Library Management System

Database Management Systems

Submitted By:

Tanbir Hasan Taz (2023100000415)
Yousuf Howlader (2020000000150)
Md Fahmidul Islam Inkiat (2023100000666)

Submitted To:

S.I.M. Adnan

Lecturer

Department of Computer Science and Engineering

Southeast University

1. Introduction

Libraries are essential institutions for organizing and providing access to knowledge resources. With the increasing number of books, users, and transactions, manual library management systems have become inefficient and prone to errors. As a result, database-driven systems are required to ensure accuracy, consistency, and efficient data handling.

The **Library Management System (LMS)** is a relational database system designed to manage information related to library users, books, authors, categories, loans, and reservations. The system supports core library operations such as book borrowing, reservation management, and tracking issue, due, and return dates. Proper use of primary and foreign keys ensures data integrity and minimizes redundancy.

This project focuses on the design and implementation of a normalized database using **Entity-Relationship modeling** and **relational schema design**. The database is normalized up to **Third Normal Form (3NF)** to improve efficiency and maintain data consistency. SQL is used to implement the database schema and perform meaningful queries.

The scope of this project is limited to backend database design and implementation. However, the system can be extended in the future by integrating a front-end application and additional features.

Objectives and Scope

- To design a well-structured relational database for a library management system.
- To efficiently manage library resources including books, users, authors, categories, loans, and reservations.
- To apply Entity-Relationship (ER) modeling, normalization up to Third Normal Form (3NF), and SQL-based implementation.
- To support efficient querying for library operations such as borrowing history, book availability, overdue tracking, and reservation records.

2. Requirements Description

The Vehicle Rental Management System involves the following entities and users:

Entities

- **User:** Stores personal and contact information of library members.
- **Book:** Stores details of books available in the library.
- **Author:** Stores information about book authors.
- **Category:** Stores book classification details.
- **Loan:** Manages book borrowing transactions between users and the library.
- **Reservation:** Records book reservation information.
- **Book_Author:** Manages the many-to-many relationship between books and authors.

Users

- **Admin (Librarian):** Manages books, authors, categories, and system data.
- **Library Member:** Borrows and reserves books.

Main Functionalities

- Add, update, and remove book records.
- Register and manage library users.
- Manage book borrowing and return transactions.
- Track loan duration, due dates, and return status.
- Manage book reservations and maintain availability records.

3. ER Diagram

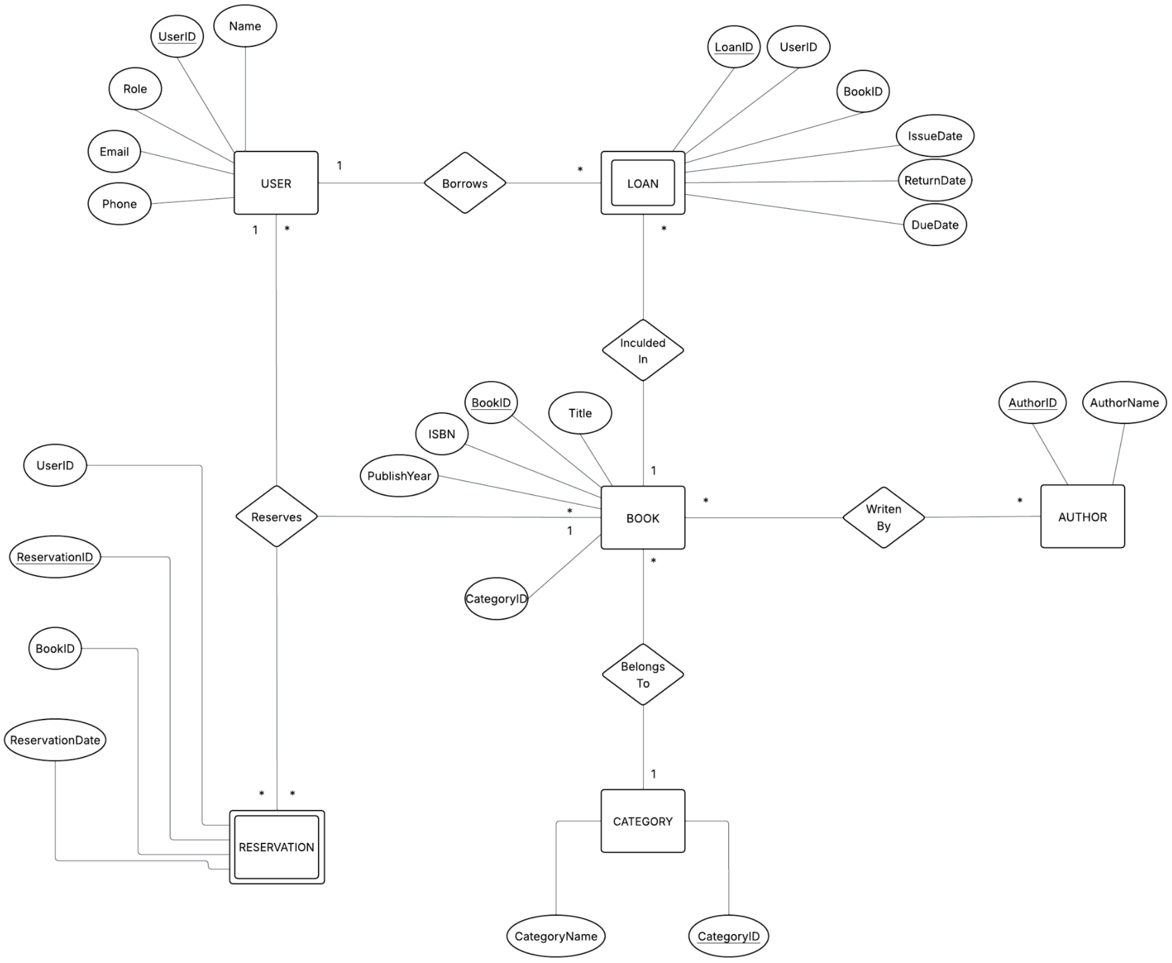
The ER diagram consists of the following entities and relationships:

Entities and Attributes

- **User** (UserID, Name, Email, Phone, Role)
- **Book** (BookID, Title, ISBN, PublishYear, CategoryID)
- **Author** (AuthorID, AuthorName)
- **Category** (CategoryID, CategoryName)
- **Loan** (LoanID, IssueDate, DueDate, ReturnDate, UserID, BookID)
- **Reservation** (ReservationID, ReservationDate, UserID, BookID)
- **Book_Author** (BookID, AuthorID)

Relationships

- A **User** can have many **Loans** (1:M).
- A **Book** can be loaned many times (1:M).
- A **User** can make many **Reservations** (1:M).
- A **Book** can be reserved many times (1:M).
- A **Book** can be written by many **Authors**, and an **Author** can write many **Books** (M:N), resolved using the **Book_Author** junction table.
- Each **Book** belongs to one **Category**, while a **Category** can include many **Books** (1:M).



4. Relational Schema

- User (UserID, Name, Email, Phone, Role)
- Category (CategoryID, CategoryName)
- Author (AuthorID, AuthorName)
- Book (BookID, Title, ISBN, PublishYear, CategoryID → Category.CategoryID)
- Book_Author (BookID → Book.BookID, AuthorID → Author.AuthorID)
- Loan (LoanID, IssueDate, DueDate, ReturnDate, UserID → User.UserID, BookID → Book.BookID)
- Reservation (ReservationID, ReservationDate, UserID → User.UserID, BookID → Book.BookID)

5. Normalization and Final Relational Schema

Rule of 1NF:

✔ Each attribute contains atomic (single) values

✔ No repeating groups or multivalued attributes

To satisfy 1NF, the data is divided into separate tables where each field stores a single value.

Tables created in 1NF:

User (UserID, Name, Email, Phone, Role)

Book (BookID, Title, ISBN, PublishYear, Category, Author)

Loan (LoanID, IssueDate, DueDate, ReturnDate, UserID, BookID)

Reservation (ReservationID, ReservationDate, UserID, BookID)

At this stage, all attributes contain atomic values, but redundancy still exists.

Rule of 2NF:

✔ Must be in 1NF
✔ No partial dependency (applies to composite primary keys)

The **Book–Author** relationship is many-to-many, so a separate junction table is created.

Tables created in 2NF:

User (UserID, Name, Email, Phone, Role)

Category (CategoryID, CategoryName)

Author (AuthorID, AuthorName)

Book (BookID, Title, ISBN, PublishYear, CategoryID)

Book_Author (BookID, AuthorID)

Loan (LoanID, IssueDate, DueDate, ReturnDate, UserID, BookID)

Reservation (ReservationID, ReservationDate, UserID, BookID)

All non-key attributes are fully dependent on their respective primary keys.

Rule of 3NF:

✔ Must be in 2NF

✔ No transitive dependency

(Non-key attributes must not depend on other non-key attributes)

In the **Book** table:

- BookID → CategoryName (repeating value)

So, **Category** is separated into its own table.

All remaining attributes depend only on their primary keys, and no transitive dependency exists.

User (UserID, Name, Email, Phone, Role)

Category (CategoryID, CategoryName)

Author (AuthorID, AuthorName)

Book (BookID, Title, ISBN, PublishYear, CategoryID → Category.CategoryID)

Book_Author (BookID → Book.BookID, AuthorID → Author.AuthorID)

Loan (LoanID, IssueDate, DueDate, ReturnDate, UserID → User.UserID, BookID → Book.BookID)

Reservation (ReservationID, ReservationDate, UserID → User.UserID, BookID → Book.BookID)

6. Database Schema

1. user

Stores personal and contact information of library members.

- **UserID** INT(11) (PK)
- **Name** VARCHAR(100)
- **Email** VARCHAR(100)
- **Phone** VARCHAR(15)
- **Role** VARCHAR(50)

2. category

Stores book classification information.

- CategoryID INT(11) **(PK)**
- CategoryName VARCHAR(100)

3. author

Stores information about book authors.

- AuthorID INT(11) **(PK)**
- AuthorName VARCHAR(100)

4. book

Stores detailed information about books available in the library.

- BookID INT(11) **(PK)**
- Title VARCHAR(200)
- ISBN VARCHAR(20) **(UNIQUE)**
- PublishYear YEAR
- CategoryID INT(11) **(FK)** → References category(CategoryID)

5. book_author

Junction table used to manage the many-to-many relationship between books and authors.

- BookID INT(11) **(FK)** → References book(BookID)
- AuthorID INT(11) **(FK)** → References author(AuthorID)
  Composite Primary Key: (BookID, AuthorID)

6. loan

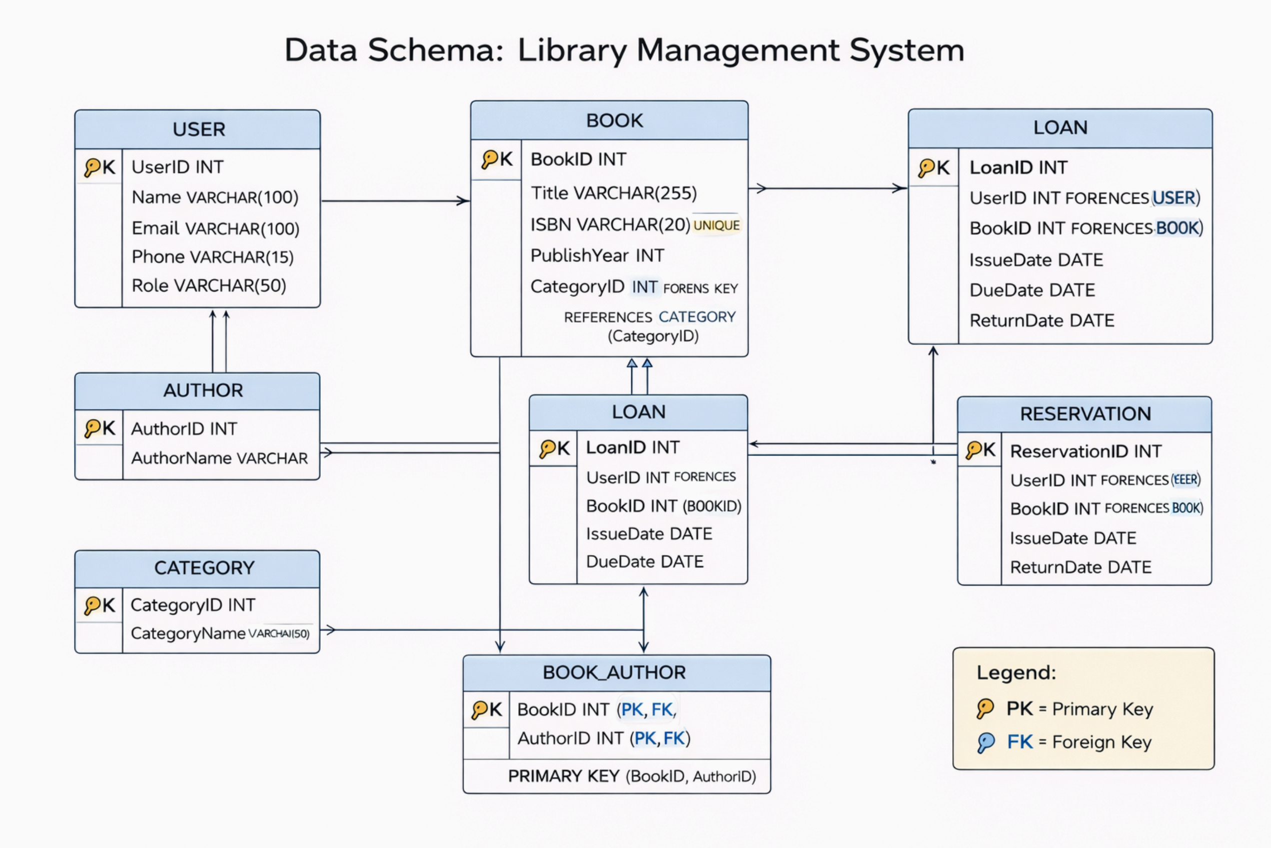
Stores information about book borrowing transactions.

