

```

CREATE DATABASE ITV_SQLPROJECT;
USE ITV_SQLPROJECT;

CREATE TABLE Authors (Author_ID TINYINT PRIMARY KEY , Author_name VARCHAR(100));
DESC Authors;

CREATE TABLE Genres (Genres_ID TINYINT PRIMARY KEY , Genre_name VARCHAR(100));
DESC Genres;

CREATE TABLE Books (Book_ID TINYINT PRIMARY KEY , Title VARCHAR(255),
Author_ID TINYINT,Foreign key (Author_ID) references Authors (Author_ID),
Genre_ID TINYINT , FOREIGN KEY(Genre_ID) references Genres(Genres_ID));
DESC Books;

CREATE TABLE USERS (User_ID TINYINT PRIMARY KEY ,User_name VARCHAR(100));
DESC Users;

CREATE TABLE Loans (Loan_ID TINYINT PRIMARY KEY ,Book_ID TINYINT , User_ID TINYINT
,Loan_date DATE , Return_Date DATE NULL,
FOREIGN KEY (Book_ID) references Books (Book_ID),
FOREIGN KEY (User_ID) references Users (User_ID));
DESC Loans;

INSERT INTO Authors VALUES (1 , 'John Doe'),(2 , 'Jane Smith'),(3 , 'Michael
Johnson'),(4, 'sarah Brown'),(5 , 'david Lee');

INSERT INTO Genres VALUES (1 , 'Fiction'),(2 , 'Non-fiction'),(3 , 'Science
fiction'),(4, 'Mystery'),(5 , 'romance');

INSERT INTO Books VALUES
(1 , 'The Great Gatsby' ,1,1),
(2, 'To kill a mockingbird' ,2,1),
(3 , '1984',3,3),
(4, 'Sherlock Holmes',4,4),
(5 , 'Pride and prejudice',5,5);
(6, 'The Hobbit' ,1,3);

INSERT INTO Users VALUES (1 , 'Alice'),(2 , 'Bob'),(3 , 'Charlie'),(4, 'Emily'),(5 , 'james');

INSERT INTO Loans Values
(1,1,1, '2023-01-15', '2023-02-15'),
(2,2,2, '2023-02-01' , '2023-03-01'),
(3,3,3, '2023-03-10', NULL),
(4,4,4, '2023-04-05' , NULL),
(5,5,5, '2023-05-12' , NULL),
(6,6,1, '2023-06-20' , NULL);

SELECT*FROM Authors;
SELECT *FROM Genres;
SELECT *FROM Books;
SELECT *FROM Users;
SELECT*FROM Loans;

--1.Retrieve the titles of all books along with the names of their authors and genres
SELECT B.Title ,A.Author_name AS Name , G.genre_name As genre from Books B
LEFT JOIN Authors A ON B.Author_ID =A.Author_ID

```

```
LEFT JOIN Genres G ON B.Genre_ID =G.Genres_ID;
```

--2. List the names of authors who have written books borrowed by user 'Alice'

```
SELECT A.Author_name
FROM Loans l
LEFT JOIN [Users] U ON L.[User_ID] = U.[User_ID]
LEFT JOIN Books B ON L.Book_ID =B.Book_ID
LEFT JOIN Authors A ON B.Author_ID = A.Author_ID
WHERE[User_Name]='Alice'
```

--3 Display the titles of books along with the names of their authors and the total number of times each book has been borrowed.

```
SELECT B.Title ,A.Author_name AS Name ,count(L.Book_ID) as [no of times taken]
from Books B
LEFT JOIN Authors A ON B.Author_ID =A.Author_ID
LEFT JOIN Genres G ON B.Genre_ID =G.Genres_ID
LEFT JOIN loans L ON B.Book_ID=L.Book_ID
group by B.Title,A.Author_name,L.Book_ID
```

--Retrieve the names of users along with the titles of books they borrowed, the loan dates, and the return dates, if any.

```
SELECT B.Title ,A.Author_name AS Name , U.[User_name] As Username , L.loan_date,
l.return_date from Loans L
LEFT JOIN [Users] U ON L.[User_ID] =U.[user_ID]
LEFT JOIN Books B ON L.book_ID =B.Book_ID
LEFT JOIN Authors A ON B.Author_ID = A.Author_ID ;
```

--List the titles of books along with the names of their authors, genres, and the loan dates for each loan made by user 'Alice'

```
SELECT B.Title ,A.Author_name ,G.Genre_name ,L.loan_date
FROM Loans l
LEFT JOIN [Users] U ON L.[User_ID] =U.[user_ID]
LEFT JOIN Books B ON L.Book_ID =B.Book_ID
LEFT JOIN Genres G ON B.genre_ID = G.genres_ID
LEFT JOIN Authors A ON B.Author_ID = A.Author_ID
WHERE[User_Name]='Alice'
```

--5. Find the names of users who borrowed books published in the 'Non-fiction' genre along with the titles of the books, the names of the authors, and the loan dates.

```
SELECT B.Title ,A.Author_name ,L.loan_date
FROM Loans l
LEFT JOIN [Users] U ON L.[User_ID] =U.[user_ID]
LEFT JOIN Books B ON L.Book_ID =B.Book_ID
LEFT JOIN Genres G ON B.genre_ID = G.genres_ID
LEFT JOIN Authors A ON B.Author_ID = A.Author_ID
WHERE[Genre_name]='Non-fictional'
```

--Find the titles of books with the highest number of loans.

```
select * from books b
where book_id= (select top 1 book_id from loans group by book_id having count(book_id)>1
order by book_id desc )
```

