**Student Name: Adeshina Ayodele**

**Exercise 1**

**You have been provided with three files with thousands of rows- Customer Sales Data, US Road Accident 1 and US Road Accident2. Please upload/import the three files into the PostgreSQL workspace environment using COPY syntax**

**Customer Sales Data.**

-- customer sales data

-- To create the schema for the table.

-- Ensure that every column title and its data type is captured

CREATE TABLE customer\_sales\_data (

index SERIAL PRIMARY KEY,

date DATE,

year INT,

month VARCHAR(10),

customer\_age INT,

customer\_gender TEXT,

country TEXT,

state TEXT,

product\_category TEXT,

sub\_category TEXT,

quantity INT,

unit\_cost NUMERIC,

unit\_price NUMERIC,

cost NUMERIC,

revenue NUMERIC

);

-- To Delete the table

DROP TABLE Customer\_sales\_data;

-- To copy the file into the table created above.

-- Encounted error in the format for date hence I used a modifier

SET datestyle = 'ISO, MDY'; -- Set date format to MM/DD/YY

COPY customer\_sales\_data (

index, date, year, month, customer\_age, customer\_gender, country, state,

product\_category, sub\_category, quantity, unit\_cost, unit\_price, cost, revenue

)

FROM '/Users/Public/Customer\_sales\_data.csv'

WITH CSV HEADER;

-- To see the entire table

select\* FROM customer\_sales\_data ;

**US Road Accident1**

-- US Accident 1

create table US\_Accident\_1(

Airport\_Code char(4),

Amenity BOOLEAN,

Astronomical\_Twilight TEXT CHECK (Astronomical\_Twilight IN ('Day', 'Night')),

Bump BOOLEAN,

Calculation1 TEXT,

City TEXT,

Civil\_Twilight TEXT CHECK (Astronomical\_Twilight IN ('Day', 'Night')),

count\_of\_Bump INT,

Count\_of\_Crossing INT,

count\_Traffic\_Signal INT,

Country TEXT,

County TEXT ,

Crossing BOOLEAN,

Description TEXT,

End\_Lat NUMERIC ,

End\_Lng NUMERIC ,

End\_Time TIMESTAMP,

Give\_Way BOOLEAN,

ID VARCHAR (10) PRIMARY KEY,

Junction BOOLEAN,

Nautical\_Twilight TEXT CHECK (Astronomical\_Twilight IN ('Day', 'Night')),

No\_Exit BOOLEAN,

Number INT,

Railway BOOLEAN,

Roundabout BOOLEAN,

Severity INT,

Side TEXT,

Source VARCHAR(15),

Start\_Time TIMESTAMP,

State CHAR(2),

Station BOOLEAN,

Stop BOOLEAN,

Street TEXT,

Sunrise\_Sunset TEXT,

Temperature\_F NUMERIC(10,1),

Timezone TEXT,

Traffic\_Calming BOOLEAN,

Traffic\_Signal BOOLEAN,

Turning\_Loop BOOLEAN,

Visibility\_mi NUMERIC,

Weather\_Condition TEXT

);

-- To Delete the table

DROP TABLE US\_Accident\_1;

-- To copy the file into the table created above.

COPY US\_Accident\_1(Airport\_Code,Amenity,Astronomical\_Twilight,Bump,Calculation1,City,Civil\_Twilight,count\_of\_Bump,Count\_of\_Crossing,count\_Traffic\_Signal,Country,County,Crossing,Description,End\_Lat,End\_Lng,End\_Time,Give\_Way,ID,Junction,Nautical\_Twilight,No\_Exit,Number,Railway,Roundabout,Severity,Side,Source,Start\_Time,State,Station,Stop,Street,Sunrise\_Sunset,Temperature\_f,Timezone,Traffic\_Calming,Traffic\_Signal,Turning\_Loop, Visibility\_mi,Weather\_Condition)

FROM '/Users/Public/US\_Accident\_1.csv'

WITH ENCODING 'ISO-8859-1'

CSV HEADER;

-- To see the entire table

select\* FROM US\_Accident\_1 LIMIT (10);

