXT
customer_id	TEXT
order_status	TEXT
order_purchase_timestamp	TEXT
order_delivered_carrier_date	TEXT
order_delivered_customer_date	TEXT
order_estimated_delivery_date	TEXT

8. Reviews.csv

Column Name	Data Type
review_id	TEXT
order_id	TEXT
review_score	INT
review_comment_title	TEXT
review_comment_message	TEXT
review_creation_date	TEXT
review_answer_timestamp	TEXT

9. Products.csv

Column Name	Data Type
product_id	TEXT
product_category_name	TEXT
product_name_lenght	INT
product_description_lenght	INT
product_photos_qty	INT
product_weight_g	INT
product_length_cm	INT
product_height_cm	INT
product_width_cm	INT



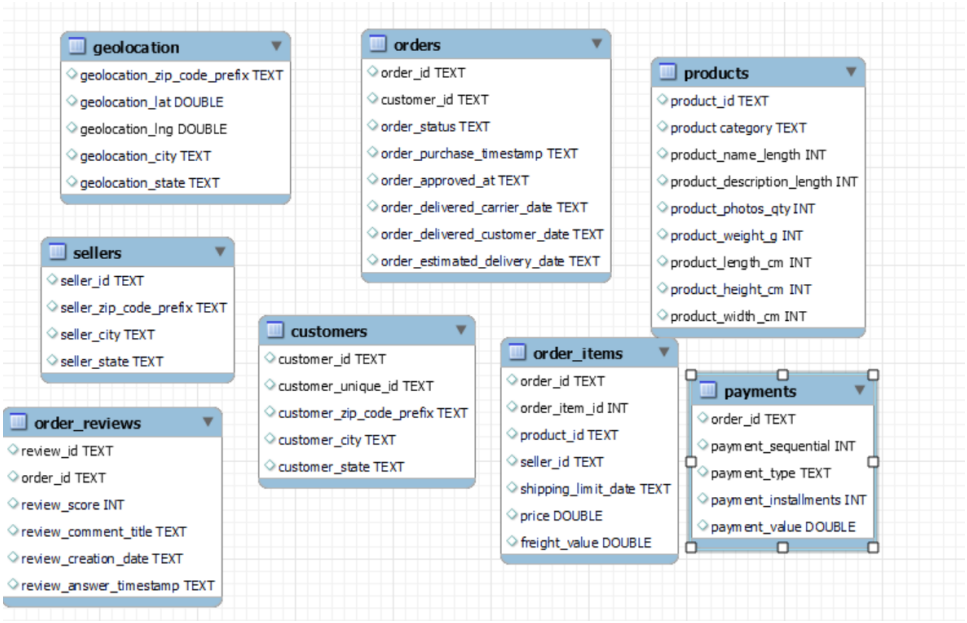


Fig1. Snip of ER Diagram

1. **Time period for which the data is given.**

```
10
11 •  select min(order_purchase_timestamp) as min_date , max(order_purchase_timestamp) as max_date from orders;
12
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

min_date	max_date
2016-09-04 21:15:19	2018-10-03 18:55:29

2. **Cities and States of customers ordered during the given period.**

```
14 •  select distinct geolocation_city,geolocation_state from geolocation;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

geolocation_city	geolocation_state
sao paulo	SP
jacarei	SP
goiania	GO
macae	RJ
curitiba	PR
sao jose dos campos	SP

-- 1.

-- 1.1 Data type of columns in a table

-- Customers

```
select column_name,data_type from information_schema.columns where
table_name = 'Customers';

-- Geolocation

select column_name,data_type from information_schema.columns where
table_name = 'Geolocation';

-- Order_items

select column_name,data_type from information_schema.columns where
table_name = 'Order_items';

-- order_reviews

select column_name,data_type from information_schema.columns where
table_name = 'order_reviews';

-- orders

select column_name,data_type from information_schema.columns where
table_name = 'orders';

-- payments

select column_name,data_type from information_schema.columns where
table_name = 'payments';

-- products

select column_name,data_type from information_schema.columns where
table_name = 'products';

-- sellers

select column_name,data_type from information_schema.columns where
table_name = 'sellers';



-- 1.2. Time period for which the data is given.

select min(order_purchase_timestamp) as min_date,
max(order_purchase_timestamp) as max_date from orders;



-- 1.3 Cities and states of customers ordered during the given period
```

select count(distinct(geolocation_city)) as city,count(distinct(geolocation_state))as state from geolocation;

select distinct(geolocation_city) as city, (geolocation_state)as state from geolocation;

