```
-- Student Contribution: Mustafa Alsaegh & Hemraj Yadav

-- Create the LibraryManagementSystem Database
CREATE DATABASE LibraryManagementSystem;

-- Switch to the new database
USE LibraryManagementSystem;

-- Create Tables

-- auth_user Table: user info table
CREATE TABLE auth_user (
    id INT NOT NULL AUTO_INCREMENT,
    password VARCHAR(128) NOT NULL,
    is_superuser TINYINT(1) NOT NULL,
    username VARCHAR(150) NOT NULL,
    first_name VARCHAR(30) NOT NULL,
    last_name VARCHAR(30) NOT NULL,
    email VARCHAR(254) NOT NULL,
    content_type_id INT NULL,
    PRIMARY KEY (id),
    UNIQUE (username),
    FOREIGN KEY (content_type_id) REFERENCES django_content_type (id)
);

-- library_book Table: This table stores information about the books available in
the library.
CREATE TABLE library_book (
    id INT NOT NULL AUTO_INCREMENT,
    title VARCHAR(100) NOT NULL,
    author VARCHAR(100) NOT NULL,
    isbn VARCHAR(13) NOT NULL,
    PRIMARY KEY (id),
    CHECK (isbn >= 0)
);


-- library_studentextra Table: This table stores additional information about the
students(user info).
CREATE TABLE library_studentextra (
    id INT NOT NULL AUTO_INCREMENT,
    user_id INT NOT NULL,
    enrollment VARCHAR(30) NOT NULL UNIQUE,
    PRIMARY KEY (id),
    FOREIGN KEY (user_id) REFERENCES auth_user(id)
);

-- Table to store book issued
CREATE TABLE library_issuedbook (
  id INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
  issuedate DATE NOT NULL,
  expirydate DATE NOT NULL,
  enrollment INT NOT NULL,
  isbn INT NOT NULL,
  FOREIGN KEY (enrollment) REFERENCES library_studentextra (id) ON DELETE CASCADE
ON UPDATE CASCADE,
  FOREIGN KEY (isbn) REFERENCES library_book (id) ON DELETE CASCADE ON UPDATE
CASCADE
);
```

```sql
-- INSERTING INFO to tables
-- Sample data for auth_user table
INSERT INTO auth_user (password, is_superuser, username, first_name, last_name,
email)
VALUES ('password1', 1, 'admin', 'John', 'Doe', 'admin@example.com');

-- Sample data for library_studentextra table
INSERT INTO library_studentextra (user_id)
VALUES (1);

-- Sample data for library_book table
INSERT INTO library_book (title, author, isbn)
VALUES ('The Catcher in the Rye', 'J.D. Salinger', '0316769177'),
    ('To Kill a Mockingbird', 'Harper Lee', '9780061120084'),
    ('1984', 'George Orwell', '9780451524935'),
    ('The Great Gatsby', 'F. Scott Fitzgerald', '9780743273565'),
    ('Pride and Prejudice', 'Jane Austen', '9780486284736');
;

-- Sample data for library_issuedbook table
INSERT INTO library_issuedbook (issuedate, expirydate, enrollment, isbn)
VALUES ('2023-03-01', '2023-03-31', '20230001', '0316769177');


-- BELOW TABLES GENERATED & HANDLED BY DJANGO (RUN first)
-- django model will automatically handle generating the SQL code needed for logins
and other related authentications.
-- We wrote it here to show what the model will contain

-- django_content_type Table: This table stores information about the content types
used by the Django models.
CREATE TABLE django_content_type (
    id INT NOT NULL AUTO_INCREMENT,
    app_label VARCHAR(100) NOT NULL,
    model VARCHAR(100) NOT NULL,
    PRIMARY KEY (id),
    UNIQUE (app_label, model)
);

-- django_admin_log Table
CREATE TABLE django_admin_log (
    id INT NOT NULL AUTO_INCREMENT,
    action_time DATETIME NOT NULL,
    user_id INT NOT NULL,
    content_type_id INT NULL,
    object_id LONGTEXT NULL,
    object_repr VARCHAR(200) NOT NULL,
    action_flag SMALLINT UNSIGNED NOT NULL,
    change_message LONGTEXT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (user_id) REFERENCES auth_user (id),
    FOREIGN KEY (content_type_id) REFERENCES django_content_type (id)
);
-- auth_permission Table
-- This table links users with the permissions they have.
CREATE TABLE auth_permission (
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    content_type_id INT NOT NULL,
    PRIMARY KEY (id),
    UNIQUE (name, content_type_id),
    FOREIGN KEY (content_type_id) REFERENCES django_content_type (id));

-- auth_user_user_permissions Table:
CREATE TABLE auth_user_user_permissions (
    id INT NOT NULL AUTO_INCREMENT,
    user_id INT NOT NULL,
    permission_id INT NOT NULL,
    PRIMARY KEY (id),
    UNIQUE (user_id, permission_id),
    FOREIGN KEY (user_id) REFERENCES auth_user (id),
    FOREIGN KEY (permission_id) REFERENCES auth_permission (id)
);




-- Create Constraints

-- auth_group_permission Table
ALTER TABLE auth_group_permission
ADD CONSTRAINT fk_auth_group_permission_group
FOREIGN KEY (group_id)
REFERENCES auth_group(id);
```