- LoanID INT(11) **(PK)**
- IssueDate DATE
- DueDate DATE
- ReturnDate DATE
- UserID INT(11) **(FK)** → References user(UserID)
- BookID INT(11) **(FK)** → References book(BookID)

7. reservation

Stores book reservation records.

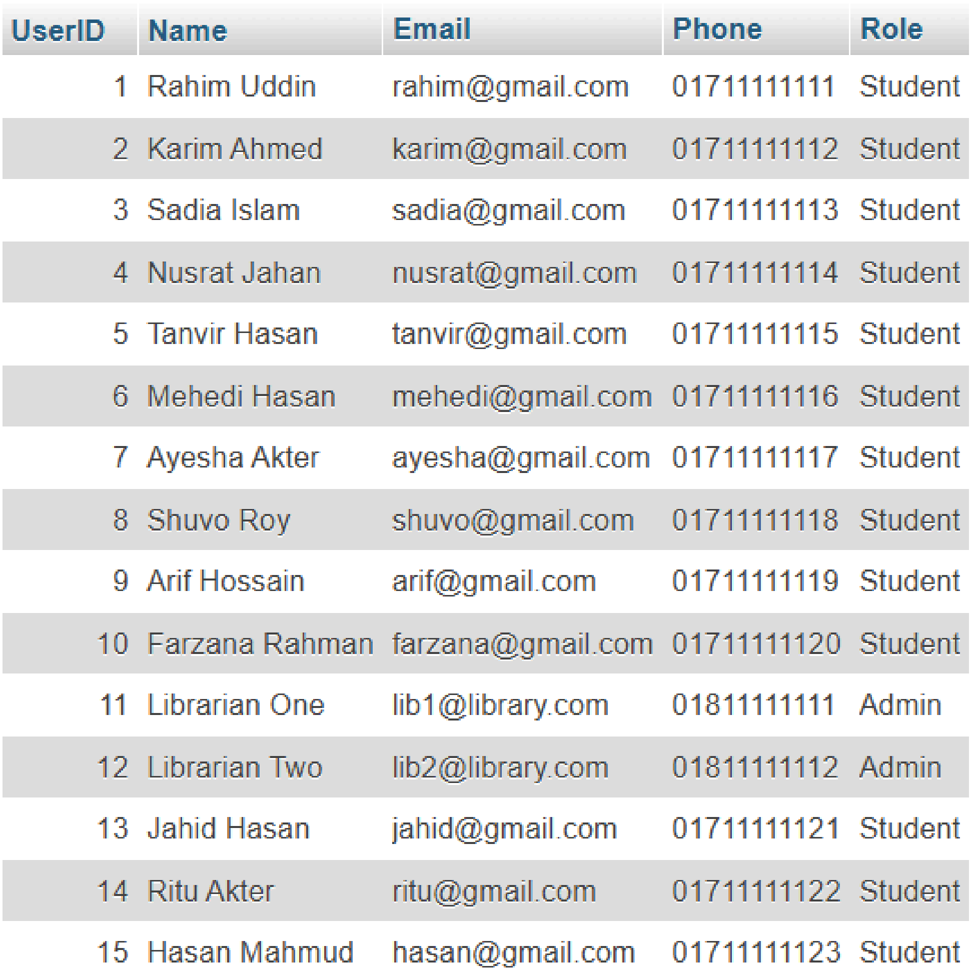
- ReservationID INT(11) **(PK)**
- ReservationDate DATE
- UserID INT(11) **(FK)** → References user(UserID)
- BookID INT(11) **(FK)** → References book(BookID)



7. Database Implementation (SQL Scripts)

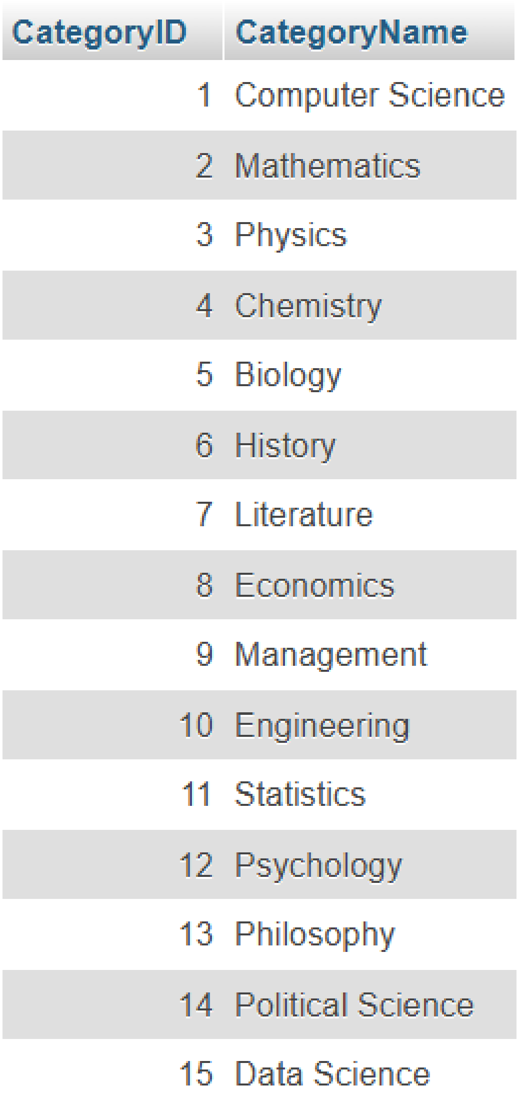
```
CREATE DATABASE LibraryManagementSystem;


CREATE TABLE user (
   UserID INT(11) PRIMARY KEY,
   Name VARCHAR(100) NOT NULL,
   Email VARCHAR(100) UNIQUE,
   Phone VARCHAR(15),
   Role VARCHAR(50)
);
```



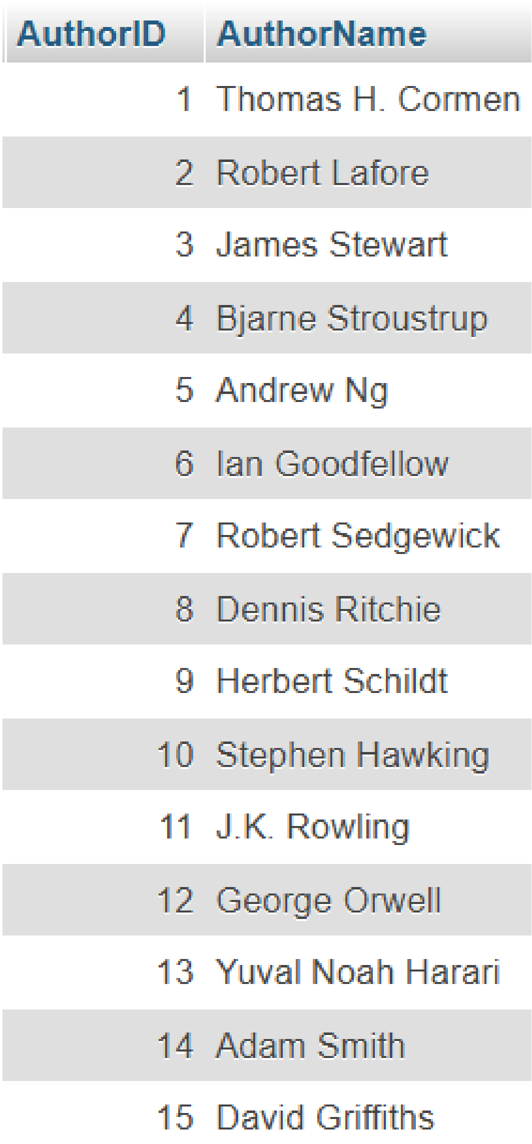
UserID	Name	Email	Phone	Role
1	Rahim Uddin	rahim@gmail.com	01711111111	Student
2	Karim Ahmed	karim@gmail.com	01711111112	Student
3	Sadia Islam	sadia@gmail.com	01711111113	Student
4	Nusrat Jahan	nusrat@gmail.com	01711111114	Student
5	Tanvir Hasan	tanvir@gmail.com	01711111115	Student
6	Mehedi Hasan	mehedi@gmail.com	01711111116	Student
7	Ayesha Akter	ayesha@gmail.com	01711111117	Student
8	Shuvo Roy	shuvo@gmail.com	01711111118	Student
9	Arif Hossain	arif@gmail.com	01711111119	Student
10	Farzana Rahman	farzana@gmail.com	01711111120	Student
11	Librarian One	lib1@library.com	01811111111	Admin
12	Librarian Two	lib2@library.com	01811111112	Admin
13	Jahid Hasan	jahid@gmail.com	01711111121	Student
14	Ritu Akter	ritu@gmail.com	01711111122	Student
15	Hasan Mahmud	hasan@gmail.com	01711111123	Student

```
CREATE TABLE category (
    CategoryID INT(11) PRIMARY KEY,
    CategoryName VARCHAR(100) NOT NULL
);
```



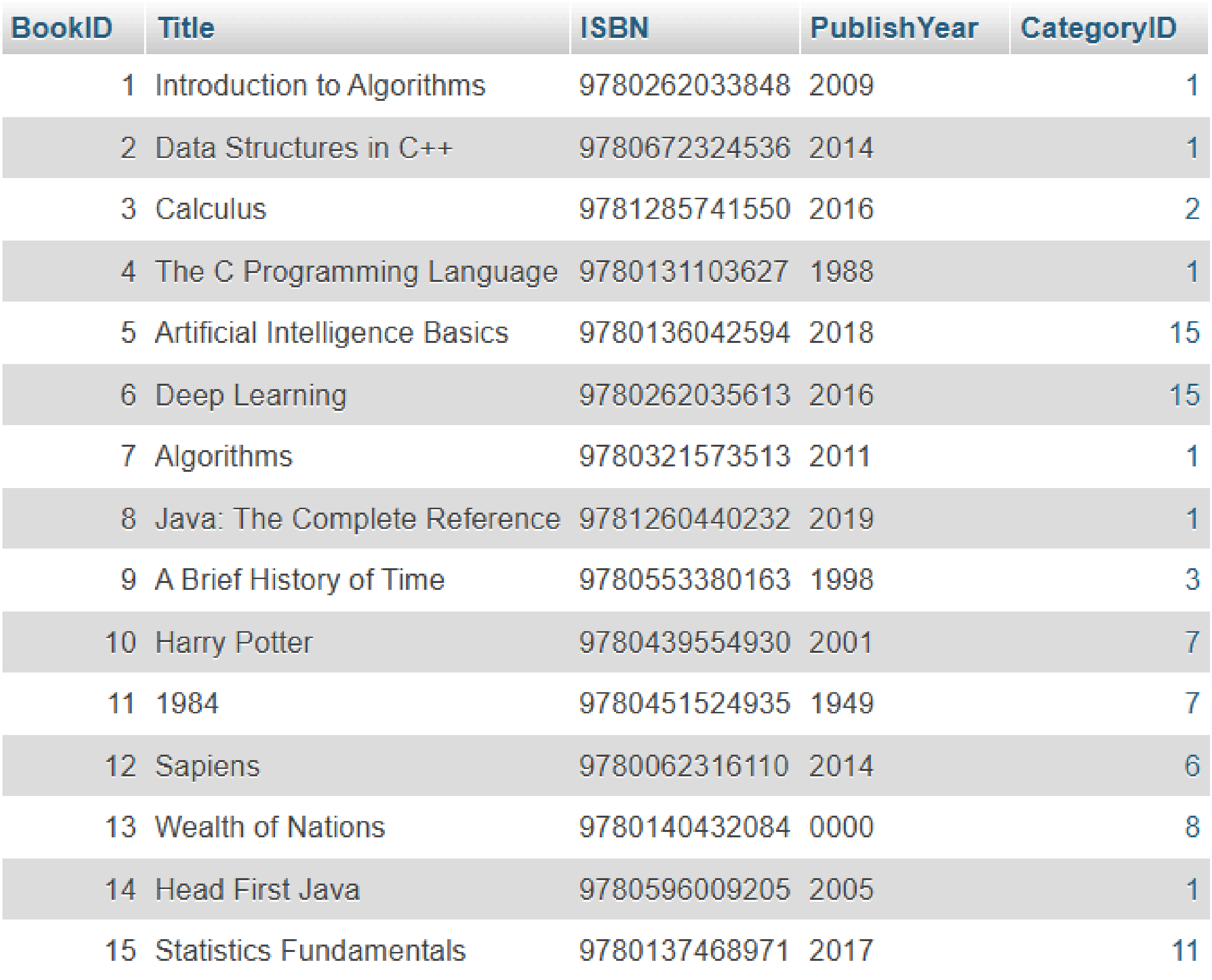
CategoryID	CategoryName
1	Computer Science
2	Mathematics
3	Physics
4	Chemistry
5	Biology
6	History
7	Literature
8	Economics
9	Management
10	Engineering
11	Statistics
12	Psychology
13	Philosophy
14	Political Science
15	Data Science

```
CREATE TABLE  author (
    AuthorID INT(11) PRIMARY KEY,
    AuthorName VARCHAR(100) NOT NULL
);
```



