

School of Computer Science and Engineering

DBA Open ended experiment project Report

*On*

***ONLINE LIBRARY MANAGEMENT SYSTEM***

**Team Members**

*USN ROLL NO NAME*

01FE20BCS019 119 Shreya Devagiri

01FE20BCS035 135 Sneha Pamali

01FE20BCS017 117 Ananya Patil

**Content**

**1 Introduction**

1.1 Online Library

* 1. Database Management System
  2. Problem Statement
  3. Objectives
  4. Motivation

1. **Requirement Collection and Analysis**
   1. Introduction
   2. Data requirement
   3. Functional Requirement

2.4 Non-Functional Requirement

1. **Database Design**
   1. Introduction

3.2 Conceptual Design

1. **Implementation and Results**
   1. Introduction
   2. Database Tables

4.3 Results

**Chapter 1: Introduction**

**1.1 Online Library**

Manual process of keeping student records, book records, account details, managing employee is very difficult. There are various problems also faced by the student in library such as finding any particular book, information whether book is available or not, for what time this book will be available, searching of books using ISBN number etc. To eliminate this manual system, library management system has been developed. Library Management System will handle all the current issues faced by the students and by its admin personnel.  To store all the information in the database from where user will place their query and get the results on the basis of their query. Only valid users will be able to access this Library Management System. Through this Library Management System, it will be easy to manage accounts and various details of particular student and employees working under library along with the records of book.  The current Library Management System does not eliminate the process of searching books within the library campus. Students have to find books manually. They have to wait until they are not provided with their library card and token. For receiving book, they have to show their library card and wait in line for their turns. The admin personnel also have to look manually on which day which person will take the charge within library to manage the overall work.

**1.2 Database Management System**

The database is a collection of inter-related data which is used to retrieve, insert, and delete the data efficiency. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

Database Management system is a software which is used to manage the database

For example: Oracle, MySQL etc. are very popular commercial database which is used in different applications.

DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.

It provides protection and security to the database. In the case of multiple users, it also maintains data consistency. The DBMS provides various functions that allow entry, storage and retrieval of large quantities of information and provides ways to manage how that information is organized.

DBMS allows the users to do following tasks:

**Data Definition:** It is used for creation, modification, and removal of definition that defines the organization of data in the database.

* **Update:** It is used for the insertion, modification, and deletion of the actual data in the database.
* **Data Retrieval:** It is used to retrieve the data from the database which can be used by applications for various purposes.
* **User Administration:** It is used for registering and monitoring users, maintain data integrity, enforcing data security, dealing with concurrency control, monitoring performance and recovering information corrupted by unexpected failure.

**Characteristics of DBMS**

* It uses a digital repository established on a server to store and manage the information.
* It can provide a clear and logical view of the process that manipulates data.
* DBMS contains automatic backup and recovery procedures.
* It contains ACID properties which maintain data in a healthy state in case of failure.
* It can reduce the complex relationship between data.
* It is used to support manipulation and processing of data.
* It is used to provide security of data.
* It can view the database from different viewpoints according to the requirements of the user.

**Advantages of DBMS**

* **Controls database redundancy:** It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.
* **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.
* **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.
* **Reduce time:** It reduces development time and maintenance need.
* **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.
* **multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces

**Relational Database Management System**

RDBMS [3] is a database management system (DBMS) based on the relational model of data. Most databases in widespread use today are based on this model. RDBMSs have been a common option for the storage of information in databases used for financial records, manufacturing and logistical information, personnel data, and other applications since the 1980s. Relational databases have often replaced legacy data models like hierarchical databases and network databases because they were easier to implement and administer. Nonetheless, relational databases received continued, unsuccessful challenges by object database management systems in the 1980s and 1990s, (which were introduced in an attempt to address the so-called object-relational impedance mismatch between relational databases and object-oriented application programs), as well as by XML database management systems in the 1990s.However, due to the expanse of technologies, such as horizontal scaling of computer clusters, NoSQL databases have recently become popular as an alternative to RDBMS databases.

**1.3 Problem Statement**

This project aims to design and implement the database for Online Library management system to maintain the following activities.

* Library has books which are uniquely identified by its book name and book code.
* Library keeps the record of students who access the books with student serial number, student full name.
* Library maintain the record of staff with their staff ID, name, salary, address, designation, working hours.
* Library books which are written by authors their data is also stored in database with author ID and author name.
* Library books can be accessed by faculty as well, so their data with faculty name, department, unique ID, address, phone numbers is stored

**1.4 Objectives of the project**

Following are the main objectives of the study of this project work

• To study and implement the basic database concepts.

• To understand the database design process.

• To Explore the activities of Online Library Management System for designing the database.

• To study and familiar about the Structured Query Languages

**1.5 Motivation**

**1. Increase efficiency**

A library management system enhances the efficiency of both the librarians and the library users. It also enables librarians to easily catalog books and keep proper records of books issued, reissued, and those not returned. On the other hand, for a library user, they can easily find the availability of any book that they need.

**2. Reduces the Library management cost**

With a library management system, you can easily eliminate the no of employees or staffs and keep different manual files without using papers, thereby reducing the cost. You can also store a lot of data in a single system so that you can reduce the manual files.

