D.1 THE DATABASE

Queries:

```
CREATE DATABASE management_system

CREATE TABLE `admin` (
  `Admin_ID` int(11) NOT NULL,
  `Admin_name` varchar(50) NOT NULL,
  `Admin_Email` varchar(100) NOT NULL,
  `Password` varchar(100) NOT NULL,
  `role` varchar(50) DEFAULT 'admin'
);

CREATE TABLE `book` (
  `Book_ID` int(11) NOT NULL,
  `Bookname` varchar(255) NOT NULL,
  `Authors` varchar(255) NOT NULL,
  `Publishers` varchar(255) NOT NULL,
  `Publication_date` date DEFAULT NULL,
  `Copies_available` int(11) DEFAULT 0,
  `ISBN` varchar(20) NOT NULL
);

CREATE TABLE `book_category` (
  `Book_ID` int(11) NOT NULL,
  `Categories_ID` int(11) NOT NULL
);
```

```
CREATE TABLE `book_like` (
  `User_ID` int(11) NOT NULL,
  `Book_ID` int(11) NOT NULL,
  `Liked_At` timestamp NOT NULL DEFAULT
current_timestamp()
);

CREATE TABLE `borrow_record` (
  `Record_ID` int(11) NOT NULL,
  `User_ID` int(11) DEFAULT NULL,
  `Borrower_Name` varchar(255) DEFAULT NULL,
  `Book_ID` int(11) DEFAULT NULL,
  `Borrow_date` date DEFAULT NULL,
  `Return_date` date DEFAULT NULL,
  `Actual_return_date` date DEFAULT NULL
);

CREATE TABLE `category` (
  `Categories_ID` int(11) NOT NULL,
  `Categories_name` varchar(100) NOT NULL
);

CREATE TABLE `comment` (
  `Comment_ID` int(11) NOT NULL,
  `User_ID` int(11) DEFAULT NULL,
  `Book_ID` int(11) DEFAULT NULL,
  `Comment_text` text DEFAULT NULL,
```

```
  `created_at` datetime NOT NULL DEFAULT
current_timestamp()
);

CREATE TABLE `reservation` (
  `Reservation_ID` int(11) NOT NULL,
  `User_ID` int(11) DEFAULT NULL,
  `Book_ID` int(11) DEFAULT NULL,
  `Expiry_date` date DEFAULT NULL
);

CREATE TABLE `user` (
  `User_ID` int(11) NOT NULL,
  `Username` varchar(50) NOT NULL,
  `Email` varchar(100) NOT NULL,
  `Password` varchar(100) NOT NULL,
  `Role` enum('user','admin') DEFAULT 'user',
  `Created_time` datetime DEFAULT current_timestamp()
);

ALTER TABLE `admin`
  ADD PRIMARY KEY (`Admin_ID`),
  ADD UNIQUE KEY `Admin_Email` (`Admin_Email`);

ALTER TABLE `book`
  ADD PRIMARY KEY (`Book_ID`),
  ADD UNIQUE KEY `ISBN` (`ISBN`);
```

```
ALTER TABLE `book_category`
  ADD PRIMARY KEY (`Book_ID`,`Categories_ID`),
  ADD KEY `Categories_ID` (`Categories_ID`);


ALTER TABLE `book_like`
  ADD PRIMARY KEY (`User_ID`,`Book_ID`),
  ADD KEY `Book_ID` (`Book_ID`);

ALTER TABLE `borrow_record`
  ADD PRIMARY KEY (`Record_ID`),
  ADD KEY `User_ID` (`User_ID`),
  ADD KEY `Book_ID` (`Book_ID`);

ALTER TABLE `category`
  ADD PRIMARY KEY (`Categories_ID`);

ALTER TABLE `comment`
  ADD PRIMARY KEY (`Comment_ID`),
  ADD KEY `User_ID` (`User_ID`),
  ADD KEY `Book_ID` (`Book_ID`);

ALTER TABLE `reservation`
  ADD PRIMARY KEY (`Reservation_ID`),
  ADD KEY `User_ID` (`User_ID`),
  ADD KEY `Book_ID` (`Book_ID`);
```

```
ALTER TABLE `user`
  ADD PRIMARY KEY (`User_ID`),
  ADD UNIQUE KEY `Email` (`Email`);
```