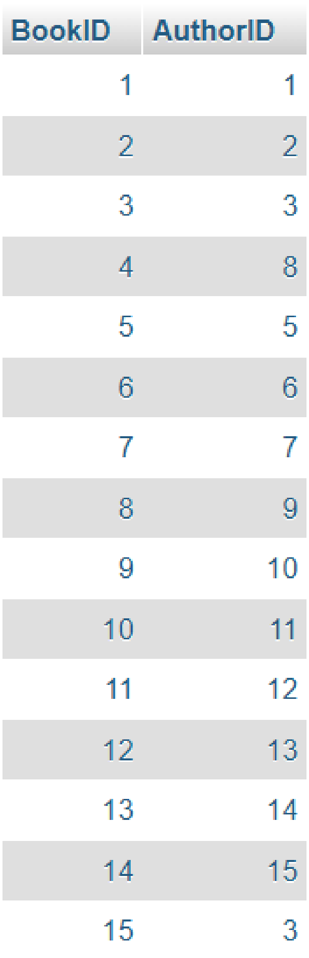
AuthorID	AuthorName
1	Thomas H. Cormen
2	Robert Lafore
3	James Stewart
4	Bjarne Stroustrup
5	Andrew Ng
6	Ian Goodfellow
7	Robert Sedgewick
8	Dennis Ritchie
9	Herbert Schildt
10	Stephen Hawking
11	J.K. Rowling
12	George Orwell
13	Yuval Noah Harari
14	Adam Smith
15	David Griffiths

```
CREATE TABLE book (
   BookID INT(11) PRIMARY KEY,
   Title VARCHAR(200) NOT NULL,
   ISBN VARCHAR(20) UNIQUE,
   PublishYear YEAR,
   CategoryID INT(11),
   FOREIGN KEY (CategoryID) REFERENCES category(CategoryID)
);
```



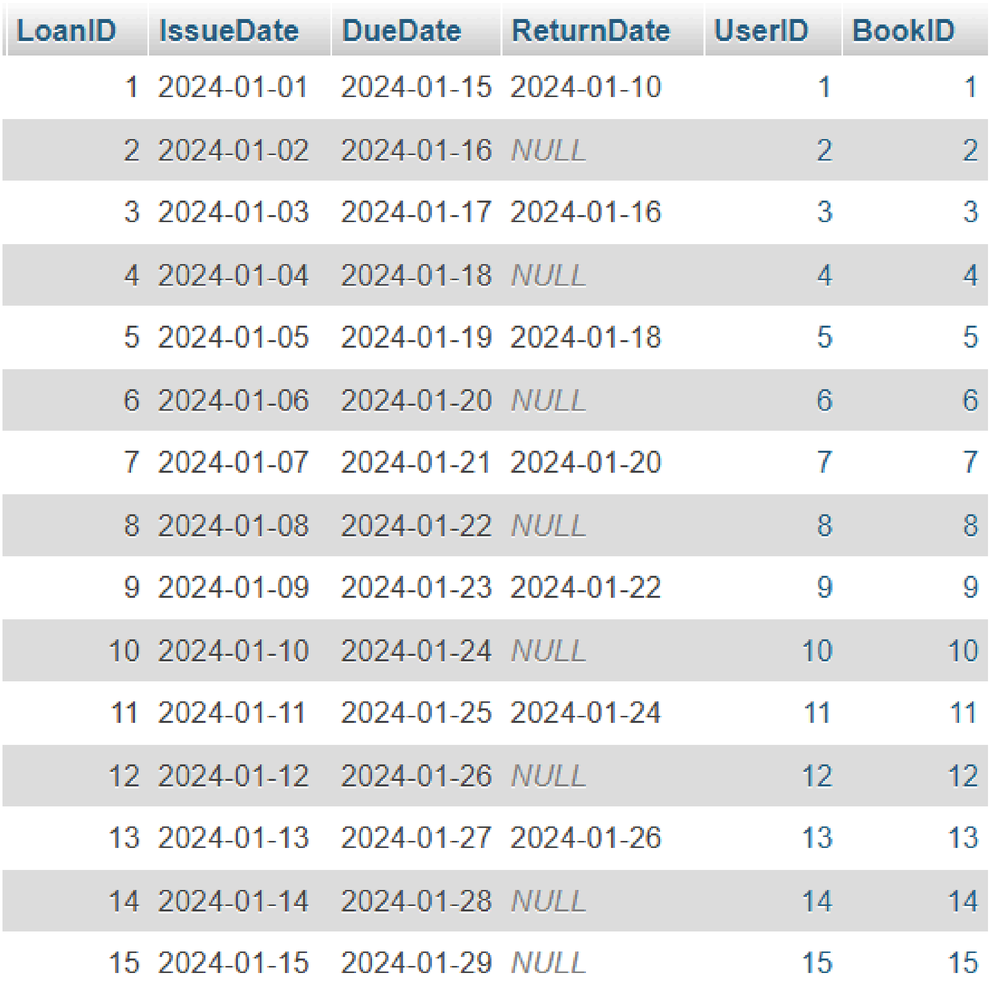
BookID	Title	ISBN	PublishYear	CategoryID
1	Introduction to Algorithms	9780262033848	2009	1
2	Data Structures in C++	9780672324536	2014	1
3	Calculus	9781285741550	2016	2
4	The C Programming Language	9780131103627	1988	1
5	Artificial Intelligence Basics	9780136042594	2018	15
6	Deep Learning	9780262035613	2016	15
7	Algorithms	9780321573513	2011	1
8	Java: The Complete Reference	9781260440232	2019	1
9	A Brief History of Time	9780553380163	1998	3
10	Harry Potter	9780439554930	2001	7
11	1984	9780451524935	1949	7
12	Sapiens	9780062316110	2014	6
13	Wealth of Nations	9780140432084	0000	8
14	Head First Java	9780596009205	2005	1
15	Statistics Fundamentals	9780137468971	2017	11

```
CREATE TABLE book_author (
   BookID INT(11),
   AuthorID INT(11),
   PRIMARY KEY (BookID, AuthorID),
   FOREIGN KEY (BookID) REFERENCES book(BookID),
   FOREIGN KEY (AuthorID) REFERENCES author(AuthorID)
);
```



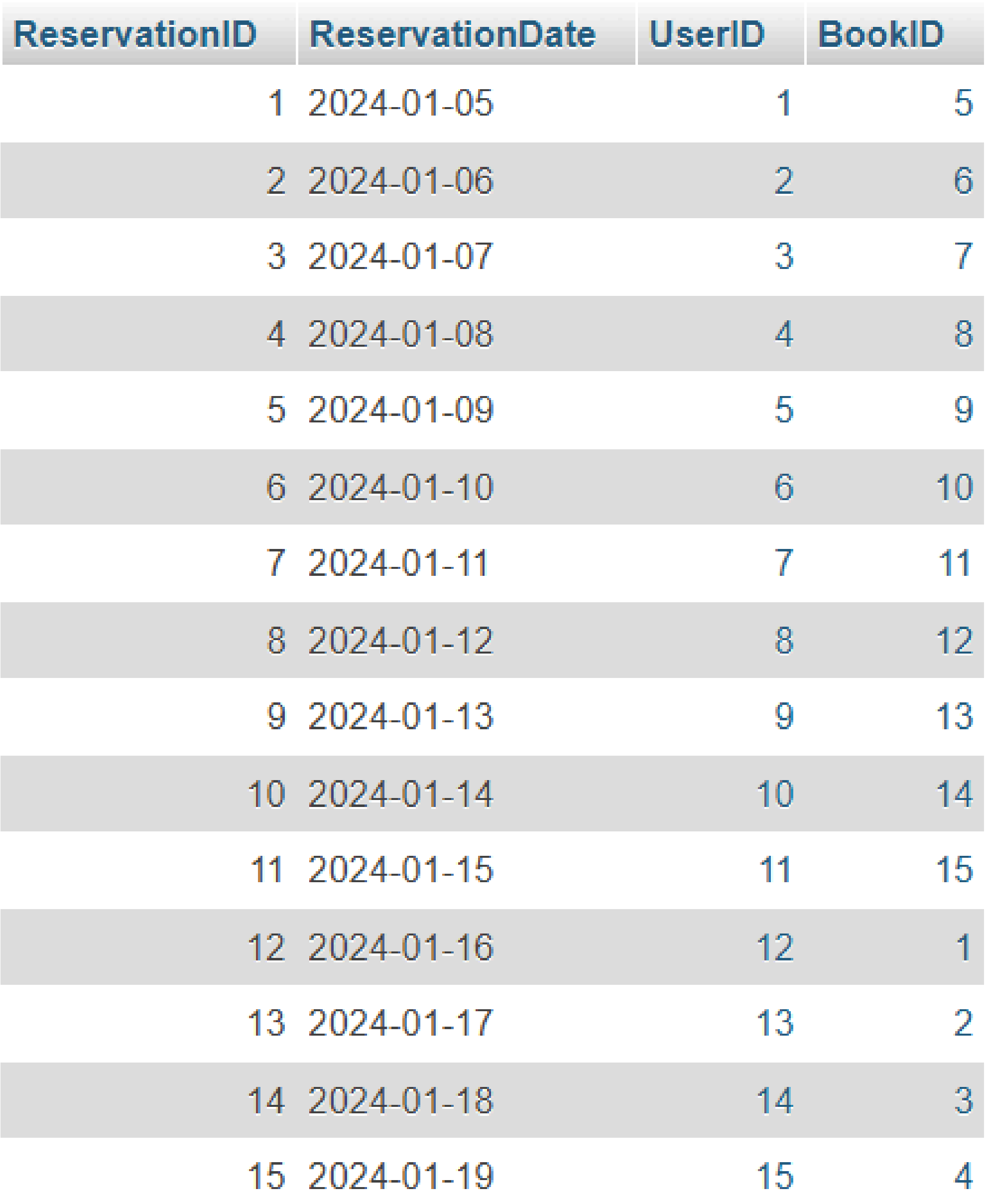
BookID	AuthorID
1	1
2	2
3	3
4	8
5	5
6	6
7	7
8	9
9	10
10	11
11	12
12	13
13	14
14	15
15	3

```
CREATE TABLE loan (
   LoanID INT(11) PRIMARY KEY,
   IssueDate DATE NOT NULL,
   DueDate DATE NOT NULL,
   ReturnDate DATE,
   UserID INT(11),
   BookID INT(11),
   FOREIGN KEY (UserID) REFERENCES user(UserID),
   FOREIGN KEY (BookID) REFERENCES book(BookID)
);
```



LoanID	IssueDate	DueDate	ReturnDate	UserID	BookID
1	2024-01-01	2024-01-15	2024-01-10	1	1
2	2024-01-02	2024-01-16	*NULL*	2	2
3	2024-01-03	2024-01-17	2024-01-16	3	3
4	2024-01-04	2024-01-18	*NULL*	4	4
5	2024-01-05	2024-01-19	2024-01-18	5	5
6	2024-01-06	2024-01-20	*NULL*	6	6
7	2024-01-07	2024-01-21	2024-01-20	7	7
8	2024-01-08	2024-01-22	*NULL*	8	8
9	2024-01-09	2024-01-23	2024-01-22	9	9
10	2024-01-10	2024-01-24	*NULL*	10	10
11	2024-01-11	2024-01-25	2024-01-24	11	11
12	2024-01-12	2024-01-26	*NULL*	12	12
13	2024-01-13	2024-01-27	2024-01-26	13	13
14	2024-01-14	2024-01-28	*NULL*	14	14
15	2024-01-15	2024-01-29	*NULL*	15	15

```
CREATE TABLE reservation (
  ReservationID INT(11) PRIMARY KEY,
  ReservationDate DATE NOT NULL,
  UserID INT(11),
  BookID INT(11),
  FOREIGN KEY (UserID) REFERENCES user(UserID),
  FOREIGN KEY (BookID) REFERENCES book(BookID)
);
```

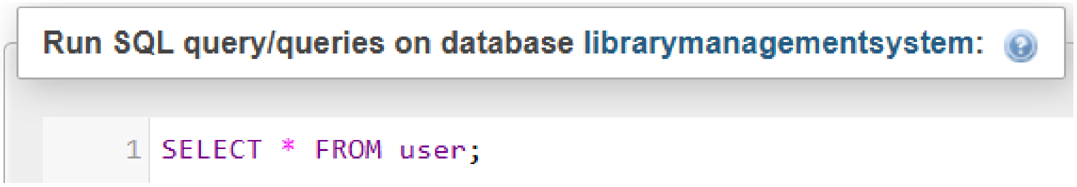


ReservationID	ReservationDate	UserID	BookID
1	2024-01-05	1	5
2	2024-01-06	2	6
3	2024-01-07	3	7
4	2024-01-08	4	8
5	2024-01-09	5	9
6	2024-01-10	6	10
7	2024-01-11	7	11
8	2024-01-12	8	12
9	2024-01-13	9	13
10	2024-01-14	10	14
11	2024-01-15	11	15
12	2024-01-16	12	1
13	2024-01-17	13	2
14	2024-01-18	14	3
15	2024-01-19	15	4

