A Project Report
On

Resources Database Management System

Course: Database Management Systems (UNC502)

Submitted by:

Shivankar Jamwal (102286006)
Parth Badhwar (102106073)
Tushar Dhingra (102156007)

Group: 3EC3

BE Third Year, Electronics and Communication Engineering

Submitted To:

Dr. Rakesh Kumar

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Department of Electronics and Communication Thapar Institute of Engineering & Technology,(Deemed To Be University), Patiala, Punjab

(JAN-JUNE 2024)

Table of Contents

1. Abstract

In the contemporary landscape of information management, Library Resource Database Management Systems (LRDBMS) emerge as indispensable tools for the effective organization, retrieval, and maintenance of vast repositories of resources within libraries, academic institutions, and various organizational settings. This report provides a comprehensive exploration of LRDBMS, elucidating its fundamental components, functionalities, and overarching significance. By integrating advanced database management techniques, LRDBMS facilitates streamlined cataloging, classification, and indexing of diverse resources, thereby enhancing user accessibility and promoting efficient utilization of information assets.

Furthermore, this abstract delves into the multifaceted benefits of LRDBMS, ranging from improved user experience to enhanced administrative efficiency. Through robust search functionalities and intuitive interfaces, LRDBMS empowers users to navigate through extensive resource collections with ease, fostering a conducive environment for research, learning, and collaboration. Additionally, LRDBMS automates administrative tasks such as inventory management, tracking, and resource acquisition, allowing library staff to allocate their time and resources more effectively towards value-added services and patron support.

Moreover, this abstract highlights the evolving technological landscape surrounding LRDBMS, including advancements in data analytics, artificial intelligence, and cloud computing. These innovations promise to further augment the capabilities of LRDBMS, enabling personalized recommendations, predictive analytics, and seamless integration with emerging information technologies. However, alongside these opportunities, LRDBMS also faces challenges such as data privacy concerns, interoperability issues, and evolving user expectations, necessitating continuous innovation and adaptation to meet evolving needs.

In conclusion, this report aims to provide a comprehensive understanding of LRDBMS and its pivotal role in modern information management. By elucidating its functionalities, benefits, and emerging trends, this abstract seeks to equip stakeholders with the knowledge and insights necessary to harness the full potential of LRDBMS in optimizing resource management, fostering knowledge dissemination, and advancing scholarly pursuits in the digital age.

2. Introduction

*"**Efficiently Organizing Knowledge**: An Overview of Resources Database Management Systems in Library Settings"*

Problem statement

Libraries face the challenge of managing vast collections of resources efficiently while ensuring accurate organization and seamless accessibility. Traditional methods of cataloging and tracking resources are labor-intensive and error-prone, making it difficult to cope with the growing volume and complexity of library holdings. This necessitates the implementation of a robust Resources Database Management System (DBMS) tailored to the unique needs of libraries. Such a system would streamline library operations, improve resource discovery, and enhance the overall user experience.

Objective

This report aims to provide a comprehensive overview of Resources DBMS in library settings, focusing on its significance, functionalities, and benefits. Key objectives include:

- Understanding the role of Resources DBMS in addressing the challenges associated with resource organization, cataloging, and retrieval within libraries.
- Exploring the features and capabilities of Resources DBMS in facilitating efficient library operations, including acquisition, circulation, and user interaction tracking.
- Analyzing the impact of Resources DBMS on enhancing user experience, promoting collaboration, and facilitating knowledge dissemination within library environments.
- Assessing the challenges and considerations involved in the implementation of Resources DBMS in library settings, including data privacy, interoperability, and user training.

Need

- **Efficient Organization:** A Library DBMS facilitates the efficient organization of vast amounts of information including books, journals, magazines, multimedia materials, and more. It enables librariesto categorize, classify, and manage these resources systematically.

- **Accessibility:** With a DBMS, library patrons can easily access the library's resources through an onlinecatalog or search interface. This accessibility enhances user experience by allowing them to find relevant materials quickly and conveniently, regardless of their physical location.

- **Resource Management:** A Library DBMS assists librarians in effectively managing their resources, including tracking circulation, managing subscriptions, and handling acquisitions. This helps ensure that resources are utilized optimally and efficiently.

- **Automation of Tasks:** By automating various administrative tasks such as cataloging, inventory management, and circulation tracking, a DBMS reduces the manual workload on library staff. This freesup their time to focus on providing better services to patrons.

- **Data Integrity and Security:** A DBMS maintains data integrity by ensuring that library records are accurate, consistent, and up-to-date. It also offers security features to protect sensitive patron information and prevent unauthorized access to library resources.

- **Analytics and Reporting:** Library DBMS often come with built-in analytical tools that allow librarians to analyze usage patterns, track trends, and generate reports. This data-driven approach helps libraries make informed decisions about resource allocation and collection development.

- **Adaptability and Scalability:** As libraries evolve and expand their collections and services, a flexibleand scalable DBMS can adapt to changing needs and accommodate growth without significant disruptions.

- **Compliance and Standards:** A Library DBMS helps libraries comply with industry standards and best practices for cataloging, metadata management, and data interoperability, ensuring compatibilitywith other library systems and databases.

Overall, a robust library DBMS plays a pivotal role in modernizing library services, fostering collaboration, and enriching the user experience in an increasingly digital age.