**3. Saves Time**

Time is precious for both the librarians and the users. By an integrated library management system, a librarian can easily record and go through the history of individual library users without wasting the time. On the other hand, the library users can easily find the location and the availability of each book faster than before. So, both the librarian and the users can accomplish more with less time.

##### **4. Increases the productivity of library workers**

Library automation will increase the productivity. With the growing ease of doing work, staffs can engage in their activities in the library without the library incurring any additional cost.

##### **5. Enhances reporting and monitoring**

Self-updating records with an automated library management system give rise to dynamic reporting and oversight capabilities. This will support efficient bookings, material circulation, and user tracking. Library Management System Software for Library Management allows management of materials within the system so that you can easily know what exactly is available, and what has been borrowed.

**Chapter 2: Requirement Collection and Analysis**

**2.1 Introduction**

The most critical aspect of specification is the gathering and compilation of system and user requirements. This process is normally done in conjunction with managers and users. The major goal in requirements gathering process is to:

• Collect the data used by the organization

• Identify relationships/conditions to be applied on the data

• Identify future data needs

• Determine how the data is used and generated.

• Identify the functions that are performed on the data

The starting place for data collection is gathering existing forms and reviewing policies and systems. Then, ask users what the data means, and determine their daily processes

Following subsections discuss the data requirements and functional and non-functional requirements identified based on the following activities collected from the library users.

* Library has books which are uniquely identified by its book name, book code, publisher name, rack number, price, subject code.
* Library keeps the record of students who access the more than one book with student serial number, student full name, address, phone number.
* Library maintain the record of staff with their staff ID, name, salary, address, designation, working hours.
* Many Library books which are written by many authors and their data is also stored in database with author ID and author name.
* Library books can be accessed by faculty as well, so their data with faculty name, department, unique ID, address, phone numbers is stored.
* Many staff manage the books according to rack number and subject code.
* Books are issued to students and faculty depending upon issued date, expiry date and fine charges if return of book exceeds the expiry date exceeds.
* Each staff can manage many students.

**2.2 Data Requirement**

Data requirement describes the data to be stored in the database pertaining to activities of the library requirement as described in section 2.1. Details of the data stored in the database is shown in the table 2.1 and Table 2.2.

Table 2.1: Data to be stored in database

|  |  |  |
| --- | --- | --- |
| SI  NO. | Group | Data related to each group |
| 1 | Book | Book name, Book code, Publisher name, rack number, price, subject code |
| 2 | Staff | Staff name, Staff ID, Address, Designation, Working hours, Salary. |
| 3. | Faculty | Faculty name, Faculty ID, Department, Phone number, Address. |
| 4 | Student | SSN, Student name, Phone number, Address. |
| 5 | Author | Author name, Author ID |

Table 2.2: Conditions on data

|  |  |
| --- | --- |
| SI NO. | CONDITIONS |
| 1 | Many books are managed by many staff members |
| 2 | Many books are written by many Authors |
| 3 | Each student and faculty can be issued with many books at a time |
| 4 | Each staff manages multiple students |
| 5 | If the return of book exceeds expiry date, then fine must be given. |

**2.3 Functional Requirement**

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it’s important to make them clear both for the development team and the stakeholders (clients). Table 2.3 shows the different types users of driving school database application and their respective responsibilities (tasks). Table 2.4 shows the different functions and user can perform on the database

Table 2.3: Categories of Users and their tasks

|  |  |  |
| --- | --- | --- |
| SI NO. | Users | Responsibility |
| 1 | Staff | Responsible for managing books and student |
| 2 | Student | Responsibility of return back the book |
| 3 | Author | Writing book |
| 4 | Faculty | Return back book within expiry date else need to pay fine |

|  |  |  |
| --- | --- | --- |
| SI NO. | Functions | User |
| 1 | Insert the records into the database | Library administrator, library head, Library assistant |
| 2 | Delete the records from database | Library administrator, library head, Library assistant |
| 3 | Issuing book with start and end date depending upon expiry date fine issued | Library staff |

Table 2.4: Functions of each user

**Functional requirements**

**Faculty:**

1)Retrieve the faculty name, department using set operator which belong to other department rather than ‘CS’.

ANS:-

select fname,dept,book\_name

from faculty,BOOK

where faculty.book\_code=book.book\_code

EXCEPT

select fname,dept,book\_name

from faculty,BOOK

where DEPT='CS' AND faculty.book\_code=book.book\_code;

2)Retrieve the faculty name, faculty id in descending order with ID greater than ‘155’.

ANS:-

select fname,fid

from faculty,book,managed\_by

where fid>'155' and faculty.book\_code=book.book\_code and book.book\_code=managed\_by.book\_code

order by fid desc;

3)Retrieve book name and faculty name who issued respective book whose price is less than the average price of all other books.

ANS:-

select book\_name,fname

from book,faculty

where faculty.book\_code=book.book\_code and price< (select avg(price)

from faculty,BOOK

where faculty.book\_code=book.book\_code);

4)Retrieve the author name who had written the costliest book.

ANS:-

select author\_name

from AUTHOR

where author\_id=(select author\_id

from written

where book\_code=(select book\_code

from BOOK

where price=(select max(price)

from book)));

5)Retrieve faculty name issued with a book its name and fine associated due to late return

6)Retrieve faculty id and department number issued with book along with issued date and expiry date.