8. SQL Queries

1. **View all users.**

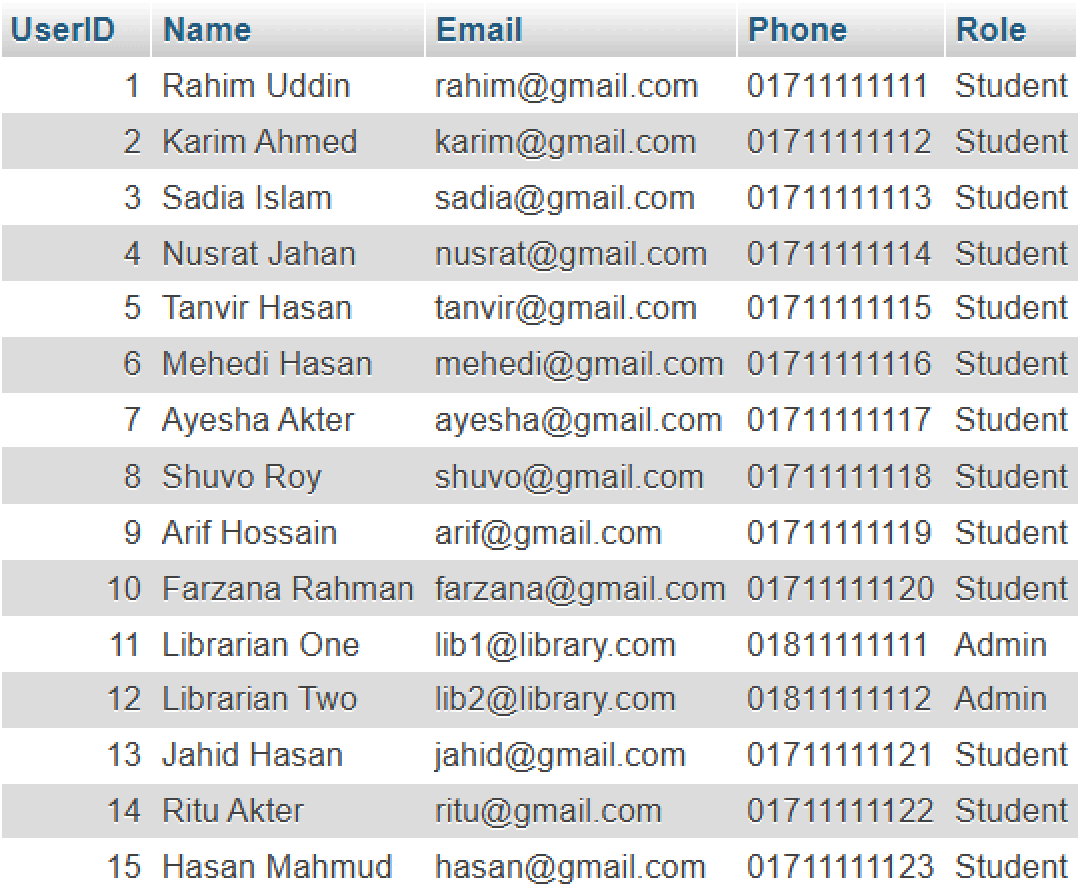
SQL Command:



Run SQL query/queries on database librarymanagementsystem:

```
SELECT * FROM user;
```

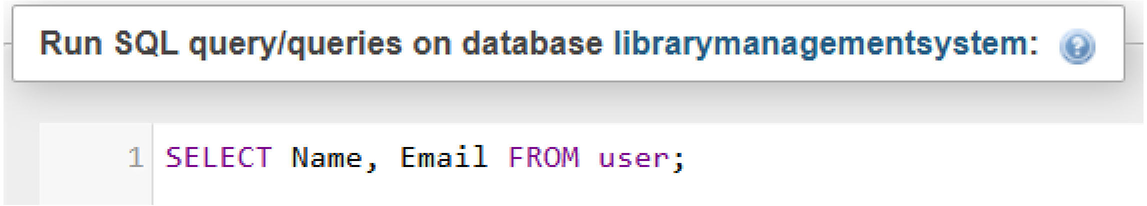
Output:



UserID	Name	Email	Phone	Role
1	Rahim Uddin	rahim@gmail.com	01711111111	Student
2	Karim Ahmed	karim@gmail.com	01711111112	Student
3	Sadia Islam	sadia@gmail.com	01711111113	Student
4	Nusrat Jahan	nusrat@gmail.com	01711111114	Student
5	Tanvir Hasan	tanvir@gmail.com	01711111115	Student
6	Mehedi Hasan	mehedi@gmail.com	01711111116	Student
7	Ayesha Akter	ayesha@gmail.com	01711111117	Student
8	Shuvo Roy	shuvo@gmail.com	01711111118	Student
9	Arif Hossain	arif@gmail.com	01711111119	Student
10	Farzana Rahman	farzana@gmail.com	01711111120	Student
11	Librarian One	lib1@library.com	01811111111	Admin
12	Librarian Two	lib2@library.com	01811111112	Admin
13	Jahid Hasan	jahid@gmail.com	01711111121	Student
14	Ritu Akter	ritu@gmail.com	01711111122	Student
15	Hasan Mahmud	hasan@gmail.com	01711111123	Student

2. **Display user name and email.**

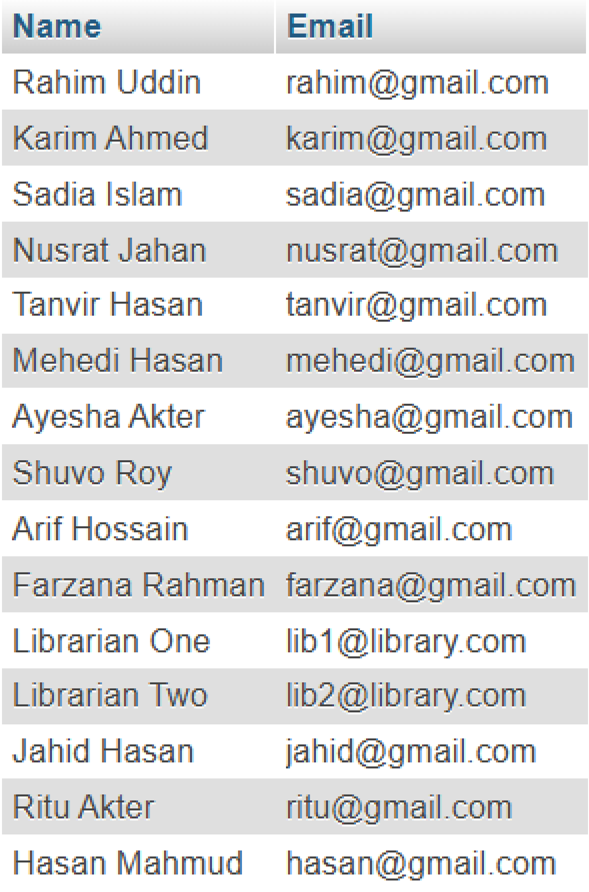
SQL Command:



Run SQL query/queries on database librarymanagementsystem:

```sql
SELECT Name, Email FROM user;
```

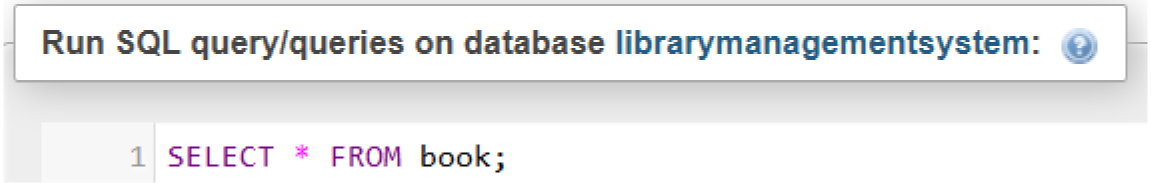
Output:



Name	Email
Rahim Uddin	rahim@gmail.com
Karim Ahmed	karim@gmail.com
Sadia Islam	sadia@gmail.com
Nusrat Jahan	nusrat@gmail.com
Tanvir Hasan	tanvir@gmail.com
Mehedi Hasan	mehedi@gmail.com
Ayesha Akter	ayesha@gmail.com
Shuvo Roy	shuvo@gmail.com
Arif Hossain	arif@gmail.com
Farzana Rahman	farzana@gmail.com
Librarian One	lib1@library.com
Librarian Two	lib2@library.com
Jahid Hasan	jahid@gmail.com
Ritu Akter	ritu@gmail.com
Hasan Mahmud	hasan@gmail.com

3. **View all books.**

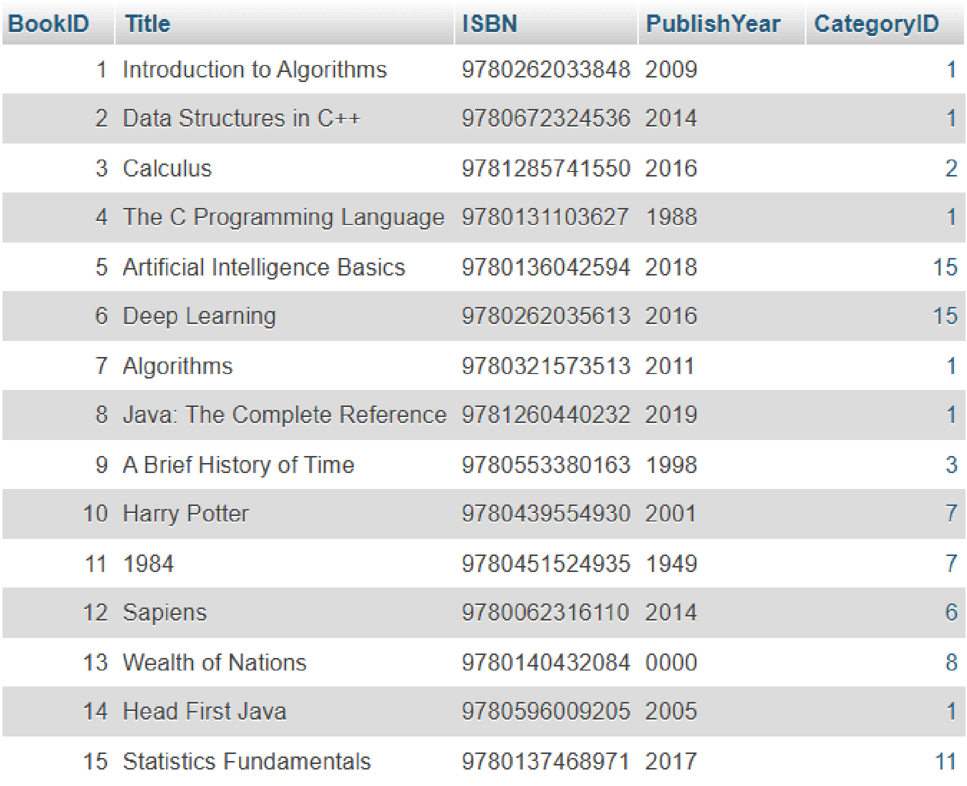
SQL Command:



Run SQL query/queries on database librarymanagementsystem:

```sql
SELECT * FROM book;
```

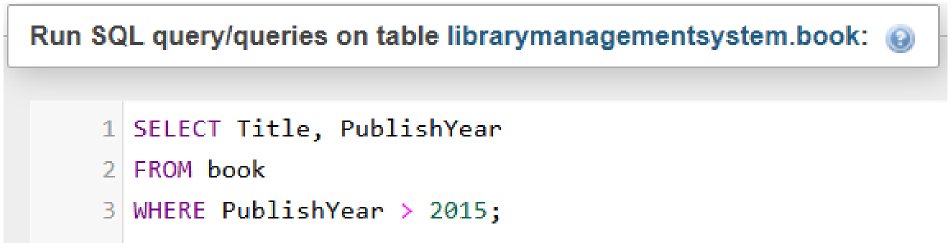
Output:



BookID	Title	ISBN	PublishYear	CategoryID
1	Introduction to Algorithms	9780262033848	2009	1
2	Data Structures in C++	9780672324536	2014	1
3	Calculus	9781285741550	2016	2
4	The C Programming Language	9780131103627	1988	1
5	Artificial Intelligence Basics	9780136042594	2018	15
6	Deep Learning	9780262035613	2016	15
7	Algorithms	9780321573513	2011	1
8	Java: The Complete Reference	9781260440232	2019	1
9	A Brief History of Time	9780553380163	1998	3
10	Harry Potter	9780439554930	2001	7
11	1984	9780451524935	1949	7
12	Sapiens	9780062316110	2014	6
13	Wealth of Nations	9780140432084	0000	8
14	Head First Java	9780596009205	2005	1
15	Statistics Fundamentals	9780137468971	2017	11

