--Creating the datbase and the tables

CREATE DATABASE "E Commerce Data"

WITH

OWNER = postgres

ENCODING = 'UTF8'

CONNECTION LIMIT = -1;

CREATE TABLE public.customers

(customer\_id character varying,

customer\_zip\_code\_prefix bigint,

PRIMARY KEY (customer\_id));

CREATE TABLE public.geolocation

(geolocation\_zip\_code\_prefix bigint,

geolocation\_city character varying,

geolocation\_state character varying,

PRIMARY KEY (geolocation\_zip\_code\_prefix));

CREATE TABLE public.order\_items

(order\_id character varying,

order\_item\_id integer,

product\_id character varying,

seller\_id character varying,

price numeric,

PRIMARY KEY (order\_id, order\_item\_id));

CREATE TABLE public.order\_payments

(order\_id character varying,

payment\_sequential integer,

payment\_type character varying,

payment\_installments integer,

payment\_value numeric,

PRIMARY KEY (order\_id, payment\_sequential));

CREATE TABLE public.orders

(order\_id character varying,

customer\_id character varying,

order\_status character varying,

order\_purchase\_timestamp timestamp without time zone,

order\_approved\_at timestamp without time zone,

order\_delivered\_carrier\_date timestamp without time zone,

order\_delivered\_customer\_data timestamp without time zone,

PRIMARY KEY (order\_id)

);

CREATE TABLE public.products

(

product\_id character varying,

product\_category\_name character varying,

PRIMARY KEY (product\_id)

);

CREATE TABLE public.reviews

(

review\_id character varying,

review\_score integer,

review\_creation\_date timestamp without time zone,

review\_answer\_timestamp timestamp without time zone,

PRIMARY KEY (review\_id)

);

CREATE TABLE public.sellers

(

seller\_id character varying,

seller\_zip\_code\_prefix integer,

PRIMARY KEY (seller\_id)

);

CREATE TABLE public.product\_category\_translation

(

product\_category\_name character varying,

product\_category\_name\_english character varying,

PRIMARY KEY (product\_category\_name)

);

CREATE TABLE public.mql

(

mql\_id character varying,

first\_contact\_date date,

origin character varying,

PRIMARY KEY (mql\_id)

);

CREATE TABLE public.closed\_deals

(

mql\_id character varying,

seller\_id character varying,

won\_date timestamp without time zone,

business\_segment character varying,

lead\_type character varying,

lead\_behavior\_profile character varying,

business\_type character varying,

PRIMARY KEY (mql\_id)

);

--Adding constraints

ALTER TABLE orders

ADD CONSTRAINT fk\_orders\_customers

FOREIGN KEY (customer\_id)

REFERENCES customers (customer\_id);

ALTER TABLE order\_items

ADD CONSTRAINT fk\_order\_product

FOREIGN KEY (product\_id)

REFERENCES products (product\_id);

ALTER TABLE order\_items

ADD CONSTRAINT fk\_order\_seller

FOREIGN KEY (seller\_id)

REFERENCES sellers (seller\_id);

ALTER TABLE closed\_deals

ADD CONSTRAINT fk\_deals\_seller

FOREIGN KEY (seller\_id)

REFERENCES sellers (seller\_id);

ALTER TABLE customers

ADD CONSTRAINT fk\_cust\_zip

FOREIGN KEY (customer\_zip\_code\_prefix)

REFERENCES geolocation (geolocation\_zip\_code\_prefix);

ALTER TABLE sellers

ADD CONSTRAINT fk\_seller\_zip

FOREIGN KEY (seller\_zip\_code\_prefix)

REFERENCES geolocation (geolocation\_zip\_code\_prefix);

--Loading the data

COPY closed\_deals

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_closed\_deals\_dataset.csv'

DELIMITER ','

CSV HEADER;

COPY customers

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_customers\_dataset.csv'

DELIMITER ','

CSV HEADER;

COPY geolocation

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_geolocation\_dataset.csv'

DELIMITER ','

CSV HEADER;

COPY mql

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_marketing\_qualified\_leads\_dataset.csv'

