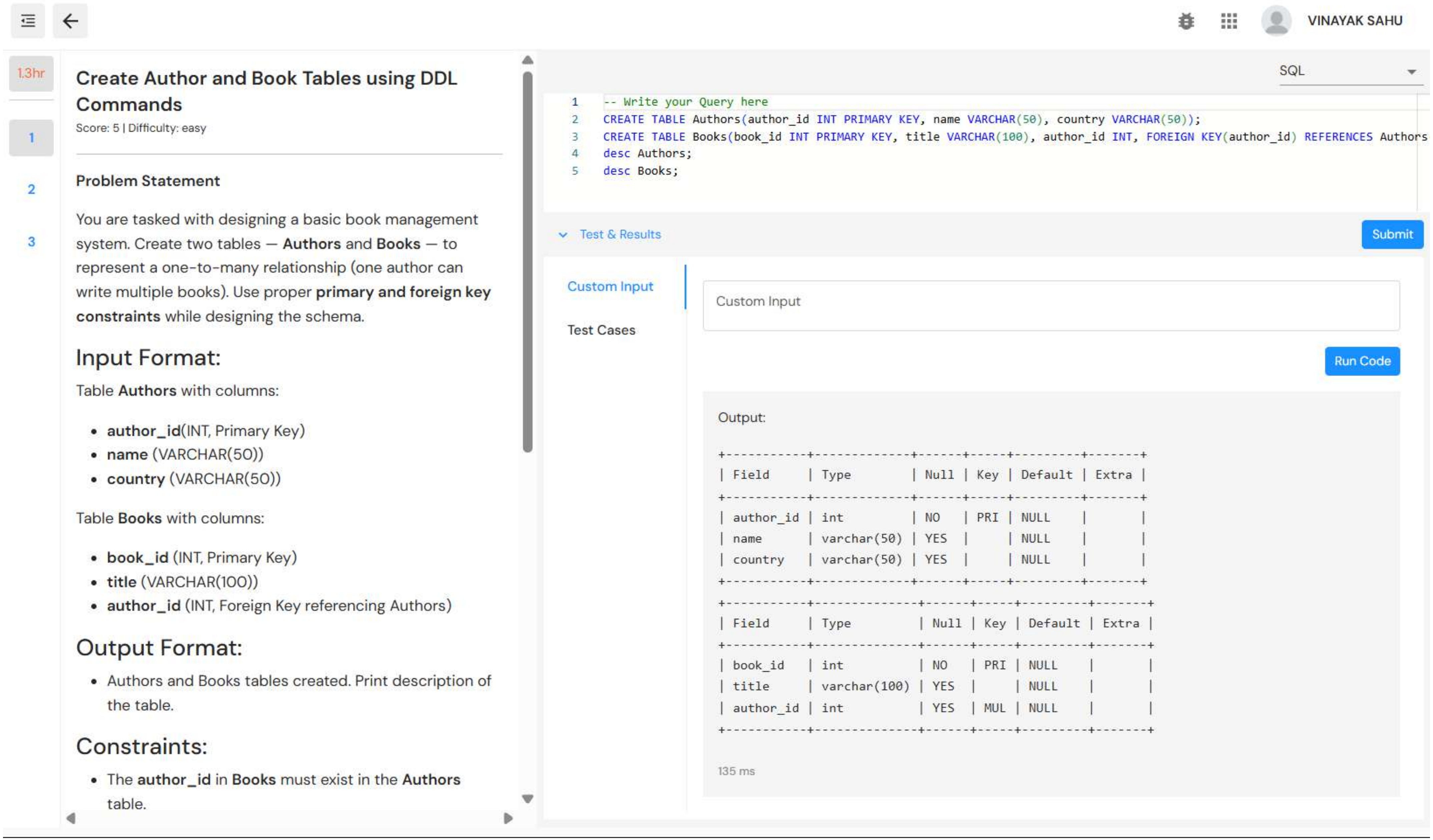


Create Author and Book Tables using DDL Commands

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 Format:

- Authors and Books tables created. Print description of the table.