**US Road Accident 2**

-- US Accident 2

CREATE TABLE US\_Accident\_2 (

Airport\_Code VARCHAR(10),

Amenity BOOLEAN,

Astronomical\_Twilight TEXT CHECK (Astronomical\_Twilight IN ('Day', 'Night')),

Bump BOOLEAN,

Calculation1 TEXT,

City TEXT,

Civil\_Twilight TEXT CHECK (Civil\_Twilight IN ('Day', 'Night')),

Count\_of\_Bump INT,

Count\_of\_Crossing INT,

Count\_Traffic\_Signal INT,

Country TEXT,

County TEXT,

Crossing BOOLEAN,

Description TEXT,

End\_Lat NUMERIC,

End\_Lng NUMERIC,

End\_Time TIMESTAMP,

Give\_Way BOOLEAN,

ID TEXT PRIMARY KEY,

Junction BOOLEAN,

Nautical\_Twilight TEXT CHECK (Nautical\_Twilight IN ('Day', 'Night')),

No\_Exit BOOLEAN,

Number INT,

Weather\_Timestamp TIMESTAMP,

Wind\_Direction TEXT,

Zipcode TEXT,

Count\_of\_accidents INT,

Count\_of\_county INT,

Distance\_mi NUMERIC,

Humidity\_percent NUMERIC,

Number\_of\_Records INT,

Precipitation\_in NUMERIC,

Pressure\_in NUMERIC,

Records INT,

Start\_Lat NUMERIC,

Start\_Lng NUMERIC,

TMC INT,

Wind\_Chill\_F NUMERIC,

Wind\_Speed\_mph NUMERIC

);

-- To Delete the table

DROP TABLE US\_Accident\_2;

-- To copy the file into the table created above.

COPY US\_Accident\_2

FROM '/Users/Public/US\_Accident\_2.csv'

WITH ENCODING 'ISO-8859-1'

CSV HEADER;

-- To see the entire table

select\* FROM US\_Accident\_2 LIMIT (10);

**Exercise 2**

Using the MOVIES and BoxOffice Tables

-- customer sales data

-- To create the schema for the table.

-- Ensure that every column title and its data type is captured

CREATE TABLE customer\_sales\_data (

index SERIAL PRIMARY KEY,

date DATE,

year INT,

month VARCHAR(10),

customer\_age INT,

customer\_gender TEXT,

country TEXT,

state TEXT,

product\_category TEXT,

sub\_category TEXT,

quantity INT,

unit\_cost NUMERIC,

unit\_price NUMERIC,

cost NUMERIC,

revenue NUMERIC

);

-- To Delete the table

DROP TABLE Customer\_sales\_data;

-- To copy the file into the table created above.

-- Encounted error in the format for date hence I used a modifier

SET datestyle = 'ISO, MDY'; -- Set date format to MM/DD/YY

COPY customer\_sales\_data (

index, date, year, month, customer\_age, customer\_gender, country, state,

product\_category, sub\_category, quantity, unit\_cost, unit\_price, cost, revenue

)

FROM '/Users/Public/Customer\_sales\_data.csv'

WITH CSV HEADER;

-- To see the entire table

select\* FROM customer\_sales\_data ;

-- US Accident 1

create table US\_Accident\_1(

Airport\_Code char(4),

Amenity BOOLEAN,

Astronomical\_Twilight TEXT CHECK (Astronomical\_Twilight IN ('Day', 'Night')),

Bump BOOLEAN,

Calculation1 TEXT,

City TEXT,

Civil\_Twilight TEXT CHECK (Astronomical\_Twilight IN ('Day', 'Night')),

count\_of\_Bump INT,

Count\_of\_Crossing INT,

count\_Traffic\_Signal INT,

Country TEXT,

County TEXT ,

Crossing BOOLEAN,

Description TEXT,

End\_Lat NUMERIC ,

End\_Lng NUMERIC ,

End\_Time TIMESTAMP,

Give\_Way BOOLEAN,

ID VARCHAR (10) PRIMARY KEY,

Junction BOOLEAN,

Nautical\_Twilight TEXT CHECK (Astronomical\_Twilight IN ('Day', 'Night')),

No\_Exit BOOLEAN,

Number INT,

Railway BOOLEAN,

Roundabout BOOLEAN,

Severity INT,

Side TEXT,

Source VARCHAR(15),

Start\_Time TIMESTAMP,

State CHAR(2),

Station BOOLEAN,

Stop BOOLEAN,

Street TEXT,

Sunrise\_Sunset TEXT,

Temperature\_F NUMERIC(10,1),

Timezone TEXT,

Traffic\_Calming BOOLEAN,

Traffic\_Signal BOOLEAN,

Turning\_Loop BOOLEAN,

Visibility\_mi NUMERIC,

Weather\_Condition TEXT

);

-- To Delete the table

DROP TABLE US\_Accident\_1;

-- To copy the file into the table created above.