D.2 THE DATA

```
INSERT INTO `admin` (`Admin_ID`, `Admin_name`,
`Admin_Email`, `Password`, `role`) VALUES
(1, 'admin', '1756211215@163.com',
'$2y$10$yL5WwmPKa.NFS9gfaip02uzF7cCQc9omkstZsu
OSz8xsn0lQPsE7a', 'admin'),
(2, 'smile', '1756211215@gmail.com',
'$2y$10$OQImojKcWvgs79ES7c6kvuct6lpt/a0i85ZDONZi
MRTGjWzTkdwGe', 'admin'),
(3, 'apple', '15197388050@163.com',
'$2y$10$sXiVJ3hMmkFVedgVEhRknuGe6.plFCtR.R1gJA
sXKcRX6NuABRNMK', 'admin');
```

```
INSERT INTO `book` (`Book_ID`, `Bookname`, `Authors`,
`Publishers`, `Publication_date`, `Copies_available`,
`ISBN`) VALUES
```

(1, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Scribner', '1925-04-10', 6, '9780743273565'),
(2, 'To Kill a Mockingbird', 'Harper Lee', 'J.B. Lippincott & Co.', '1960-07-11', 3, '9780060935467'),
(3, '1984', 'George Orwell', 'Secker & Warburg', '1949-06-08', 3, '9780451524935'),
(4, 'The Catcher in the Rye', 'J.D. Salinger', 'Little, Brown and Company', '1951-07-16', 4, '9780316769488'),
(5, 'Citizen: An American Lyric', 'Claudia Rankine', 'Graywolf Press', '2014-10-07', 4, '9781555976903'),
(6, 'A Theory of Justice', 'John Rawls', 'Harvard University Press', '1971-01-01', 3, '9780674000780');

INSERT INTO `book_category` (`Book_ID`, `Categories_ID`) VALUES
(1, 15),
(2, 15),
(3, 15),
(4, 15),
(5, 16),
(6, 20);

INSERT INTO `book_like` (`User_ID`, `Book_ID`, `Liked_At`) VALUES
(123, 3, '2025-05-01 13:24:31'),
(123, 4, '2025-05-04 14:33:39'),

```
(123, 5, '2025-05-04 14:33:39'),
(123, 6, '2025-05-01 13:24:33'),
(1234567890, 1, '2025-05-04 14:33:18'),
(1234567890, 2, '2025-05-04 14:33:19'),
(1234567890, 3, '2025-05-04 13:01:15'),
(1234567890, 4, '2025-05-04 14:33:16'),
(1234567890, 5, '2025-05-04 14:33:15'),
(1234567890, 6, '2025-05-04 14:33:14');


INSERT INTO `borrow_record` (`Record_ID`, `User_ID`,
`Borrower_Name`, `Book_ID`, `Borrow_date`,
`Return_date`, `Actual_return_date`) VALUES
(29, 123, 'apple', 1, '2025-05-02', '2025-05-16',
'2025-05-02'),
(30, 123, 'apple', 4, '2025-05-02', '2025-05-16',
'2025-05-02'),
(32, 123, 'apple', 3, '2025-05-02', '2025-05-16', NULL),
(33, 1234567891, 'admin', 6, '2025-05-02', '2025-05-16',
NULL),
(35, 1234567890, 'lisongjie', 3, '2025-05-04', '2025-05-18',
NULL);

INSERT INTO `category` (`Categories_ID`,
`Categories_name`) VALUES
(15, 'Fiction'),
(16, 'Poetry'),
```

(17, 'Prose'),
(18, 'Drama'),
(19, 'History'),
(20, 'Philosophy'),
(21, 'Science');

