

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:

The screenshot shows the byteXL SQL editor interface. On the left, a sidebar contains 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 area displays the results of SQL queries. The first query, 'DESCRIBE Authors;', shows the schema for the Authors table with columns: author_id (int, NO, PRI, NULL), name (varchar(50), YES, NULL), and country (varchar(50), YES, NULL). The second query, 'DESCRIBE Books;', shows the schema for the Books table with columns: book_id (int, NO, PRI, NULL), title (varchar(100), YES, NULL), and author_id (int, YES, MUL, NULL). The right panel shows the 'Test & Results' section with a 'Custom Input' field and a 'Run Code' button. The output of the queries is displayed in a table format, showing the field names, types, nullability, keys, and defaults.

Field	Type	Null	Key	Default	Extra
author_id	int	NO	PRI	NULL	
name	varchar(50)	YES		NULL	
country	varchar(50)	YES		NULL	

books Table:

Field	Type	Null	Key	Default	Extra
book_id	int	NO	PRI	NULL	
title	varchar(100)	YES		NULL	
author_id	int	YES	MUL	NULL	

Sample Test Cases

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	

144 ms

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:

The screenshot shows the ByteXL interface with the problem statement and the output of the SQL queries. The problem statement is: "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." The input format is: "Pre-existing Authors and Books table structures from Problem 1." The output format is: "Authors Table" and "Books Table". The output shows the following tables:

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

TEST CASE 2:

The screenshot shows the ByteXL interface with the test case results. The test case is: "Insert Sample Records into Author and Book Tables". The test case status is: "Passed". The test case info is: "Test Case 1".

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:

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 (highlighted), Classes, Editor, Lab, Assessment, Nimbus, and Nimbus Submissions. 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, input format, and table definitions for Authors and Books. The SQL query is shown in a code editor, and the execution results are displayed in a table format.

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)

SQL Query:

```
1 SELECT
2   Books.title,
3   Authors.name,
4   Authors.country
5 FROM
6   Books
7 INNER JOIN
```

Test & Results

Custom Input: Custom Input

Run Code

Output:

title	name	country
Data Science Basics	Ashish	India
AI in Education	Smaran	USA
SQL Simplified	Ashish	India

157 ms

TEST CASE 3:

29hr

1

2

3

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)

Output Format:

- A list of books with their **title, name of the author, and**

SQL

```
1 SELECT Books.title, Authors.name, Authors.country
2 FROM Books INNER JOIN Authors ON Books.author_id = Authors.author_id;
3
```

Test & Results

Submit

Custom Input

Test Cases

Test Case	Status	Test Case Info
Test Case 1	Passed	