2. **In-depth Exploration:**
    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?**

```
37    -- 2
38    -- 2.1 Growing trend in Brazil
39    SELECT Extract(YEAR from order_purchase_timestamp) as year,
40     Extract(Month from order_purchase_timestamp) as month, count(1) as
41    SalesCount FROM orders GROUP BY year,month
42    order by year asc,month asc ;
43
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	month	SalesCount
2016	9	4
2016	10	189
2016	12	1
2017	1	467
2017	2	1047

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

Brazillians shop more in Evening.

```
43    select count(distinct order_id) as order_number,
44    case
45    when date_format(order_purchase_timestamp, '%H:%i,%s') between '00:00:00' and '05:59:59' then 'Dawn '
46    when date_format(order_purchase_timestamp , '%H:%i,%s') between '06:00:00' and '11:59:59' then 'Morning '
47    when date_format(order_purchase_timestamp , '%H:%i,%s') between '12:00:00' and '18:59:59' then 'Evening '
48    when date_format(order_purchase_timestamp , '%H:%i,%s') between '19:00:00' and '23:59:59' then 'Night '
49    end as Phase_of_day from orders group by Phase_of_day order by order_number desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_number	Phase_of_day
26412	Evening
16980	Night
13510	Morning
2824	Dawn
162	NULL

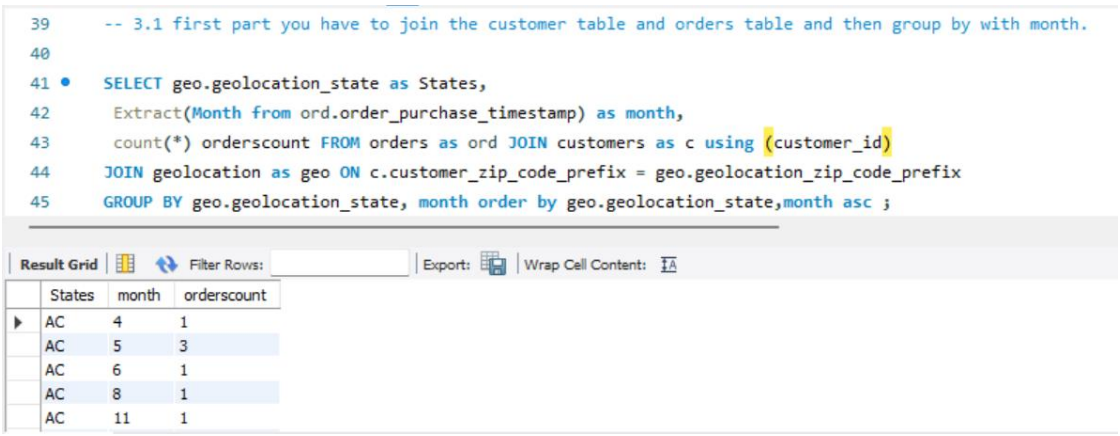
-- 2

-- 2.1 Growing trend in Brazil

SELECT Extract(YEAR from order_purchase_timestamp) as year,

```
 Extract(Month from order_purchase_timestamp) as month, count(1) as

SalesCount FROM orders GROUP BY year,month

order by year asc,month asc ;



-- 2.2 Time when Brazillian cutomers tend to buy

select count(distinct order_id) as order_number,

case

when date_format(order_purchase_timestamp, '%H:%i,%s') between
'00:00:00' and '05:59:59' then 'Dawn '

when date_format(order_purchase_timestamp , '%H:%i,%s') between
'06:00:00' and '11:59:59' then 'Morning '

when date_format(order_purchase_timestamp , '%H:%i,%s') between
'12:00:00' and '18:59:59' then 'Evening '

when date_format(order_purchase_timestamp , '%H:%i,%s') between
'19:00:00' and '23:59:59' then 'Night '

end as Phase_of_day from orders group by Phase_of_day order by
order_number desc;

-- Therefore, Brazillians shop more in Evening.
```

