EXPERIMENT-01

AIM:

(i) Create Author and Book Tables using DDL Commands

(ii) Insert Sample Records into Author and Book Tables

(iii) Retrieve Book Titles Along with Author Information Using INNER JOIN

OBJECTIVE:

The objective of this experiment is to understand the core components of database schema design, particularly the creation and linking of tables using primary and foreign keys.
It also aims to strengthen the practical knowledge of DDL (Data Definition Language) and DML (Data Manipulation Language) operations, including table creation, data insertion, and joining tables to retrieve meaningful insights.
By performing this experiment on the ByteSQL platform, students will gain hands-on experience in relational database management and writing efficient SQL queries for real-world data modeling scenarios.

PROCEDURE:

- Launch the ByteSQL platform to perform SQL operations in an interactive environment.
- Use CREATE TABLE statements to define the Authors table with the following fields:

i. author_id (Primary Key)

ii. name (VARCHAR)

iii. country (VARCHAR)

- Define the Books table using CREATE TABLE with the fields:

i. book_id (Primary Key)

ii. title (VARCHAR)

iii. author_id (Foreign Key referencing Authors.author_id)

- Insert sample data into the Authors table using INSERT INTO commands with at least three distinct authors.
- Insert sample data into the Books table using INSERT INTO commands while ensuring each book is linked to a valid author via the author_id foreign key.
- Use an INNER JOIN SQL query to combine both tables and retrieve the book titles, author names, and author countries, matching records based on the common author_id.
- Validate the results by ensuring that each book is correctly displayed with its corresponding author's information as per the join condition.

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:

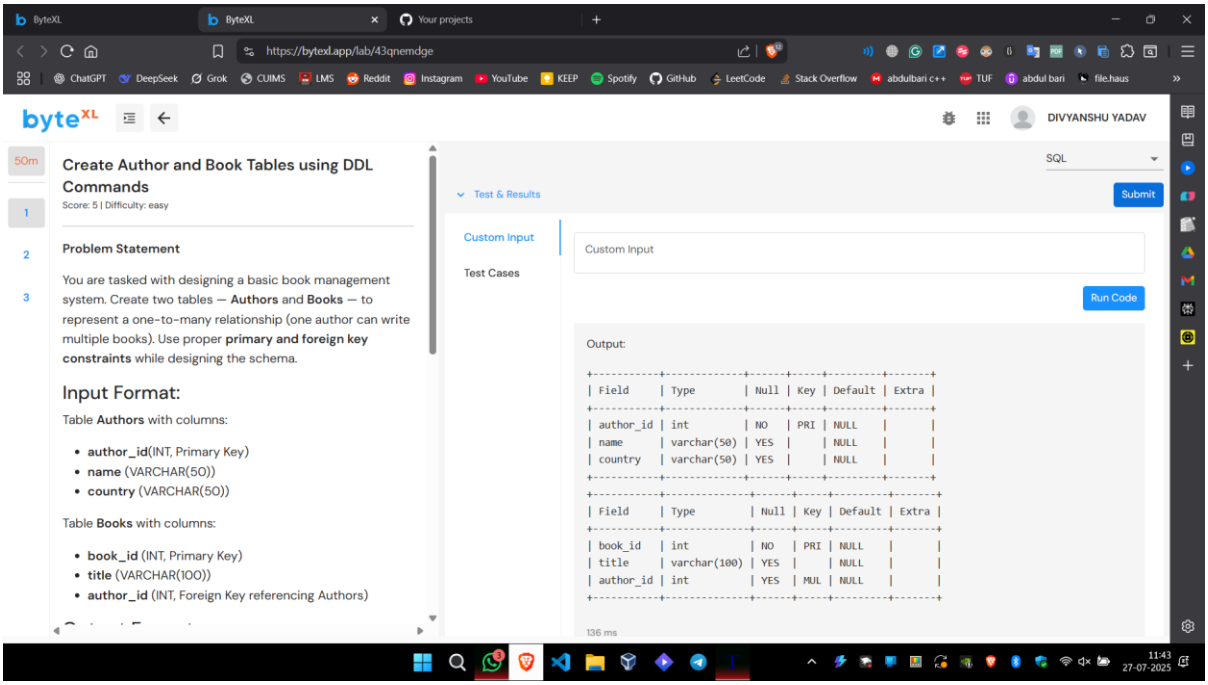
CREATE TABLE Authors (author_id INT PRIMARY KEY, name VARCHAR(50), country VARCHAR(50));

CREATE TABLE Books (book_id INT PRIMARY KEY, title VARCHAR(100), author_id INT, FOREIGN KEY (author_id) REFERENCES Authors(author_id));

