CREATE DATABASE JMLDB;  
USE JMLDB;  
  
CREATE TABLE Author (  
 authorID INT AUTO\_INCREMENT PRIMARY KEY,  
 Name VARCHAR(255) NOT NULL,  
 Nationality VARCHAR(255)  
);  
  
CREATE TABLE Employees (  
 LibrarianID INT AUTO\_INCREMENT PRIMARY KEY,  
 FirstName VARCHAR(255) NOT NULL,  
 LastName VARCHAR(255) NOT NULL,  
 Email VARCHAR(255) UNIQUE NOT NULL,  
 Phone VARCHAR(15),  
 AccessCode VARCHAR(255) NOT NULL,  
 Position VARCHAR(255)  
);  
  
CREATE TABLE Categories (  
 categoryID INT AUTO\_INCREMENT PRIMARY KEY,  
 categoryName VARCHAR(255) NOT NULL,  
 LibrarianID INT,  
 FOREIGN KEY (LibrarianID) REFERENCES Employees(LibrarianID)  
);  
  
CREATE TABLE Books (  
 bookID INT AUTO\_INCREMENT PRIMARY KEY,  
 title VARCHAR(255) NOT NULL,  
 publisher VARCHAR(255),  
 authorID INT,  
 year INT,  
 categoryID INT,  
 FOREIGN KEY (authorID) REFERENCES Author(authorID),  
 FOREIGN KEY (categoryID) REFERENCES Categories(categoryID)  
);  
  
CREATE TABLE Members (  
 MemberID INT AUTO\_INCREMENT PRIMARY KEY,  
 FirstName VARCHAR(255) NOT NULL,  
 LastName VARCHAR(255) NOT NULL,  
 Email VARCHAR(255) UNIQUE NOT NULL,  
 PhoneNumber VARCHAR(15),  
 DateOfVinculation DATE,  
 ID VARCHAR(255) UNIQUE NOT NULL,  
 InDebt BOOLEAN  
);  
  
CREATE TABLE Loans (  
 MemberID INT,  
 BookID INT,  
 BorrowDate DATE,  
 DueDate DATE,  
 ReturnDate DATE,  
 LibrarianID INT,  
 PRIMARY KEY (MemberID, BookID),  
 FOREIGN KEY (MemberID) REFERENCES Members(MemberID),  
 FOREIGN KEY (BookID) REFERENCES Books(bookID),  
 FOREIGN KEY (LibrarianID) REFERENCES Employees(LibrarianID)  
);  
  
CREATE TABLE Reservations (  
 ReservationID INT AUTO\_INCREMENT PRIMARY KEY,  
 BookID INT,  
 MemberID INT,  
 ReservationDate DATE,  
 FOREIGN KEY (BookID) REFERENCES Books(bookID),  
 FOREIGN KEY (MemberID) REFERENCES Members(MemberID)  
);  
  
CREATE TABLE Fines (  
 MemberID INT,  
 DueDate DATE,  
 DaysWithoutReturn INT,  
 AmountInDebt DECIMAL(10, 2),  
 InDebt BOOLEAN,  
 BookID INT,  
 PRIMARY KEY (MemberID, DueDate),  
 FOREIGN KEY (MemberID) REFERENCES Members(MemberID),  
 FOREIGN KEY (BookID) REFERENCES Books(bookID)  
);  
  
  
INSERT INTO Employees (FirstName, LastName, Email, Phone, AccessCode, Position) VALUES  
('John', 'Doe', 'johndoe@juliomariolibrary.com', '123-456-7890', 'AC123', 'Librarian'),  
('Jane', 'Smith', 'janesmith@juliomariolibrary.com', '234-567-8901', 'AC124', 'Assistant Librarian'),  
('Emily', 'Johnson', 'emilyjohnson@juliomariolibrary.com', '345-678-9012', 'AC125', 'Librarian'),  
('Michael', 'Brown', 'michaelbrown@juliomariolibrary.com', '456-789-0123', 'AC126', 'Librarian'),  
('Sarah', 'Davis', 'sarahdavis@juliomariolibrary.com', '567-890-1234', 'AC127', 'Librarian'),  
('David', 'Miller', 'davidmiller@juliomariolibrary.com', '678-901-2345', 'AC128', 'Assistant Librarian'),  
('Laura', 'Wilson', 'laurawilson@juliomariolibrary.com', '789-012-3456', 'AC129', 'Librarian'),  
('James', 'Taylor', 'jamestaylor@juliomariolibrary.com', '890-123-4567', 'AC130', 'Librarian'),  
('Sophia', 'Anderson', 'sophiaanderson@juliomariolibrary.com', '901-234-5678', 'AC131', 'Librarian'),  
('Daniel', 'Thomas', 'danielthomas@juliomariolibrary.com', '012-345-6789', 'AC132', 'Librarian');  
  