**Student**

1)Retrieve the student’s name and ssn issued with a book which is in rack number ‘02’.

ANS: -

select sname,ename,ssn,book\_name

from student,book

where rack\_no='2' and student.book\_code=book.book\_code;

2)Retrieve the student names and book code along with book name with start date and end date of book being issued.

ANS: -

select sname, ename, ssn, book\_name, start\_date\_, end\_date

from student,BOOK

where student.book\_code=book.book\_code;

3)Retrieve the student ssn and name issued with book which is written by author ‘Donald Neamen’

ANS: -

select sname,ename,ssn,book\_name

from student,book,written,AUTHOR

where author\_name='Donald Neamen' and student.book\_code=book.book\_code and book.book\_code=written.book\_code and written.author\_id=author.author\_id;

4)Retrieve the student names and price of each book one is issued with in descending order.

ANS: -

select sname,ename,book\_name,price

from book,STUDENT

where student.book\_code=book.book\_code

order by price desc;

5)Retrieve the student’s name, staff id, student ssn issued with book which is managed by staff in rack no ‘851’

ANS: -

select sname,staff\_id,ssn,book\_name

from student,book,managed\_by

where rack='851' and student.book\_code=book.book\_code and book.book\_code=managed\_by.book\_code;

6)Retrieve the student names, ssn, phone number who are issued with book along with subject code.

**BOOK AND MANAGED\_BY**

1. Retrieve the information of the book, which is managed by the staff named ‘Manish Ashok’.

ANS: -

select \*

from BOOK

where book\_code in (select book\_code

from managed\_by

where staff\_id in (select staff\_id

from staff

where full\_name='Manish Ashok'));

1. Retrieve the salary if the staff who manages the book, whose expiry date is ’25-AUG-2020’

ANS: -

select salary

from STAFF

where staff\_id in (select staff\_id

from managed\_by

where book\_code in (select book\_code

from BOOK

where expiry\_date='25-AUG-2020'));

1. Retrieve the author’s name and author’s id whose book code is ‘101’

ANS: -

select a.author\_id , author\_name

from author a , managed\_by m , written w

where a.author\_id=w.author\_id

and w.book\_code=m.book\_code

and m.book\_code='101';

1. Retrieve the staff if who manages the book having maximum penalty fee

ANS: -

select m.staff\_id

from managed\_by m ,BOOK b

where m.book\_code=b.book\_code

and b.penalty\_fee in (select max(penalty\_fee)

from BOOK);

1. Create the view table of books whose penalty fee is greater than Rs 12 and it’s price is more than Rs 500

**WRITTEN**

1. Retrieve the book-name, publisher-name and author-id which is present in rack number 6.

ANS: -

SELECT book\_name,publisher\_name,author\_id

from written,book

where written.book\_code=book.book\_code and rack\_no=6;

1. Retrieve the subject code, book name with its author name and price written by David C Lay.

ANS: -

SELECT subject\_code,book\_name, price

from book, author, written

where publisher\_name='David C Lay'and AUTHOR.author\_id=written.author\_id and book.book\_code= written.book\_code;

1. Retrieve all the author-name whose subject code is 510.  
   ANS: -

SELECT author\_name

from author,written,book

where subject\_code=510 and written.book\_code=book.book\_code AND author.author\_id=written.author\_id ;

**STAFF**

1. Retrieve all the staff names and student full names who stay in 'Vidyagiri'.

ANS: -

SELECT sname,ename,full\_name

from staff, student

where staff.ssn=student.ssn and staff.address='Vidyagiri' and student.address='Vidyagiri';

1. Retrieve all the student names and staff names who issued the book COA and also display its rack number.

ANS: -

SELECT sname as STUDENT\_NAME, full\_name as STAFF\_ISSUED,book.rack\_no

from student,staff,book

where student.ssn=staff.ssn and book.book\_code=student.book\_code and book.book\_name='COA';

1. Retrieve all the staff names who are Database manager and salary is more than 45000 Rs/- and Clerk whose salary is more than equal to 10,000 Rs/-.

ANS: -

SELECT staff\_id,full\_name

from STAFF

WHERE STAFF.designation='Database manager' AND STAFF.salary>45000

UNION

SELECT staff\_id,full\_name

from STAFF

WHERE STAFF.designation='Clerk'and STAFF.salary>=10000;

1. Retrieve all the book names with author-name with subject number managed by staff and the rack it belongs where penalty-fee paid by the student is greater than or equal to 15.

ANS: -

SELECT book.book\_name,book.publisher\_name,full\_name as STAFF\_MANAGER,book.subject\_code,managed\_by.rack

from book,staff,managed\_by

where managed\_by.staff\_id=staff.staff\_id and managed\_by.book\_code=book.book\_code and penalty\_fee>=15;

**2.4 Non-Functional Requirement**

Non-functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. In this project we are addressing the security in order to secure the data and accessibility.

**Chapter 3: Database Design**

##### **3.1. Introduction**

The requirements gathering and specification provides you with a high-level understanding of the organization, its data, and the processes that you must model in the database. Database design involves constructing a suitable model of for the information. Since the design process is complicated, especially for large databases, database design is divided into three phases:

* Conceptual database design
* Logical database design
* Physical database design.