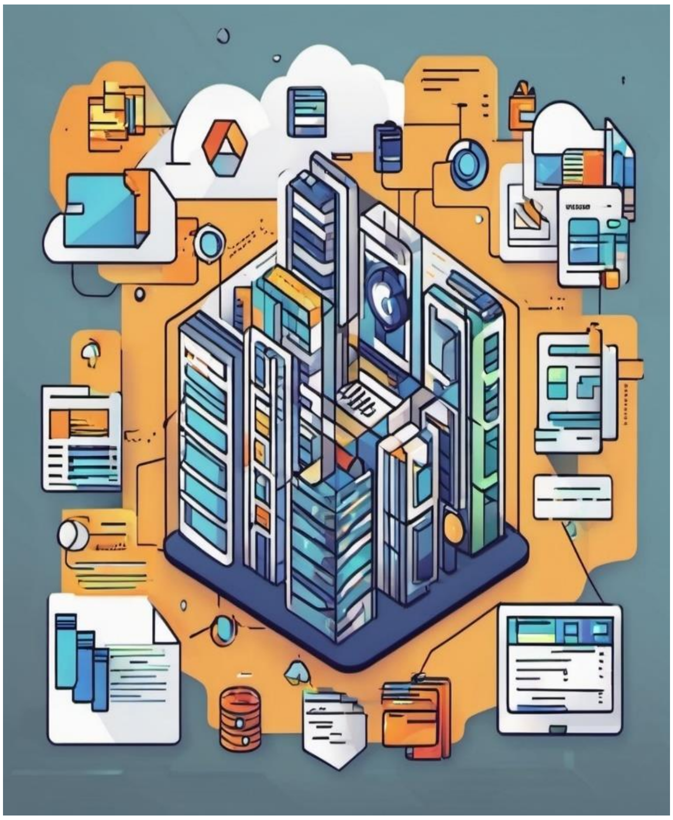


Fig 1. Resources database management system

3. Entities

- **What are Entities ?**

Entities are the real-world objects stored in a database, represented as tables. Attributes are the properties of these entities, describing their characteristics.

Together, entities and attributes structure the data in a database, organizing it into meaningful units for efficient storage and retrieval.

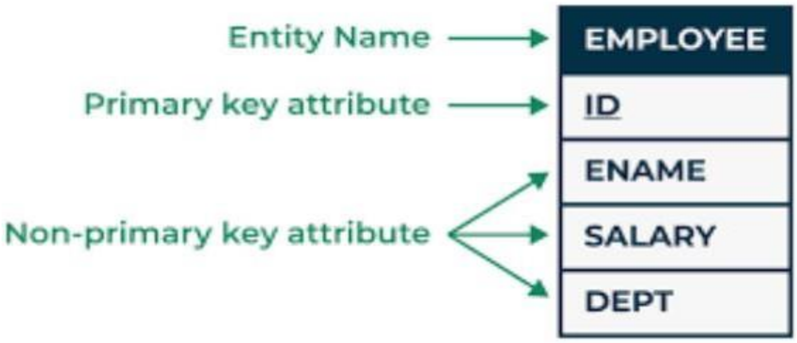


Fig 2. Entity with its attributes

- **Entities with their respective attributes**

Here in our topic, we have four entities : Student, Lib, Subscription, Book and their respective attributes mentioned below.

Student :

Roll No	Name	M_No	Issued_books	Fine

Lib:

ISBN	Copies	Delay_cost	Lost_Cost	Book_Name	Author	Publisher

Subscription:

Book_Id	Roll_No	Issue_Date	Return_Date	Actual_Return_Date

Book :

