

EXPERIMENT 1

AIM –

Create Author and Book tables and insert sample records into author and book tables and then retrieve book titles along with author information using Inner Join.

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;

byteXL

Home

Dashboard

Feedback Requests

Reports

Student Reports

Learning

AI Mentor (Beta)

Courses

Classes

Editor

Lab

Assessment

Nimbus

Nimbus Submissions

Nimbus Apps

Community

Organizations

Branches

Rathnare

44m

Create Author and Book Tables using DDL

Commands

Score: 5 | Difficulty: easy

1

2

3

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.
- Use appropriate data types and constraints.
- name** and **country** should allow up to 50 characters.

1

2

3

4

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;

Test & Results

Submit

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	

132 ms

QUERY 2 –

```
INSERT INTO Authors (Author_id, name, country)
VALUES
```

```
(1, 'Ashish', 'India'),
```

```
(2, 'Smaran', 'USA'),
```

```
(3, 'Vaibhav', 'UK');
```

```
-- Insert into Books table (match book_id values
exactly)
```

```
INSERT INTO Books (Book_id, title, author_id) VALUES
```

```
(101, 'Data Science Basics', 1),
```

```
(102, 'AI in Education', 2),
```

```
(103, 'SQL Simplified', 1);
```

```
-- Display the tables
```

```
SELECT * FROM Authors;
```

```
SELECT * FROM Books;
```

byteXL

Home

Dashboard

Feedback Requests

Reports

Student Reports

Learning

AI Mentor (Beta)

Courses

Classes

Editor

Lab

Assessment

Nimbus

Nimbus Submissions

Nimbus App

Community

Organizations

Branches

Database

44m

Insert Sample Records into Author and Book Tables

Score: 5 | Difficulty: easy

1

2

3

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,

Test & Results

Submit

SQL

```

1 INSERT INTO Authors(author_id, name, country) VALUES(1,'Ashish','India'),(2,'Smaran','USA'),(3,'Vaibhav','UK');
2 INSERT INTO Books (book_id,title,author_id) VALUES(101,'Data Science Basics',1),(102,'AI in Education',2),(103,'SQL Simplified',1);
3 SELECT * FROM Authors;
4 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 |
+-----+-----+-----+

```

162 ms

QUERY 3 –

SELECT

B.title AS title,

A.name AS name,

A.country AS country

FROM

Books AS B

INNER JOIN

Authors AS A ON B.author_id = A.author_id;

The screenshot displays the byteXL SQL editor interface. On the left is a sidebar with navigation links: Home, Dashboard, Feedback Requests, Reports, Student Reports, Learning, AI Mentor (Beta), Courses, Classes, Editor, Lab, Assessment, Nimbus, Nimbus Submissions, Nimbus Apps, Community, Organizations, Branches, and Retry/Pass. The main content area is titled 'Retrieve Book Titles Along with Author Information Using INNER JOIN' with a score of 5 and difficulty of 'easy'. It includes a 'Problem Statement' about joining Authors and Books tables, 'Input Format' details for the tables, 'Output Format' requirements, and 'Constraints'. On the right, the SQL editor shows the following query:

```
1 SELECT
2 B.title AS title,
3 A.name AS name,
4 A.country AS country
5 FROM
6 Books AS B
7 INNER JOIN
8 Authors AS A ON
9 B.author_id=A.author_id;
```

At the bottom right of the editor is a 'Submit' button.

Output:

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

179 ms