);

EXPERIMENT: 01

Name: Paras Deep Singh Maan

UID: 23BCC70010

Subject: Advanced Database Management Systems

Subject Code: 23CSP-333

Section & Group: 23BCC-1 (A)

AIM:

To understand and implement the Author-Book Relationship using SQL JOIN operations and basic SQL queries.

THEORY:

SQL JOIN statements are employed to merge data from two or more tables by linking them through related columns, most commonly a primary key in one table and a corresponding foreign key in another. As databases are generally normalized to eliminate redundancy, JOIN operations become essential in assembling complete and insightful datasets.

Types of JOINS:

- **INNER JOIN**: Returns only the rows where there is a match in both tables.
- **LEFT JOIN**: Retrieves all rows from the left table along with the matching rows from the right table. If no match is found, the result includes NULLs for the right table's columns.
- **RIGHT JOIN**: Retrieves all rows from the right table and the corresponding matched rows from the left table. NULLs are used for unmatched entries from the left table.
- **FULL JOIN (FULL OUTER JOIN)**: Combines all rows from both tables. Where matches are not found, NULL values are used to fill in the missing data from the opposite table.

SQL QUERIES:

1. Create the 'authors' and 'books' tables:

```
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)
);
DESCRIBE authors;
DESCRIBE books;
```

2. Insert values into the tables and display them:

```
INSERT INTO authors 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;
SELECT * FROM books;
```

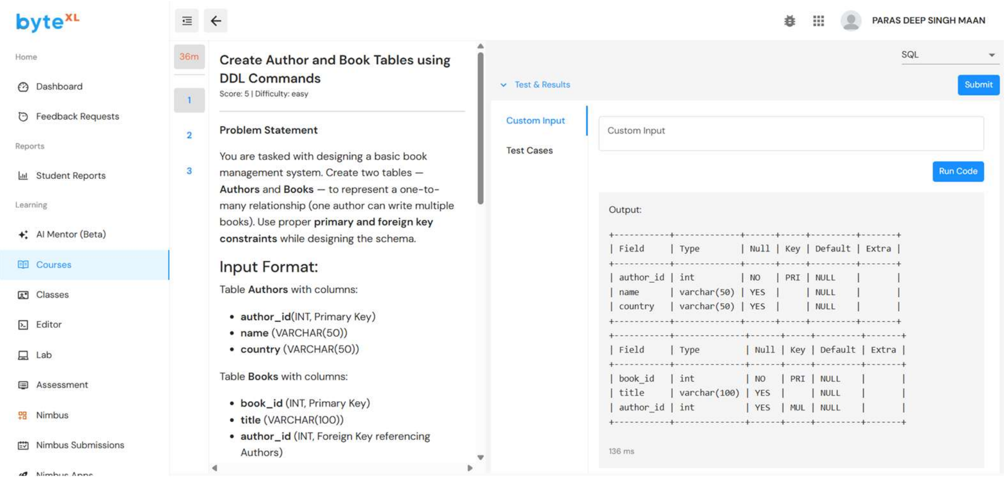
3. Retrieve titles of all books along with the author's name and country:

```
SELECT title, name, country
FROM authors a
INNER JOIN books b ON a.author_id = b.author_id;
```

CONCLUSION:

SQL JOIN clauses are vital for effectively combining related data from normalized tables, enabling the retrieval of comprehensive datasets. Choosing the appropriate JOIN type—INNER, LEFT, RIGHT, or FULL—is crucial depending on whether unmatched records are to be included. A sound understanding of JOIN operations enhances the efficiency of database queries and supports the development of meaningful reports in relational database systems.

SCREENSHOTS:



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))