Book_Id	Availability	ISBN

4. ER Diagram

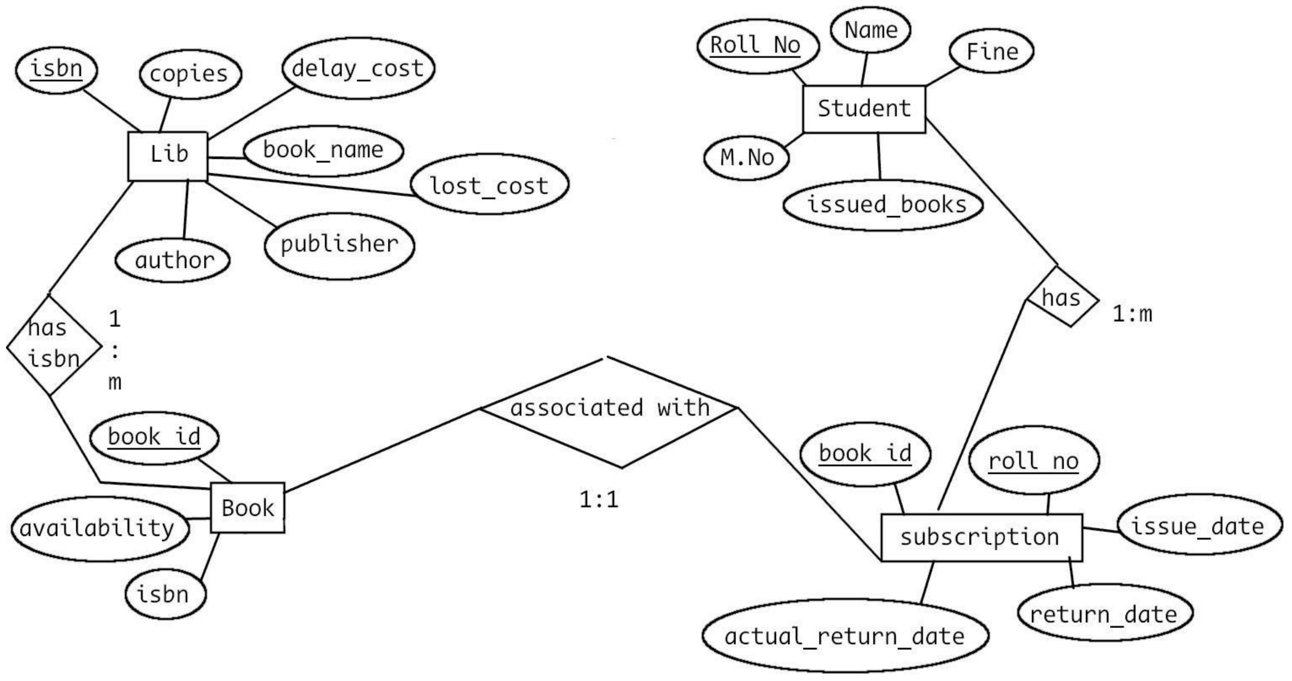


Fig 3. ER diagram

5. Relationships in ER diagram

ER diagram contains following relationships:

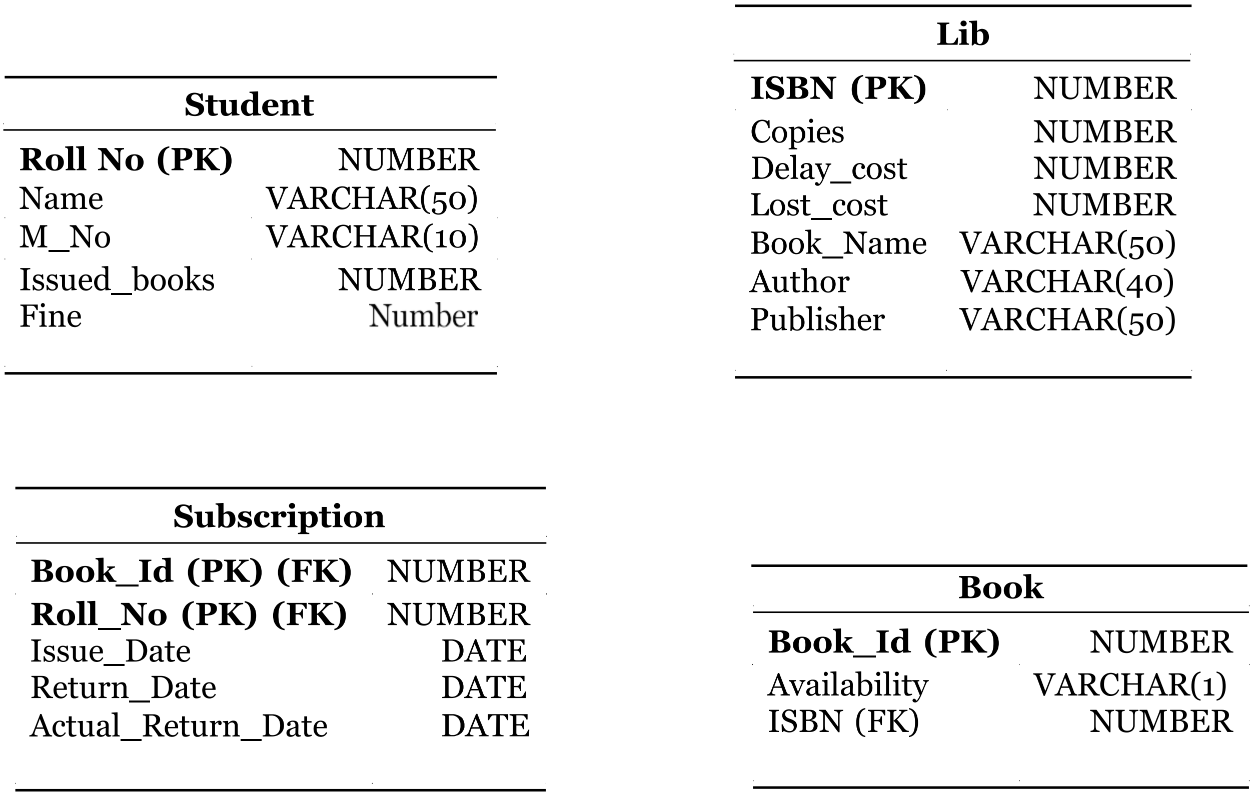
ENTITY 1	NAME OF RELATIONSHIP	ENTITY 2	CARDINALITY
Lib	has_isbn	Book	1:m
Book	associated_with	Subscription	1:1
Student	has	Subscription	1:m

Table 1. Relationship table

Type of the binary relationship	Relationships in the system
one-to-one	1) Each Subscription leads to borrow ofonly one book
one-to-many	1) A library has many books 2) A student can have multiple subscriptions.

Table 2. Relationship type table

6. ER Diagram to Tables



Student	
Roll No (PK)	NUMBER
Name	VARCHAR(50)
M_No	VARCHAR(10)
Issued_books	NUMBER
Fine	Number

Lib	
ISBN (PK)	NUMBER
Copies	NUMBER
Delay_cost	NUMBER
Lost_cost	NUMBER
Book_Name	VARCHAR(50)
Author	VARCHAR(40)
Publisher	VARCHAR(50)

Subscription	
Book_Id (PK) (FK)	NUMBER
Roll_No (PK) (FK)	NUMBER
Issue_Date	DATE
Return_Date	DATE
Actual_Return_Date	DATE

Book	
Book_Id (PK)	NUMBER
Availability	VARCHAR(1)
ISBN (FK)	NUMBER

Table 3. Conversion from ER diagram to tables

7. Normalization

Normalization is a process used in database design to organize data in a way that reduces redundancy and dependency, leading to a more efficient and reliable database structure. It involves breaking down large tables into smaller, more manageable ones and establishing relationships between them to minimize data duplication and improve data integrity. Essentially, normalization ensures that each piece of data is stored in only one place, reducing the risk of inconsistencies and anomalies within the database.

