

DATABASE MANAGEMENT INTERN

AS A PART OF



Brainwave
MATRIX SOLUTIONS

PRESENTING TASK 1

Submitted By:

Shafak

Date: 17 Dec 2024

Design and Implement a Relational Database for a Library Management System.

Designing and implementing a relational database for a Library Management System involves several steps, including defining the requirements, designing the schema, and implementing the database. Below is a comprehensive guide to help you through this process.

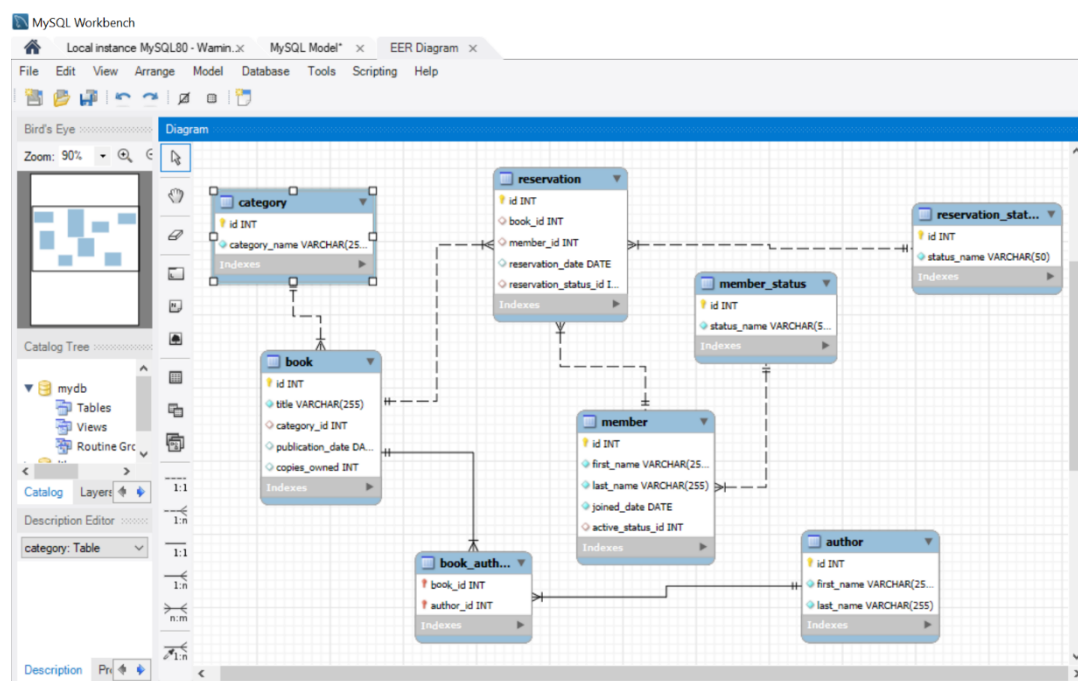
Step 1: Requirements Gathering

Before designing the database, we need to identify the key entities and their relationships. Here are some common requirements for a Library Management System:

1. **Books:** Each book has a title, author(s), ISBN, publication year, genre, and availability status.
2. **Members:** Each member has a name, membership ID, contact information, and membership status.
3. **Loans:** Members can borrow books, which involves tracking the loan date, return date, and due date.
4. **Authors:** Authors can write multiple books, and a book can have multiple authors.
5. **Categories:** Books can belong to multiple categories (e.g., Fiction, Non-Fiction, Science, etc.).

Relationships

- **Books to Loans:** One-to-Many (One book can have many loans)
- **Members to Loans:** One-to-Many (One member can have many loans)
- **Books to Authors:** Many-to-Many (One book can have many authors, and one author can write many books)



- **Books to Categories:** Many-to-Many (One book can belong to many categories, and one category can have many books)

Step 2: Database Schema Design

Based on the requirements, we can define the following tables:

- **book:** This table stores information about books in the library.
 - **id:** Primary key, unique identifier for each book.
 - **title:** Title of the book.
 - **category_id:** Foreign key referencing the **category** table, indicating the category of the book.
 - **publication_date:** Date of publication of the book.
 - **copies_owned:** Number of copies of the book owned by the library.
- **author:** This table stores information about authors.
 - **id:** Primary key, unique identifier for each author.
 - **first_name:** First name of the author.
 - **last_name:** Last name of the author.
- **category:** This table stores information about different categories of books.
 - **id:** Primary key, unique identifier for each category.
 - **category_name:** Name of the book category.
- **member:** This table stores information about members of the library.
 - **id:** Primary key, unique identifier for each member.
 - **first_name:** First name of the member.
 - **last_name:** Last name of the member.
 - **joined_date:** Date when the member joined the library.
 - **active_status_id:** Foreign key referencing the **member_status** table, indicating the membership status of the member.
- **book_author:** This table establishes the relationship between books and authors.
 - **book_id:** Foreign key referencing the **book** table.
 - **author_id:** Foreign key referencing the **author** table.
- **reservation:** This table stores information about reservations made by members for books.
 - **id:** Primary key, unique identifier for each reservation.

