Queries of ECommerce Business Analysis – Devina

-- 1. Create the tables

-- customers\_dataset, geolocation\_dataset, order\_items\_dataset, order\_payments\_dataset, order\_reviews\_dataset, orders\_dataset, product\_dataset, sellers\_dataset

CREATE TABLE IF NOT EXISTS customers\_dataset (

customer\_id VARCHAR UNIQUE PRIMARY KEY,

customer\_unique\_id VARCHAR,

customer\_zip\_code\_prefix VARCHAR,

customer\_city VARCHAR,

customer\_state VARCHAR

);

-- Primary Key creation for the table of geolocation\_dataset will be done after importing the data

CREATE TABLE IF NOT EXISTS geolocation\_dataset (

geolocation\_zip\_code\_prefix VARCHAR,

geolocation\_lat NUMERIC,

geolocation\_lng NUMERIC,

geolocation\_city VARCHAR,

geolocation\_state VARCHAR

);

CREATE TABLE IF NOT EXISTS order\_items\_dataset (

order\_id VARCHAR,

order\_item\_id INT,

product\_id VARCHAR,

seller\_id VARCHAR,

shipping\_limit\_date TIMESTAMP,

price NUMERIC,

freight\_value NUMERIC

);

CREATE TABLE IF NOT EXISTS order\_payments\_dataset (

order\_id VARCHAR,

payment\_sequential INT,

payment\_type VARCHAR,

payment\_installments INT,

payment\_value NUMERIC

);

CREATE TABLE IF NOT EXISTS order\_reviews\_dataset (

review\_id VARCHAR,

order\_id VARCHAR,

review\_score INT,

review\_comment\_title VARCHAR,

review\_comment\_message TEXT,

review\_creation\_date TIMESTAMP,

review\_answer\_timestamp TIMESTAMP

);

CREATE TABLE IF NOT EXISTS orders\_dataset (

order\_id VARCHAR UNIQUE PRIMARY KEY,

customer\_id VARCHAR,

order\_status VARCHAR,

order\_purchase\_timestamp TIMESTAMP,

order\_approved\_at TIMESTAMP,

order\_delivered\_carrier\_date TIMESTAMP,

order\_delivered\_customer\_date TIMESTAMP,

order\_estimated\_delivery\_date TIMESTAMP

);

CREATE TABLE IF NOT EXISTS product\_dataset (

idx INT,

product\_id VARCHAR UNIQUE PRIMARY KEY,

product\_category\_name VARCHAR,

product\_name\_length NUMERIC,

product\_description\_length NUMERIC,

product\_photos\_qty NUMERIC,

product\_weight\_g NUMERIC,

product\_length\_cm NUMERIC,

product\_height\_cm NUMERIC,

product\_width\_cm NUMERIC

);

CREATE TABLE IF NOT EXISTS sellers\_dataset (

seller\_id VARCHAR UNIQUE PRIMARY KEY,

seller\_zip\_code\_prefix VARCHAR,

seller\_city VARCHAR,

seller\_state VARCHAR

);

-- 2. Import all the corresponding dataset into the tables

-- 3. Clean duplicates of geolocation\_zip\_code\_prefix on table geolocation\_dataset to set it as Primary Key

-- 3.1. Add a new column containing unique number as the identifier.

ALTER TABLE geolocation\_dataset ADD COLUMN id SERIAL;

-- 3.2. Delete the duplicates by getting the duplicate value with lowest identifier number.

DELETE FROM geolocation\_dataset

WHERE id IN

(SELECT id

FROM

(SELECT id,

ROW\_NUMBER() OVER (PARTITION BY geolocation\_zip\_code\_prefix

ORDER BY id) AS row\_num

FROM geolocation\_dataset) t

WHERE t.row\_num > 1);

-- 3.3. Now that the identifier column isn't required anymore, drop the column to save space on our dataset.

ALTER TABLE geolocation\_dataset DROP COLUMN id;

-- 3.4 Set the geolocation\_zip\_code\_prefix as the Primary Key.

ALTER TABLE geolocation\_dataset ADD PRIMARY KEY (geolocation\_zip\_code\_prefix);

-- 4. Add Foreign Key to the corresponding tables to generate ERD.

-- 4.1. order\_id on orders\_dataset as Foreign Key for order\_payments\_dataset, order\_reviews\_dataset, and order\_items\_dataset.

ALTER TABLE order\_payments\_dataset ADD FOREIGN KEY (order\_id) REFERENCES orders\_dataset (order\_id);

ALTER TABLE order\_reviews\_dataset ADD FOREIGN KEY (order\_id) REFERENCES orders\_dataset (order\_id);

ALTER TABLE order\_items\_dataset ADD FOREIGN KEY (order\_id) REFERENCES orders\_dataset (order\_id);

-- 4.2. customer\_id on customers\_dataset as Foreign Key for orders\_dataset

ALTER TABLE orders\_dataset ADD FOREIGN KEY (customer\_id) REFERENCES customers\_dataset (customer\_id);

-- 4.3. product\_id on product\_dataset as Foreign Key for order\_items\_dataset

ALTER TABLE order\_items\_dataset ADD FOREIGN KEY (product\_id) REFERENCES product\_dataset (product\_id);

-- 4.4. seller\_id on sellers\_dataset as Foreign Key for order\_items\_dataset

ALTER TABLE order\_items\_dataset ADD FOREIGN KEY (seller\_id) REFERENCES sellers\_dataset (seller\_id);

-- 4.5. geolocation\_zip\_code\_prefix on geolocation\_dataset as Foreign Key for sellers\_dataset and customers\_dataset

-- Ideally, this query should be good enough if there is no any problem with the data:

ALTER TABLE sellers\_dataset ADD FOREIGN KEY (seller\_zip\_code\_prefix) REFERENCES geolocation\_dataset (geolocation\_zip\_code\_prefix);

ALTER TABLE customers\_dataset ADD FOREIGN KEY (customer\_zip\_code\_prefix) REFERENCES geolocation\_dataset (geolocation\_zip\_code\_prefix);

-- But since the Primary Key is missing some zip codes available on customers\_dataset and sellers\_dataset, you can manually add the foreign key using the interface by right-clicking on the table (customers\_dataset and sellers\_dataset), choose properties, open Constraints tab, and customize the references.

-- It is advised to update the geolocation\_dataset with the missing zip code prefix to uphold the reliabilities of the database.