COPY US\_Accident\_1(Airport\_Code,Amenity,Astronomical\_Twilight,Bump,Calculation1,City,Civil\_Twilight,count\_of\_Bump,Count\_of\_Crossing,count\_Traffic\_Signal,Country,County,Crossing,Description,End\_Lat,End\_Lng,End\_Time,Give\_Way,ID,Junction,Nautical\_Twilight,No\_Exit,Number,Railway,Roundabout,Severity,Side,Source,Start\_Time,State,Station,Stop,Street,Sunrise\_Sunset,Temperature\_f,Timezone,Traffic\_Calming,Traffic\_Signal,Turning\_Loop, Visibility\_mi,Weather\_Condition)

FROM '/Users/Public/US\_Accident\_1.csv'

WITH ENCODING 'ISO-8859-1'

CSV HEADER;

-- To see the entire table

select\* FROM US\_Accident\_1 LIMIT (10);

-- US Accident 2

CREATE TABLE US\_Accident\_2 (

Airport\_Code VARCHAR(10),

Amenity BOOLEAN,

Astronomical\_Twilight TEXT CHECK (Astronomical\_Twilight IN ('Day', 'Night')),

Bump BOOLEAN,

Calculation1 TEXT,

City TEXT,

Civil\_Twilight TEXT CHECK (Civil\_Twilight IN ('Day', 'Night')),

Count\_of\_Bump INT,

Count\_of\_Crossing INT,

Count\_Traffic\_Signal INT,

Country TEXT,

County TEXT,

Crossing BOOLEAN,

Description TEXT,

End\_Lat NUMERIC,

End\_Lng NUMERIC,

End\_Time TIMESTAMP,

Give\_Way BOOLEAN,

ID TEXT PRIMARY KEY,

Junction BOOLEAN,

Nautical\_Twilight TEXT CHECK (Nautical\_Twilight IN ('Day', 'Night')),

No\_Exit BOOLEAN,

Number INT,

Weather\_Timestamp TIMESTAMP,

Wind\_Direction TEXT,

Zipcode TEXT,

Count\_of\_accidents INT,

Count\_of\_county INT,

Distance\_mi NUMERIC,

Humidity\_percent NUMERIC,

Number\_of\_Records INT,

Precipitation\_in NUMERIC,

Pressure\_in NUMERIC,

Records INT,

Start\_Lat NUMERIC,

Start\_Lng NUMERIC,

TMC INT,

Wind\_Chill\_F NUMERIC,

Wind\_Speed\_mph NUMERIC

);

-- To Delete the table

DROP TABLE US\_Accident\_2;

-- To copy the file into the table created above.

COPY US\_Accident\_2

FROM '/Users/Public/US\_Accident\_2.csv'

WITH ENCODING 'ISO-8859-1'

CSV HEADER;

-- To see the entire table

select\* FROM US\_Accident\_2 LIMIT (10);

--Using the MOVIES and BoxOffice Tables

--1.Find the movie with a row id of 6

select \*

From movies

WHERE movie\_id = 6;

--2.Find the movies released in the years between 2000 and 2010

SELECT \*

FROM MOVIES

WHERE YEAR BETWEEN '2000'AND'2010';

--3. List all movies and their combined sales in millions of dollars

SELECT title,

ROUND((boxoffice.domestic\_sales + boxoffice.international\_sales)/1000000, 2)AS Combined\_Sales

FROM MOVIES

JOIN BOXOFFICE

ON MOVIES.movie\_id = BOXOFFICE.movie\_id;

--4 List all movies and their ratings in percent

SELECT title,

ROUND(boxoffice.rating\*10, 2)AS Rating\_Percentage

FROM MOVIES

JOIN BOXOFFICE

ON MOVIES.movie\_id = BOXOFFICE.movie\_id;

-- 5. Find all the Toy Story movies

SELECT \*

FROM MOVIES

WHERE title like'Toy Story%';