In our project work we are addressing the conceptual database design using ER modelling and logical database design using the implementation data model called Relational model.

###### **3.2. Conceptual Database Design**

Conceptual database design involves modelling the collected information at a high-level of abstraction without using a particular data model or DBMS. This model allows for easy communication between end-users and database developers and has a clear method to convert from high-level model to relational model. The most popular model for conceptual database design is the Entity Relationship model which describes data as attribute, entity and relationship.

Table 3.1 shows the list of attributes Table 3.2 shows the list of entity types and table 3.3 shows the list of relationship types identified for the requirement discussed in the section 2.1.

**Table 3.1: List of attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.  No. | Attribute Name | Attribute Type | Entity Type |
| 1 | Student ssn | Simple | Student |
| 2 | Start name | Simple | Student |
| 3 | End name | Simple | Student |
| 4 | Address | simple | Student |
| 5 | Phone no | simple | Student |
| 6 | Faculty Id | simple | Faculty |
| 7 | First name | simple | Faculty |
| 8 | Last name | simple | Faculty |
| 9 | Phone no | simple | Faculty |
| 10 | Department | simple | Faculty |
| 11 | Address | simple | Faculty |
| 12 | Staff id | Simple | Staff |
| 13 | Full name | Simple | Staff |
| 14 | Designation | Simple | Staff |
| 15 | Working hours | Simple | Staff |
| 16 | Salary | Simple | Staff |
| 17 | Address | Simple | Staff |
| 18 | Book code | Simple | Book |
| 19 | Book name | Simple | Book |
| 20 | Subject code | Simple | Book |
| 21 | rack no | Simple | Book |
| 22 | Publisher name | Simple | Book |
| 23 | Price | Simple | Book |
| 24 | Author id | Simple | Author |
| 25 | Author name | Simple | Author |
| 26 | Rack no | Simple | Relationship type |
| 27 | Subject no | Simple | Relationship type |
| 28 | Start date | Simple | Relationship type |
| 29 | End date | Simple | Relationship type |
| 30 | Penalty fee | Simple | Relationship type |
| 31 | Issued date | Simple | Relationship type |
| 32 | Expiry date | Simple | Relationship type |
| 33 | fine | Simple | Relationship type |

**Table 3.2: List of Entity Types**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl. No. | Entity Type Name | Type of Entity Type | Justification |
| 1 | Book | Strong | It has key attribute |
| 2 | Staff | Strong | It has key attribute |
| 3 | Student | Strong | It has key attribute |
| 4 | Faculty | Strong | It has key attribute |
| 5 | Author | Strong | It has key attribute |

**Table 3.3: List of Relationship Types**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sl.  No. | Relationship Type  Name | Type of  Relationship  Type | Justification | Participating  Entity Type with cardinality ratio | Participation |
| 1 | Managed by | Simple |  | Book | Total |
| Staff | Total |
| 2 | Written | Simple |  | Author | Total |
| Book | Total |
| 3 | issued to faculty | Simple |  | Book | Total |
| Faculty | Total |
| 4 | issued to student | Simple |  | Book | Total |
| Student | Total |
| 5 | manages | Simple |  | Staff | Total |
| Student | Total |

**3.2.1 E-R Diagram, Schema Diagram and Normalization**

Entity relationship diagram of the proposed system as described in the requirement analysis is shown in the figure 3.1and figure 3.2 shows the schema diagram obtained after converting ER diagram to relational model.