Table: Student

- **1st Normal Form:** There is no multi-valued attribute in the table (in this project, we are considering only one mobile number per student), so it is in 1st Normal form.
- **2nd Normal Form:** There is no partial dependency in the table as all the fields are dependent only on Roll No of student. Hence, the table is in 2nd Normal form.
- **3rd Normal Form:** Since there is no transitive dependency in the table (all fields are dependent only the primary key), the table is in 3rd Normal Form.
- **Boyce-Codd Normal Form:** Since every field of the table is dependent only on primary key,it is in BCNF.

Table: Lib

- **1st Normal Form:** There is no multi-valued attribute in the table, so it is in1st Normal form.
- **2nd Normal Form:** There is no partial dependency in the table as all the fields are dependent only on ISBN number. Hence, the table is in 2nd Normal form.
- **3rd Normal Form:** Since there is no transitive dependency in the table (all fields are dependent only the primary key), the table is in 3rd Normal Form.
- **Boyce-Codd Normal Form:** Since every field of the table is dependent only on primary key,it is in BCNF.

Table: Book

- **1st Normal Form:** There is no multi-valued attribute in the table, so it is in1st Normal form.
- **2nd Normal Form:** There is no partial dependency in the table as all the fields are dependent only on Book ID of a book. Hence, the table is in 2nd Normal form.
- **3rd Normal Form:** Since there is no transitive dependency in the table (all fields are dependent only the primary key), the table is in 3rd Normal Form.
- **Boyce-Codd Normal Form:** Since every field of the table is dependent only on primary key,it is in BCNF.

Table: Subscription

- **1st Normal Form:** There is no multi-valued attribute in the table, so it is in1st Normal form.
- **2nd Normal Form:** There is no partial dependency in the table as all the fields are dependent on the Book ID (A single book as a single subscription associated with it). Hence, the table is in 2nd Normal form.
- **3rd Normal Form:** Since there is no transitive dependency in the table (all fields are dependent only the primary key), the table is in 3rd Normal Form.
- **Boyce-Codd Normal Form:** Since every field of the table is dependent only on primary key,it is in BCNF.

8. Implementation

- **Table Creation**

```
CREATE TABLE
 student (
   rollno number PRIMARY KEY,
   name varchar(40),
   m_no varchar(10),
   fine number,
   issued_books number CHECK (issued_books <= 10)
 );

CREATE TABLE
 lib (
   isbn number PRIMARY KEY,
   bookname  varchar(50),
   author varchar(40),
   publication varchar(20),
   copies number,
   lost_cost number,
   delay_cost number
 );

CREATE TABLE
 book (
   bookid number GENERATED BY DEFAULT ON NULL AS
   IDENTITY PRIMARY KEY,
   isbn number,
   availability varchar(1) CHECK (
     (availability = 'A')
     OR (availability = 'O')
   )
 );
```

```
CREATE TABLE
  subscription (
    bookid number,
    rollno number,
    issue_date date,
    return_date date,
    actual_return_date date,
    PRIMARY KEY (bookid, rollno)
  );
```

- **Addition of foreign keys**

```
ALTER TABLE book ADD CONSTRAINT book_fk FOREIGN KEY (isbn)
REFERENCES lib (isbn);

ALTER TABLE subscription ADD CONSTRAINT
subscription_fk_roll FOREIGN KEY (rollno)
REFERENCES student (rollno);

ALTER TABLE subscription ADD CONSTRAINT
subscription_fk_book FOREIGN KEY (bookid)
REFERENCES book (bookid);
```

- **Insertion of data**

-- 1.student

```
INSERT INTO student VALUES (1, 'ALFRED', 623623623, 0, 0);
INSERT INTO student VALUES (2, 'JAMES', 659659659,0, 0);
INSERT INTO student VALUES (3, 'GEORGE', 654654654, 0, 0);
INSERT INTO student VALUES (4, 'TOM', 658658658,0, 0);
INSERT INTO student VALUES (5, 'PETER', 652652652,0, 0);
INSERT INTO student VALUES (6, 'JENNY', 651651651,0, 0);
INSERT INTO student VALUES (7, 'ROSE', 657657657,0, 0);
INSERT INTO student VALUES (8, 'MONICA', 639639639, 0, 0);
INSERT INTO student VALUES (9, 'PHOEBE', 678678678, 0, 0);
```

INSERT INTO student VALUES (10, 'RACHEL', 687687687,0, 0);
INSERT INTO student VALUES (11, 'ISHAAN', 9237463826,0, 0);