DELIMITER ','

CSV HEADER;

COPY order\_items

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_order\_items\_dataset.csv'

DELIMITER ','

CSV HEADER;

COPY order\_payments

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_order\_payments\_dataset.csv'

DELIMITER ','

CSV HEADER;

COPY order\_reviews

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_order\_reviews\_dataset.csv'

DELIMITER ','

CSV HEADER;

COPY orders

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_orders\_dataset.csv'

DELIMITER ','

CSV HEADER;

COPY product\_category\_translation

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\product\_category\_name\_translation.csv'

DELIMITER ','

CSV HEADER;

COPY products

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_products\_dataset.csv'

DELIMITER ','

CSV HEADER;

COPY reviews

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_reviews\_dataset.csv'

DELIMITER ','

CSV HEADER;

COPY sellers

FROM 'E:\MS ES DS\SEM 2\CSE 460\Project Data\olist\_sellers\_dataset.csv'

DELIMITER ','

CSV HEADER;

--Viewing sample data

SELECT \* FROM reviews

SELECT \* FROM mql

--Query executions

SELECT \* FROM reviews

WHERE (EXTRACT(YEAR FROM review\_creation\_date) = 2018 AND

EXTRACT(MONTH FROM review\_creation\_date) = 2)

INSERT INTO customers(customer\_id, customer\_zip\_code\_prefix)

VALUES ('f78374342g34837gg47846274ghhh3hkk', 8775)

SELECT \* FROM customers

WHERE customer\_id = 'f78374342g34837gg47846274ghhh3hkk'

INSERT INTO reviews(review\_id,

review\_score,

review\_creation\_date,

review\_answer\_timestamp)

VALUES('F8786VY875656G5feef67746576vd',

3,

'2022-02-21 00:00:00',

'2022-01-30 16:23:00')

SELECT \* FROM reviews

WHERE EXTRACT(YEAR FROM review\_creation\_date) = 2022

DELETE FROM closed\_deals

WHERE lead\_type IS NULL

SELECT \* FROM closed\_deals

WHERE lead\_type IS NULL

UPDATE order\_items

SET price = 60

WHERE product\_id = '4244733e06e7ecb4970a6e2683c13e61'

SELECT order\_id, product\_id, price FROM order\_items

WHERE product\_id = '4244733e06e7ecb4970a6e2683c13e61'

SELECT t2.product\_category, SUM(t1.num\_orders) num\_orders FROM

(SELECT COUNT(order\_id) AS num\_orders, product\_id

FROM order\_items

GROUP BY product\_id) AS t1

JOIN

(SELECT product\_id, product\_category\_name\_english AS product\_category

FROM products pd

JOIN product\_category\_translation pct

ON pd.product\_category\_name = pct.product\_category\_name) AS t2

ON t1.product\_id = t2.product\_id

GROUP BY t2.product\_category

ORDER BY num\_orders DESC

SELECT t2.business\_type,

t2.ct\*100 /(SELECT SUM(ct) FROM

(SELECT business\_type, COUNT(business\_type) ct

FROM closed\_deals

GROUP BY business\_type) t1

) pct\_of\_count

FROM

(SELECT business\_type, COUNT(business\_type) ct

FROM closed\_deals

GROUP BY business\_type) t2

SELECT geolocation\_state state,count(customer\_id) num\_cust FROM customers c

JOIN geolocation g

ON c.customer\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix

GROUP BY g.geolocation\_state

ORDER BY num\_cust DESC limit 5

SELECT T2.seller\_id,T1.geolocation\_city, T1.geolocation\_state, T2.sales

FROM

(SELECT seller\_id, count(order\_id) sales

FROM order\_items

GROUP BY seller\_id

) T2

JOIN

(SELECT s.seller\_id, g.geolocation\_city, g.geolocation\_state

FROM sellers s

JOIN geolocation g

ON s.seller\_zip\_code\_prefix = g.geolocation\_zip\_code\_prefix) T1

ON T1.seller\_id =T2.seller\_id

ORDER BY sales DESC LIMIT 10