SQL Project

Navin Singh

Library Management System

Create database Library;

Use library;

For Books_table:

- create table books_table(book_id int primary key,

title varchar(50), author varchar(50),

genre char(50),publication_date varchar(20),

isbn int(50), quantity_available int(10));

- desc books_table;
- insert into books_table (book_id,title,author,genre,publication_date,isbn,quantity_available)
  1) values (1,'Harry Potter','J.K. Rowling', 'Fantasy','1997-06-26',9788700631625,5),
  2) (2,'To Kill a Mockingbird', 'Harper Lee', 'Fiction','1960-07-11',9780061120084,3),
  3) (3,'The Great Gatsby','F.Scott Fizgerald','Classic','1925-04-10',9780743273565,2),
  4) (4, 'Pride and Prejudice', 'Jane Austen', 'Romance', '1813-01-28', '9780141439518', 6),

5) (5, 'The Catcher in the Rye', 'J.D. Salinger', 'Fiction', '1951-07-16', '9780316769488', 15),

6) (6, 'Harry Potter and the Philosopher\'s Stone', 'J.K. Rowling', 'Fantasy', '1997-06-26', '9780747532743', 20),

7) (7, 'To the Lighthouse', 'Virginia Woolf', 'Modernist', '1927-05-05', '9780156907392', 5),

8) (8, 'Moby-Dick', 'Herman Melville', 'Adventure', '1851-10-18', '9780142437247', 7),

9) (9, 'The Hobbit', 'J.R.R. Tolkien', 'Fantasy', '1937-09-21', '9780618260300', 9),

10) (10, 'The Lord of the Rings', 'J.R.R. Tolkien', 'Fantasy', '1954-07-29', '9780618640157', 11);

- select * from books_table;

For Authors Table:

- create table authors (author_id int primary key, author_name varchar(50));
- desc authors;
- insert into authors (author_id, author_name)

values (1,'J.K. Rowling'),

(2,'Harper Lee'),

(3,'F.Scott Fitzgerald')

(4, 'George Orwell'),

(5, 'J.D. Salinger'),

(6, 'J.K. Rowling'),

(7, 'Virginia Woolf'),

(8, 'Herman Melville'),

(9, 'J.R.R. Tolkien'),

(10, 'Leo Tolstoy');

- select * from authors;

FOR MEMBERS TABLE:

- create table members (member_id int primary key, member_name varchar(50),

member_email varchar(50),

member_phone int);

- desc members;

- insert into members (member_id, member_name, member_email, member_phone)

values (1, 'John Doe', 'john.deo@example.com', 123-456-7890),

(2,'Jane Smith', 'jane.smith@example.com', 987-654-3210);

(3, 'Alice Johnson', 'alice.johnson@example.com', '555-555-5555'),

(4, 'Bob Brown', 'bob.brown@example.com', '999-999-9999'),

(5, 'Emily Davis', 'emily.davis@example.com', '111-222-3333'),

(6, 'Michael Wilson', 'michael.wilson@example.com', '444-444-4444'),

(7, 'Emma Garcia', 'emma.garcia@example.com', '777-777-7777'),

(8, 'David Martinez', 'david.martinez@example.com', '888-888-8888'),

(9, 'Olivia Rodriguez', 'olivia.rodriguez@example.com', '666-666-6666'),

(10, 'James Lee', 'james.lee@example.com', '333-333-3333');

- select * from members;

FOR BORROWINGS TABLE:

- create table borrowings (borrowing_id int primary key,

book_id int, foreign key (book_id) references books_table(book_id),

member_id int, foreign key (member_id) references members (member_id),

borrowing_date varchar(50),

return_date varchar(50),

is_returned varchar(50));

- desc borrowings;

- insert into borrowings (borrowing_id, book_id, member_id, borrowing_date, return_date, is_returned)
- values (1,1,1,'2024-02-10','2024-02-20','true'),

(2,2,2,'2024-02-15','Null','false'),

(3, 3, 3, '2024-03-13', '2024-03-27', 0),

(4, 4, 4, '2024-03-14', '2024-03-28', 0),

(5, 5, 5, '2024-03-15', '2024-03-29', 1),

(6, 6, 6, '2024-03-16', '2024-03-30', 0),

(7, 7, 7, '2024-03-17', '2024-03-31', 1),

(8, 8, 8, '2024-03-18', '2024-04-01', 0),

(9, 9, 9, '2024-03-19', '2024-04-02', 0),

(10, 10, 10, '2024-03-20', '2024-04-03', 1);

- select * from borrowings;

FOR PUBLISHERS TABLE:

- create table publishers (publisher_id int primary key,

publisher_name varchar(50),

publisher_country varchar(50));

- desc publishers;

- insert into publishers (publisher_id,publisher_name, publisher_country)
- values (1,'Penguin Random House', 'United State'),

(2,'Harpet Collins','United Kindom'),

(3, 'Hachette Livre', 'France'),

(4, 'Macmillan Publishers', 'United Kingdom'),

(5, 'Simon & Schuster', 'United States'),

(6, 'Springer Nature', 'Germany'),

(7, 'Wiley', 'United States'),

(8, 'Oxford University Press', 'United Kingdom'),

(9, 'Pearson Education', 'United Kingdom'),

(10, 'Cambridge University Press', 'United Kingdom');

- select * from publishers;

FOR BOOK_COPIES TABLE:

- create table book_copies (copy_id int primary key, book_id int,

foreign key(book_id) references books_table(book_id),

copy_number int, conditions varchar(20), shelf_location varchar(20));

- desc book_copies;

- insert into book_copies (copy_id,book_id,copy_number,conditions,shelf_location)
- values (1,1,001,'Good', 'A1'),

(2,1,002,'Fair','B3'),

(3, 2, '201', 'Fair', 'B4'),

(4, 2, '202', 'Good', 'C2'),

(5, 3, '301', 'Excellent', 'D1'),

(6, 3, '302', 'Good', 'D1'),

(7, 4, '401', 'Fair', 'A2'),

(8, 4, '402', 'Fair', 'B1'),

(9, 5, '501', 'Excellent', 'C3'),

(10, 5, '502', 'Good', 'B3');

- select * from book_copies;

FOR AUTHORS_BOOKS_MAPPING TABLE:

- create table Authors_Books_Mapping (author_book_id int primary key,

author_id int, foreign key(author_id) references authors(author_id),

book_id int, foreign key(book_id) references books_table(book_id));

- desc authors_books_mapping;

- insert into authors_books_mapping (author_book_id, author_id, book_id)
- values (1,1,1),

(2,2,2),

(3, 3, 3),

(4, 4, 4),

(5, 5, 5),

(6, 6, 6),

(7, 7, 7),

(8, 8, 8),

(9, 9, 9),

(10, 10, 10);

- select * from authors_books_mapping;

FOR REVIEW TABLE:

- create table reviews (review_id int primary key,

book_id int, foreign key(book_id) references books_table(book_id),

member_id int, foreign key(member_id) references members(member_id), rating float,

review_text varchar(50), review_date varchar(20));

- desc reviews;

- insert into reviews (review_id, book_id, member_id, rating, review_text, review_date)
- values (1,1,1,4.5,'A classic masterpiece', '2024-02-12'),

(2,2,2,5.0,'Absolutely loved it!', '2024-02-18'),

(3, 3, 3, 5, 'One of the best dystopian novels ever written.', '2023-07-20'),

(4, 4, 4, 4, 'Beautifully written romance with memorable characters.', '2023-08-25'),

(5, 5, 5, 3, 'Interesting read, but the protagonist was hard to relate to.', '2023-09-30'),

(6, 6, 6, 5, 'Magical! Captivating from the very first page.', '2023-10-05'),

(7, 7, 7, 4, 'A challenging but rewarding read.', '2023-11-10'),

(8, 8, 8, 4, 'Epic adventure on the high seas!', '2023-12-15'),

(9, 9, 9, 5, 'A delightful journey into Middle-earth.', '2024-01-20'),

(10, 10, 10, 5, 'Masterpiece! A timeless classic.', '2024-02-25');

- select * from reviews;

FOR TRANSACTION TABLE:

- create table transactions (transaction_id varchar(50) primary key,

member_id int, foreign key(member_id) references members(member_id),

transaction_date varchar(50),

transaction_type varchar(20), amount_paid int);

- desc transactions;

- insert into transactions (transaction_id, member_id, transaction_date, transaction_type

,amount_paid)

- values (1,1,'2024-02-10','Borrow',0),

(2,2,'2024-02-15','Borrow',0),

(3, 3, '2023-03-15', 'Borrow', 30.75),

(4, 4, '2023-04-20', 'Borrow', 20.00),

(5, 5, '2023-05-25', 'Borrow', 28.49),

(6, 6, '2023-06-30', 'Borrow', 18.75),

(7, 7, '2023-07-05', 'Borrow', 35.99),

(8, 8, '2023-08-10', 'Borrow', 22.50),

(9, 9, '2023-09-15', 'Borrow', 40.25),

(10, 10, '2023-10-20', 'Borrow', 25.00);

- select * from transactions;

FOR CHECK ALL TABLES

1) select * from authors;
2) select * from authors_books_mapping;
3) select * from book_copies;
4) select * from books_table;
5) select * from borrowings;
6) select * from members;
7) select * from publishers;
8) select * from reviews;
9) select * from transactions;

1. List all books borrowed by a specific member:
   - select * from books_table, transactions where member_id=1;
   - select title from books_table, transactions where member_id=1;
2. Find the most popular genres:
   - select genre, rating from books_table, reviews where rating=5;
   - select genre, rating from books_table, reviews order by rating desc limit 1;
3. Identify books with the highest average rating:
   - select title, rating from books_table , reviews order by rating desc;
4. List all members who have borrowed more than 5 books:
   - select member_name, copy_number from members,book_copies where copy_number > 5;
5. List all members who have borrowed less than 5 books:
   - select member_name, copy_number from members,book_copies where copy_number < 5;
6. Retrieve the top-rated books with at least 5 reviews:
   - select title, rating from books_table, reviews order by rating desc limit 5;
7. Calculate the total revenue generated from book purchases:
   - select sum(amount_paid) as "Total Revenue " from transactions;
8. List all books with their respective authors and publishers:
   - select distinct title,author_name,publisher_name from books_table, authors, publishers;
9. Find books that are currently available for borrowing:
   - select title,quantity_available from books_table;
10. Identify members who have overdue books:
    - select distinct member_name,title,is_returned from members,books_table, borrowings where is_returned='false';

11. List the top 10 most borrowed books:

- select title, borrowing_id from books_table, borrowings where borrowing_id > 10;

12. Calculate the average number of days a book is borrowed for:

- Sorry (No data available)

13. Find the total number of books published in each year:

- select title,author,publication_date from books_table;

14. Identify members who have borrowed books more than once:

- select member_name, copy_number from members,book_copies where copy_number > 1;

15. List all books with their respective authors and average ratings:

- select title,author_name,rating from books_table,authors,reviews;

16. Calculate the total number of copies available for each book:

- select title,quantity_available from books_table;

17. Create a view of transaction table and provide privilege to another user. The user can view only member id and transaction date and privilege should be to select id who made transaction on any specific date

- create view transaction_view as select member_id,transaction_date from transactions;

- grant select on transaction_view to 'sriram'@'localhost';

Thanks

Navin Singh