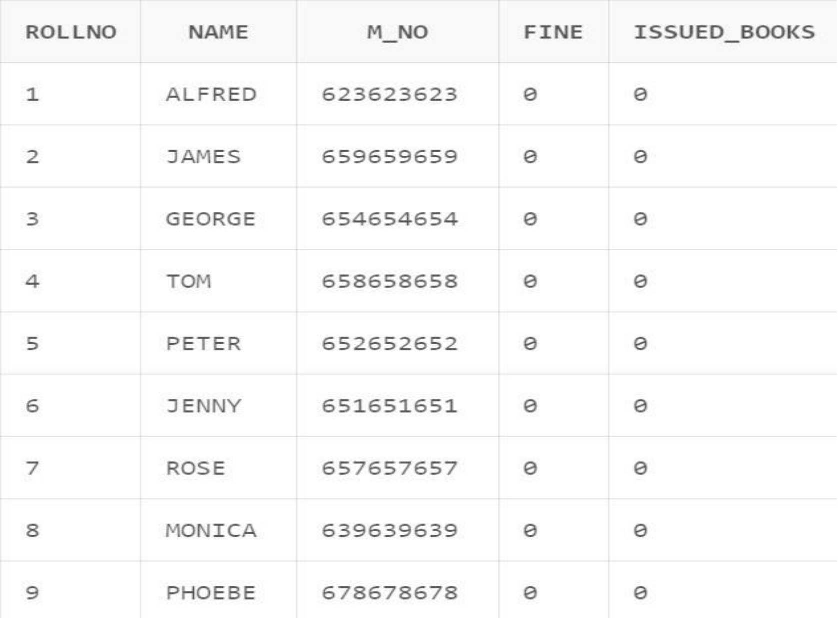
INSERT INTO student VALUES (12, 'CHIRAG', 8736482648,0, 0);

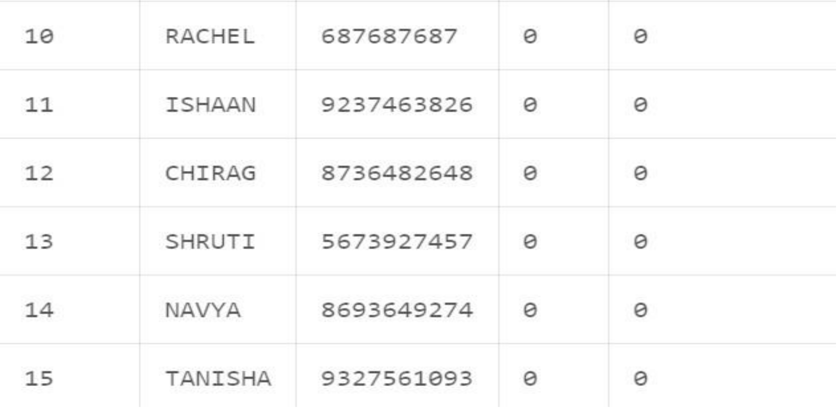
INSERT INTO student VALUES (13, 'SHRUTI', 5673927457,0, 0);

INSERT INTO student VALUES (14, 'NAVYA', 8693649274,0, 0);

INSERT INTO student VALUES (15, 'TANISHA', 9327561093,0, 0);

Select * from student;





ROLLNO	NAME	M_NO	FINE	ISSUED_BOOKS
1	ALFRED	623623623	0	0
2	JAMES	659659659	0	0
3	GEORGE	654654654	0	0
4	TOM	658658658	0	0
5	PETER	652652652	0	0
6	JENNY	651651651	0	0
7	ROSE	657657657	0	0
8	MONICA	639639639	0	0
9	PHOEBE	678678678	0	0
10	RACHEL	687687687	0	0
11	ISHAAN	9237463826	0	0
12	CHIRAG	8736482648	0	0
13	SHRUTI	5673927457	0	0
14	NAVYA	8693649274	0	0
15	TANISHA	9327561093	0	0

-- 2.lib

INSERT INTO lib VALUES(1200,'Lord Of Chaos', 'Robert Jordan','MacMillan', 1, 200, 1); INSERT

INTO lib VALUES(1201,'Fires Of Heaven', 'Robert Jordan','MacMillan', 11, 200, 1);

INSERT INTO lib VALUES(1202,'The Adventures of Tom Sawyer', 'Mark Twain','Sun Publications', 5, 200, 1);

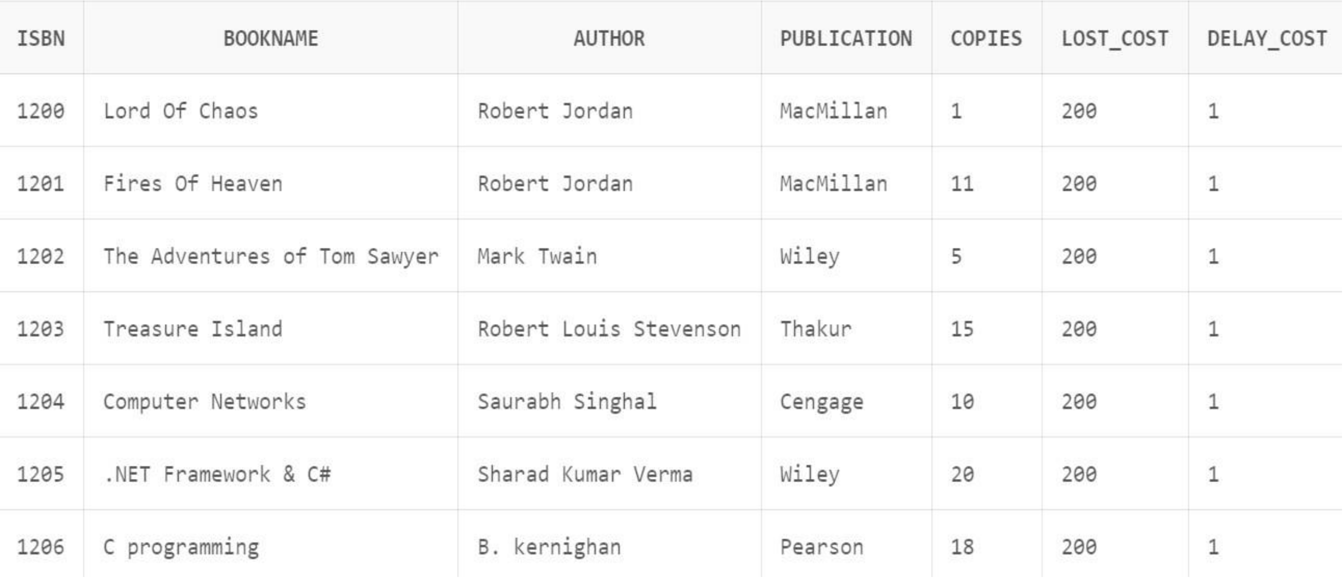
INSERT INTO lib VALUES(1203,'Treasure Island', 'Robert Louis Stevenson','T.Publications', 15,200, 1);