4. **Display book titles published after 2015.**

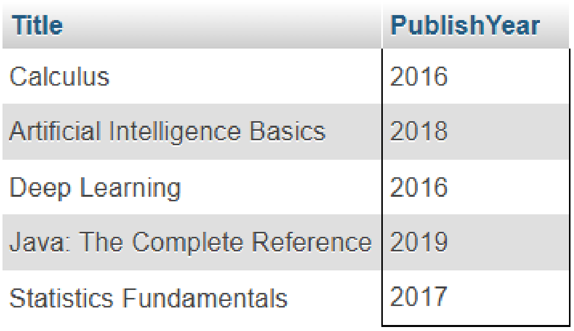
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT Title, PublishYear
FROM book
WHERE PublishYear > 2015;
```

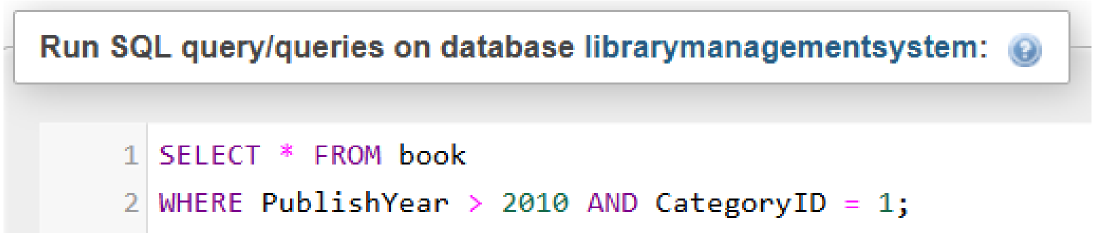
Output:



Title	PublishYear
Calculus	2016
Artificial Intelligence Basics	2018
Deep Learning	2016
Java: The Complete Reference	2019
Statistics Fundamentals	2017

5. **Books published after 2010 AND belonging to CategoryID = 1.**

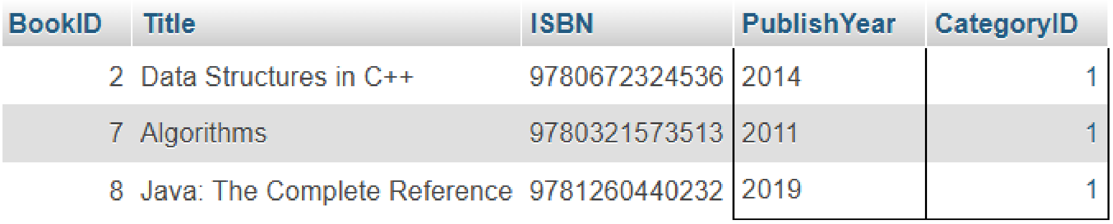
SQL Command:



Run SQL query/queries on database librarymanagementsystem:

```
SELECT * FROM book
WHERE PublishYear > 2010 AND CategoryID = 1;
```

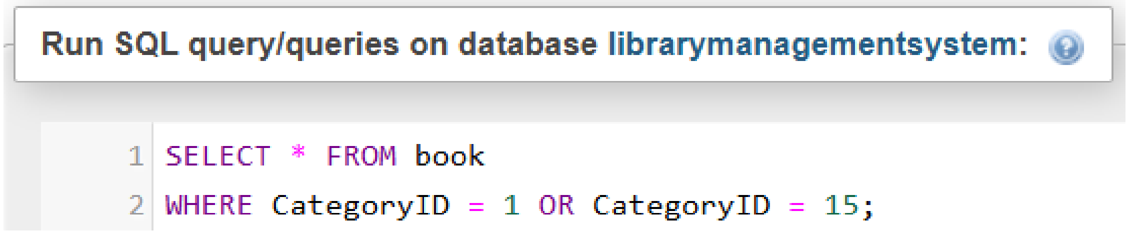
Output:



BookID	Title	ISBN	PublishYear	CategoryID
2	Data Structures in C++	9780672324536	2014	1
7	Algorithms	9780321573513	2011	1
8	Java: The Complete Reference	9781260440232	2019	1

6. **Books from CategoryID 1 OR CategoryID 15.**

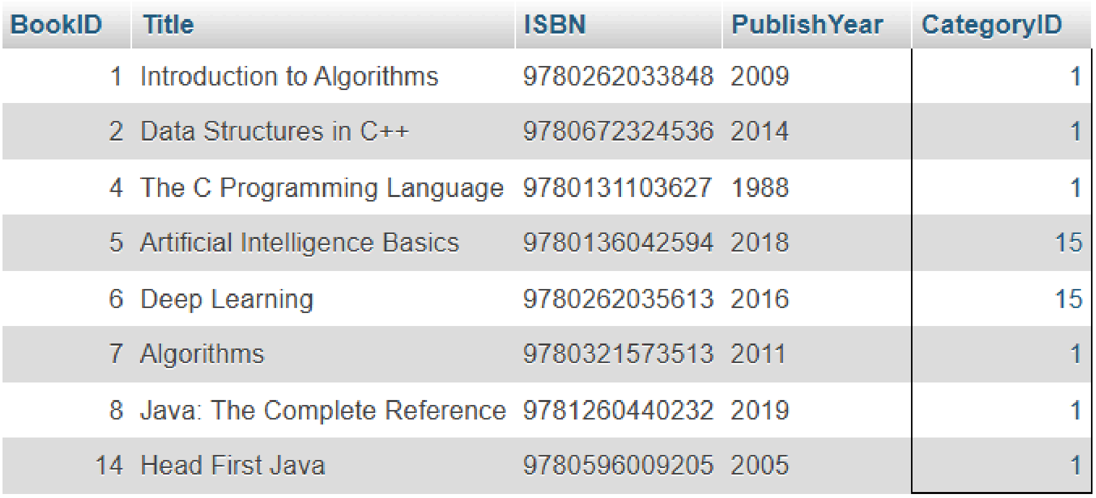
SQL Command:



Run SQL query/queries on database librarymanagementsystem:

```
SELECT * FROM book
WHERE CategoryID = 1 OR CategoryID = 15;
```

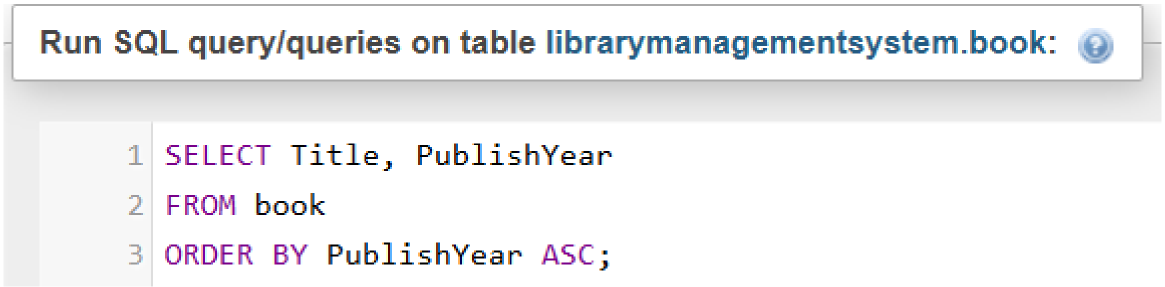
Output:



BookID	Title	ISBN	PublishYear	CategoryID
1	Introduction to Algorithms	9780262033848	2009	1
2	Data Structures in C++	9780672324536	2014	1
4	The C Programming Language	9780131103627	1988	1
5	Artificial Intelligence Basics	9780136042594	2018	15
6	Deep Learning	9780262035613	2016	15
7	Algorithms	9780321573513	2011	1
8	Java: The Complete Reference	9781260440232	2019	1
14	Head First Java	9780596009205	2005	1

7. **Books ordered by publish year (ascending).**

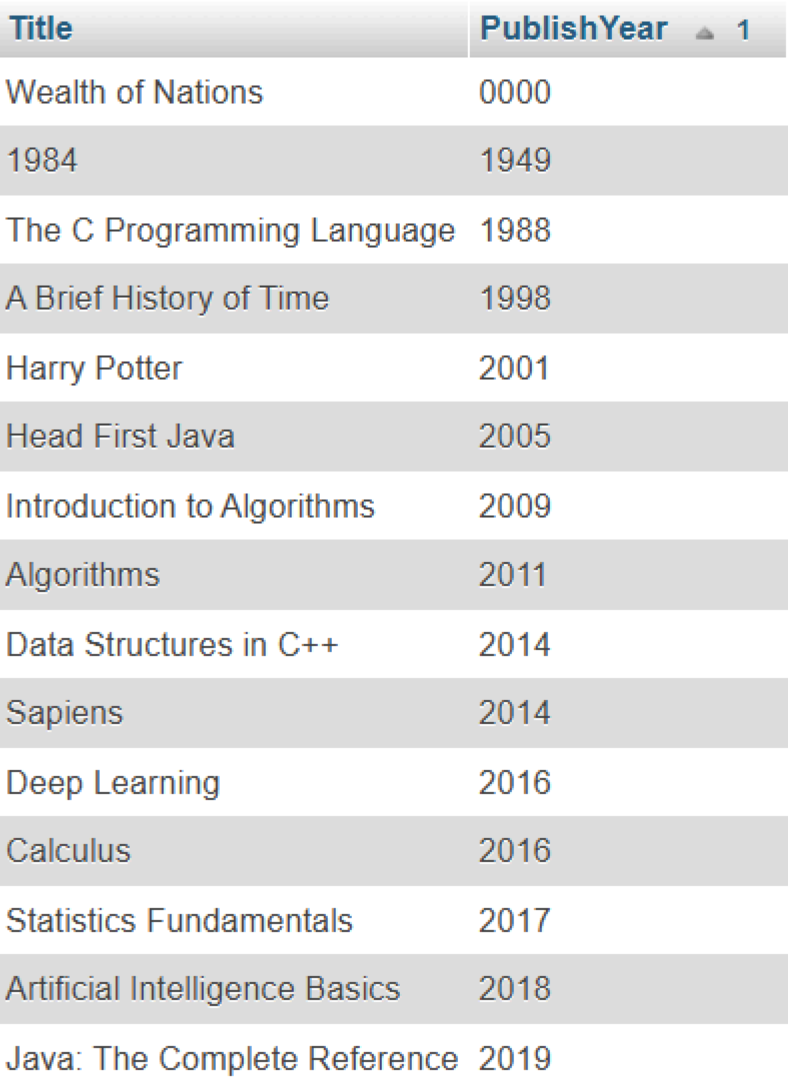
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT Title, PublishYear
FROM book
ORDER BY PublishYear ASC;
```

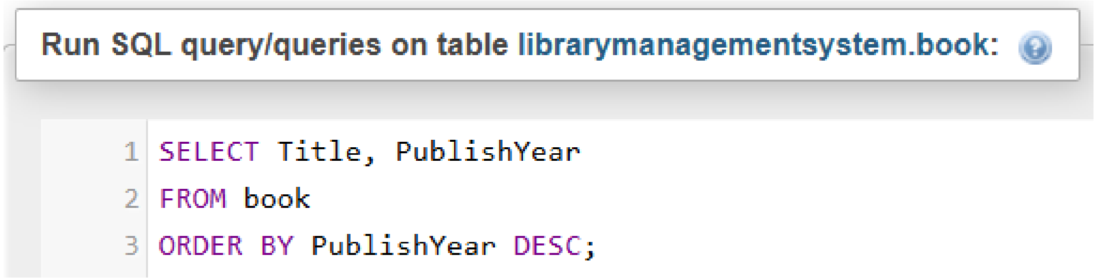
Output:



Title	PublishYear ▲ 1
Wealth of Nations	0000
1984	1949
The C Programming Language	1988
A Brief History of Time	1998
Harry Potter	2001
Head First Java	2005
Introduction to Algorithms	2009
Algorithms	2011
Data Structures in C++	2014
Sapiens	2014
Deep Learning	2016
Calculus	2016
Statistics Fundamentals	2017
Artificial Intelligence Basics	2018
Java: The Complete Reference	2019

