

The screenshot shows the ByteXL web application interface. The browser tabs at the top include 'ByteXL', 'Welcome to Chandigarh University', and others. The address bar shows 'bytexl.app/lab/43qnemdgc'. The application has a sidebar on the left with navigation links: Home, Dashboard, Feedback Requests, Reports, Student Reports, Learning, AI Mentor (Beta), Courses (highlighted), Classes, Editor, Lab, Assessment, Nimbus, Nimbus Submissions, and Nimbus Apps. The main content area is titled 'Create Author and Book Tables using DDL Commands' with a score of 5 and difficulty of easy. The problem statement asks to design a basic book management system with two tables: Authors and Books. The SQL editor contains the following code:

```
1 create table Authors(author_id int primary key,name varchar(50),country varchar(50));
2 create table Books(book_id int primary key,title varchar(100),author_id int,foreign key(author_id) references Authors(author_id));
3 desc Authors;
4 desc Books;
```

Below the editor is the 'Test & Results' section, which includes a table with the following data:

Test Case	Status	Test Case Info
Test Case 1	Passed	



JITEN MOURYA

SQL



Test & Results

Submit

Custom Input

Custom Input

Test Cases

Run Code

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	

126 ms

Experimental learning online platform

ByteXL

ByteXL

Welcome to Chandigarh University

byteXL.app/lab/43qnemdgc

byteXL

Home

Dashboard

Feedback Requests

Reports

Student Reports

Learning

AI Mentor (Beta)

Courses

Classes

Editor

Lab

Assessment

Nimbus

Nimbus Submissions

Nimbus Apps

1.4hr

Create Author and Book Tables using DDL Commands

Score: 51 | 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

SQL

```
1 create table Authors(author_id int primary key,name varchar(50),country varchar(50));
2 create table Books(book_id int primary key,title varchar(100),author_id int,foreign key(author_id) references Authors(author_id));
3 desc Authors;
4 desc Books;
```

Test & Results

Submit

31°C

Search

10:09 AM