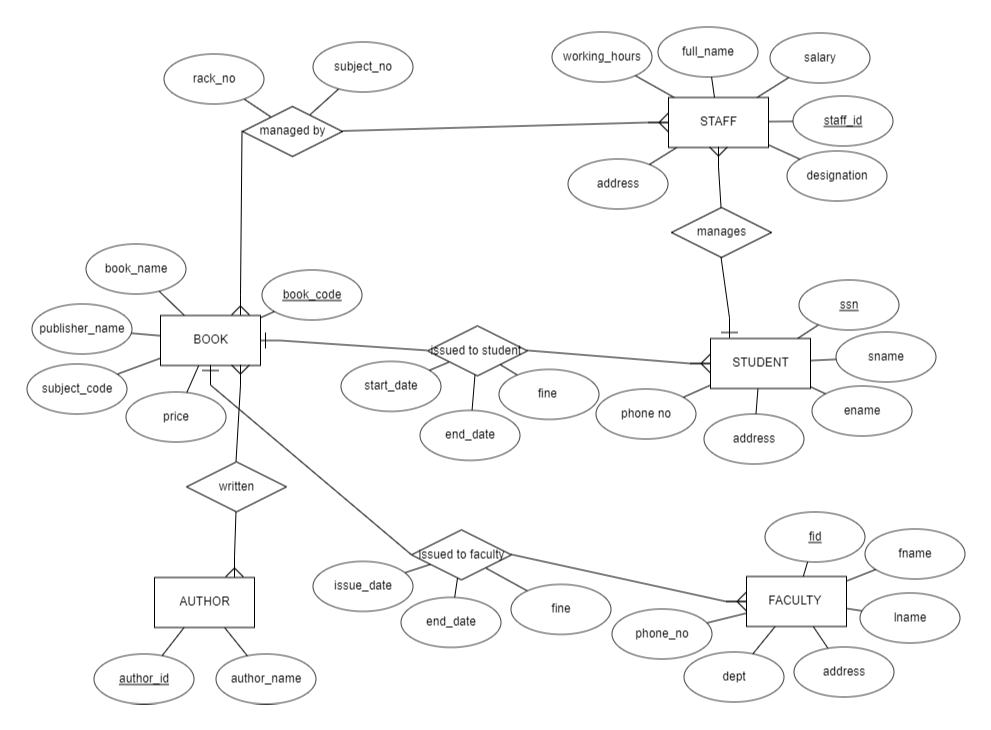


FIG 3.1:ER DIAGRAM

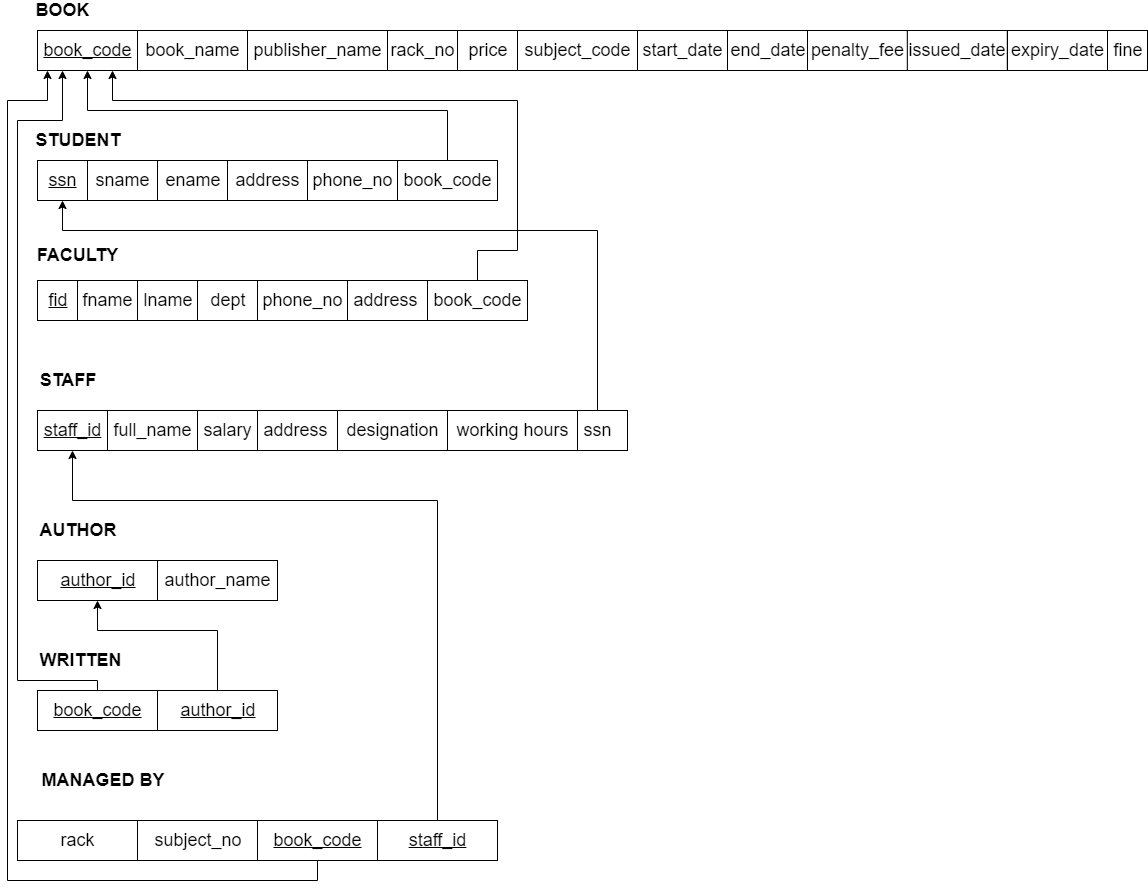


FIG 3.2: SCHEMA DIAGRAM OF THE FIG 3.1

1.BOOK

(book\_code, book\_name, publisher\_name, subject\_code, rack\_no, price, issued\_date, expiry\_date, start\_date, end\_date, penalty\_fee, fine)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

2.STUDENT

( ssn, sname, ename, address, phone\_no, book\_code)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

3. FACULTY

( fid, fname, lname, address, dept, phone\_no, book\_code)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

4.STAFF

( staff\_id , ssn, full\_name, salary, working\_hours, designation,address)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

5.AUTHOR

(author\_id, author\_name)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

6.WRITTEN

(book\_code, author\_id)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

7. MANAGED BY

( book\_code, staff\_id, subject\_no, rack)

• The Relation is in 1NF as it has atomic valued attributes

• The Relation is in 2NF since, every attribute is fully functionally dependent on the key.