8. **Books ordered by publish year (descending).**

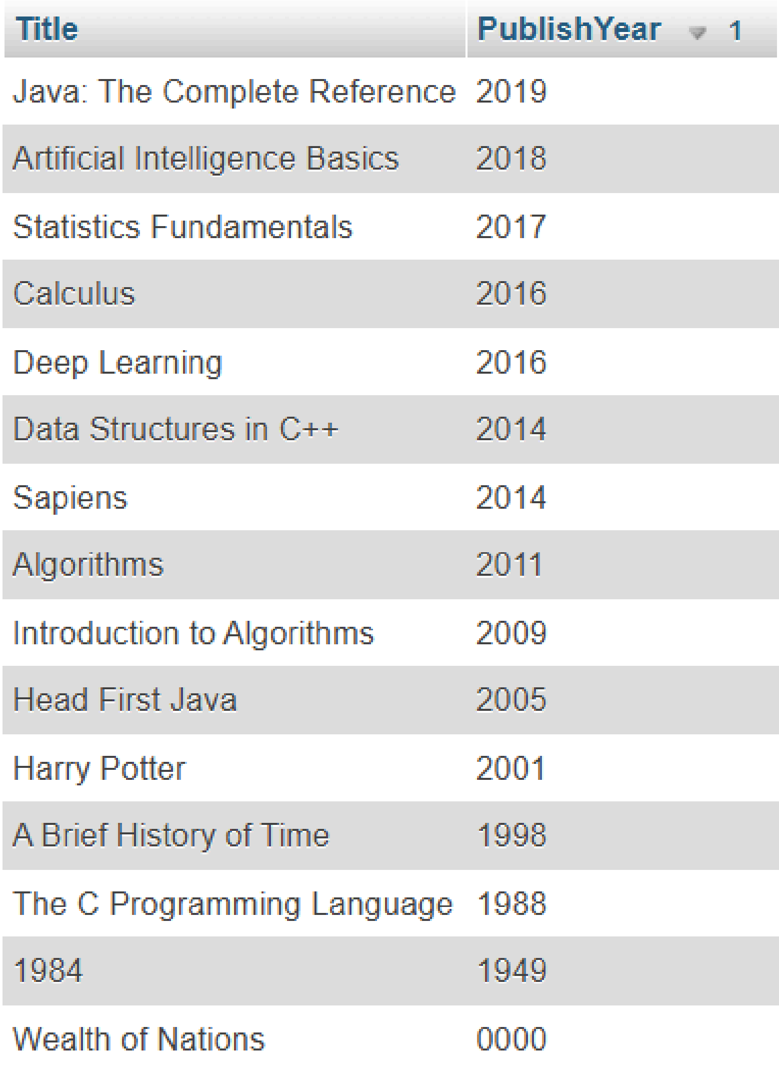
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```sql
SELECT Title, PublishYear
FROM book
ORDER BY PublishYear DESC;
```

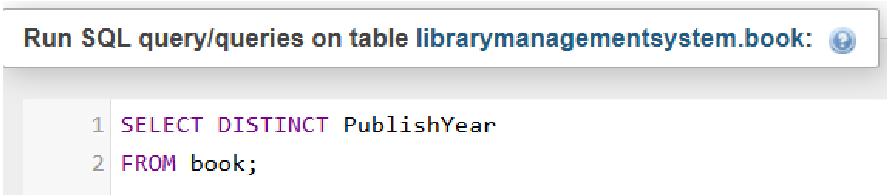
Output:



Title	PublishYear ▼ 1
Java: The Complete Reference	2019
Artificial Intelligence Basics	2018
Statistics Fundamentals	2017
Calculus	2016
Deep Learning	2016
Data Structures in C++	2014
Sapiens	2014
Algorithms	2011
Introduction to Algorithms	2009
Head First Java	2005
Harry Potter	2001
A Brief History of Time	1998
The C Programming Language	1988
1984	1949
Wealth of Nations	0000

9. **Display distinct publish years.**

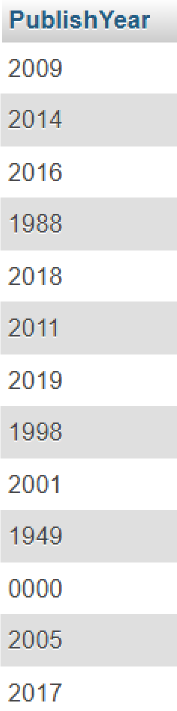
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT DISTINCT PublishYear
FROM book;
```

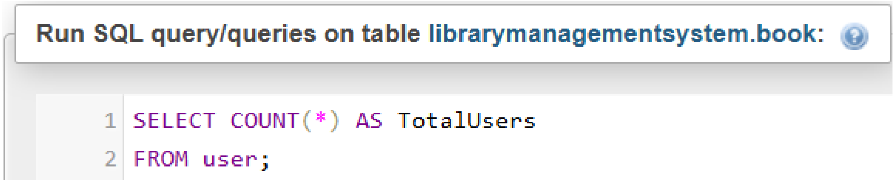
Output:



PublishYear
2009
2014
2016
1988
2018
2011
2019
1998
2001
1949
0000
2005
2017

10. **Total number of users.**

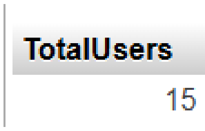
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT COUNT(*) AS TotalUsers
FROM user;
```

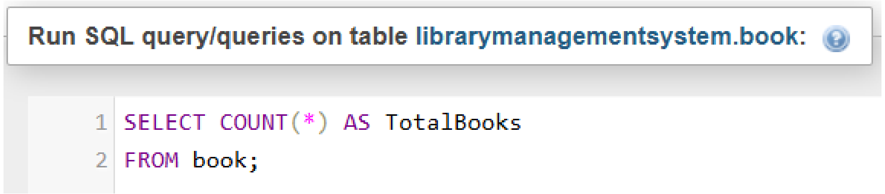
Output:



TotalUsers
15

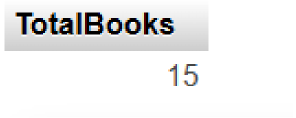
11. **Total number of books.**

SQL Command:


Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT COUNT(*) AS TotalBooks
FROM book;
```

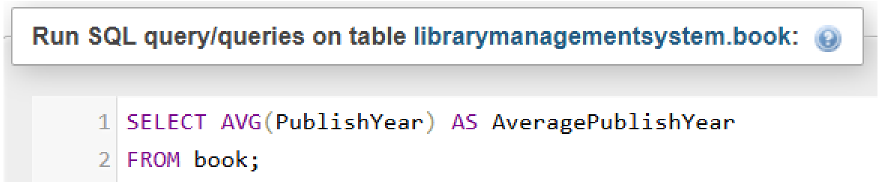
Output:



TotalBooks
15

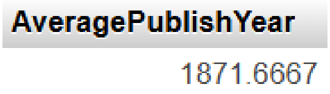
12. **Average publish year of books.**

SQL Command:


Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT AVG(PublishYear) AS AveragePublishYear
FROM book;
```

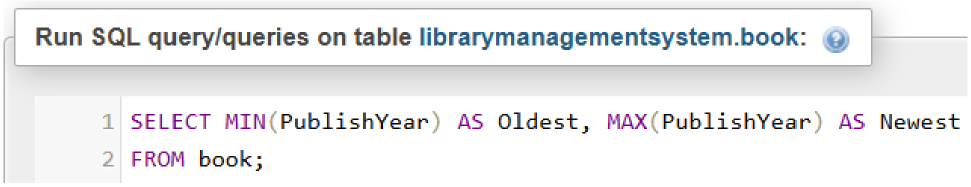
Output:



AveragePublishYear
1871.6667

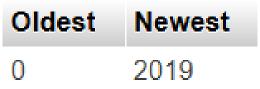
13. **Oldest and newest book publish year.**

SQL Command:


Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT MIN(PublishYear) AS Oldest, MAX(PublishYear) AS Newest
FROM book;
```

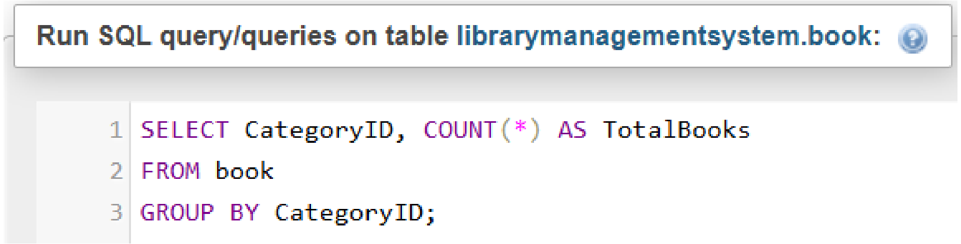
Output:



Oldest	Newest
0	2019

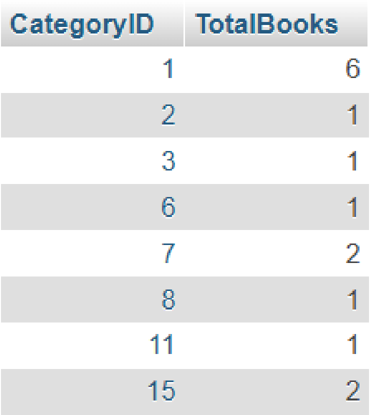
14. **Number of books in each category**

SQL Command:


Run SQL query/queries on table librarymanagementsystem.book:

```sql
SELECT CategoryID, COUNT(*) AS TotalBooks
FROM book
GROUP BY CategoryID;
```

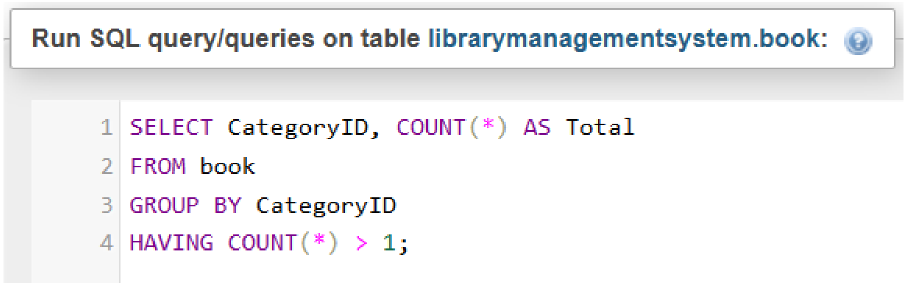
Output:



CategoryID	TotalBooks
1	6
2	1
3	1
6	1
7	2
8	1
11	1
15	2

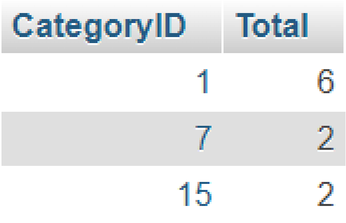
15. **Categories having more than one book**

SQL Command:


Run SQL query/queries on table librarymanagementsystem.book:

```sql
SELECT CategoryID, COUNT(*) AS Total
FROM book
GROUP BY CategoryID
HAVING COUNT(*) > 1;
```

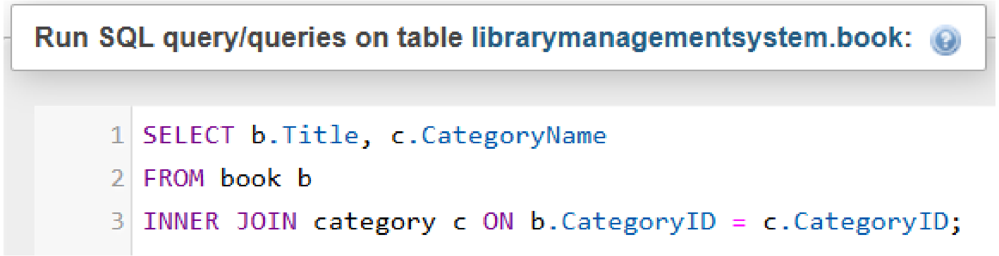
Output:



CategoryID	Total
1	6
7	2
15	2

16. **Display books with their category names.**

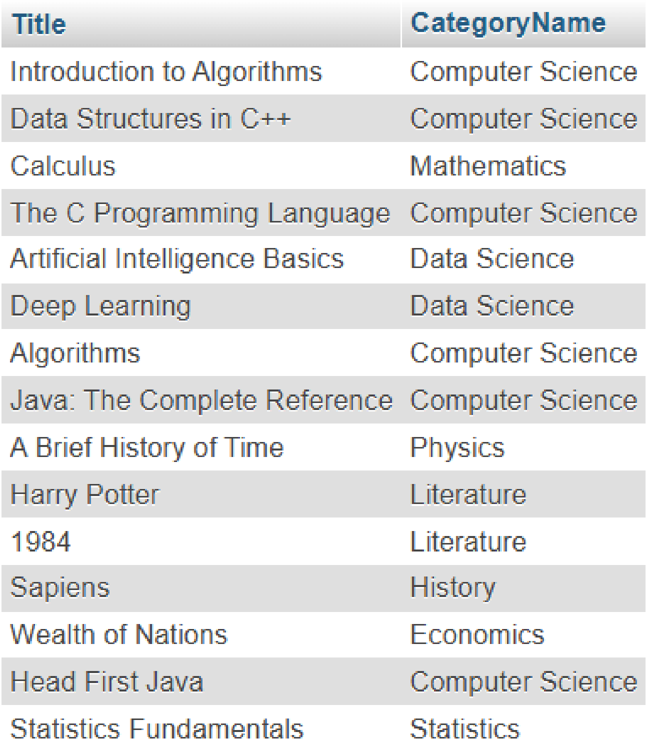
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT b.Title, c.CategoryName
FROM book b
INNER JOIN category c ON b.CategoryID = c.CategoryID;
```

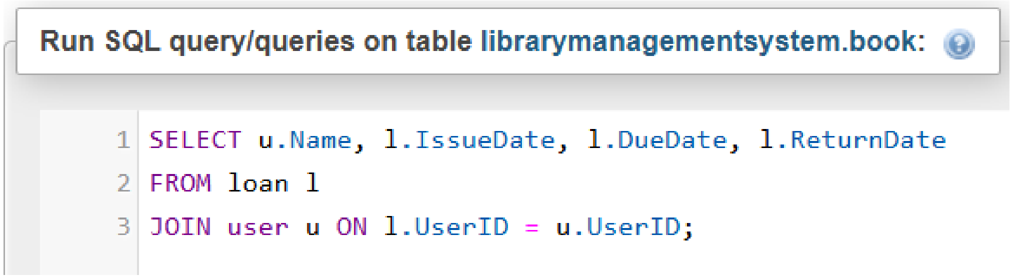
Output:



Title	CategoryName
Introduction to Algorithms	Computer Science
Data Structures in C++	Computer Science
Calculus	Mathematics
The C Programming Language	Computer Science
Artificial Intelligence Basics	Data Science
Deep Learning	Data Science
Algorithms	Computer Science
Java: The Complete Reference	Computer Science
A Brief History of Time	Physics
Harry Potter	Literature
1984	Literature
Sapiens	History
Wealth of Nations	Economics
Head First Java	Computer Science
Statistics Fundamentals	Statistics

17. **Display loan details with user name.**

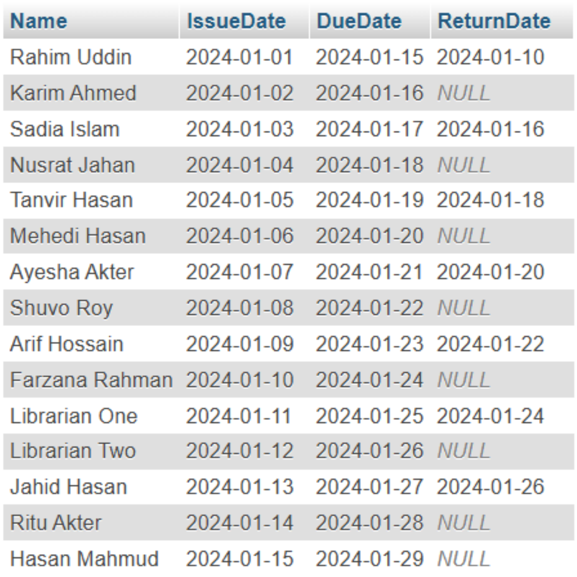
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT u.Name, l.IssueDate, l.DueDate, l.ReturnDate
FROM loan l
JOIN user u ON l.UserID = u.UserID;
```