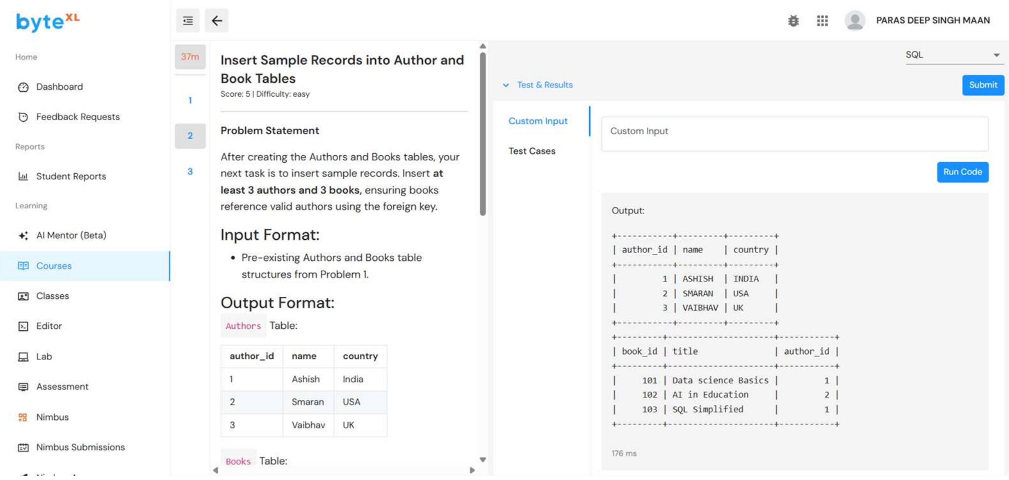
Table **Books** with columns:

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

Output:

```
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| author_id | int         | NO   | PRI | NULL    |       |
| name      | varchar(50) | YES  |     | NULL    |       |
| country   | varchar(50) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| book_id   | int          | NO   | PRI | NULL    |       |
| title     | varchar(100) | YES  |     | NULL    |       |
| author_id | int          | YES  | MUL | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
```

136 ms



Insert Sample Records into Author and Book Tables

Score: 5 | Difficulty: easy

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:

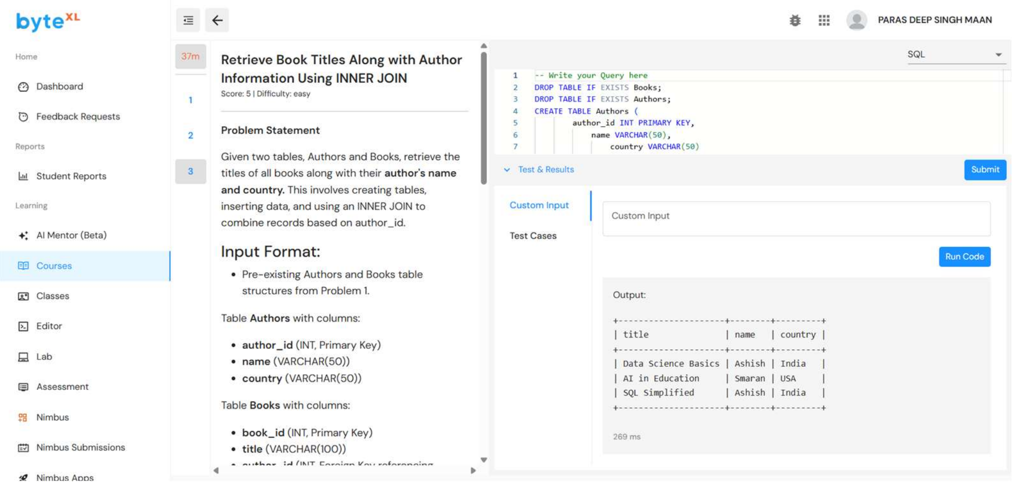
author_id	name	country
1	Ashish	India
2	Smaran	USA
3	Vaibhav	UK

`Books` Table:

Output:

```
+-----------+---------+---------+
| author_id | name    | country |
+-----------+---------+---------+
|         1 | ASHISH  | INDIA   |
|         2 | SMARAN  | USA     |
|         3 | VAIBHAV | UK      |
+-----------+---------+---------+
+---------+--------------------+-----------+
| book_id | title              | author_id |
+---------+--------------------+-----------+
|     101 | Data science Basics |         1 |
|     102 | AI in Education    |         2 |
|     103 | SQL Simplified     |         1 |
+---------+--------------------+-----------+
```

176 ms



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
-- Write your Query here
DROP TABLE IF EXISTS Books;
DROP TABLE IF EXISTS Authors;
CREATE TABLE Authors (
        author_id INT PRIMARY KEY,
            name VARCHAR(50),
                country VARCHAR(50)
```

Output:

```
+--------------------+--------+---------+
| title              | name   | country |
+--------------------+--------+---------+
| Data Science Basics | Ashish | India   |
| AI in Education    | Smaran | USA     |
| SQL Simplified     | Ashish | India   |
+--------------------+--------+---------+
```

269 ms