EXPERIMENT 8

AIM:

Set up a PostgreSQL database and create tables to store relational data. Perform basic CRUD operations using SQL queries.

THEORY:

In this experiment, we explored how to set up and work with a PostgreSQL relational database management system (RDBMS) to store and manage structured data in the context of a hospital database. PostgreSQL is an advanced open-source database system that provides robust support for SQL (Structured Query Language), which is the standard language for relational database management. It is widely used in both academic and professional environments due to its reliability, stability, and strong compliance with SQL standards.

The primary goal of this experiment was to model and implement a real-world scenario using relational database principles. We focused on a hospital system where a table named patients was created to store various attributes related to patient records, including patient name, age, gender, diagnosis, doctor's name, and hospital name. Each patient record was uniquely identified using a SERIAL primary key, which ensures that each row in the table is assigned a unique identifier automatically.

To define the structure of the data, we used the CREATE TABLE command in PostgreSQL. This allowed us to set the column names, data types (such as VARCHAR for text and INTEGER for numbers), and constraints like the PRIMARY KEY. After defining the schema, we moved on to implementing the four essential SQL operations known as CRUD: Create, Read, Update, and Delete.

The INSERT INTO statement was used to add new patient records to the table. This operation is critical for feeding data into the database. Next, we used the SELECT * FROM command to read or retrieve all the records from the table. This query helped us verify the data insertion and understand the current state of the table.

We then performed the UPDATE operation, which allowed us to change specific information in a record—for example, updating a patient's diagnosis based on new findings. The WHERE clause played a vital role here, as it helped us target only the desired row(s) without affecting others. Finally, we executed the DELETE FROM command to remove a record from the table. This command is useful for eliminating outdated or incorrect data.

Throughout the experiment, we also learned the importance of working in the correct database context (using \c database_name), listing tables with \dt, and using basic SQL syntax effectively. By executing these commands step by step, we gained a deeper understanding of how data can be efficiently organized, stored, and manipulated within a relational database environment.

Moreover, we applied the database design to an Indian healthcare setting by including data entries related to Indian hospitals and doctors. This helped contextualize our understanding and demonstrated the practical applications of PostgreSQL in managing critical information in real-world sectors such as healthcare.

```
Server [localhost]: localhost
Database [postgres]: postgres
Port [5432]: 5432
Username [postgres]: postgres
Password for user postgres:

psql (17.4)
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.
```

CREATE DATABASE

```
postgres=# \dt
Did not find any relations.
postgres=# CREATE TABLE patients (
postgres(#     patient_id SERIAL PRIMARY KEY,
postgres(#     name VARCHAR(100),
postgres(#     age INTEGER,
postgres(#     gender VARCHAR(10),
postgres(#     diagnosis VARCHAR(150),
postgres(#     doctor_name VARCHAR(100),
postgres(#     hospital_name VARCHAR(100)
postgres(# );
CREATE TABLE
postgres=# SELECT COUNT(*) FROM patients;
```

INSERT VALUES

```
postgres=# INSERT INTO patients (name, age, gender, diagnosis, doctor_name, hospital_name) VALUES
postgres-# ('Ravi Mehra', 45, 'Male', 'Hypertension', 'Dr. Shalini Gupta', 'AIIMS Delhi'),
postgres-# ('Pooja Nair', 30, 'Female', 'Diabetes Type 2', 'Dr. Kiran Rao', 'Apollo Hospital, Chennai'),
postgres-# ('Imran Sheikh', 60, 'Male', 'Cardiac Arrest', 'Dr. Abhay Kulkarni', 'Fortis Hospital, Mumbai'),
postgres-# ('Lakshmi Devi', 55, 'Female', 'Arthritis', 'Dr. Ramesh Iyer', 'CMC Vellore');
INSERT 0 4
```

VIEW RELATION

```
postgres=# SELECT * FROM patients;
 patient_id |     name     | age | gender |    diagnosis    |    doctor_name     |      hospital_name
------------+--------------+-----+--------+-----------------+--------------------+--------------------------
          1 | Ravi Mehra   |  45 | Male   | Hypertension    | Dr. Shalini Gupta  | AIIMS Delhi
          2 | Pooja Nair   |  30 | Female | Diabetes Type 2 | Dr. Kiran Rao      | Apollo Hospital, Chennai
          3 | Imran Sheikh |  60 | Male   | Cardiac Arrest  | Dr. Abhay Kulkarni | Fortis Hospital, Mumbai
          4 | Lakshmi Devi |  55 | Female | Arthritis       | Dr. Ramesh Iyer    | CMC Vellore
(4 rows)
```

UPDATE

```
postgres=# UPDATE patients
postgres-# SET diagnosis = 'Chronic Arthritis'
postgres-# WHERE name = 'Lakshmi Devi';
UPDATE 1
```

DELETE

```
postgres=# DELETE FROM patients
postgres-# WHERE name = 'Imran Sheikh';
DELETE 1
```

FINAL TABLE

```
postgres=# SELECT * FROM patients;
 patient_id |     name     | age | gender |     diagnosis     |    doctor_name    |      hospital_name
------------+--------------+-----+--------+-------------------+-------------------+--------------------------
          1 | Ravi Mehra   |  45 | Male   | Hypertension      | Dr. Shalini Gupta | AIIMS Delhi
          2 | Pooja Nair   |  30 | Female | Diabetes Type 2   | Dr. Kiran Rao     | Apollo Hospital, Chennai
          4 | Lakshmi Devi |  55 | Female | Chronic Arthritis | Dr. Ramesh Iyer   | CMC Vellore
(3 rows)
```

CONCLUSION:

Through this experiment, we successfully implemented a hospital management scenario using PostgreSQL. We created a table, inserted Indian patient data, and performed SQL-based CRUD operations effectively. This hands-on practice helped us understand how relational databases can be used to manage real-world data in the healthcare sector. We also gained confidence in writing and executing SQL queries to handle structured information in a secure and efficient manner.