Experiment 1: SQL Data Definition and Manipulation

AIM

To create the `Authors` and `Books` tables using Data Definition Language (DDL) commands, insert sample records into these tables, and retrieve book titles along with author information using an `INNER JOIN`.

OBJECTIVE

This experiment aims to demonstrate the fundamental concepts of database management, including:

1. **DDL (Data Definition Language):** Creating tables and defining their structure, including primary and foreign keys.
2. **DML (Data Manipulation Language):** Inserting data into tables.
3. **Joins:** Combining data from multiple tables based on related columns using an `INNER JOIN`. This is a core concept for querying data from a normalized database schema.

PROCEDURE / ALGORITHM

1. **Create Tables:** Use the `CREATE TABLE` command to define the `Authors` and `Books` tables. Define `author_id` as the primary key in the `Authors` table and `book_id` as the primary key in the `Books` table. Establish a foreign key relationship by defining `author_id` in the `Books` table as a foreign key that references `author_id` in the `Authors` table.
2. **Insert Data:** Use the `INSERT INTO` command to populate the `Authors` and `Books` tables with sample data.
3. **Retrieve Data:** Use the `SELECT` statement with an `INNER JOIN` to retrieve specific columns (`Books.title`, `Authors.name`, `Authors.country`) from both tables, combining the rows where `Books.author_id` matches `Authors.author_id`.

Q1: Problem Statement

Create the `Authors` and `Books` tables using DDL commands.

QUERY

```
CREATE TABLE Authors_Khushi(author_id INT PRIMARY KEY, name VARCHAR(50), country VARCHAR(50));
CREATE TABLE Books(book_id INT PRIMARY KEY, title VARCHAR(100), author_id INT, FOREIGN KEY(author_id) REFERENCES Authors_Khushi(author_id));
desc Authors_Khushi;
desc Books;
```

OUTPUT

```
postgres=# CREATE TABLE Authors_Khushi(author_id INT PRIMARY KEY, name VARCHAR(50), country VARCHAR(50));
CREATE TABLE
postgres=# CREATE TABLE Books(book_id INT PRIMARY KEY, title VARCHAR(100), author_id INT, FOREIGN KEY(author_id) REFERENCES Authors_Khushi(author_id));
CREATE TABLE
```

```
postgres=# \d Authors_Khushi;
                  Table "public.authors_khushi"
  Column   |         Type          | Collation | Nullable | Default
-----------+-----------------------+-----------+----------+---------
 author_id | integer               |           | not null |
 name      | character varying(50) |           |          |
 country   | character varying(50) |           |          |
Indexes:
    "authors_khushi_pkey" PRIMARY KEY, btree (author_id)
Referenced by:
    TABLE "books" CONSTRAINT "books_author_id_fkey" FOREIGN KEY (author_id) REFERENCES authors_khushi(author_id)


postgres=# \d Books;
                     Table "public.books"
  Column   |          Type          | Collation | Nullable | Default
-----------+------------------------+-----------+----------+---------
 book_id   | integer                |           | not null |
 title     | character varying(100) |           |          |
 author_id | integer                |           |          |
Indexes:
    "books_pkey" PRIMARY KEY, btree (book_id)
Foreign-key constraints:
    "books_author_id_fkey" FOREIGN KEY (author_id) REFERENCES authors_khushi(author_id)
```

Q2: Problem statement

Insert sample records into the `Authors` and Books tables.

QUERY

insert into Authors_Khushi Values
(1, 'Ashish', 'India'),
(2, 'Smaran', 'USA'),
(3, 'Vaibhav', 'UK');
Insert into Books Values
(101, 'Data Science Basics', 1),
(102, 'AI in Education', 2),
(103, 'SQL Simplified', 1);
Select * from Authors_Khushi;
Select * from Books;

OUTPUT

```
postgres=# insert into Authors_Khushi Values
postgres-# (1, 'Ashish', 'India'),
postgres-# (2, 'Smaran', 'USA'),
postgres-# (3, 'Vaibhav', 'UK');
INSERT 0 3
postgres=# Insert into Books Values
postgres-# (101, 'Data Science Basics', 1),
postgres-# (102, 'AI in Education', 2),
postgres-# (103, 'SQL Simplified', 1);
INSERT 0 3
```

```
postgres=# Select * from Authors_Khushi;
 author_id |  name   | country
-----------+---------+---------
         1 | Ashish  | India
         2 | Smaran  | USA
         3 | Vaibhav | UK
(3 rows)


postgres=# Select * from Books;
 book_id |        title        | author_id
---------+---------------------+-----------
     101 | Data Science Basics |         1
     102 | AI in Education     |         2
     103 | SQL Simplified      |         1
(3 rows)
```

Q3: Problem Statement

Retrieve book titles along with author information using an Inner Join.

QUERY

SELECT Books.title, Authors_Khushi.name, Authors_Khushi.country FROM Books INNER JOIN Authors_Khushi ON Books.author_id = Authors_Khushi.author_id;

OUTPUT

```
postgres=# SELECT Books.title, Authors_Khushi.name, Authors_Khushi.country FROM Books INNER JOIN Authors_Khushi ON Books.author_id = Authors_Khushi.author_id;
       title        |  name  | country
--------------------+--------+---------
 Data Science Basics | Ashish | India
 AI in Education     | Smaran | USA
 SQL Simplified      | Ashish | India
(3 rows)
```