Constraints:

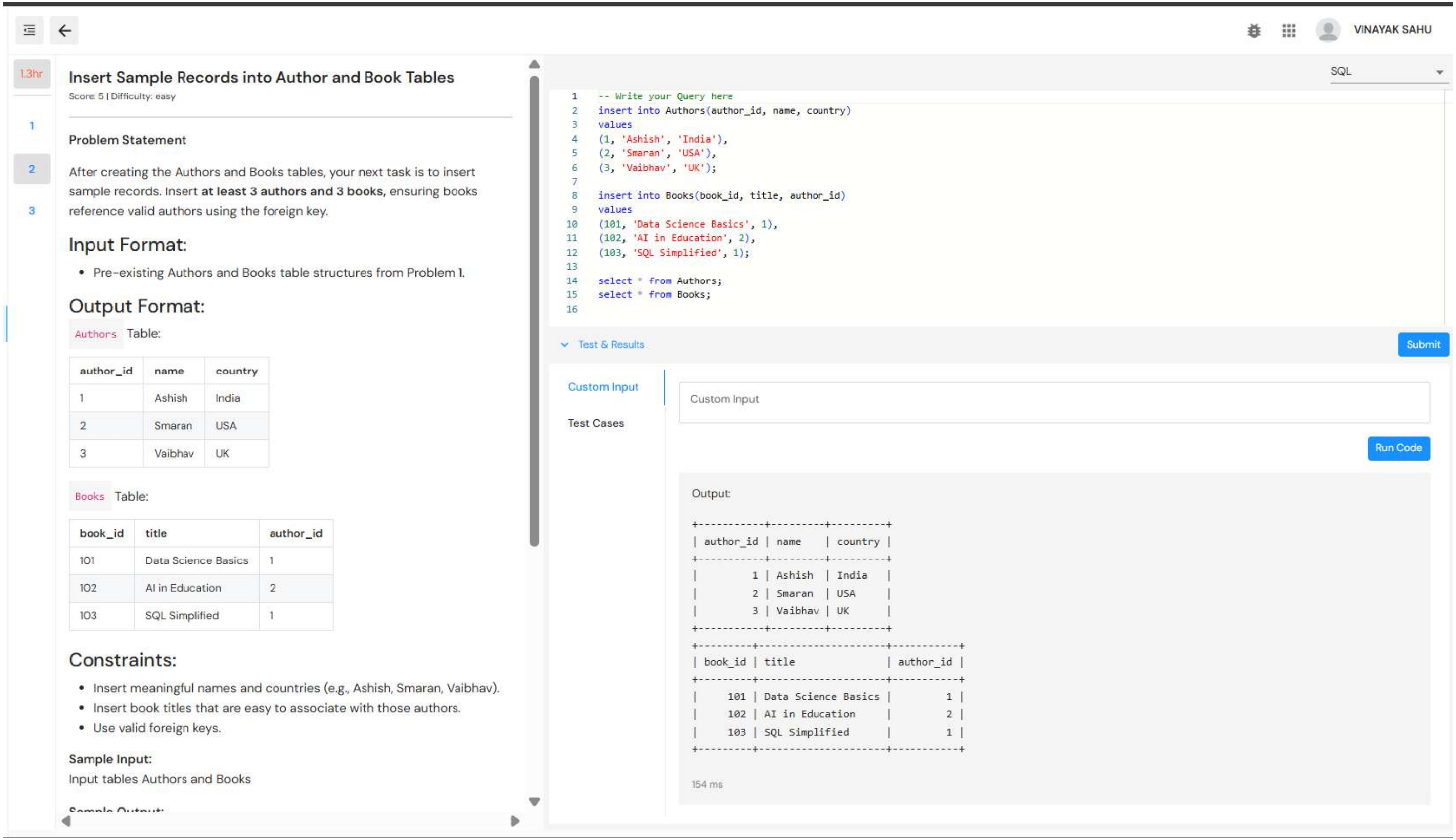
- The **author_id** in **Books** must exist in the **Authors** table.

```sql
-- Write your Query here
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
desc Authors;
desc Books;
```