INSERT INTO Categories (categoryName, LibrarianID) VALUES  
('Fiction', 1), -- categoryID 1  
('Science Fiction', 2), -- categoryID 2  
('Fantasy', 1), -- categoryID 3  
('Horror', 2), -- categoryID 4  
('Mystery', 3), -- categoryID 5  
('Biography', 3), -- categoryID 6  
('History', 4), -- categoryID 7  
('Philosophy', 4), -- categoryID 8  
('Non-Fiction', 5), -- categoryID 9  
('Romance', 5); -- categoryID 10  
  
INSERT INTO Author (Name, Nationality) VALUES  
('J.K. Rowling', 'British'), -- authorID 1  
('George Orwell', 'British'), -- authorID 2  
('J.R.R. Tolkien', 'British'), -- authorID 3  
('Isaac Asimov', 'American'), -- authorID 4  
('Arthur C. Clarke', 'British'), -- authorID 5  
('Stephen King', 'American'), -- authorID 6  
('Agatha Christie', 'British'), -- authorID 7  
('Mark Twain', 'American'), -- authorID 8  
('Ernest Hemingway', 'American'), -- authorID 9  
('F. Scott Fitzgerald', 'American'), -- authorID 10  
('Sarah J. Maas', 'American'); -- authorID 11  
  
INSERT INTO Books (title, publisher, authorID, year, categoryID) VALUES  
('Harry Potter and the Philosopher\'s Stone', 'Bloomsbury', 1, 1997, 3),  
('1984', 'Secker & Warburg', 2, 1949, 1),  
('The Hobbit', 'George Allen & Unwin', 3, 1937, 3),  
('Foundation', 'Gnome Press', 4, 1951, 2),  
('Rendezvous with Rama', 'Gollancz', 5, 1973, 2),  
('The Shining', 'Doubleday', 6, 1977, 4),  
('Murder on the Orient Express', 'Collins Crime Club', 7, 1934, 5),  
('Adventures of Huckleberry Finn', 'Chatto & Windus', 8, 1884, 1),  
('The Old Man and the Sea', 'Charles Scribner\'s Sons', 9, 1952, 6),  
('The Great Gatsby', 'Charles Scribner\'s Sons', 10, 1925, 1),  
('A Court of Thorns and Roses', 'Bloomsbury', 11, 2015, 3),  
('A Court of Mist and Fury', 'Bloomsbury', 11, 2016, 3),  
('A Court of Wings and Ruin', 'Bloomsbury', 11, 2017, 3),  
('A Court of Frost and Starlight', 'Bloomsbury', 11, 2018, 3),  
('A Court of Silver Flames', 'Bloomsbury', 11, 2021, 3),  
('The Shining', 'Doubleday', 6, 1977, 4),  
('Murder on the Orient Express', 'Collins Crime Club', 7, 1934, 5),  
('Adventures of Huckleberry Finn', 'Chatto & Windus', 8, 1884, 1),  
('The Old Man and the Sea', 'Charles Scribner\'s Sons', 9, 1952, 6),  
('The Great Gatsby', 'Charles Scribner\'s Sons', 10, 1925, 1),  
('A Court of Thorns and Roses', 'Bloomsbury', 11, 2015, 3),  
('A Court of Mist and Fury', 'Bloomsbury', 11, 2016, 3),  
('A Court of Wings and Ruin', 'Bloomsbury', 11, 2017, 3),  
('A Court of Frost and Starlight', 'Bloomsbury', 11, 2018, 3),  
('A Court of Silver Flames', 'Bloomsbury', 11, 2021, 3);  
  
INSERT INTO Members (FirstName, LastName, Email, PhoneNumber, DateOfVinculation, ID, InDebt) VALUES  
('Alice', 'Lopez', 'alicegreen@gmail.com', '123-456-7890', '2020-01-15', 'ID001', FALSE),  
('Bob', 'Macky', 'bobwhite@hotmail.com', '234-567-8901', '2020-02-20', 'ID002', TRUE),  
('Charlie', 'Brown', 'charlieblack@yahoo.com', '345-678-9012', '2020-03-25', 'ID003', FALSE),  
('Daisy', 'Mcdougal', 'daisybrown@gmail.com', '456-789-0123', '2020-04-30', 'ID004', TRUE),  
('Edward', 'Cullen', 'edwardblue@gmail.com', '567-890-1234', '2020-05-05', 'ID005', FALSE),  
('Fiona', 'Enchanted', 'fionayellow@icloud.com', '678-901-2345', '2020-06-10', 'ID006', FALSE),  
('George', 'Ramirez', 'georgered@hotmail.com', '789-012-3456', '2020-07-15', 'ID007', TRUE),  
('Hannah', 'Hoops', 'hannahpurple@yahoo.com', '890-123-4567', '2020-08-20', 'ID008', FALSE),  
('Ian', 'Luck', 'ianorange@test.com', '901-234-5678', '2020-09-25', 'ID009', TRUE),  
('Jane', 'Aire', 'janepink@urosario.com', '012-345-6789', '2020-10-30', 'ID010', FALSE);  
  