INSERT INTO lib VALUES(1204,'Computer Networks', 'Saurabh Singhal','Cengage', 10, 200, 1);

INSERT INTO lib VALUES(1205,'.NET Framework & C#', 'Sharad Kumar Verma','Wiley', 20, 200, 1);

INSERT INTO lib VALUES(1206,'C programming', 'B. kernighan','Pearson', 18, 200, 1);

SELECT * FROM lib;



ISBN	BOOKNAME	AUTHOR	PUBLICATION	COPIES	LOST_COST	DELAY_COST
1200	Lord Of Chaos	Robert Jordan	MacMillan	1	200	1
1201	Fires Of Heaven	Robert Jordan	MacMillan	11	200	1
1202	The Adventures of Tom Sawyer	Mark Twain	Wiley	5	200	1
1203	Treasure Island	Robert Louis Stevenson	Thakur	15	200	1
1204	Computer Networks	Saurabh Singhal	Cengage	10	200	1
1205	.NET Framework & C#	Sharad Kumar Verma	Wiley	20	200	1
1206	C programming	B. kernighan	Pearson	18	200	1

-- 3.book

INSERT INTO book VALUES(NULL, 1200, 'A');

INSERT INTO book VALUES(NULL, 1201, 'A');

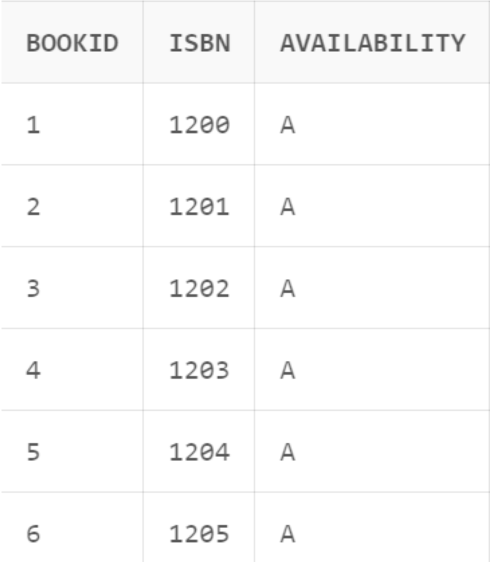
INSERT INTO book VALUES(NULL, 1202, 'A');

INSERT INTO book VALUES(NULL, 1203, 'A');

INSERT INTO book VALUES(NULL, 1204, 'A');

INSERT INTO book VALUES(NULL, 1205, 'A');

SELECT * FROM book;

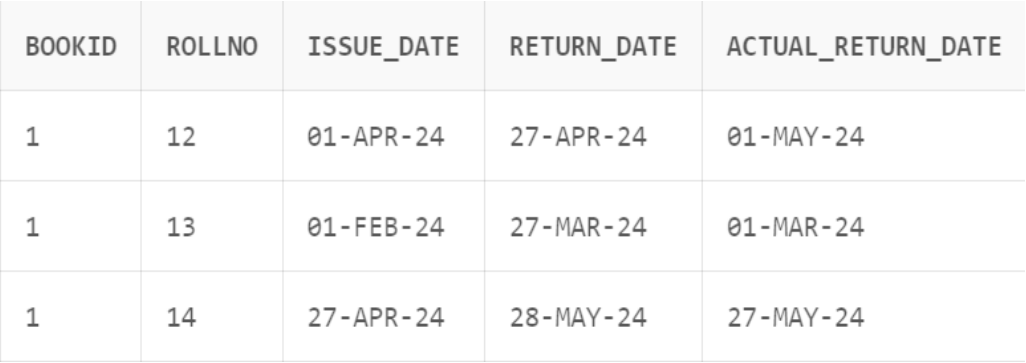


BOOKID	ISBN	AVAILABILITY
1	1200	A
2	1201	A
3	1202	A
4	1203	A
5	1204	A
6	1205	A

-- 4.subscription

```
INSERT INTO subscription VALUES(1,12,
  to_date('01-04-2024','dd-mm-yyyy'),
  to_date('27-04-2024','dd-mm-yyyy'),
  to_date('01-05-2024','dd-mm-yyyy'));

INSERT INTO subscription VALUES(1,13,
  to_date('01-02-2024','dd-mm-yyyy'),
  to_date('27-03-2024','dd-mm-yyyy'),
  to_date('01-03-2024','dd-mm-yyyy'));
INSERT INTO subscription VALUES(1,14,
to_date('27-04-2024','dd-mm-yyyy'),
  to_date('28-05-2024','dd-mm-yyyy'),
  to_date('27-05-2024','dd-mm-yyyy'));
```