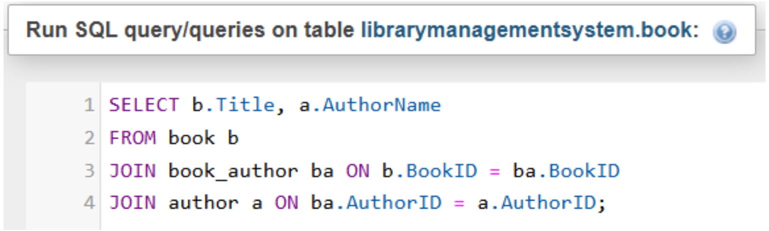
Output:



Name	IssueDate	DueDate	ReturnDate
Rahim Uddin	2024-01-01	2024-01-15	2024-01-10
Karim Ahmed	2024-01-02	2024-01-16	*NULL*
Sadia Islam	2024-01-03	2024-01-17	2024-01-16
Nusrat Jahan	2024-01-04	2024-01-18	*NULL*
Tanvir Hasan	2024-01-05	2024-01-19	2024-01-18
Mehedi Hasan	2024-01-06	2024-01-20	*NULL*
Ayesha Akter	2024-01-07	2024-01-21	2024-01-20
Shuvo Roy	2024-01-08	2024-01-22	*NULL*
Arif Hossain	2024-01-09	2024-01-23	2024-01-22
Farzana Rahman	2024-01-10	2024-01-24	*NULL*
Librarian One	2024-01-11	2024-01-25	2024-01-24
Librarian Two	2024-01-12	2024-01-26	*NULL*
Jahid Hasan	2024-01-13	2024-01-27	2024-01-26
Ritu Akter	2024-01-14	2024-01-28	*NULL*
Hasan Mahmud	2024-01-15	2024-01-29	*NULL*

18. **Display book title with author name.**

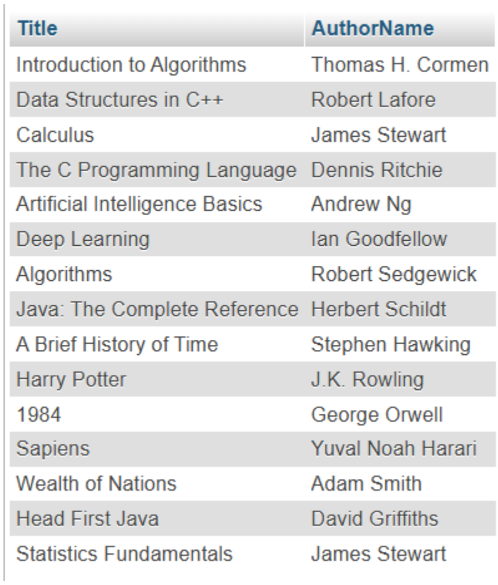
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```sql
SELECT b.Title, a.AuthorName
FROM book b
JOIN book_author ba ON b.BookID = ba.BookID
JOIN author a ON ba.AuthorID = a.AuthorID;
```

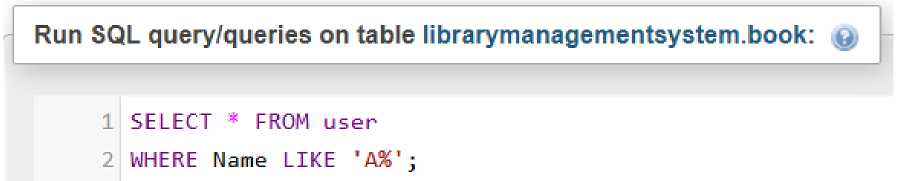
Output:



Title	AuthorName
Introduction to Algorithms	Thomas H. Cormen
Data Structures in C++	Robert Lafore
Calculus	James Stewart
The C Programming Language	Dennis Ritchie
Artificial Intelligence Basics	Andrew Ng
Deep Learning	Ian Goodfellow
Algorithms	Robert Sedgewick
Java: The Complete Reference	Herbert Schildt
A Brief History of Time	Stephen Hawking
Harry Potter	J.K. Rowling
1984	George Orwell
Sapiens	Yuval Noah Harari
Wealth of Nations	Adam Smith
Head First Java	David Griffiths
Statistics Fundamentals	James Stewart

19. **Users whose name starts with 'A'**

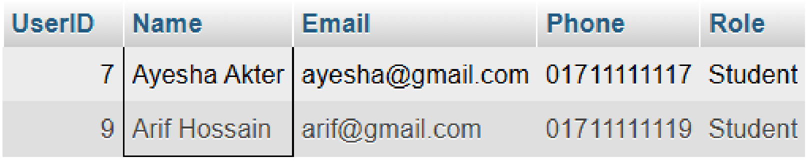
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT * FROM user
WHERE Name LIKE 'A%';
```

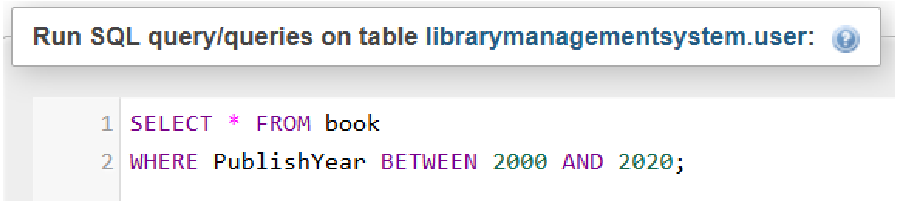
Output:



UserID	Name	Email	Phone	Role
7	Ayesha Akter	ayesha@gmail.com	01711111117	Student
9	Arif Hossain	arif@gmail.com	01711111119	Student

20. **Books published between 2000 and 2020**

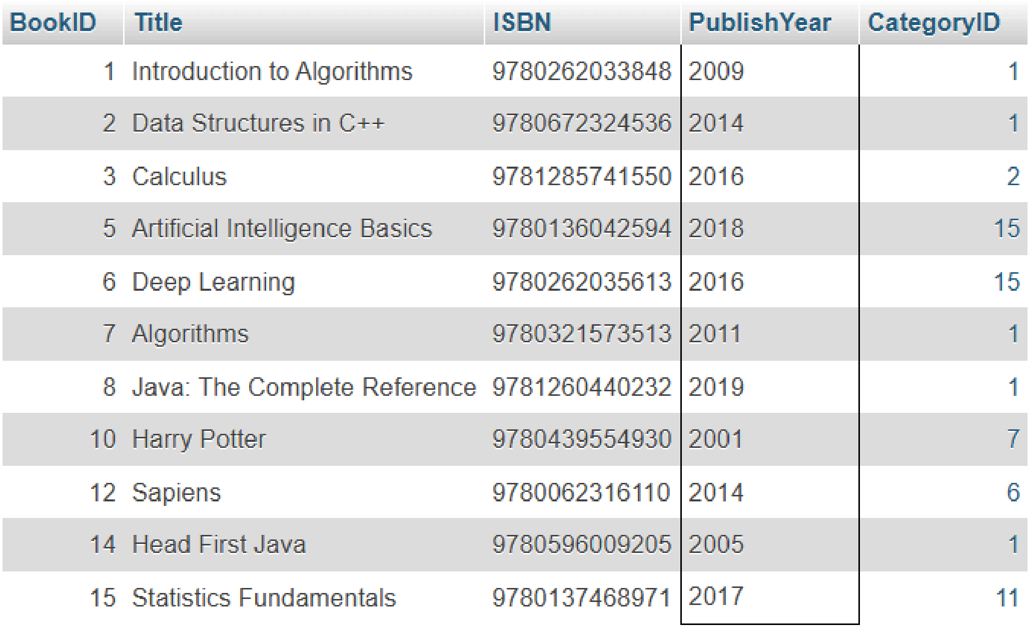
SQL Command:



Run SQL query/queries on table librarymanagementsystem.user:

```
SELECT * FROM book
WHERE PublishYear BETWEEN 2000 AND 2020;
```

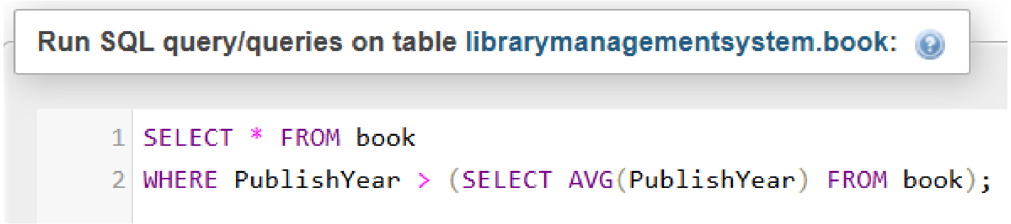
Output:



BookID	Title	ISBN	PublishYear	CategoryID
1	Introduction to Algorithms	9780262033848	2009	1
2	Data Structures in C++	9780672324536	2014	1
3	Calculus	9781285741550	2016	2
5	Artificial Intelligence Basics	9780136042594	2018	15
6	Deep Learning	9780262035613	2016	15
7	Algorithms	9780321573513	2011	1
8	Java: The Complete Reference	9781260440232	2019	1
10	Harry Potter	9780439554930	2001	7
12	Sapiens	9780062316110	2014	6
14	Head First Java	9780596009205	2005	1
15	Statistics Fundamentals	9780137468971	2017	11