  
INSERT INTO Loans (MemberID, BookID, BorrowDate, DueDate, ReturnDate, LibrarianID) VALUES  
(1, 25, '2024-01-01', '2024-01-15', '2024-01-11', 1),  
(2, 16, '2024-01-02', '2024-01-16', NULL, 2),  
(3, 1, '2024-01-03', '2024-01-17', '2024-02-01', 3),  
(2, 18, '2024-01-04', '2024-01-18', NULL, 4),  
(5, 23, '2024-01-05', '2024-01-19', '2024-01-18', 5),  
(6, 19, '2024-03-06', '2024-03-20', '2024-03-19', 6),  
(7, 13, '2024-02-07', '2024-02-21', '2024-03-19', 7),  
(7, 9, '2024-05-08', '2024-05-22', '2024-05-11', 8),  
(10, 24, '2024-02-09', '2024-02-23', NULL, 9),  
(10, 11, '2024-06-10', '2024-06-24', '2024-06-23', 10);  
  
INSERT INTO Fines (MemberID, DueDate, DaysWithoutReturn, AmountInDebt, InDebt, BookID) VALUES  
(5, '2024-04-15', 1, 0, FALSE, 1),  
(6, '2024-05-13', 25, 25, TRUE, 3),  
(8, '2024-02-19',6,0, FALSE, 15),  
(5, '2024-03-24', 7, 7, TRUE, 13),  
(5, '2024-06-19', 2, 0, FALSE, 14),  
(6, '2024-05-11', 24, 0.00, FALSE, 23),  
(4, '2024-04-14', 3, 3, TRUE, 12),  
(3, '2024-02-27', 0, 0.00, FALSE, 22),  
(8, '2024-03-24', 35, 35, TRUE, 17),  
(10, '2024-06-24', 2, 0, FALSE, 5);  
  
INSERT INTO Reservations (BookID, MemberID, ReservationDate) VALUES  
(20, 3, '2024-06-28'),  
(24, 4, '2024-06-02'),  
(16, 7, '2024-06-11'),  
(8, 9, '2024-06-17'),  
(4, 10, '2024-06-24'),  
(2, 2, '2024-06-19'),  
(11, 6, '2024-06-17'),  
(19, 5, '2024-07-09'),  
(20, 2, '2024-07-04'),  
(14, 10, '2024-07-03');  
  
#what is the author with the majority of books?  
SELECT a.Name, *COUNT*(b.bookID) AS BookCount  
FROM Author a  
JOIN Books b ON a.authorID = b.authorID  
GROUP BY a.Name  
ORDER BY BookCount DESC  
LIMIT 1;  
#What is the category davis is on charge on  
SELECT c.categoryName  
FROM Categories c  
JOIN Employees e ON c.LibrarianID = e.LibrarianID  
WHERE e.LastName = 'Davis';  
  
#information of all members with loans  
SELECT DISTINCT m.\*  
FROM Members m  
JOIN Loans l ON m.MemberID = l.MemberID;  
  
#What is the amount of money in debt?  
SELECT *SUM*(AmountInDebt) AS TotalDebt  
FROM Fines  
WHERE InDebt = TRUE;  
  
#Who is the member who has more delays?  
SELECT m.FirstName, m.LastName, *SUM*(f.DaysWithoutReturn) AS TotalDelays  
FROM Members m  
JOIN Fines f ON m.MemberID = f.MemberID  
GROUP BY m.MemberID  
ORDER BY TotalDelays DESC  
LIMIT 1;  
  
#What are the loans in charge of Laura?  
SELECT l.\*  
FROM Loans l  
Right JOIN Employees e ON l.LibrarianID = e.LibrarianID  
WHERE e.FirstName = 'Laura' AND e.LastName = 'Wilson';  
  
SELECT l.\*, m.FirstName AS MemberFirstName, m.LastName AS MemberLastName, m.PhoneNumber as PhoneNumber  
FROM Loans l  
JOIN Employees e ON l.LibrarianID = e.LibrarianID  
JOIN Members m ON l.MemberID = m.MemberID  
WHERE e.FirstName = 'Laura' AND e.LastName = 'Wilson';  
  
#What is the category with the largest number of books?  
SELECT c.categoryName, *COUNT*(b.bookID) AS BookCount  
FROM Categories c  
JOIN Books b ON c.categoryID = b.categoryID  
GROUP BY c.categoryName  
ORDER BY BookCount DESC  
LIMIT 1;  
  
#what are the titles that repeat in the library?  
SELECT title, *COUNT*(bookID) AS Count  
FROM Books  
GROUP BY title  
HAVING Count > 1;  
  
#what is the average number of days for returned books?  
SELECT *AVG*(*DATEDIFF*(ReturnDate, BorrowDate)) AS AverageDays  
FROM Loans  
WHERE ReturnDate IS NOT NULL;  
  
  
#8. List all employees and their categories, including the ones that does not have a category on charge so we can balance the charges between the employees.  
SELECT e.FirstName, e.LastName, c.categoryName  
FROM Employees e  
RIGHT JOIN Categories c ON e.LibrarianID = c.LibrarianID;