

Experiment-1

1) Aim

To design and implement a basic **Book Management System** using SQL by:

- Creating two related tables — **Authors** and **Books** — with proper keys and rules.
- Adding sample data to both tables while keeping the link between them correct.
- Writing an **INNER JOIN** query to display book titles along with the author's name and country.

2) Objective

A **relational database** is a way to store data in tables where each table represents a specific type of information, and connections between them are made using keys.

- The **Authors** table holds details like author ID, name, and country.
- The **Books** table stores book ID, title, and the author who wrote the book.

The link between Authors and Books is **one-to-many** — one author can write many books, but each book is written by only one author. This link is made using a **foreign key**:

- `author_id` in the Books table refers to `author_id` in the Authors table.

To ensure data is accurate and consistent:

- **Primary Keys** uniquely identify each record.
- **Foreign Keys** make sure that each book is connected to a valid author.

3) Procedure / Algorithm

Step 1: Design the Database Structure

1.1 Create a table named **Authors** with these columns:

- author_id (Primary Key, INT)
- name (VARCHAR(50))
- country (VARCHAR(50))

1.2 Create a table named **Books** with these columns:

- book_id (Primary Key, INT)
- title (VARCHAR(100))
- author_id (Foreign Key, INT) referring to Authors(author_id)

Step 2: Add Sample Data

2.1 Insert at least three sample records into the **Authors** table with names and countries.

2.2 Insert at least three sample records into the **Books** table, using valid author_id values from the Authors table.

Step 3: Retrieve Data Using JOIN

3.1 Write an **INNER JOIN** query to get:

- Book title
- Author name
- Author country

3.2 Run the query to show which book is written by which author.

Step 4: Check Integrity and Results

4.1 Make sure foreign key rules are followed (no books without a matching author).

4.2 Confirm that the final output correctly shows related data from both tables.

5)Problem Statement->

Problem statement-1:

Design a basic Book Management System by creating two relational tables: Authors and Books. The system must represent a one-to-many relationship, where one author can write multiple books, but each book is associated with only one author. Use appropriate primary key and foreign key constraints to maintain referential integrity between the tables.

Query-1:

```
CREATE TABLE Authors(  
    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(author_id) );  
  
DESC Authors;  
  
DESC Books;
```

Output-1:

29m

1

2

3

Create Author and Book Tables using DDL Commands

Score: 5 | Difficulty: easy

Problem Statement

You are tasked with designing a basic book management system. Create two tables — **Authors** and **Books** — to represent a one-to-many relationship (one author can write multiple books). Use proper **primary and foreign key constraints** while designing the schema.

Input Format:

Table **Authors** with columns:

- author_id**(INT, Primary Key)
- name** (VARCHAR(50))
- country** (VARCHAR(50))

Table **Books** with columns:

- book_id** (INT, Primary Key)
- title** (VARCHAR(100))
- author_id** (INT, Foreign Key referencing Authors)

SQL

```

1 CREATE TABLE Authors (
2   author_id INT PRIMARY KEY,
3   name VARCHAR(50),
4   country VARCHAR(50)
5 );
6
7 CREATE TABLE Books (
8   book_id INT PRIMARY KEY,
9   title VARCHAR(100),
10  author_id INT,
11  FOREIGN KEY (author_id) REFERENCES Authors(author_id)
12 );
13
14 DESC Authors;
15 DESC Books;
16

```

Test & Results

Submit

Custom Input

Test Cases

Test Case	Status	Test Case Info
Test Case 1	Passed	

Problem statement-2:

After creating the Authors and Books tables, your next task is to insert sample records into both tables. You must add at least three authors and three books, ensuring that each book correctly references an existing author through the `author_id` field.

Query-2:

```

insert into Authors(author_id,name,country)

values(1,'Ashish','India'),

(2,'Smaran','USA'),

(3,'Vaibhav','UK');

insert into Books(book_id,title,author_id)

values(101,'Data Science Basics',1),

(102,'AI in Education',2),

```

```
(103,'SQL Simplified',1);

select * from Authors;select * from Books;
```

Output-2:

The screenshot shows a SQL practice interface with a problem statement and a test case result.

Problem Statement: After creating the Authors and Books tables, your next task is to insert sample records. Insert **at least 3 authors** and **3 books**, ensuring books reference valid authors using the foreign key.

Input Format:

- Pre-existing Authors and Books table structures from Problem 1.

Output Format:

Authors Table:

author_id	name	country
1	Ashish	India
2	Smaran	USA
3	Vaibhav	UK

Books Table:

book_id	title	author_id
101	Data Science Basics	1
102	AI in Education	2
103	SQL Simplified	1

SQL Query:

```
-- Write your Query here
1 insert into Authors(author_id,name,country)
2 values(1,'Ashish','India'),(2,'Smaran','USA'),(3,'Vaibhav','UK');
3
4
5 insert into Books(book_id,title,author_id)
6 values(101,'Data Science Basics',1),(102,'AI in Education',2),(103,'SQL Simplified',1);
7 select * from Authors;
8 select * from Books;
```

Test & Results:

Test Case	Status	Test Case Info
Test Case 1	Passed	

Problem statement-3:

Using the existing Authors and Books tables, your task is to retrieve a list of books along with their corresponding author's name and country. This requires performing an INNER JOIN on the author_id field to combine data from both tables based on their relationship.

Query-3:

```
select Books.title,Authors.name,Authors.country from Books
Inner Join Authors ON Books.author_id=Authors.author_id;
```

Output-3:

30m

1

2

3

Retrieve Book Titles Along with Author Information Using INNER JOIN

Score: 5 | Difficulty: easy

Problem Statement

Given two tables, Authors and Books, retrieve the titles of all books along with their **author's name and country**. This involves creating tables, inserting data, and using an INNER JOIN to combine records based on author_id.

Input Format:

- Pre-existing Authors and Books table structures from Problem 1.

Table **Authors** with columns:

- author_id** (INT, Primary Key)
- name** (VARCHAR(50))
- country** (VARCHAR(50))

Table **Books** with columns:

- book_id** (INT, Primary Key)
- title** (VARCHAR(100))
- author_id** (INT Foreign Key referencing

SQL

```
1 --y here
2
3 --,Authors.name,Authors.country from Books inner join Authors ON Books.author_id=Authors.author_id;
```

Test & Results

Submit

Custom Input

Test Cases

Test Case	Status	Test Case Info
Test Case 1	Passed	