3. **Evolution of E-commerce orders in the Brazil region:**
    1. **Get month on month orders by states**



```
39    -- 3.1 first part you have to join the customer table and orders table and then group by with month.
40
41 •  SELECT geo.geolocation_state as States,
42     Extract(Month from ord.order_purchase_timestamp) as month,
43     count(*) orderscount FROM orders as ord JOIN customers as c using (customer_id)
44    JOIN geolocation as geo ON c.customer_zip_code_prefix = geo.geolocation_zip_code_prefix
45    GROUP BY geo.geolocation_state, month order by geo.geolocation_state,month asc ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

States	month	orderscount
AC	4	1
AC	5	3
AC	6	1
AC	8	1
AC	11	1

2. **Distribution of customers across the states in Brazil**

```
47      -- 3.2 Distribution of customers across the states in Brazil
48  •   SELECT customer_state,count(*) customercount
49      FROM customers
50      GROUP BY customer_state
51      order by customercount desc;
52
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_state	customercount
SP	41746
RJ	12852
MG	11635
RS	5466
PR	5045

-- 3

-- 3.1 first part you have to join the customer table and orders table and then group by with month.

SELECT geo.geolocation_state as States,

Extract(Month from ord.order_purchase_timestamp) as month,

count(*) orderscount FROM orders as ord JOIN customers as c using (customer_id)

JOIN geolocation as geo ON c.customer_zip_code_prefix = geo.geolocation_zip_code_prefix

GROUP BY geo.geolocation_state, month order by geo.geolocation_state,month asc ;

-- 3.2 Distribution of customers across the states in Brazil

SELECT customer_state,count(*) customercount

FROM customers

GROUP BY customer_state

order by customercount desc;

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

```
39 • with S1 as (
40     SELECT Round(Sum(pay.payment_value),2) as sum2017
41     FROM orders as ord
42     JOIN payments as pay using (order_id) Where Extract(Year from ord.order_purchase_timestamp) = 2017
43    and Extract(Month from ord.order_purchase_timestamp) BETWEEN 1 and 8
44     ),
45   S2 as (
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Sumof2018	Sumof2017	increaseValue
16113.67	6903.67	133.41

    2. **Mean & Sum of price and freight value by customer state**

```
57 • SELECT cust.customer_state,round(avg(price),2) as priceaverage ,
58   round(avg(freight_value),2) as avgfreightvalue,
59    round(sum(price),2) as sumprice , round(sum(freight_value),2) as
60   sumfreightvalue
61   FROM order_items as ord
62   JOIN customers as cust ON sel.seller_zip_code_prefix = cust.customer_zip_code_prefix
63   Group BY cust.customer_state;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_state	priceaverage	avgfreightvalue	sumprice	sumfreightvalue
SP	67.21	10.55	4771.84	748.87
PR	165.5	15.8	331	31.6

-- 4

-- 4.1 % increase in cost of orders from 2017 to 2018

with S1 as (

SELECT Round(Sum(pay.payment_value),2) as sum2017

FROM orders as ord

JOIN payments as pay using (order_id)

Where Extract(Year from ord.order_purchase_timestamp) = 2017

```
 and Extract(Month from ord.order_purchase_timestamp) BETWEEN 1 and 8

),

S2 as (

 SELECT Round(Sum(pay.payment_value),2) as sum2018

 FROM orders as ord

 JOIN payments as pay using (order_id)

 Where Extract(Year from ord.order_purchase_timestamp) = 2018

 and Extract(Month from ord.order_purchase_timestamp) BETWEEN 1 and 8

)

Select sum2018 as Sumof2018,sum2017 as Sumof2017, Round((sum2018-

sum2017)/sum2017*100,2) as increaseValue from S1,S2;


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

SELECT cust.customer_state,round(avg(price),2) as priceaverage ,

round(avg(freight_value),2) as avgfreightvalue,

 round(sum(price),2) as sumprice , round(sum(freight_value),2) as

sumfreightvalue

FROM order_items as ord

JOIN customers as cust ON sel.seller_zip_code_prefix =
cust.customer_zip_code_prefix

Group BY cust.customer_state;
```

