**SQL Mini Project – Library Management Database**

- Biki Nath Newa

**1. Creating Database:**

CREATE DATABASE librarymanagementsystem

**2. Creating Tables:**

CREATE TABLE Books(  
Book\_ID INT PRIMARY KEY AUTO\_INCREMENT,   
Title VARCHAR(30) NOT NULL,  
Author VARCHAR(30) NOT NULL,  
ISBN BIGINT NOT NULL,   
Genre VARCHAR(30) NOT NULL,  
Location VARCHAR(5),  
Quantity INT,  
Available INT);

CREATE TABLE Users(  
User\_ID INT PRIMARY KEY AUTO\_INCREMENT,   
Username VARCHAR(30) NOT NULL,   
Email VARCHAR(30) NOT NULL UNIQUE,  
Date\_Created Date NOT NULL,   
Verified Boolean DEFAULT FALSE NOT NULL,   
Books\_Borrowed INT NULL);

CREATE TABLE Loans(  
Loan\_ID INT PRIMARY KEY AUTO\_INCREMENT,  
Book\_ID INT,  
User\_ID INT,  
Loan\_Date DATE NOT NULL,  
Return\_Date DATE NOT NULL,  
Overdue Boolean DEFAULT FALSE,  
Penalty\_Fee INT DEFAULT 0,  
FOREIGN KEY (Book\_ID) REFERENCES Books(Book\_ID),  
FOREIGN KEY (User\_ID) REFERENCES Users(User\_ID));

CREATE TABLE Ordered(  
Order\_ID INT PRIMARY KEY AUTO\_INCREMENT,  
Title VARCHAR(30) NOT NULL,  
Author VARCHAR(30) NOT NULL,  
ISBN INT NOT NULL,   
Genre VARCHAR(30) NOT NULL,  
Price INT NOT NULL);

CREATE TABLE Wish\_List(  
Wish\_ID INT PRIMARY KEY AUTO\_INCREMENT,  
User\_ID INT,  
Title VARCHAR(30) NOT NULL,  
FOREIGN KEY (User\_ID) REFERENCES Users(User\_ID));

**3. Insert Values:**

INSERT INTO Books(Title, Author, ISBN, Genre, Location, Quantity, Available)  
VALUES  
("Leviathan", "Scott Westerfeld", 9780143206088, "Sci-Fi", "S-3", 2, 1),  
("Lord Of The Rings", "J.R.R. Tolkien", 9780544003415, "Fantasy", "F-1", 1, 1),  
("Jack Reacher", "Tom Clancy", 9780536359315, "Thriller", "T-4", 1, 0),  
("Dune", "Frank Hebert", 9780441172719, "Sci-Fi", "S-2", 3, 2),  
("Long Walk To Freedom", "Nelson Mandela", 9780349106533, "Autobiography", "A-5", 1, 1),  
("Lord Of The Flies", "William Golding", 9780399501487, "Dystopian", "D-4", 1, 0);

A screenshot of a computer

AI-generated content may be incorrect.

INSERT INTO Users(Username, Email, Date\_Created, Verified, Books\_Borrowed)  
VALUES  
("Biki Nath Newa", "bikinathnewa@gmail.com", "2024-01-04", TRUE, 1),  
("Ram Prasad Pandey", "rampp@gmail.com", "2024-01-11", FALSE, 0),  
("Jenisha Joshi", "jennyj@gmail.com", "2024-02-01", TRUE, 2),  
("Rahul Maskey", "rmaskey@gmail.com", "2024-02-03", TRUE, 0);

A screenshot of a computer

AI-generated content may be incorrect.

INSERT INTO Loans(Book\_ID, User\_ID, Loan\_Date, Return\_Date)  
VALUES  
(4, 1, "2025-02-25", "2025-03-03"),  
(1, 3, "2025-02-23", "2025-03-01"),  
(6, 3, "2025-03-03", "2025-03-10");



INSERT INTO Ordered(Title, Author, ISBN, Genre, Price)  
VALUES  
("Atomic Habits", "James Clear", 9780735211292, "Self-Help", 8000),  
("Eragon", "Christopher Paolini", 0375826696, "Fantasy", 7500),  
("Lord Loss", "Darren Shan", 9780316114998, "Horror", 6000);

A screenshot of a computer

AI-generated content may be incorrect.

INSERT INTO Wish\_List(User\_ID, Title)  
VALUES  
(1, "The Time Machine"),  
(1, "Atomic Habits"),  
(3, "Twilight"),  
(4, "Tao Te Ching"),  
(4, "Dune");

A screenshot of a computer

AI-generated content may be incorrect.

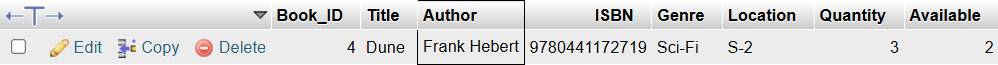
**4. SQL Operations:**

SELECT Title, Available FROM Books WHERE Title = "Lord Of The Flies"; /->Search for book by name

A close up of a computer screen

AI-generated content may be incorrect.

SELECT \* FROM books WHERE Author = "Frank Hebert"; /-> Search for book by author

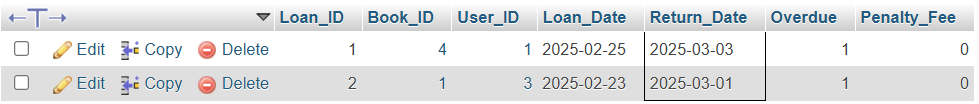


SELECT users.User\_ID, users.Username, loans.Loan\_ID, books.Title, books.Book\_ID   
FROM users  
INNER JOIN loans  
ON users.User\_ID = loans.User\_ID #-> Search for loaned books by Username  
INNER JOIN books  
ON loans.Book\_ID = books.Book\_ID  
WHERE Username = "Jenisha Joshi";

A screenshot of a computer

AI-generated content may be incorrect.

UPDATE loans set Overdue = TRUE where Return\_Date < NOW(); #-> Check for overdue books  
SELECT \* from loans WHERE Return\_Date < NOW();



UPDATE loans SET Penalty\_Fee = (DATEDIFF(CURRENT\_DATE,Return\_Date) \* 50) WHERE Return\_Date < CURRENT\_DATE; #-> Calculate penalty fee for overdue books (50 rupees per day)  
SELECT \* from loans;

A screenshot of a computer

AI-generated content may be incorrect.

SELECT wish\_list.Wish\_ID, users.Username, books.Title, books.Available  
FROM wish\_list  
INNER JOIN users   
ON users.User\_ID = wish\_list.User\_ID #-> Search for available books from wish list  
INNER JOIN books  
ON wish\_list.Title = books.Title  
WHERE Available > 0;

A close up of words

AI-generated content may be incorrect.