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    |       |
+-----------+--------------+------+-----+---------+-------+
```

135 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:

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.

Sample Input:

Input tables Authors and Books

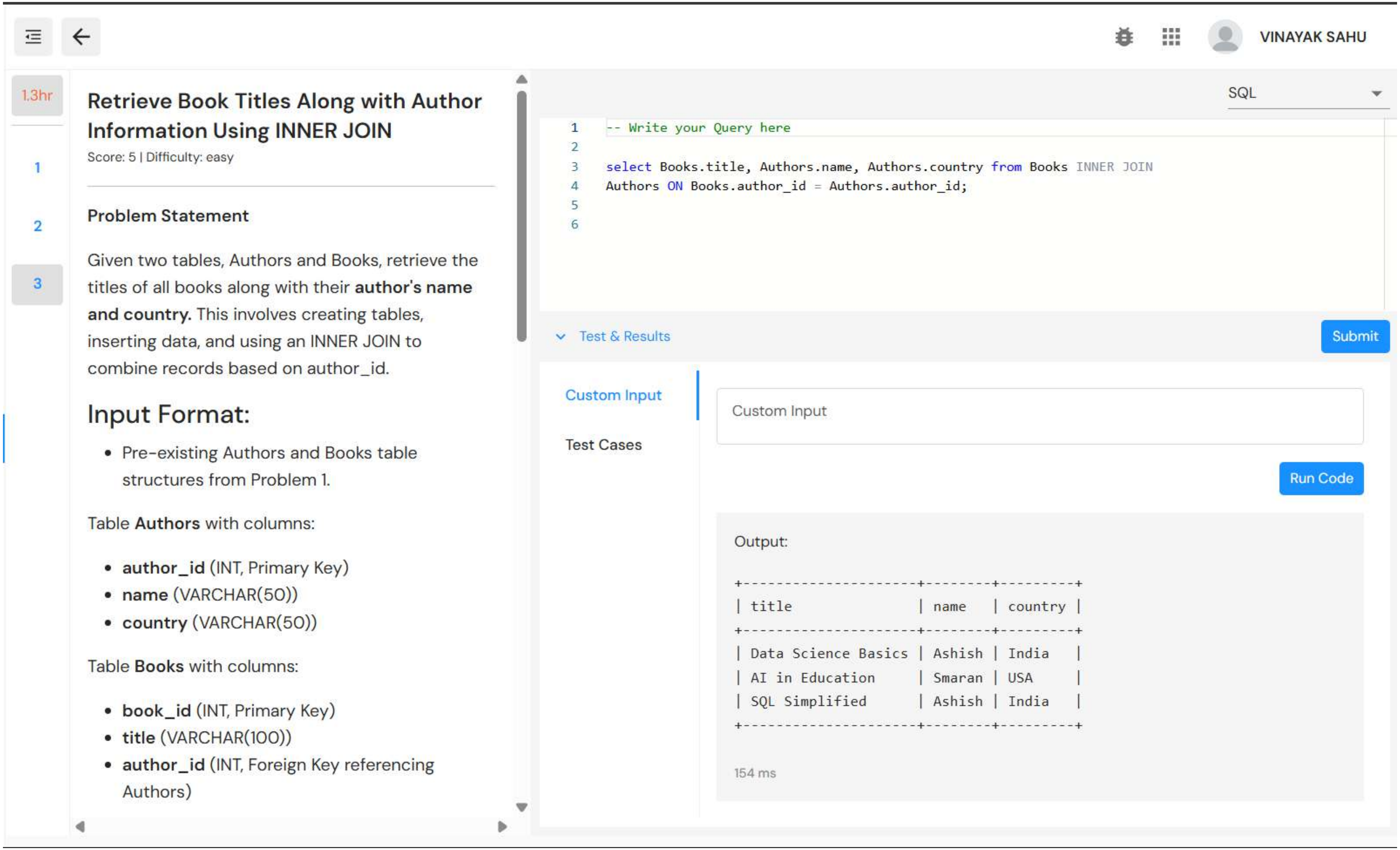
Sample Output:

```sql
-- Write your Query here
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:

```
+-----------+---------+---------+
| 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 |
+---------+---------------------+-----------+
```

154 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))
- **author_id** (INT, Foreign Key referencing Authors)

```sql
-- Write your Query here

select Books.title, Authors.name, Authors.country from Books INNER JOIN
Authors ON Books.author_id = Authors.author_id;


```

Output:

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

154 ms