- **book_id:** Foreign key referencing the **book** table, identifying the reserved book.
- **member_id:** Foreign key referencing the **member** table, identifying the member making the reservation.
- **reservation_date:** Date when the reservation was made.
- **reservation_status_id:** Foreign key referencing the **reservation_status** table, indicating the current status

SQL Queries based on ER Diagram

Here are some SQL queries based on the ER diagram provided:

Step 1: Create the Database

Copy code

```
CREATE DATABASE library;
```

```
USE library;
```

Step 2: Create the Tables

```
-- Create 'category' table
```

```
CREATE TABLE category (
    id INT AUTO_INCREMENT PRIMARY KEY,
    category_name VARCHAR(255) NOT NULL
);
```

```
-- Create 'author' table
```

```
CREATE TABLE author (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL
);
```

```
-- Create 'member_status' table
```

```
CREATE TABLE member_status (
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```

        status_name VARCHAR(50) NOT NULL
    );

-- Create 'member' table
CREATE TABLE member (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(255) NOT NULL,
    last_name VARCHAR(255) NOT NULL,
    joined_date DATE NOT NULL,
    active_status_id INT,
    FOREIGN KEY (active_status_id) REFERENCES member_status(id)
);

-- Create 'book' table
CREATE TABLE book (
    id INT AUTO_INCREMENT PRIMARY KEY,
    title VARCHAR(255) NOT NULL,
    category_id INT,
    publication_date DATE,
    copies_owned INT,
    FOREIGN KEY (category_id) REFERENCES category(id)
);

-- Create 'book_author' table (many-to-many relationship between books and authors)
CREATE TABLE book_author (
    book_id INT,
    author_id INT,
    PRIMARY KEY (book_id, author_id),
    FOREIGN KEY (book_id) REFERENCES book(id),
    FOREIGN KEY (author_id) REFERENCES author(id)
);

```

```
);
```

```
-- Create 'reservation_status' table
```

```
CREATE TABLE reservation_status (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    status_name VARCHAR(50) NOT NULL  
);
```

```
-- Create 'reservation' table
```

```
CREATE TABLE reservation (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    book_id INT,  
    member_id INT,  
    reservation_date DATE,  
    reservation_status_id INT,  
    FOREIGN KEY (book_id) REFERENCES book(id),  
    FOREIGN KEY (member_id) REFERENCES member(id),  
    FOREIGN KEY (reservation_status_id) REFERENCES reservation_status(id)  
);
```

Step 3: Insert Sample Data

```
-- Insert data into 'category' table
```

```
INSERT INTO category (category_name) VALUES  
(  
    ('Science Fiction'),  
    ('Fantasy'),  
    ('Mystery'),  
    ('Biography'),  
    ('Technology');  
);
```

```
-- Insert data into 'author' table
```

```
INSERT INTO author (first_name, last_name) VALUES
('Isaac', 'Asimov'),
('J.K.', 'Rowling'),
('Agatha', 'Christie'),
('Stephen', 'Hawking'),
('Elon', 'Musk');
```

```
-- Insert data into 'member_status' table
```

```
INSERT INTO member_status (status_name) VALUES
('Active'),
('Inactive');
```

```
-- Insert data into 'member' table
```

```
INSERT INTO member (first_name, last_name, joined_date, active_status_id) VALUES
('John', 'Doe', '2022-01-15', 1),
('Jane', 'Smith', '2023-03-22', 1),
('Alice', 'Johnson', '2021-11-10', 2);
```

```
-- Insert data into 'book' table
```

```
INSERT INTO book (title, category_id, publication_date, copies_owned) VALUES
('The Foundation', 1, '1951-05-10', 3),
('Harry Potter and the Philosopher\'s Stone', 2, '1997-06-26', 5),
('Murder on the Orient Express', 3, '1934-01-01', 4),
('A Brief History of Time', 4, '1988-04-01', 2),
('Tesla: A Biography', 5, '2018-11-01', 6);
```

```
-- Insert data into 'book_author' table
```

```
INSERT INTO book_author (book_id, author_id) VALUES
(1, 1),
(2, 2),
```

(3, 3),

(4, 4),

(5, 5);

-- Insert data into 'reservation_status' table

INSERT INTO reservation_status (status_name) VALUES

('Reserved'),

('Checked Out'),

('Cancelled');

-- Insert data into 'reservation' table

INSERT INTO reservation (book_id, member_id, reservation_date, reservation_status_id)
VALUES

(1, 1, '2023-12-01', 1),

(2, 2, '2023-12-15', 2),

(3, 3, '2023-12-20', 3);