INSERT INTO `comment` (`Comment_ID`, `User_ID`, `Book_ID`, `Comment_text`, `created_at`) VALUES
(10, 123, 1, 'This is a great book on programming.', '2025-04-29 10:00:00'),
(11, 123, 3, 'This is a bad book on programming.', '2025-04-30 10:00:00'),
(12, 123456, 2, 'this book has good plot.', '2025-04-30 23:10:45'),
(13, 123, 3, 'good', '2025-05-01 21:22:44'),
(15, 123, 6, 'good', '2025-05-04 21:34:12'),
(16, 123, 6, 'this book is so good', '2025-05-04 21:34:43');

INSERT INTO `reservation` (`Reservation_ID`, `User_ID`, `Book_ID`, `Expiry_date`) VALUES
(4, 123, 3, '2025-05-07'),
(5, 123, 5, '2025-05-07'),
(7, 123, 1, '2025-05-07'),
(8, 123, 2, '2025-05-07'),
(9, 123, 6, '2025-05-07'),
(12, 1234567890, 3, '2025-05-11');

```
INSERT INTO `user` (`User_ID`, `Username`, `Email`,
`Password`, `Role`, `Created_time`) VALUES
(123, '123', 'j13787810907@163.com',
'$2y$10$g6vAcOqmT9.axZ91DPh4BeGy0a110UZrpOdBG
sf7Did1CCLnDu4Qi', 'user', '2025-04-28 02:51:30'),
(123456, 'smile', '123421@163.com',
'$2y$10$IKktHnjytgFn4IBvq8eS4OtedH1xoC5W1lbxwJ1m
7qOuDo2uXoYnm', 'user', '2025-04-30 08:57:32'),
(123456789, 'apple', '13787810907j@gmail.com',
'$2y$10$05twlZbb9y7chpAtWvr3HuTJdtB0SkxMBglSb0tg
hk1gyqolB89KC', 'user', '2025-04-30 08:56:52'),
(1234567890, 'lisongjie', '1756211215@gmail.com',
'$2y$10$VH7BNk8BnZowlT3sdBK9w.gTwaGFw2TFrh2OK
ddPV51C3TZBFaZfO', 'user', '2025-05-01 18:05:30'),
(1234567891, 'admin', '1234567891@163.com',
'$2y$10$FluX0S8kyKydHHG3mmVDFO7WdCZ8Uo6VoLp
pV7Jjb2lol.14wKWwG', 'user', '2025-05-02 21:07:18');
```

D.3. QUERIES

```
SELECT * FROM book;
```

This SQL query selects all columns and all rows from the book table. It is used to display a complete list of all books stored in the library database, including details such as the book title, author, publisher, publication date, number of available copies, and the ISBN number.

```
SELECT * FROM admin;
```

This SQL query retrieves all data from the `admin` table. It is used to view all information about the administrative users of the library management system, which may include usernames, passwords, contact information, or other credentials related to system access and control.

```
SELECT * FROM borrow_record;
```

This SQL query displays every record from the `borrow_record` table. It is useful for tracking the borrowing activity of users, such as which books were borrowed, who borrowed them, the borrowing dates, due dates, and potentially the return status of each borrowed book.

```
SELECT Categories_name, COUNT(*) AS TotalBooks
FROM category
GROUP BY Categories_name
HAVING COUNT(*) > 0;
```

This SQL query retrieves the number of books in each category by grouping the data based on the `Categories_name` column in the `book_category` table. It then filters the results to show only those categories that have more than one book, using the `HAVING` clause.

```
SELECT *
FROM `book`
INNER JOIN `book_like` ON `book`.`Book_ID` =
`book_like`.`Book_ID`;
```

This SQL query uses an INNER JOIN to combine `book` and `book_like` tables. It fetches all columns from both tables, returning only rows where the `Book_ID` matches in both tables. This way, we can get related book and user - like data in one result.

```
SELECT `User_ID`, `Book_ID`,
(SELECT COUNT(*) FROM `reservation` r2 WHERE
r2.`Book_ID` = r1.`Book_ID`) as `reservation_count`
FROM `reservation` r1;
```

This SQL query retrieves the `User_ID`, `Book_ID` from the `reservation` table, along with a count of related reservation

records for each `Book_ID` in a subquery. It's used to show user - book reservation associations and the number of reservations per book.