21. **Books published after the average publish year.**

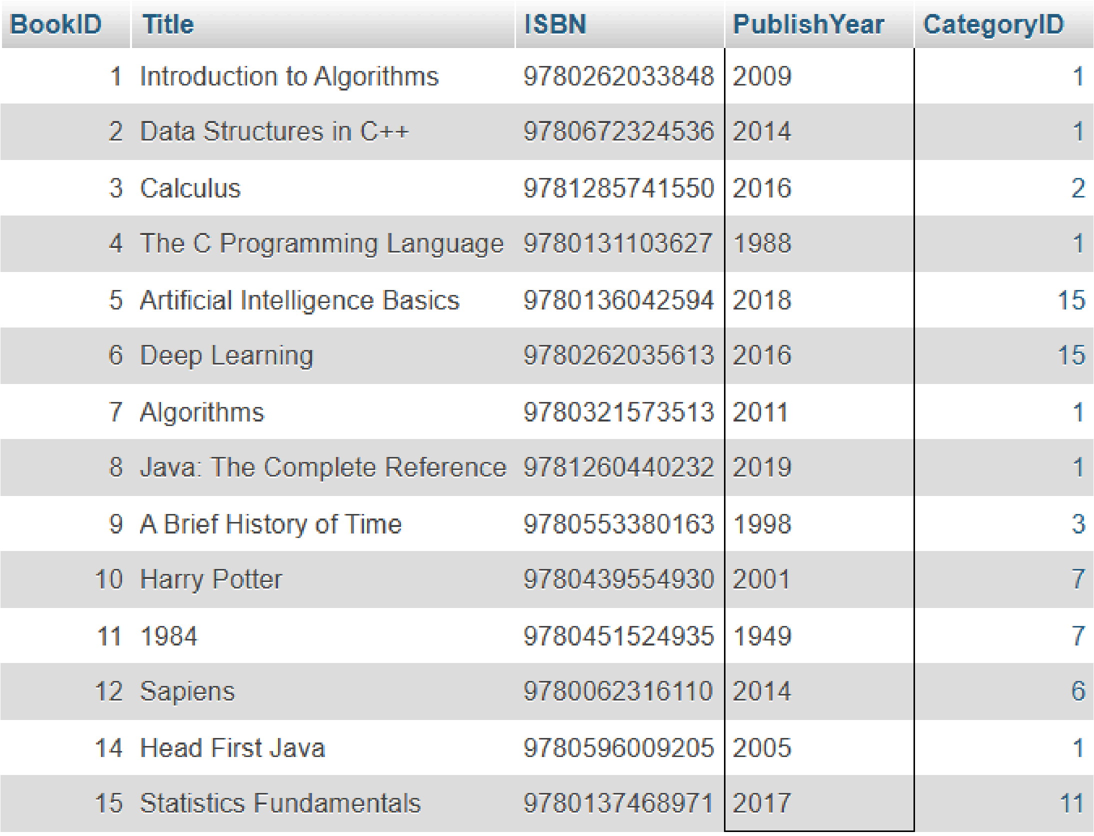
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT * FROM book
WHERE PublishYear > (SELECT AVG(PublishYear) FROM book);
```

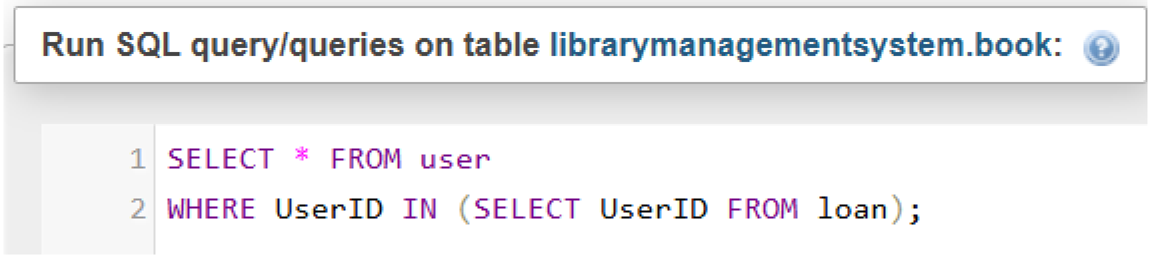
Output:



BookID	Title	ISBN	PublishYear	CategoryID
1	Introduction to Algorithms	9780262033848	2009	1
2	Data Structures in C++	9780672324536	2014	1
3	Calculus	9781285741550	2016	2
4	The C Programming Language	9780131103627	1988	1
5	Artificial Intelligence Basics	9780136042594	2018	15
6	Deep Learning	9780262035613	2016	15
7	Algorithms	9780321573513	2011	1
8	Java: The Complete Reference	9781260440232	2019	1
9	A Brief History of Time	9780553380163	1998	3
10	Harry Potter	9780439554930	2001	7
11	1984	9780451524935	1949	7
12	Sapiens	9780062316110	2014	6
14	Head First Java	9780596009205	2005	1
15	Statistics Fundamentals	9780137468971	2017	11

22. **Users who borrowed at least one book.**

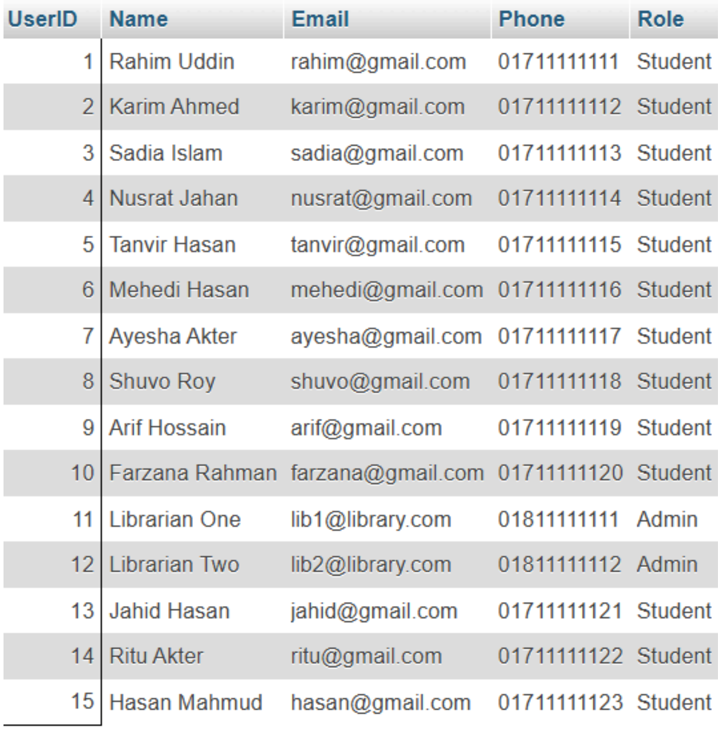
SQL Command:



Run SQL query/queries on table librarymanagementsystem.book:

```
SELECT * FROM user
WHERE UserID IN (SELECT UserID FROM loan);
```

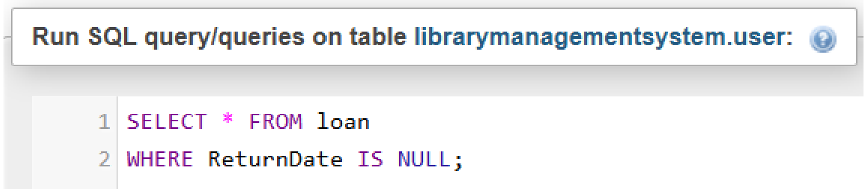
Output:



UserID	Name	Email	Phone	Role
1	Rahim Uddin	rahim@gmail.com	01711111111	Student
2	Karim Ahmed	karim@gmail.com	01711111112	Student
3	Sadia Islam	sadia@gmail.com	01711111113	Student
4	Nusrat Jahan	nusrat@gmail.com	01711111114	Student
5	Tanvir Hasan	tanvir@gmail.com	01711111115	Student
6	Mehedi Hasan	mehedi@gmail.com	01711111116	Student
7	Ayesha Akter	ayesha@gmail.com	01711111117	Student
8	Shuvo Roy	shuvo@gmail.com	01711111118	Student
9	Arif Hossain	arif@gmail.com	01711111119	Student
10	Farzana Rahman	farzana@gmail.com	01711111120	Student
11	Librarian One	lib1@library.com	01811111111	Admin
12	Librarian Two	lib2@library.com	01811111112	Admin
13	Jahid Hasan	jahid@gmail.com	01711111121	Student
14	Ritu Akter	ritu@gmail.com	01711111122	Student
15	Hasan Mahmud	hasan@gmail.com	01711111123	Student

23. **Loans that are not yet returned.**

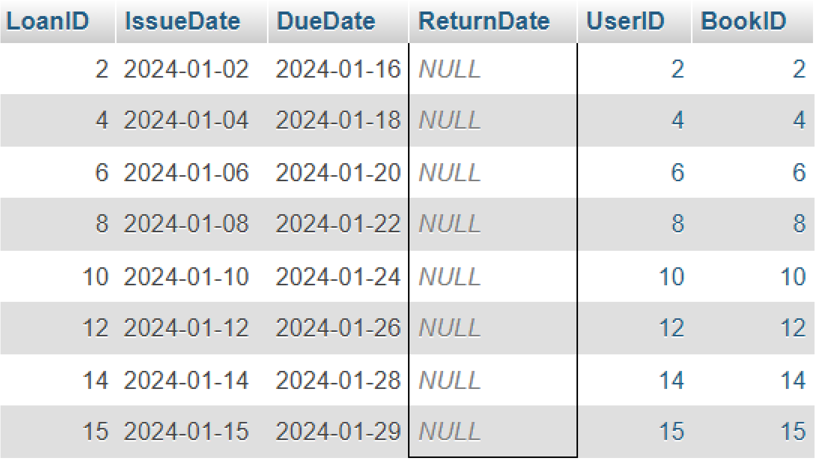
SQL Command:



Run SQL query/queries on table librarymanagementsystem.user:

```
SELECT * FROM loan
WHERE ReturnDate IS NULL;
```

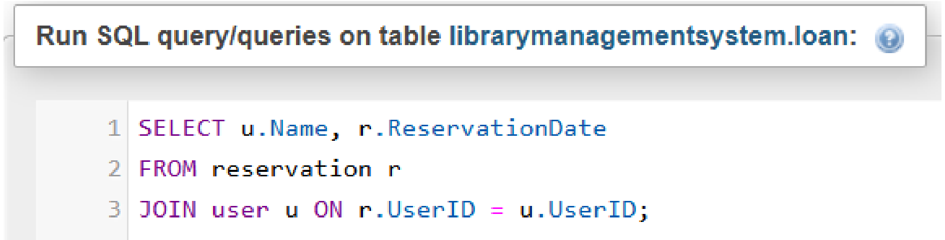
Output:



LoanID	IssueDate	DueDate	ReturnDate	UserID	BookID
2	2024-01-02	2024-01-16	NULL	2	2
4	2024-01-04	2024-01-18	NULL	4	4
6	2024-01-06	2024-01-20	NULL	6	6
8	2024-01-08	2024-01-22	NULL	8	8
10	2024-01-10	2024-01-24	NULL	10	10
12	2024-01-12	2024-01-26	NULL	12	12
14	2024-01-14	2024-01-28	NULL	14	14
15	2024-01-15	2024-01-29	NULL	15	15

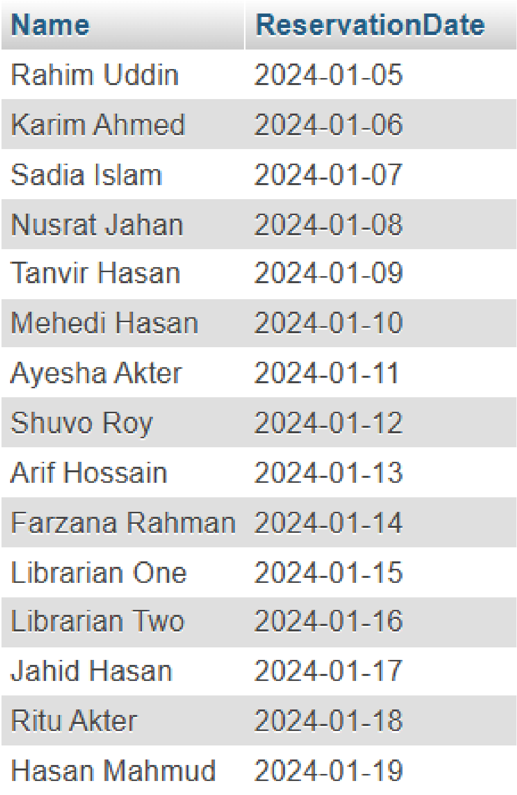
24. **Users with reserved books.**

SQL Command:


Run SQL query/queries on table librarymanagementsystem.loan:

```
SELECT u.Name, r.ReservationDate
FROM reservation r
JOIN user u ON r.UserID = u.UserID;
```

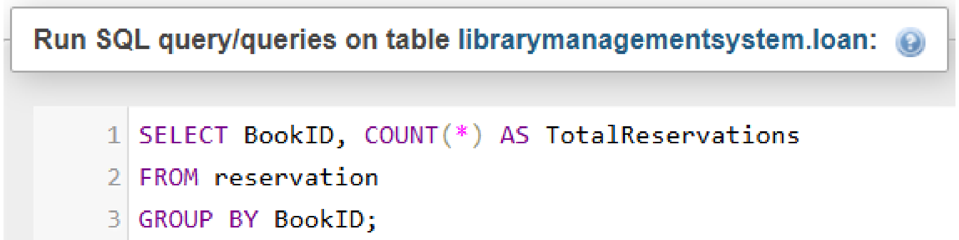
Output:



Name	ReservationDate
Rahim Uddin	2024-01-05
Karim Ahmed	2024-01-06
Sadia Islam	2024-01-07
Nusrat Jahan	2024-01-08
Tanvir Hasan	2024-01-09
Mehedi Hasan	2024-01-10
Ayesha Akter	2024-01-11
Shuvo Roy	2024-01-12
Arif Hossain	2024-01-13
Farzana Rahman	2024-01-14
Librarian One	2024-01-15
Librarian Two	2024-01-16
Jahid Hasan	2024-01-17
Ritu Akter	2024-01-18
Hasan Mahmud	2024-01-19

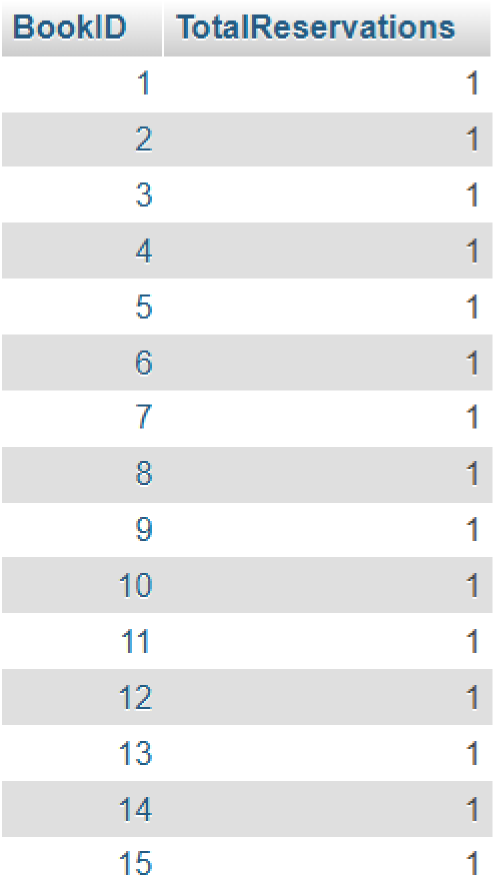
25. **Count total reservations per book.**

SQL Command:


Run SQL query/queries on table librarymanagementsystem.loan:

```
SELECT BookID, COUNT(*) AS TotalReservations
FROM reservation
GROUP BY BookID;
```

Output:



BookID	TotalReservations
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1

9. Conclusion and Future Advancement

This project provided practical experience in designing and implementing a relational database system using the Library Management System as a case study. Through this project, key database concepts such as ER modeling, normalization up to Third Normal Form (3NF), and SQL implementation were successfully applied. Challenges encountered during the project included correctly defining entity relationships and maintaining referential integrity across multiple tables.

Future enhancements of the system may include the integration of an online library portal, real-time book availability tracking, automated fine calculation for overdue books, and the development of mobile or web-based applications to improve user accessibility and system efficiency.

10. Contribution of Each Team Member

- Member 1: ER Diagram and Requirements Analysis
- Member 2: SQL Implementation and Queries
- Member 3: Database Schema and Normalization