Experiment-1

1)Aim->

To design and implement a basic book management system using SQL by:

1. Creating two related tables — `Authors` and `Books` — with appropriate constraints.
2. Inserting valid sample records into both tables while maintaining referential integrity.
3. Querying the data using an `INNER JOIN` to retrieve book titles along with corresponding author names and countries.

2)Objective->

A **relational database** is a structured collection of data organized into tables. Each table stores information about a specific entity, and relationships between entities are established through keys.

1. **Authors Table –** stores information about authors such as their ID, name, and country.
2. **Books Table –** stores details about books such as book ID, title, and the author who wrote it.

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

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

To maintain **data integrity**, we use:

- **Primary Keys** to uniquely identify each record in a table.
- **Foreign Keys** to ensure that a book can only be linked to an existing author.

3)Procedure/Algorithum->

Step-1: Design of database Schema:

1.1. Create a table named `Authors` with the following columns:
- `author_id` (Primary Key, INT)
- `name` (VARCHAR(50))
- `country` (VARCHAR(50))

1.2. Create a table named `Books` with the following columns:
- `book_id` (Primary Key, INT)
- `title` (VARCHAR(100))
- `author_id` (Foreign Key, INT) referencing `Authors(author_id)`

`Step-2: Insert Sample Data:`

2.1. Insert at least **three records** into the `Authors` table with meaningful names and countries.
2.2. Insert at least **three records** into the `Books` table, making sure that each `author_id` used exists in the `Authors` table.

Step -3: Perform Data Retrieval Using JOIN:

3.1. Write an SQL `INNER JOIN` query to retrieve the following combined information:
- Book title
- Author name
- Author country

3.2. Execute the query to display the output that links each book to its respective author.

Step 4: Verify Integrity and Output

4.1. Ensure that all foreign key constraints are satisfied (i.e., no orphaned books).
4.2. Confirm that the final result shows only valid and related records from both tables.

5)Problem Statement->

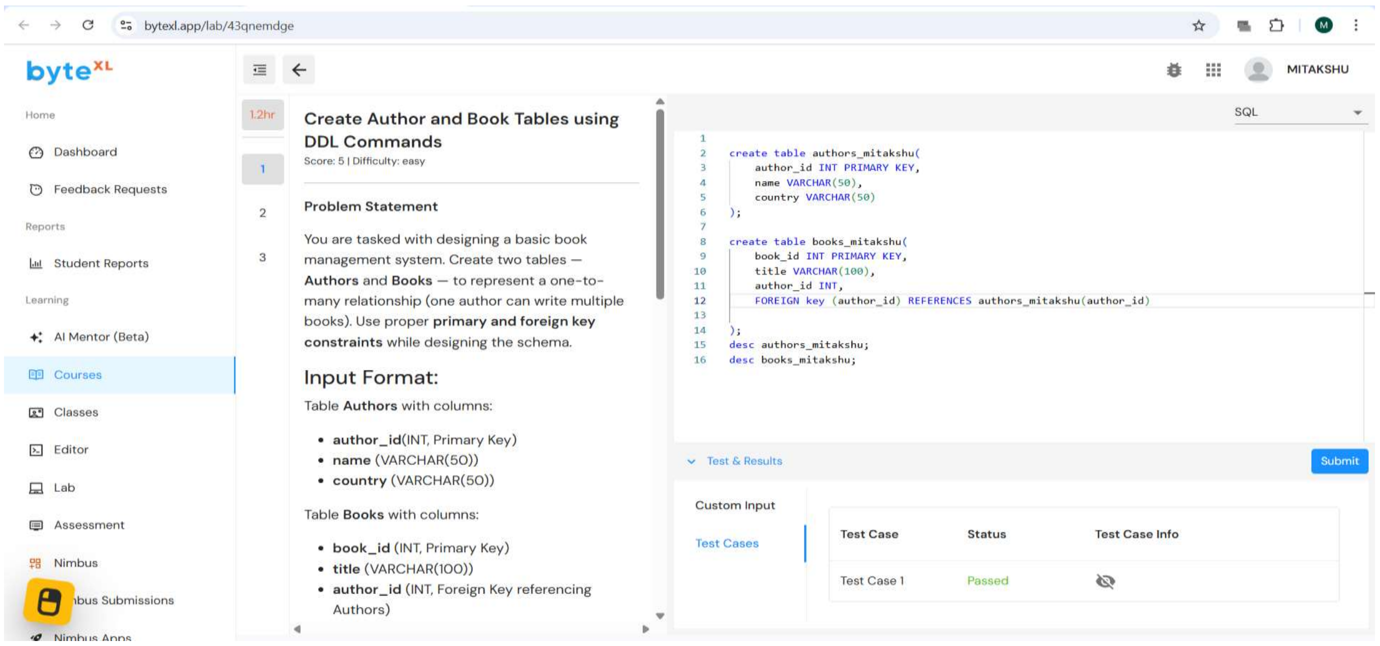
Problem statement-1:

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:



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)

```

create table authors_mitakshu(
    author_id INT PRIMARY KEY,
    name VARCHAR(50),
    country VARCHAR(50)
);

create table books_mitakshu(
    book_id INT PRIMARY KEY,
    title VARCHAR(100),
    author_id INT,
    FOREIGN key (author_id) REFERENCES authors_mitakshu(author_id)

);
desc authors_mitakshu;
desc books_mitakshu;
```

Test Case	Status	Test Case Info
Test Case 1	Passed	

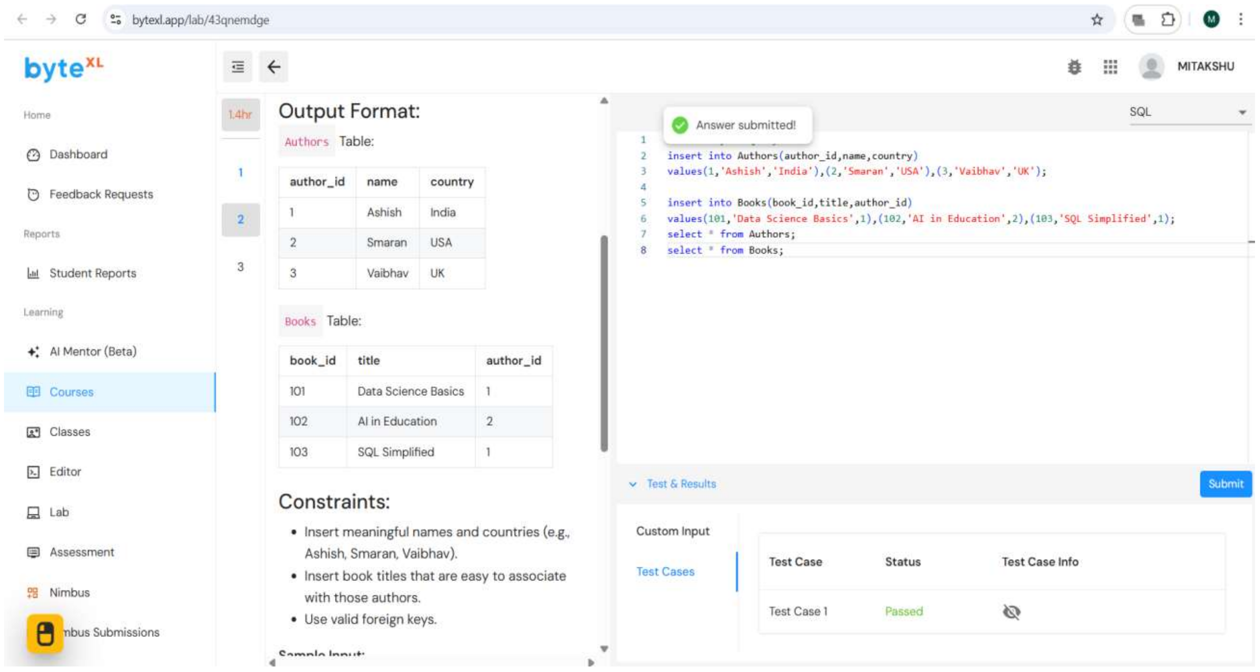
Problem statement-2:

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:


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

Constraints:

- Insert meaningful names and countries (e.g., Ashish, Smaran, Vaibhav).
- Insert book titles that are easy to associate with those authors.
- Use valid foreign keys.