--OR

SELECT \*

FROM MOVIES

JOIN BOXOFFICE

ON MOVIES.movie\_id = BOXOFFICE.movie\_id

WHERE title like'Toy Story%';

--6. Find all the movies directed by John Lasseter

SELECT \*

FROM MOVIES

WHERE director = 'John Lasseter';

--OR

SELECT \*

FROM MOVIES

JOIN BOXOFFICE

ON MOVIES.movie\_id = BOXOFFICE.movie\_id

WHERE director = 'John Lasseter';

--7. Find all the movies (and director) not directed by John Lasseter

SELECT title,director

FROM MOVIES

WHERE director != 'John Lasseter';

--OR

SELECT title,director

FROM MOVIES

JOIN BOXOFFICE

ON MOVIES.movie\_id = BOXOFFICE.movie\_id

WHERE director != 'John Lasseter';

--8. Find the domestic and international sales for each movie

SELECT title, domestic\_sales,international\_sales

FROM MOVIES

JOIN BOXOFFICE

ON MOVIES.movie\_id = BOXOFFICE.movie\_id;

--9. Show the sales numbers for each movie that did better

-- internationally rather than domestically

SELECT title,international\_sales, domestic\_sales

FROM MOVIES

JOIN BOXOFFICE

ON MOVIES.movie\_id = BOXOFFICE.movie\_id

WHERE BOXOFFICE.international\_sales > BOXOFFICE.domestic\_sales;

-- 10. List all the movies by their ratings in descending order

SELECT title,rating

FROM MOVIES

JOIN BOXOFFICE

ON MOVIES.movie\_id = BOXOFFICE.movie\_id

ORDER BY BOXOFFICE.rating;

--11. Find the number of movies each director has directed

SELECT director,COUNT(\*) AS number\_of\_movies

FROM MOVIES

GROUP BY MOVIES.director;

--12. Toy Story 4 has been released to critical acclaim! It had a rating of

--8.7, and made 340 million domestically and 270 million internationally.

--Add the record to the BoxOffice table.

-- At this point i believe it is better to have a table created to hold

-- the combined or joined tables (movies and boxoffice). This is to simplify subsequent queries.

CREATE VIEW Combined\_Table AS

SELECT

MOVIES.movie\_id AS movie\_id,

MOVIES.title,

MOVIES.director,

MOVIES.year,

MOVIES.length\_minutes,

BOXOFFICE.rating,

BOXOFFICE.domestic\_sales,

BOXOFFICE.international\_sales

FROM MOVIES

JOIN BOXOFFICE

ON MOVIES.movie\_id = BOXOFFICE.movie\_id;

-- Now to add new values.

INSERT INTO BOXOFFICE (movie\_id, rating, domestic\_sales, international\_sales)

VALUES (15,8.7,340000000,270000000);

--To check

SELECT \*

FROM BOXOFFICE

--13. The director for A Bug's Life is incorrect, it was actually directed by

--John Lasseter

UPDATE MOVIES

SET director='John Lasseter'

WHERE title = 'A Bugs Life';

--To verify

SELECT \*

FROM MOVIES

WHERE title = 'A Bugs Life';

--14. The year that Toy Story 2 was released is incorrect, it was actually

--released in 1999. You should update it.

-- There MUST have been an error in this question because the year of release is already 1999 in the table

-- So I modified the question to change it to 1914.

UPDATE MOVIES

SET year = 1914

WHERE title = 'Toy Story 2';

--To verify

SELECT \*

FROM MOVIES

WHERE title = 'Toy Story 2';

-- And then back to what it was.

UPDATE MOVIES

SET year = 1999

WHERE title = 'Toy Story 2';

--15. This database is getting too big, lets remove all movies that were

--released before 2005.

-- Before Delecting it is good practice to check first

SELECT \*

FROM Combined\_Table

WHERE year < 2005;

DELETE

FROM Combined\_Table

WHERE year < 2005;

-- On executing the script above, I learnt that ...

--Views that do not select from a single table or view are not automatically updatable.

-- So I went to the actually table has the year column.

SELECT \*

FROM MOVIES

WHERE year < 2005;

DELETE

FROM MOVIES

WHERE year < 2005;

-- To verify the resulting table

SELECT \*

FROM MOVIES