• We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

**Chapter 4: Implementation and Results**

**4.1 Introduction**

Implementation involves the construction of a database according to the specification of a logical schema. This will include the specification of an appropriate storage schema, security enforcement, external schema and so on. Implementation is influenced by the choice of available DBMSs, database tools and operating environment. There are additional tasks beyond simply creating a database schema and implementing the constraints such as data must be entered into the tables, issues relating to the users and user processes need to be addressed, and the management activities associated with wider aspects of corporate data management need to be supported. In practice, implementation of the logical schema in a given DBMS requires a very detailed knowledge of the specific features and facilities that the DBMS has to offer. In an ideal world, and in keeping with good software engineering practice, the first stage of implementation would involve matching the design requirements with the best available implementing tools and then using those tools for the implementation. In database terms, this might involve choosing vendor products with DBMS and SQL variants most suited to the database which is to be implemented. There are many relational DBMSs, available such as Oracle Database, Microsoft SQL Server, MySQL, IBM DB2, IBM Informix and Microsoft Access, use SQL. In this project we used Oracle SQL developer create the following tables of Online library management database.

**4.2 Database Tables**

Following tables table 4.1 to table 4.6 are the tables created for the schema diagram shown in figure 3.2.

Table 4.1: Book Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| book\_code | Integer | Primary Key |
| book\_name | Varchar(10) | Not Null |
| publisher\_name | Varchar(10) | Not Null |
| subject\_code | Integer | Not Null |
| rack\_no | Integer | Not Null |
| price | Integer | Not Null |
| issued\_date | Date | Not Null |
| expiry\_date | Date | Not Null |
| start\_date | Date | Not Null |
| end\_date | Date | Not Null |
| penalty\_fee | Integer | Not Null |
| fine | Integer | Not Null |

Table 4.2: Student Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| ssn | Integer | Primary Key |
| sname | Varchar(10) | Not Null |
| ename | Varchar(10) | Not Null |
| address | Varchar(10) | Not Null |
| phone\_no | Integer | Not Null |
| book\_code | Integer | Foreign key reference to Book |

Table 4.3: Faculty Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| fid | Integer | Primary Key |
| fname | Varchar(10) | Not Null |
| lname | Varchar(10) | Not Null |
| dept | Varchar(10) | Not Null |
| phone\_no | Integer | Not Null |
| address | Varchar(10) | Not Null |
| book\_code | Integer | Foreign key reference to Book |

Table 4.4: Staff Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| staff\_id | Integer | Primary Key |
| full\_name | Varchar(20) | Not Null |
| Salary | Integer | Not Null |
| address | Varchar(10) | Not Null |
| designation | Varchar(10) | Not Null |
| working\_hours | Integer | Not Null |
| ssn | Integer | Foreign key reference to Student |

Table 4.5: Author Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| author\_id | Integer | Primary Key |
| author\_name | Varchar(10) | Not Null |

Table 4.6: Written Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| book\_code | Integer | Primary Key as well as  Foreign key reference to Book |
| author\_id | Integer | Primary Key as well as  Foreign key reference to Author |

Table 4.7: Managed By Table Description

|  |  |  |
| --- | --- | --- |
| Attribute | Type | Constraints |
| rack | Integer | Not Null |
| subject\_no | Integer | Not Null |
| book\_code | Integer | Primary Key as well as  Foreign key reference to Book |
| author\_id | Integer | Primary Key as well as  Foreign key reference to Author |

Following syntax shows for creating database table shown with an example for the Table 4.1

