**LAB 7**

**Submitted by: SIMRAN**

**100377444**

**(a)**

Create the three tables with their SQL queries and paste the code in text format below. Make sure while creating the tables: [5+2.5+2.5]

• Must have primary keys in all the tables.

• The attributes which can potentially be foreign keys must have NOT NULL Constraint.

• Gender attribute in users table should CHECK for ‘M’ or ‘F’.

1.Users Table Query :

CREATE TABLE users(

user\_id serial PRIMARY KEY,

user\_name VARCHAR(255),

age INT,

gender CHAR(1) CHECK(gender = 'M' OR gender = 'F'),

occ\_id INT NOT NULL REFERENCES occupation(occ\_id),

city\_id INT NOT NULL REFERENCES city(city\_id)

);

2.Occupation Table Query :

CREATE TABLE occupation (

occ\_id INT NOT NULL PRIMARY KEY,

occ\_name VARCHAR(255)

);

3.City Table Query:

CREATE TABLE city(

city\_id INT NOT NULL PRIMARY KEY,

city\_name VARCHAR(255)

);

**(b)**

1. Write 2 different types of sql queries to find the users in city ‘Boston’. Write 2 types of queries, one using joins to find the answer and another using subqueries to find the same answer.

ANS: USING JOINS:

SELECT users.user\_id , users.user\_name

FROM users JOIN city

ON city.city\_id = users.city\_id

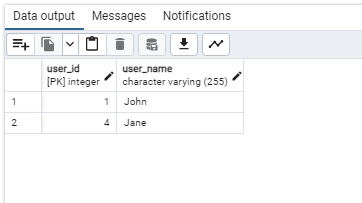
WHERE city.city\_name= 'Boston';

USING Sub-query:

SELECT user\_id , user\_name

FROM users

WHERE city\_id = (SELECT city\_id FROM city WHERE city\_name = 'Boston');



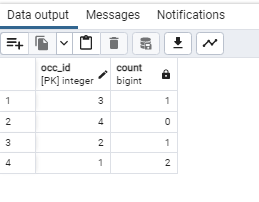
2. Write sql query to find how many users are there per occupation.

ANS:

SELECT occupation.occ\_id, COUNT(users.user\_id)

FROM users RIGHT JOIN occupation ON users.occ\_id = occupation.occ\_id

GROUP BY occupation.occ\_id;



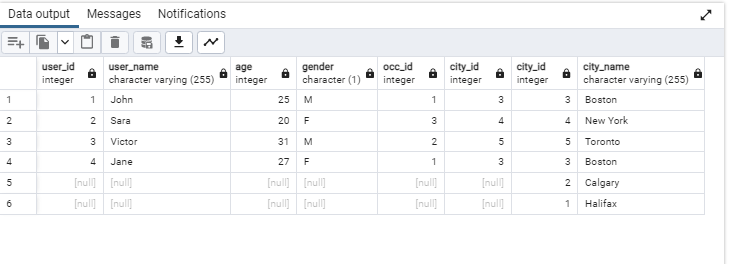
3. Perform full outer join between users and city

ANS:

SELECT \*

FROM users FULL OUTER JOIN city

ON users.city\_id = city.city\_id;



**(c)**

1. Write query to make a copy of ‘users’ table known as ‘users\_new’ without Data.

ANS:

CREATE TABLE users\_new

AS TABLE users

WITH NO DATA ;

2. Write query to insert all columns of ‘users’ to the ‘users\_new’.

ANS:

INSERT INTO users\_new

SELECT \* FROM users;

3. Write CASE query to add one more column with salary values to the ‘users’ table. Salary for Software engineer is 80,000, Accountant is 70,000 and Pharmacist is 90,000.

ANS:

ALTER TABLE users\_new

ADD salary NUMERIC(10,2);

UPDATE users\_new

SET salary =

CASE

WHEN users\_new.occ\_id = (SELECT occ\_id FROM occupation WHERE occ\_name = 'Software engineer ')THEN 80000

WHEN users\_new.occ\_id = (SELECT occ\_id FROM occupation WHERE occ\_name = 'Accountant') THEN 70000

WHEN users\_new.occ\_id = (SELECT occ\_id FROM occupation WHERE occ\_name = 'Pharmacist')THEN 90000

END;

**(d)**

1. Write query to add foreign keys constraints to ‘users’ table. Assuming you forgot to add it earlier.

ANS:

ALTER TABLE users

ADD CONSTRAINT occuID\_no FOREIGN KEY (occ\_id) REFERENCES occupation(occ\_id),

ADD CONSTRAINT cityID\_no FOREIGN KEY (city\_id) REFERENCES city(city\_id);

2. Add country column to ‘city’ table. DEFAULT constraint must be used to add Canada as a default country for cities. [Use DEFAULT Constraint to default your country to Canada, that way you only have to write the countries for cities not in Canada, Hint : remember ‘boston’ and ‘new York’ are cities in US, rest all are in Canada, Use ALTER TABLE to add column and default constraint].

ANS: Adding column:

ALTER TABLE city

ADD country VARCHAR(255) DEFAULT 'Canada';

UPDATE city

SET country =

CASE

WHEN city\_name = 'Boston' THEN 'US'

WHEN city\_name = 'New York' THEN 'US'

ELSE 'Canada'

END;

The table ‘city’ looks like:

