ASSIGNMENT- 5

NAME:A.RAKCHA

REGNO:192311421

1. Design a database to manage books, members and loans.

Model tables for books, members and loans.

Write stored procedures for list available books.

Implement triggers to update book availability when a book is checked out.

Write SQL queries to analyze list books currently checked out.

**Database Model (Tables)**

We will create three primary tables:

* **books**: Stores information about the books available in the library.
* **members**: Stores details about the library members.
* **loans**: Stores records of book loans (checked out books by members).

**Table 1: books**

sql

Copy code

CREATE TABLE books (

book\_id INT PRIMARY KEY AUTO\_INCREMENT,

title VARCHAR(255) NOT NULL,

author VARCHAR(255) NOT NULL,

available\_copies INT NOT NULL, -- Number of available copies in the library

total\_copies INT NOT NULL -- Total number of copies

);

**Table 2: members**

sql

Copy code

CREATE TABLE members (

member\_id INT PRIMARY KEY AUTO\_INCREMENT,

member\_name VARCHAR(255) NOT NULL,

contact\_info VARCHAR(255),

membership\_date DATE NOT NULL

);

**Table 3: loans**

sql

Copy code

CREATE TABLE loans (

loan\_id INT PRIMARY KEY AUTO\_INCREMENT,

book\_id INT,

member\_id INT,

loan\_date DATE NOT NULL,

return\_date DATE,

status VARCHAR(50) NOT NULL DEFAULT 'Checked Out', -- 'Checked Out' or 'Returned'

CONSTRAINT fk\_book FOREIGN KEY (book\_id) REFERENCES books(book\_id),

CONSTRAINT fk\_member FOREIGN KEY (member\_id) REFERENCES members(member\_id)

);

**2. Stored Procedures**

**Stored Procedure to List Available Books**

This stored procedure will list books that are available (i.e., those with at least one available copy).

sql

Copy code

DELIMITER //

CREATE PROCEDURE ListAvailableBooks()

BEGIN

SELECT book\_id, title, author, available\_copies

FROM books

WHERE available\_copies > 0;

END //

DELIMITER ;

You can call this procedure to list books that are available for checkout.

sql

Copy code

CALL ListAvailableBooks();

**3. Triggers**

**Trigger to Update Book Availability When a Book is Checked Out**

This trigger will update the available\_copies field in the books table every time a book is checked out (i.e., when a loan is recorded in the loans table).

sql

Copy code

DELIMITER //

CREATE TRIGGER UpdateBookAvailabilityOnLoan

AFTER INSERT ON loans

FOR EACH ROW

BEGIN

UPDATE books

SET available\_copies = available\_copies - 1

WHERE book\_id = NEW.book\_id AND available\_copies > 0;

END //

DELIMITER ;

This trigger ensures that when a book is checked out, the number of available copies of that book is reduced by 1.

**Trigger to Update Book Availability When a Book is Returned**

This trigger will update the available\_copies field in the books table when a loan record is updated to mark the book as returned.

sql

Copy code

DELIMITER //

CREATE TRIGGER UpdateBookAvailabilityOnReturn

AFTER UPDATE ON loans

FOR EACH ROW

BEGIN

IF OLD.status = 'Checked Out' AND NEW.status = 'Returned' THEN

UPDATE books

SET available\_copies = available\_copies + 1

WHERE book\_id = NEW.book\_id;

END IF;

END //

DELIMITER ;

This trigger ensures that when a book is returned, the number of available copies is incremented.

**4. SQL Queries for Analyzing Currently Checked-Out Books**

**Query to List Books Currently Checked Out**

This query will return a list of books that are currently checked out, along with member details and loan dates.

sql

Copy code

SELECT b.title, b.author, m.member\_name, l.loan\_date, l.return\_date, l.status

FROM loans l

JOIN books b ON l.book\_id = b.book\_id

JOIN members m ON l.member\_id = m.member\_id

WHERE l.status = 'Checked Out';

This query will show all books that are currently checked out, the member who checked them out, and the loan dates.

**Example Output**

| **title** | **author** | **member\_name** | **loan\_date** | **return\_date** | **status** |
| --- | --- | --- | --- | --- | --- |
| The Great Gatsby | F. Scott | John Doe | 2024-11-01 | 2024-11-15 | Checked Out |
| Moby Dick | Herman Melville | Alice Johnson | 2024-11-03 | 2024-11-17 | Checked Out |
| 1984 | George Orwell | Bob Smith | 2024-11-05 | 2024-11-19 | Checked Out |

**Conclusion**

This library management system provides an efficient way to handle books, members, and loan transactions. The key components include:

1. **Tables**:
   * books: Tracks the details of books, including the number of available copies and total copies.
   * members: Stores member information, including contact details and membership date.
   * loans: Tracks book checkouts and returns, including the loan status and return date.
2. **Stored Procedure**:
   * **ListAvailableBooks**: Lists books that are available for checkout, making it easy for library staff or members to view available books.
3. **Triggers**:
   * **UpdateBookAvailabilityOnLoan**: Updates the book's availability when a book is checked out, ensuring accurate tracking of available copies.
   * **UpdateBookAvailabilityOnReturn**: Updates the book's availability when a book is returned, making the copy available again for checkout.
4. **SQL Queries**:
   * **Books currently checked out**: Provides insights into which books are checked out, the members who borrowed them, and the loan status. This helps to track which books are in circulation and manage overdue items.

**Benefits:**

* **Automation**: Triggers automate the process of updating the availability of books, reducing manual errors and ensuring real-time accuracy.
* **Efficiency**: The stored procedure for listing available books simplifies the process for both library staff and members to find books that can be checked out.
* **Data Insights**: The queries provide valuable insights into library usage, helping administrators to track books currently checked out and manage library resources effectively.

This database model can be easily extended to include additional features such as book reservations, overdue notifications, or fines management, making it scalable for future requirements.

Top of Form

Bottom of Form