

1. Define a schema for a Library Management System with the following entities:

- Books
- Authors
- Members
- Borrow_Records

1. Authors

Represents authors of books.

Column Name	Data Type	Constraints
AuthorID	INT	PRIMARY KEY, AUTO_INCREMENT
FirstName	VARCHAR(50)	NOT NULL
LastName	VARCHAR(50)	NOT NULL
DOB	DATE	NULL
Bio	TEXT	NULL

2. Books

Represents books in the library.

Column Name	Data Type	Constraints
BookID	INT	PRIMARY KEY, AUTO_INCREMENT
Title	VARCHAR(100)	NOT NULL
AuthorID	INT	FOREIGN KEY REFERENCES Authors(AuthorID)
ISBN	VARCHAR(20)	UNIQUE, NOT NULL
Publisher	VARCHAR(50)	NULL
YearPublished	YEAR	NULL
CopiesAvailable	INT	NOT NULL, DEFAULT 1

3. Members

Represents library members.

Column Name	Data Type	Constraints
MemberID	INT	PRIMARY KEY, AUTO_INCREMENT
FirstName	VARCHAR(50)	NOT NULL
LastName	VARCHAR(50)	NOT NULL
Email	VARCHAR(100)	UNIQUE, NOT NULL
PhoneNumber	VARCHAR(15)	NULL
MembershipDate	DATE	NOT NULL

4. Borrow_Records

Represents book borrow history.

Column Name	Data Type	Constraints
RecordID	INT	PRIMARY KEY, AUTO_INCREMENT
BookID	INT	FOREIGN KEY REFERENCES Books(BookID)
MemberID	INT	FOREIGN KEY REFERENCES Members(MemberID)
BorrowDate	DATE	NOT NULL
ReturnDate	DATE	NULL
DueDate	DATE	NOT NULL
Status	VARCHAR(20)	NOT NULL, CHECK (Status IN ('Borrowed','Returned','Overdue'))

Relationships

Books ↔ **Authors** → Many-to-One

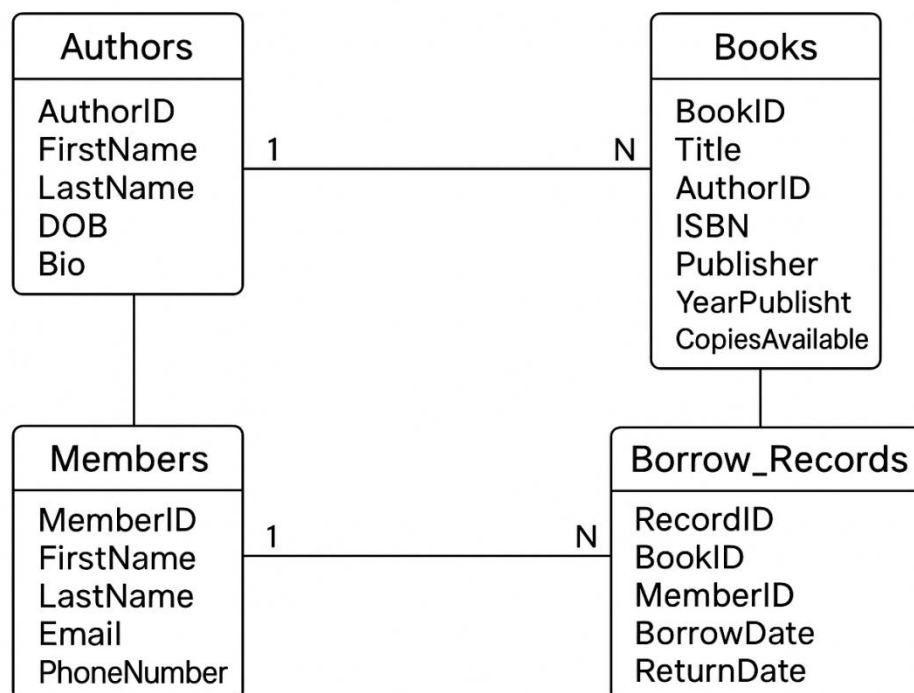
Each book has one author, but an author can write many books.

Borrow_Records ↔ **Books** → Many-to-One

A book can be borrowed many times; each borrow record refers to one book.

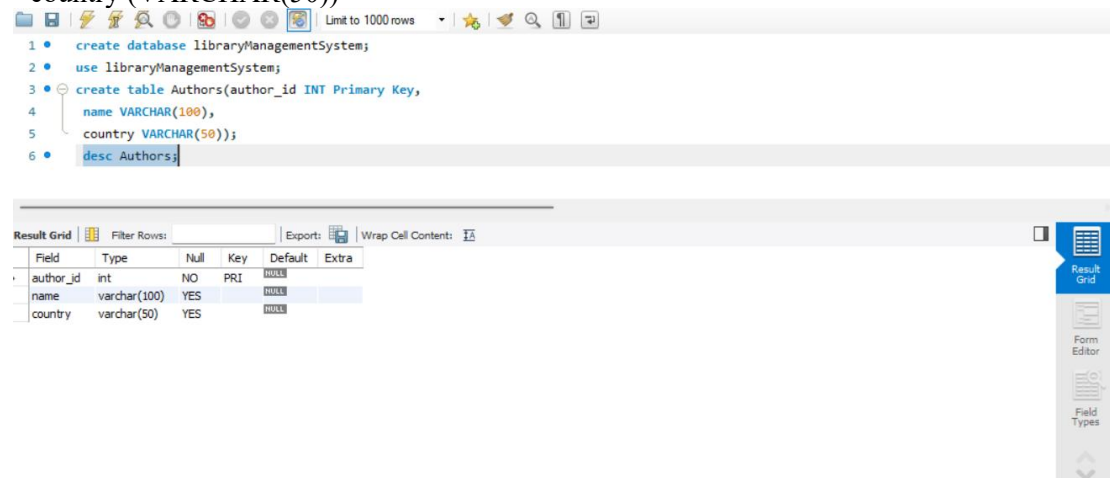
Borrow_Records ↔ **Members** → Many-to-One

A member can borrow multiple books; each borrow record refers to one member.



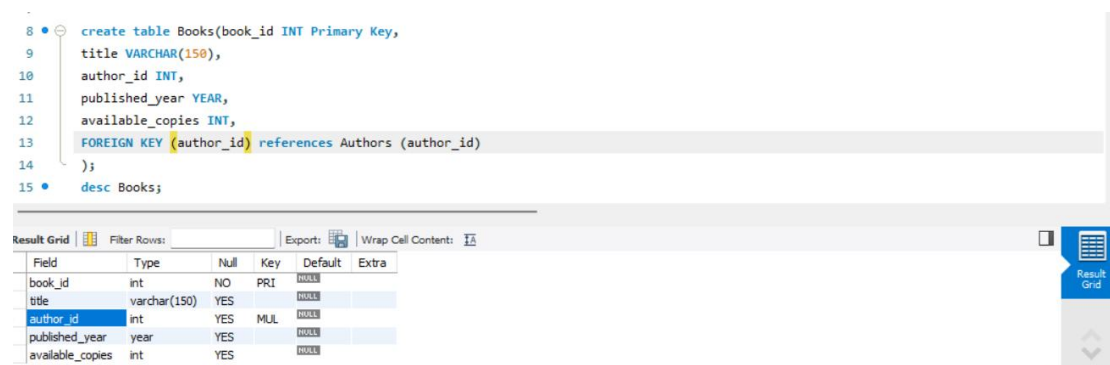
2. Write the SQL command to create a table Authors with the following fields:

- author_id (Primary Key, INT)
- name (VARCHAR(100))
- country (VARCHAR(50))



3. Write the SQL command to create a table Books with the following fields:

- book_id (Primary Key, INT)
- title (VARCHAR(150))
- author_id (Foreign Key referencing Authors)
- published_year (YEAR)
- available_copies (INT)



4. Write the SQL command to create a table Members with:

- member_id (Primary Key, INT)
- name (VARCHAR(100))
- email (VARCHAR(100), unique)
- phone (VARCHAR(15))

```

17 • create table Members(
18     member_id INT Primary Key,
19     name VARCHAR(100),
20     email VARCHAR(100) unique,
21     phone VARCHAR(15)
22 );
23
24 • DESC Members;

```

Field	Type	Null	Key	Default	Extra
member_id	int	NO	PRI		
name	varchar(100)	YES			
email	varchar(100)	YES	UNI		
phone	varchar(15)	YES			

Result 3 x Read Only

5. Write the SQL command to create a table Borrow_Records with:

- record_id (Primary Key, INT)
- member_id (Foreign Key referencing Members)
- book_id (Foreign Key referencing Books)
- borrow_date (DATE)
- return_date (DATE)

```

26 • create table Borrow_Records(
27     record_id INT Primary Key,
28     member_id int,
29     book_id int,
30     borrow_date DATE,
31     return_date DATE,
32     FOREIGN KEY (member_id) references Members (member_id),
33     FOREIGN KEY (book_id) references Books (book_id));
34
35 • DESC Borrow_Records;

```

Field	Type	Null	Key	Default	Extra
record_id	int	NO	PRI		
member_id	int	YES	MUL		
book_id	int	YES	MUL		
borrow_date	date	YES			
return_date	date	YES			

Result 4 x Read Only

6. Modify the Books table to add a column genre of type VARCHAR(50).

```

37 • Alter table books add genre varchar(50);
38 • desc books;

```

Field	Type	Null	Key	Default	Extra
book_id	int	NO	PRI		
title	varchar(150)	YES			
author_id	int	YES	MUL		
published_year	year	YES			
available_copies	int	YES			

Result 5 x Read Only

7. Write the SQL command to drop the Borrow_Records table

```

40 • Drop table Borrow_Records;

```

#	Time	Action	Message	Duration / Fetch
16	20:11:03	DESC Borrow_Records	5 row(s) returned	0.000 sec / 0.000 sec
17	20:13:25	Alter table books add genre varchar(50)	Error Code: 1146. Table 'librarymanagementsystem.book' doesn't exist	0.015 sec
18	20:14:00	Alter table books add genre varchar(50)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec
19	20:14:20	Alter table books add genre varchar(50)	Error Code: 1060. Duplicate column name 'genre'	0.000 sec
20	20:14:25	desc books	6 row(s) returned	0.000 sec / 0.000 sec
21	20:15:36	Drop table Borrow_Records	0 row(s) affected	0.016 sec

8. Insert 3 records into the Authors table.

42	•	insert into Authors (author_id,name,country) values(1,'Chaitali','India'),(2,'ronak','India'),(3,'rahul','India');	
----	---	--	--

#	Time	Action	Message	Duration / Fetch
21	20:15:36	Drop table Borrow_Records	0 row(s) affected	0.016 sec
22	20:18:23	insert authors values(1,'Chaitali','India'),(2,'ronak','India'),(3,'rahul','India')	Error Code: 1241. Operand should contain 1 column(s)	0.000 sec
23	20:20:33	insert into authors(author_id,name,country) values(1,'Chaitali','India'),(2,'ronak','India'),(3,'rahul','India')	Error Code: 1241. Operand should contain 1 column(s)	0.015 sec
24	20:21:02	insert into Authors (author_id,name,country) values(1,'Chaitali','India'),(2,'ronak','India'),(3,'rahul','India')	Error Code: 1241. Operand should contain 1 column(s)	0.000 sec
25	20:22:09	insert into Authors (author_id,name,country) values(1,'Chaitali','India'),(2,'ronak','India'),(3,'rahul','India')	Error Code: 1241. Operand should contain 1 column(s)	0.000 sec
26	20:24:04	insert into Authors (author_id,name,country) values(1,'Chaitali','India'),(2,'ronak','India'),(3,'rahul','India')	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.015 sec

9. Insert 5 books into the Books table

Note:When we use foreign key that parent must need value which we use in child class.

46	•	select * from Authors;	
47	•	select * from books;	
48			

#	Time	Action	Message	Duration / Fetch
3	22:50:22	insert into books (book_id, title, author_id, published_year, available_copies) values(1,'AnimalFam',101,2003,...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('librarymanagementsystem',...	0.031 sec
4	22:55:48	insert into Authors (author_id, name, country) values (101,'mohini','India'),(102,'Devika','India'),(103,'yash','India'...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
5	22:55:53	insert into books (book_id, title, author_id, published_year, available_copies) values(1,'AnimalFam',101,2003,...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
6	22:56:38	select * from Authors LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
7	22:56:38	select * from books LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
8	22:56:51	select * from Authors LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

book_id	title	author_id	published_year	available_copies	genre
1	AnimalFam	101	2003	5	...
2	The old man	102	2005	4	...
3	The Sea	103	2006	5	...
4	The Blue Umbrella	104	2001	5	...
5	Mr.fox	105	2005	7	...

#	Time	Action	Message	Duration / Fetch
4	22:55:48	insert into Authors (author_id, name, country) values (101,'mohini','India'),(102,'Devika','India'),(103,'yash','India'...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec
5	22:55:53	insert into books (book_id, title, author_id, published_year, available_copies) values(1,'AnimalFam',101,2003,...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
6	22:56:38	select * from Authors LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
7	22:56:38	select * from books LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
8	22:56:51	select * from Authors LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
9	22:58:33	select * from books LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

10. Insert 3 members into the Members table

49	•	insert into Members (member_id,name,email,phone) value (1,'chaitali','chaitali@gmail.com','900667755'),(2,'mohini','mohini@gmail.com','9004...	
50	•	select * from Members;	
51			

member_id	name	email	phone
1	chaitali	chaitali@gmail.com	900667755
2	mohini	mohini@gmail.com	9004433588
3	devika	devika@gmail.com	9006655799

#	Time	Action	Message	Duration / Fetch
1	22:58:33	select * from Members	3 row(s) returned	0.000 sec / 0.000 sec

11. Insert 4 borrow records into the Borrow_Records table

```

52 • insert into Borrow_Records (record_id,member_id,book_id,borrow_date,return_date) values (111,1,1,'2023-04-23','2023-05-24'),(112,2,2,'2023-
53 • select * from Borrow_Records;

```

record_id	member_id	book_id	borrow_date	return_date
111	1	1	2023-04-23	2023-05-24
112	2	2	2023-04-24	2023-05-25
113	3	3	2023-06-25	2023-07-24
NULL	NULL	NULL	NULL	NULL

Borrow_Records 6 x

12. Write an SQL query to select all books where published_year is after 2015

```

56 • select * from books where published_year >2015;
57

```

book_id	title	author_id	published_year	available_copies	genre
2	The old man	102	2016	4	NULL
4	The Blue Umbrella	104	2016	5	NULL
5	Mr.fox	105	2017	7	NULL
NULL	NULL	NULL	NULL	NULL	NULL

books 11 x

13. Write a SQL query to create a foreign key & primary key relationship between two tables.

```

create table Authors(author_id INT Primary Key,
name VARCHAR(100),
country VARCHAR(50));
desc Authors;

```

```

create table Books(book_id INT Primary Key,
title VARCHAR(150),
author_id INT,
published_year YEAR,
available_copies INT,
FOREIGN KEY (author_id) references Authors (author_id)
);

```

```

6 • desc Authors;
7
8 • create table Books(book_id INT Primary Key,

```

Field	Type	Null	Key	Default	Extra
author_id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
country	varchar(50)	YES		NULL	

Result 12 x

```

14  )
15  desc Books;
16
17  create table Members(

```

Field	Type	Null	Key	Default	Extra
book_id	int	NO	PRI		
title	varchar(150)	YES			
author_id	int	YES	MUL		
published_year	year	YES			
available_copies	int	YES			
genre	varchar(50)	YES			

Result 13 x Read Only

14. Write an SQL query to find all members who have borrowed the book with title 'Database Systems'.

```

56  select * from books where published_year >2015;
57  SELECT M.member_id, M.name, M.email
58  FROM Members M
59  JOIN Borrow_Records BR ON M.member_id = BR.member_id
60  JOIN Books B ON BR.book_id = B.book_id
61  WHERE B.title = 'Database Systems';
62

```

member_id	name	email	phone
1	chaitali	chaitali@gmail.com	900667755
2	mohini	mohini@gmail.com	9004433588
3	devika	devika@gmail.com	9006655799

Members 21 x Apply Revert

```

56  select * from books where published_year >2015;
57  SELECT M.member_id, M.name, M.email
58  FROM Members M
59  JOIN Borrow_Records BR ON M.member_id = BR.member_id
60  JOIN Books B ON BR.book_id = B.book_id
61  WHERE B.title = 'Database Systems';

```

member_id	name	email
2	mohini	mohini@gmail.com

books 22 Result 23 x Read Only

15. Update the available_copies column of a specific book (choose any book) by reducing it by 1 after it is borrowed.

```

47  select * from books;
48
49  insert into Members (member_id,name,email,phone) value (1,'chaitali','chaitali@gmail.com','900667755'),(2,'mohini','mohini@gmail.com','9004
50  select * from Members;
51

```

book_id	title	author_id	published_year	available_copies	genre
1	AnimalFarm	101	2015	5	
2	Database Systems	102	2016	5	
3	The Sea	103	2011	5	
4	Database Systems	104	2016	5	
5	Mr.fox	105	2017	7	

books 27 x Apply Revert

```

63 • UPDATE Books
64 SET available_copies = available_copies - 1
65 WHERE title = 'Database Systems'
66 AND author_id = 102 -- example author_id
67 AND available_copies > 0;
68 • select * from books;
69

```

book_id	title	author_id	published_year	available_copies	genre
1	AnimalFarm	101	2015	5	NOVEL
2	Database Systems	102	2016	4	NOVEL
3	The Sea	103	2011	5	NOVEL
4	Database Systems	104	2016	5	NOVEL
5	Mr.fox	105	2017	7	NOVEL
*	NULL	NULL	NULL	NULL	NULL

books 28 x Apply Revert

16. Delete a record from Members where member_id = 3

Note :where we have foreign key dependency of in child table that time we can't delete direct that member from table firstly, delete dependency in child table then delete that particular row of parent table.

```

70 • delete from Members where member_id = 3;
71 • delete from Borrow_Records where member_id = 3;
72
73 • select * from Members;
74

```

member_id	name	email	phone
1	chaitali	chaitali@gmail.com	900667755
2	mohini	mohini@gmail.com	9004433588
*	NULL	NULL	NULL

Members 30 x Apply Revert

17. Update a Book name record from Book table with id = 1.

```

68 • select * from books;
69

```

book_id	title	author_id	published_year	available_copies	genre
1	AnimalFarm	101	2015	5	NOVEL
2	Database Systems	102	2016	4	NOVEL
3	The Sea	103	2011	5	NOVEL
4	Database Systems	104	2016	5	NOVEL
5	Mr.fox	105	2017	7	NOVEL
*	NULL	NULL	NULL	NULL	NULL

books 31 x Apply Revert

```

75 • UPDATE Books
76 SET title = 'CPP'
77 WHERE book_id = 1;
78 • select * from books;
79

```

book_id	title	author_id	published_year	available_copies	genre
1	CPP	101	2015	5	NOVEL
2	Database Systems	102	2016	4	NOVEL
3	The Sea	103	2011	5	NOVEL
4	Database Systems	104	2016	5	NOVEL
5	Mr.fox	105	2017	7	NOVEL
*	NULL	NULL	NULL	NULL	NULL

books 32 x Apply Revert

18. Write an SQL query to list all books along with their authors' names.

```

80 • SELECT
81     B.book_id,
82     B.title,
83     A.name AS author_name,
84     A.country
85 FROM Books B
86 JOIN Authors A ON B.author_id = A.author_id;

```

book_id	title	author_name	country
1	CPP	mohini	India
2	Database Systems	Devika	India
3	The Sea	yash	India
4	Database Systems	rutu	india
5	Mr.fox	dewoo	india

Result 33 x Read Only

19. Write an SQL query to delete all books from the Books table where the published_year is before 2015.

: We need two step for above delete operation from Database.

Step:delete author data from authors table

DELETE FROM Authors WHERE author_id = 102;

Step 2: then delete the books data where published_year is before 2015

delete from books WHERE published_year < 2015;

20. Write an SQL query to **find all books that are never borrowed (i.e., no records in Borrow_Records).**

```

97 • SELECT B.book_id, B.title
98 FROM Books B
99 LEFT JOIN Borrow_Records BR ON B.book_id = BR.book_id
100 WHERE BR.book_id IS NULL;
101
102

```

book_id	title
3	The Sea
4	Database Systems
5	Mr.fox

Result 36 x Read Only