**Database Creation:**

CREATE TABLE BOOK

(

book\_code INT NOT NULL,

book\_name VARCHAR(10),

publisher\_name VARCHAR(10),

subject\_code INT NOT NULL,

rack\_no INT NOT NULL,

price INT NOT NULL,

issued\_date DATE NOT NULL,

expiry\_date DATE NOT NULL,

start\_date DATE NOT NULL,

end\_date DATE NOT NULL,

penalty\_fee INT,

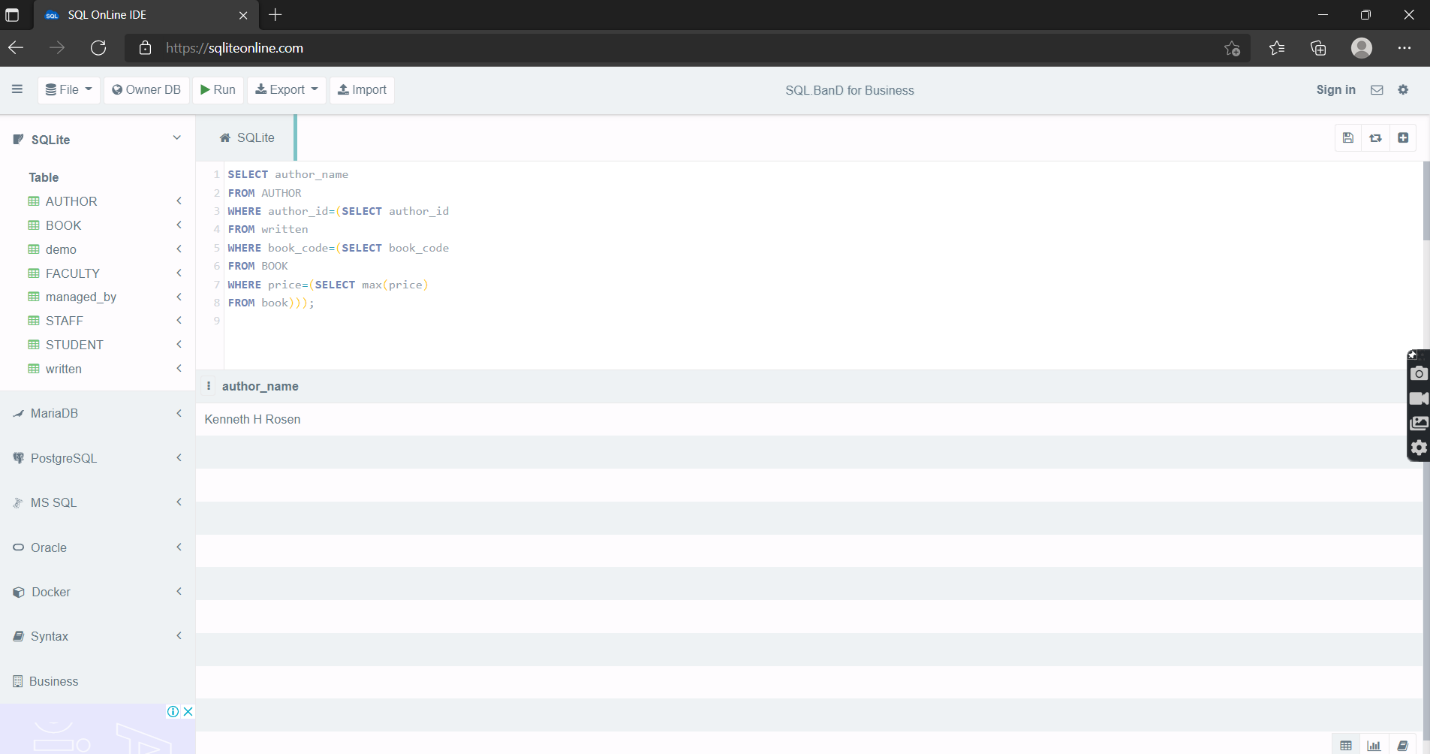
fine INT,

PRIMARY KEY (book\_code)

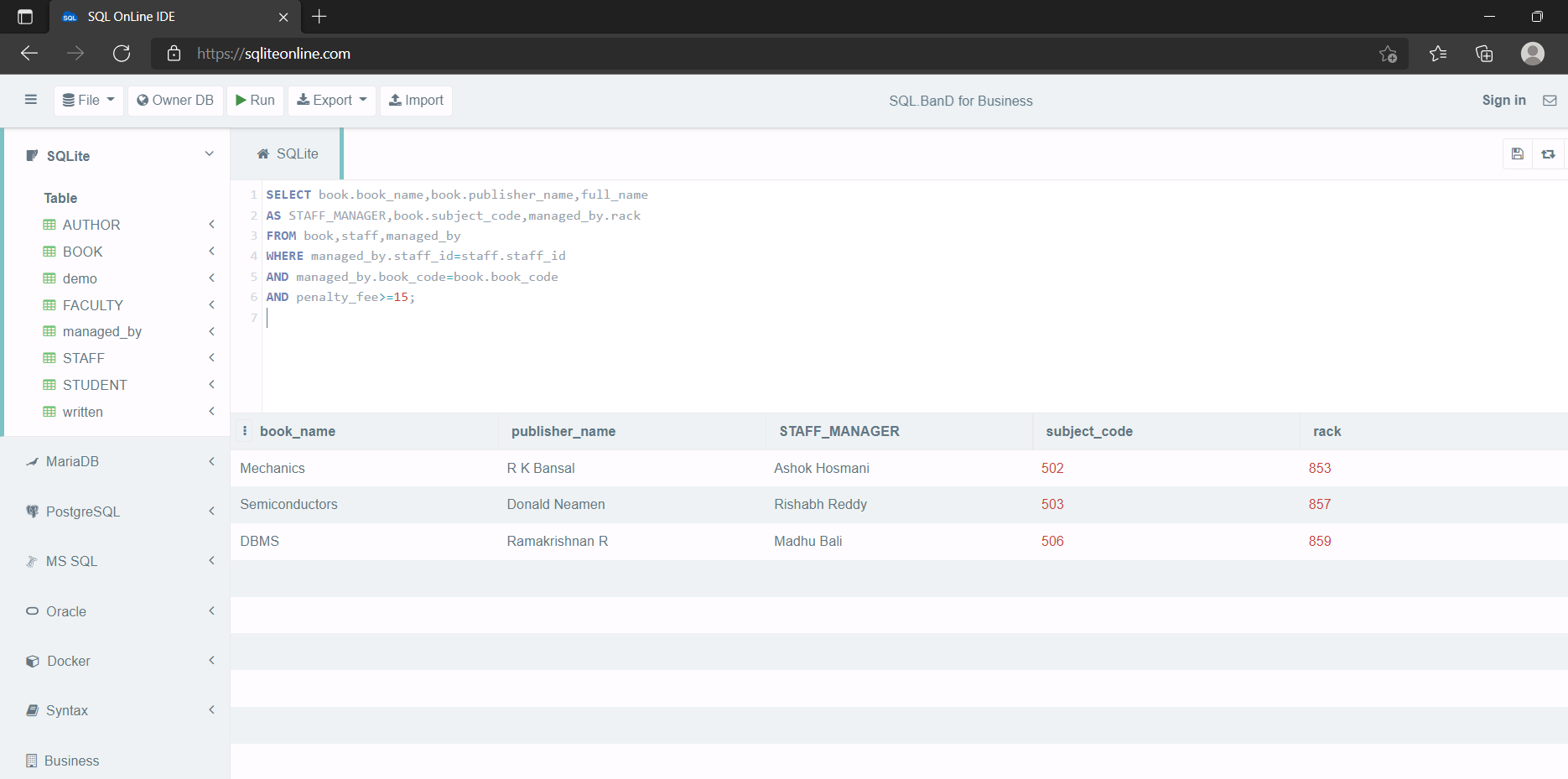
);