5. Analysis on sales, freight and delivery time

1. **Calculate days between purchasing, delivering and estimated delivery**

```sql
-- 5
-- 5.1 .Days between purchasing, delivering and estimated delivery
SELECT order_id,order_purchase_timestamp as purchasetime,
 DATEDIFF(Extract(Date FROM order_delivered_customer_date),Extract(Date FROM
order_purchase_timestamp) ,Day) as Day_between_deliver_purchase,
 DATEDIFF(Extract(Date FROM order_estimated_delivery_date),Extract(Date FROM
order_purchase_timestamp),Day) as Day_between_exstimate_purchase,
 DATEDIFF(Extract(Date FROM order_estimated_delivery_date),Extract(Date FROM
order_delivered_customer_date),Day) as Day_between_estimated_delivery,
FROM orders
WHERE order_delivered_customer_date IS NOT NULL;
```

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**

```sql
-- 5.2 days between purchasing, delivering and estimated delivery
SELECT order_id,
 TIMESTAMPDIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) AS
time_to_delivery,
abs(TIMESTAMPDIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY))
AS diff_estimated_delivery;
```

3. **Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery**

```sql
122  -- 5.3  Group data by state, take mean of freight_value, time_to_delivery, diff_estimated_delivery
123  SELECT cust.customer_state as tstate,round(avg(ord.freight_value),2) as
124  avg_freighttop
125   FROM order_items as ord
126   JOIN sellers as sel Using(seller_id)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

tstate	avg_freighttop
PR	15.8
SP	10.55

4. **Sort the data to get the following:**
   - **Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5**

```
-- 5.4
-- 5.4 a) Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
SELECT cust.customer_state,
round(avg(TIMESTAMPDIFF(order_delivered_customer_date,order_purchase_timestamp,DAY
)),2) AS time_to_delivery
FROM orders as ord
JOIN customers as cust
ON ord.customer_id = cust.customer_id
Group BY cust.customer_state
order by time_to_delivery desc limit 5;
```

- **Top 5 states with highest/lowest average time to delivery**

```
-- 5.4 b) Top 5 states with highest average time to delivery
SELECT cust.customer_state,
round(avg(TIMESTAMPDIFF(order_delivered_customer_date,order_purchase_timestamp,DAY)),
2) AS time_to_delivery
FROM orders as ord
JOIN customers as cust
ON ord.customer_id = cust.customer_id
Group BY cust.customer_state
order by time_to_delivery asc limit 5;
```

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

```
-- 5.4 c) Top 5 states with highest average time to delivery
SELECT cust.customer_state,
avg(ABS(TIMESTAMPDIFF(order_delivered_customer_date,order_estimated_delivery_date,DAY
))) AS diff_estimated_delivery
FROM orders as od
JOIN order_items as ord using(order_id)
JOIN sellers as sel Using(seller_id)
JOIN customers as cust
ON sel.seller_zip_code_prefix = cust.customer_zip_code_prefix
Where order_status ="delivered"
and order_delivered_customer_date is not null
Group BY cust.customer_state order by diff_estimated_delivery asc limit 5;
```

-- 5
-- 5.1 .Days between purchasing, delivering and estimated delivery
SELECT order_id,order_purchase_timestamp as purchasetime,
DATEDIFF(Extract(Date FROM order_delivered_customer_date),Extract(Date FROM order_purchase_timestamp) ,Day) as Day_between_deliver_purchase,
DATEDIFF(Extract(Date FROM order_estimated_delivery_date),Extract(Date FROM order_purchase_timestamp),Day) as Day_between_exstimate_purchase,
DATEDIFF(Extract(Date FROM order_estimated_delivery_date),Extract(Date FROM order_delivered_customer_date),Day) as Day_between_estimated_delivery,

```
FROM orders
WHERE order_delivered_customer_date IS NOT NULL;


-- 5.2 days between purchasing, delivering and estimated delivery
SELECT order_id,
 TIMESTAMPDIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) AS
time_to_delivery,
abs(TIMESTAMPDIFF(order_estimated_delivery_date,order_delivered_customer_date,DA
Y))
AS diff_estimated_delivery;