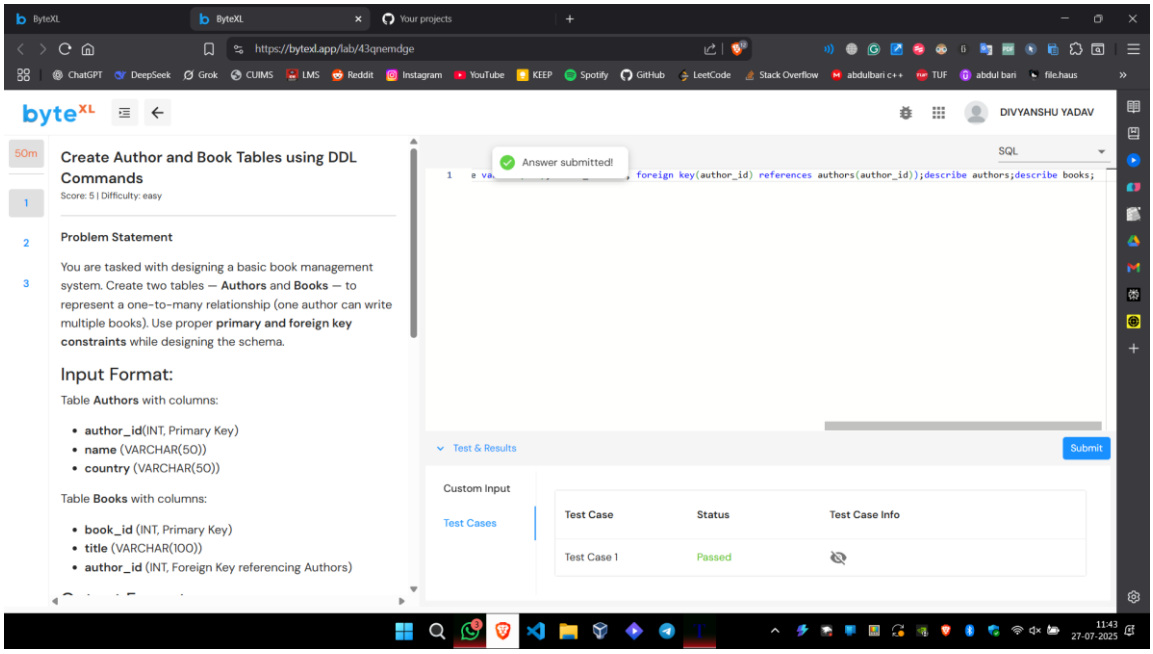
DESCRIBE Authors;

DESCRIBE Books;

OUTPUT 1:



TEST CASE 1:



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:

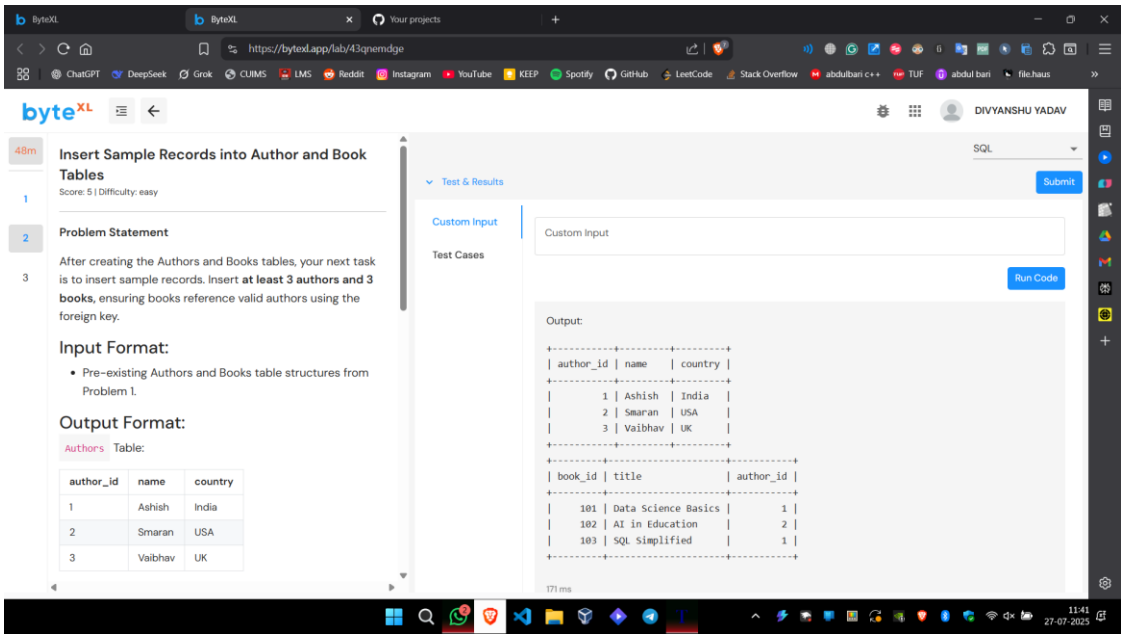
INSERT INTO Authors VALUES (1, 'Ashish', 'India'), (2, 'Smaran', 'USA'), (3, 'Vaibhav', 'UK');