SELECT * FROM subscription;



BOOKID	ROLLNO	ISSUE_DATE	RETURN_DATE	ACTUAL_RETURN_DATE
1	12	01-APR-24	27-APR-24	01-MAY-24
1	13	01-FEB-24	27-MAR-24	01-MAR-24
1	14	27-APR-24	28-MAY-24	27-MAY-24

-- Drop tables

```
DROP TABLE student;
DROP TABLE subscription;
DROP TABLE lib;
DROP TABLE book;
```

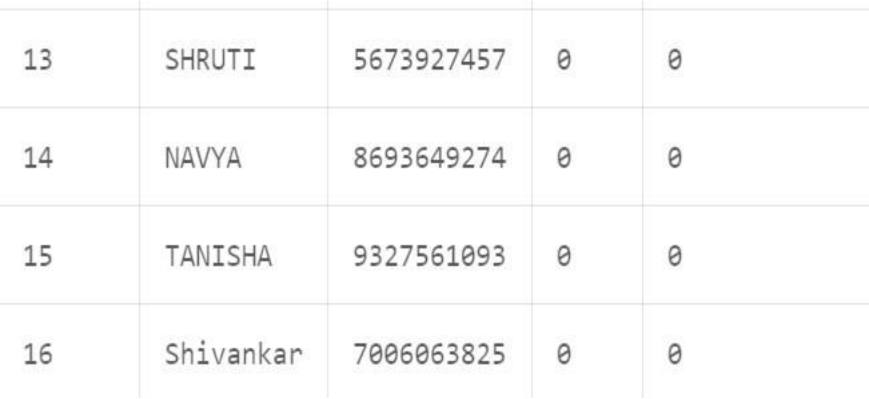
- **PL- SQL starts**

❖ ***<u>Procedure 1</u>***

-- Addition of students --

```
CREATE OR REPLACE PROCEDURE add_student(roll_no in number,
  s_name in varchar, m_no in varchar)
IS
BEGIN
  INSERT INTO student VALUES(roll_no, s_name, m_no, 0,0);
END;


DECLARE
  roll_no number;
name varchar(50);
m_no varchar(10);
BEGIN
  roll_no := 16;
  name := 'Shivankar';
  m_no := '7006063825';
  add_student(roll_no,name,m_no);
END;
```

SELECT * FROM student;



13	SHRUTI	5673927457	0	0
14	NAVYA	8693649274	0	0
15	TANISHA	9327561093	0	0
16	Shivankar	7006063825	0	0

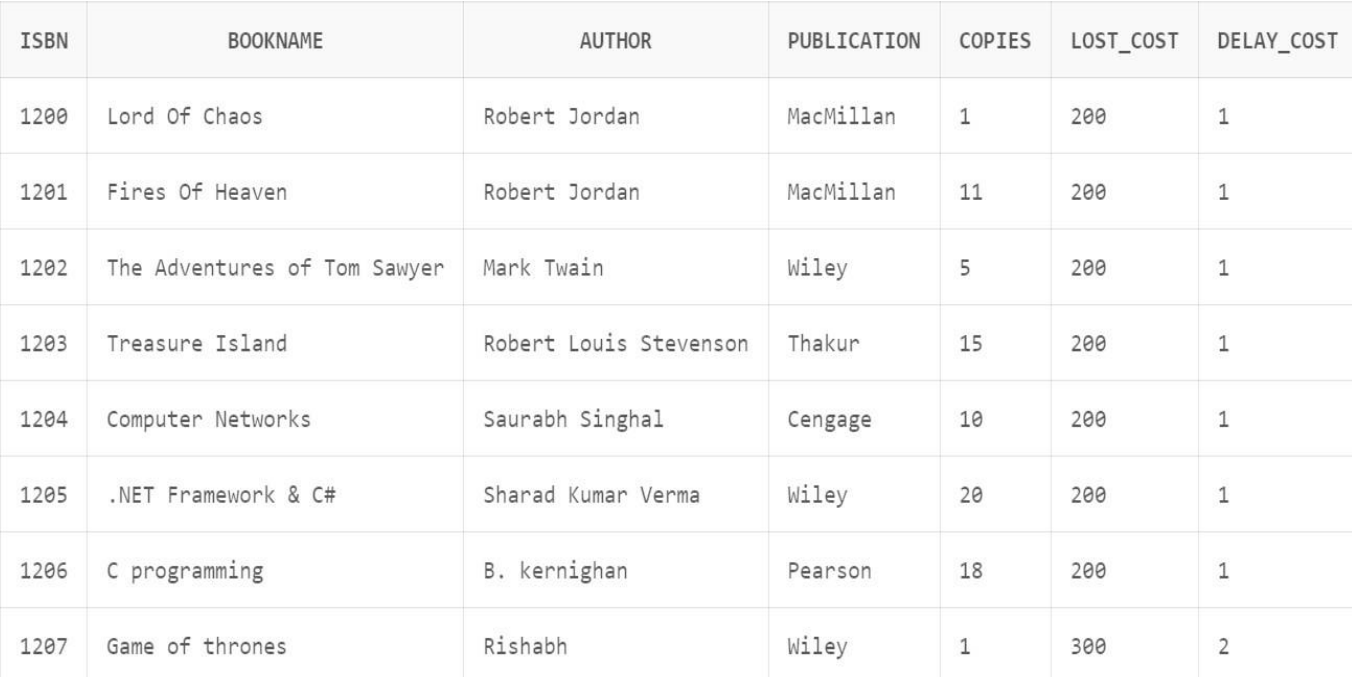
// A new student added as roll no 16.