-- 5.2 time_to_delivery & diff_estimated_delivery
With S1 as(
 SELECT order_id,
TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_timestamp,DAY) AS
time_to_delivery,
TIMESTAMPDIFF(order_estimated_delivery_date,order_delivered_customer_date,DAY) AS
diff_estimated_delivery,
 FROM orders
 WHERE order_delivered_customer_date IS NOT NULL
)
SELECT cust.customer_state, round(avg(ord.freight_value),2) as
average_freight_value,
 round(avg(OD.time_to_delivery),2) as average_time_to_delivery,
 round(avg(OD.diff_estimated_delivery),2) as average_diff_estimated_delivery
FROM order_items as ord
JOIN sellers as sel Using(seller_id)
JOIN customers as cust
ON sel.seller_zip_code_prefix = cust.customer_zip_code_prefix
JOIN S1 as OD
ON ord.order_id = OD.order_id
Group BY cust.customer_state;

-- 5.3     Group data by state, take mean of freight_value, time_to_delivery,
diff_estimated_delivery
SELECT cust.customer_state as tstate,round(avg(ord.freight_value),2) as
avg_freighttop
 FROM order_items as ord
 JOIN sellers as sel Using(seller_id)
 JOIN customers as cust
 ON sel.seller_zip_code_prefix = cust.customer_zip_code_prefix
 Group BY cust.customer_state
 order by avg_freighttop desc;

-- 5.4
-- 5.4 a) Top 5 states with highest/lowest average freight value - sort in desc/asc limit 5
```

```
SELECT cust.customer_state,
round(avg(TIMESTAMPDIFF(order_delivered_customer_date,order_purchase_timestamp,
DAY
)),2) AS time_to_delivery
FROM orders as ord
JOIN customers as cust
ON ord.customer_id = cust.customer_id
Group BY cust.customer_state
order by time_to_delivery desc limit 5;

-- 5.4 b) Top 5 states with highest average time to delivery
SELECT cust.customer_state,
round(avg(TIMESTAMPDIFF(order_delivered_customer_date,order_purchase_timestamp,
DAY)),
2) AS time_to_delivery
FROM orders as ord
JOIN customers as cust
ON ord.customer_id = cust.customer_id
Group BY cust.customer_state
order by time_to_delivery asc limit 5;

-- 5.4 c) Top 5 states with highest average time to delivery
SELECT cust.customer_state,
avg(ABS(TIMESTAMPDIFF(order_delivered_customer_date,order_estimated_delivery_dat
e,DAY
))) AS diff_estimated_delivery
FROM orders as od
JOIN order_items as ord using(order_id)
JOIN sellers as sel Using(seller_id)
JOIN customers as cust
ON sel.seller_zip_code_prefix = cust.customer_zip_code_prefix
Where order_status ="delivered"
and order_delivered_customer_date is not null
Group BY cust.customer_state order by diff_estimated_delivery asc limit 5;
```

6. Payment type analysis:

1. **Month over Month count of orders for different payment types**

```
167  -- 6
168  -- 6.1 Month over Month count of orders for different payment types
169  SELECT pay.payment_type,Extract(Month FROM order_purchase_timestamp) as
170  month,count(1)
171  FROM payments as pay JOIN orders as ord using(order_id)
172  group by month,payment_type order by pay.payment_type,month;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

payment_type	month	count(1)
31842	4	1
credit_card	1	19
credit_card	2	12
credit_card	3	14
credit_card	4	16

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

```
174  -- 6.2 Count of orders based on the no. of payment installments
175  select payment_installments,Count(order_id) as ordercount
176  from payments
177  group by payment_installments;
178
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

payment_installments	ordercount
4	38
1	239
8	20
10	30
5	23

```
-- 6
-- 6.1 Month over Month count of orders for different payment types
SELECT pay.payment_type,Extract(Month FROM order_purchase_timestamp) as
month,count(1)
FROM payments as pay JOIN orders as ord using(order_id)
group by month,payment_type order by pay.payment_type,month;


-- 6.2 Count of orders based on the no. of payment installments
select payment_installments,Count(order_id) as ordercount
from payments
group by payment_installments;
```