Answer submitted!

```sql
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;
```

Test Case	Status	Test Case Info
Test Case 1	Passed	

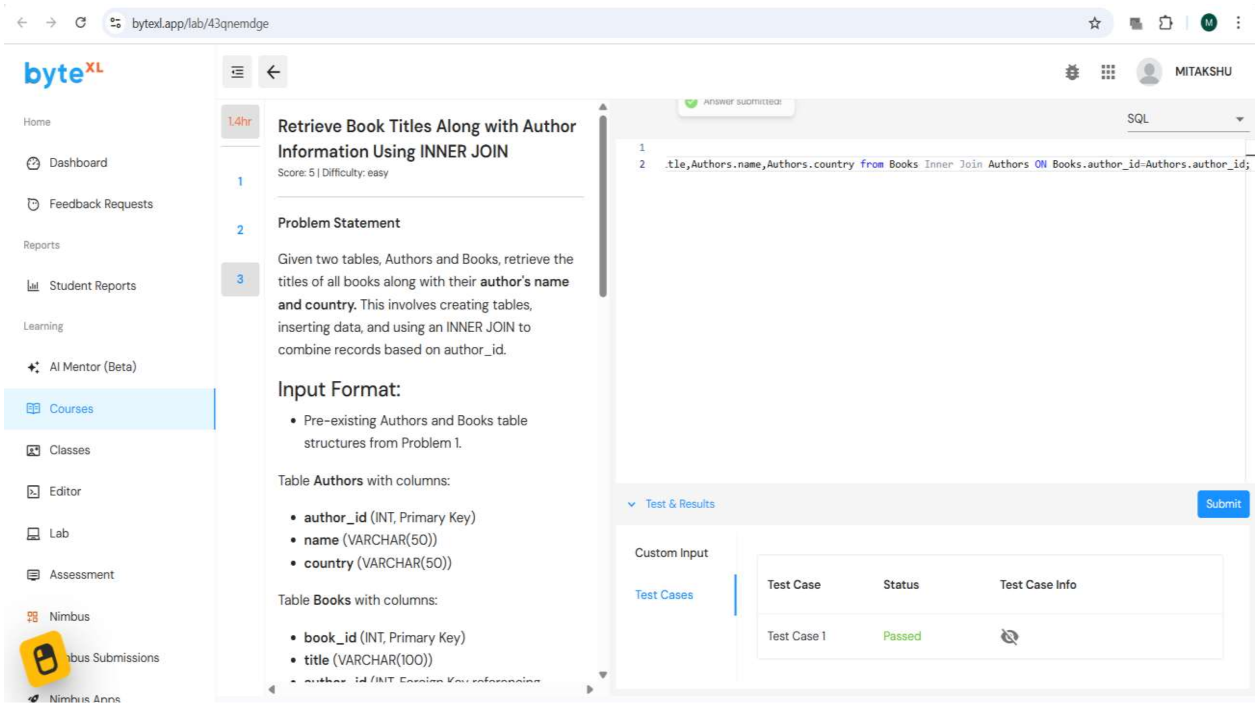
Problem statement-3:

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:



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))

```sql

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

Test Case	Status	Test Case Info
Test Case 1	Passed	

6)Learning Outcomes->

- Understand how to model one-to-many relationships using SQL.
- Learn to create database tables with primary and foreign key constraints.
- Gain skills in inserting valid data while maintaining referential integrity.
- Practice writing INNER JOIN queries to combine related data.
- Develop the ability to query and interpret results from relational tables.