**4.3. Results**

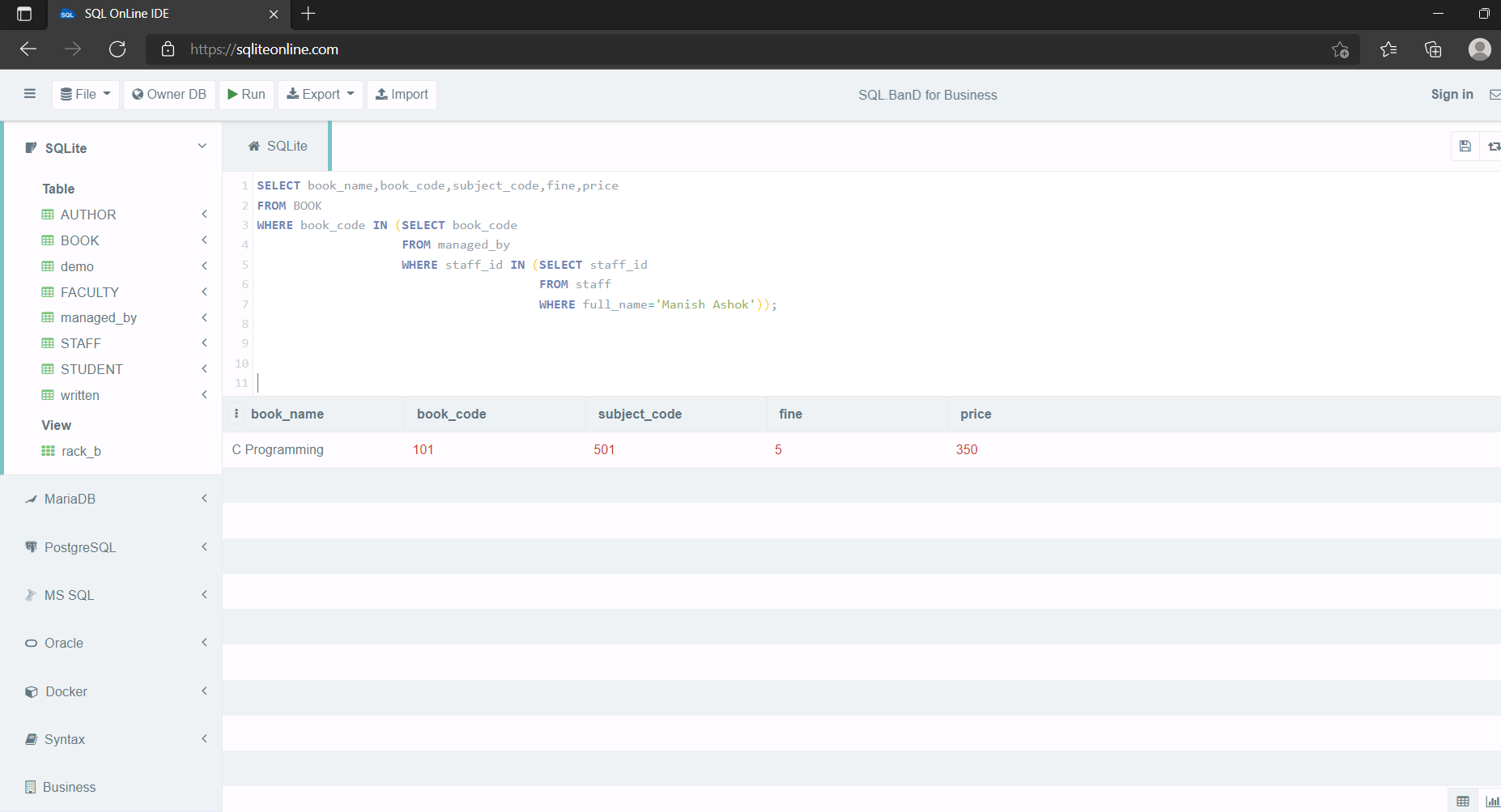
1) Retrieve the author’s name who had written the costliest book.



2) Retrieve all the book names with author-name with subject number managed by staff and the rack it belongs where penalty-fee paid by the student is greater than or equal to 15.



3) Retrieve the book name, book code, price, subject code,fine which is managed by the staff named ‘Manish Ashok'



**References: -**

[1] https://github.com/vinitshahdeo/Library-Management-System

[2] [SQL OnLine IDE (sqliteonline.com)](https://sqliteonline.com/) (https://sqliteonline.com/)

[3] Ramez Elmasri, Shamkant B. Navathe “Fundamentals of Database Systems, 7th Edition “ pearson, 2016

[4] Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems” 3rd Edition, McGraw Hill Higher Education, 2003