INSERT INTO Books VALUES (101, 'Data Science Basics', 1), (102, 'AI in Education', 2), (103, 'SQL Simplified', 1);

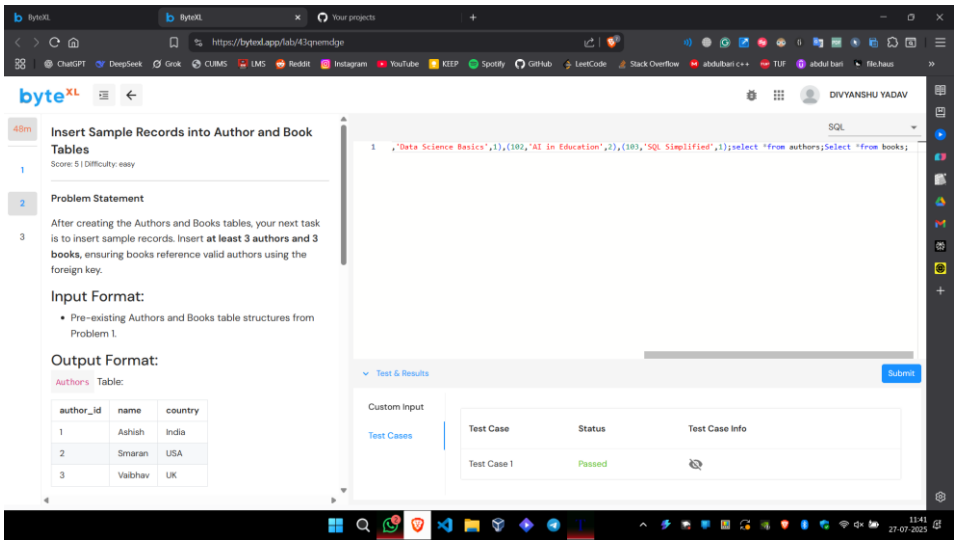
SELECT * FROM Authors;

SELECT * FROM Books;

OUTPUT 2:



TEST CASE 2:



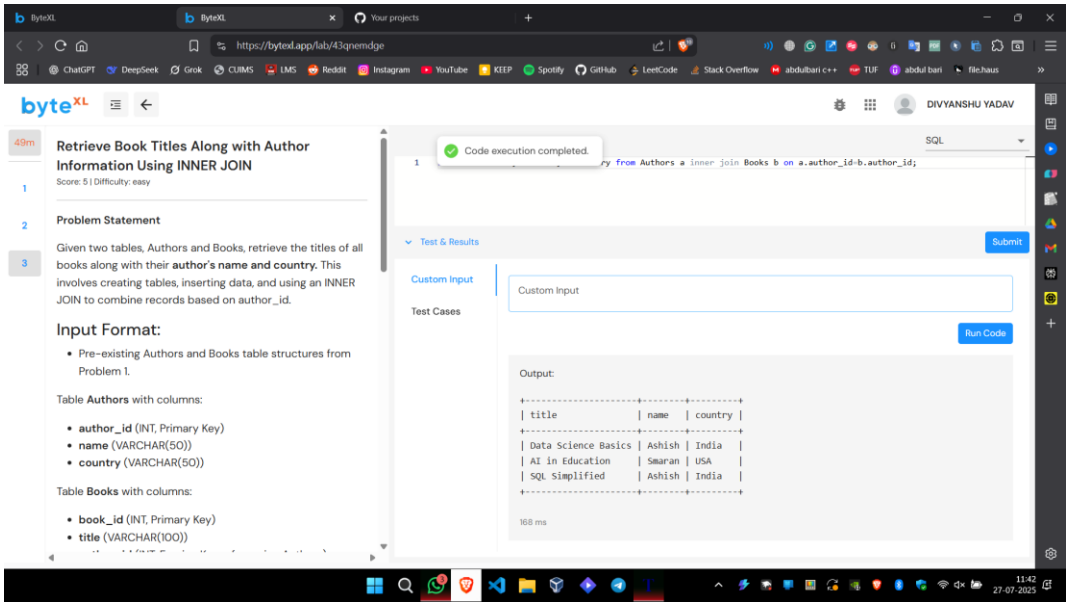
Problem Statement 3: Given two tables, Authors and Books, retrieve the titles of all books along with their author's name and country. This involves creating tables, inserting data, and using an INNER JOIN to combine records based on author_id.

Query 3:

```
SELECT Books.title, Authors.name, Authors.country

FROM Books INNER JOIN Authors ON Books.author_id = Authors.author_id;
```

OUTPUT 3:



TEST CASE 3: