Experiment 1: SQL DDL & DML

AIM

To design and create the Authors_Yash and Books tables using SQL Data Definition Language (DDL), insert sample data using DML statements, and retrieve book titles with corresponding author details through the use of an INNER JOIN operation..

OBJECTIVE

1. This experiment helps in understanding the fundamental concepts of database systems by performing the following:
2. Data Definition Language (DDL): Creating relational tables and specifying primary and foreign key constraints.
3. Data Manipulation Language (DML): Inserting records into the database.
4. JOIN operations: Using INNER JOIN to fetch data across tables linked through a foreign key, which is a key aspect of querying in normalized relational databases.

PROCEDURE / ALGORITHM

1. Table Creation: Define the structure of Authors_Yash and Books tables using the CREATE TABLE command. Set author_id as a primary key in the Authors_Yash table, and establish it as a foreign key in the Books table.
2. Inserting Records: Use the INSERT INTO command to add sample entries into both tables.
3. Data Retrieval: Apply the SELECT statement with an INNER JOIN to display the book title along with the author's name and country where the author_id field matches in both tables.

Q1: Problem Statement

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

QUERY

CREATE TABLE Authors_Yash(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_Yash(author_id));
desc Authors_Yash;
desc Books;

OUTPUT

```
postgres=# CREATE TABLE Authors_Yash(author_id INT PRIMARY KEY, name VARCHAR(50), country VARCHAR(50));
CREATE TABLE
postgres=# \d Authors_Yash
                  Table "public.authors_yash"
  Column   |         Type          | Collation | Nullable | Default
-----------+-----------------------+-----------+----------+---------
 author_id | integer               |           | not null |
 name      | character varying(50) |           |          |
 country   | character varying(50) |           |          |
Indexes:
    "authors_yash_pkey" PRIMARY KEY, btree (author_id)
```

```
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 Books;
                    Table "public.books"
  Column   |          Type          | Collation | Nullable | Default
-----------+------------------------+-----------+----------+---------
 book_id   | integer                |           | not null |
 title     | character varying(100) |           |          |
 author_id | integer                |           |          |
```

Q2: Problem statement

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

QUERY

insert into Authors_Yash Values
(1, 'Ashish', 'India'),
(2, 'Smaran', 'USA'),
(3, 'Vaibhav', 'UK');
Insert into Books Values
(101, 'Data Science Basics', 1),
(102, 'AI in Education', 2),

```
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
```

(103, 'SQL Simplified', 1);
Select * from Authors_Yash;
Select * from Books;

OUTPUT

```
postgres=# insert into Authors_Yash Values
postgres-# (1, 'Ashish', 'India'),
postgres-# (2, 'Smaran', 'USA'),
postgres-# (3, 'Vaibhav', 'UK');
INSERT 0 3
```

```
postgres=# Select * from Authors_Yash;
 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_Yash.name, Authors_Yash.country FROM Books INNER JOIN Authors_Yash ON Books.author_id = Authors_Yash.author_id;

OUTPUT

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