❖ ***Procedure 2***

-- add first book/ more books –

```
CREATE OR REPLACE PROCEDURE add_first_book(
  isbn_no in number,
  bookname in varchar,
author in varchar,
publication in varchar,
lost_cost in number,
delay_cost in number) IS
BEGIN
  INSERT INTO lib VALUES(isbn_no, bookname,author,publication,
    1,lost_cost, delay_cost);
  INSERT INTO book VALUES(NULL, isbn_no, 'A');
END;

CREATE OR REPLACE PROCEDURE add_more_books(isbn_no in number) IS
BEGIN
  INSERT INTO book VALUES(NULL, isbn_no, 'A');
  UPDATE lib SET copies = copies + 1 WHERE lib.isbn = isbn_no;
END;



DECLARE
  counter number;
isbn_no number;
bookname varchar(50);
author varchar(40);
publication varchar(20);
lost_cost number;
delay_cost number;
BEGIN
  isbn_no := 1207;
  SELECT count(*) INTO counter FROM lib WHERE lib.isbn = isbn_no;
  IF counter > 0 THEN
  add_more_books(isbn_no);

  ELSE
    bookname := 'Game of thrones';
    author := 'Rishabh';
    publication := 'Wiley';
    lost_cost := 300;
    delay_cost := 2;
    add_first_book(isbn_no,bookname,author,publication,
    lost_cost,delay_cost);
  END IF;
END;
```

SELECT * FROM lib;



ISBN	BOOKNAME	AUTHOR	PUBLICATION	COPIES	LOST_COST	DELAY_COST
1200	Lord Of Chaos	Robert Jordan	MacMillan	1	200	1
1201	Fires Of Heaven	Robert Jordan	MacMillan	11	200	1
1202	The Adventures of Tom Sawyer	Mark Twain	Wiley	5	200	1
1203	Treasure Island	Robert Louis Stevenson	Thakur	15	200	1
1204	Computer Networks	Saurabh Singhal	Cengage	10	200	1
1205	.NET Framework & C#	Sharad Kumar Verma	Wiley	20	200	1
1206	C programming	B. kernighan	Pearson	18	200	1
1207	Game of thrones	Rishabh	Wiley	1	300	2

// book with ISBN 1207 is not present so it get added with mentioned attributes.

❖ ***Procedure 3*** **--**

-- issue a book --

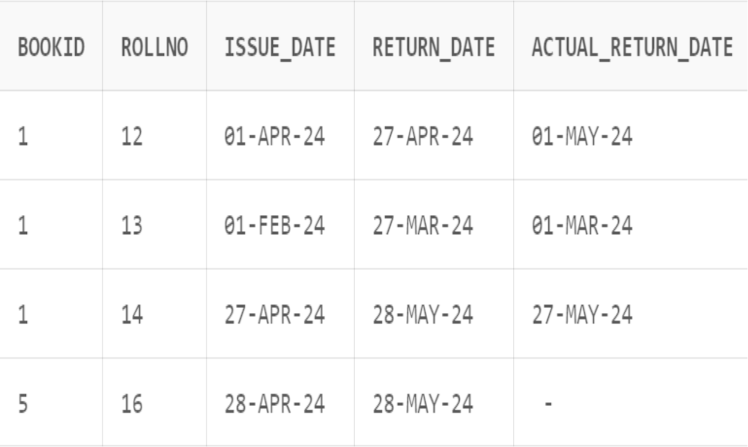
```
CREATE OR REPLACE PROCEDURE issue_book(
  roll_no in number, book_id in number, issue_date in date) IS
```

```
isbn_no number;
  BEGIN
    UPDATE student SET issued_books = issued_books+1
    WHERE rollno = roll_no;
    SELECT isbn INTO isbn_no FROM book WHERE bookid = book_id;

    UPDATE lib SET copies = copies - 1 WHERE isbn = isbn_no;
    UPDATE book SET availability = 'O' WHERE bookid = book_id;
    INSERT INTO subscription VALUES (book_id,roll_no,issue_date,
    issue_date + 30,NULL);
  END;



  DECLARE
  roll_no number;
  book_id number;
  i_date varchar(15);
  issue_date date;
  BEGIN
    roll_no := 16;
    book_id := 4;
    i_date := '2024-04-27';
    issue_date := to_date(i_date, 'yyyy-mm-dd');
  issue_book(roll_no,book_id,issue_date);
  END;
```

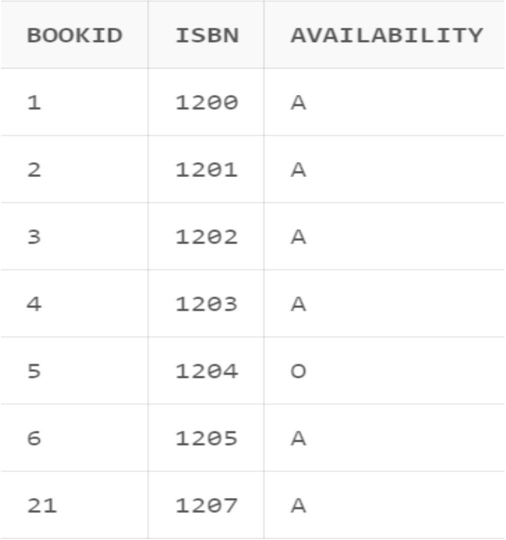
SELECT * FROM subscription;

SELECT * FROM lib;

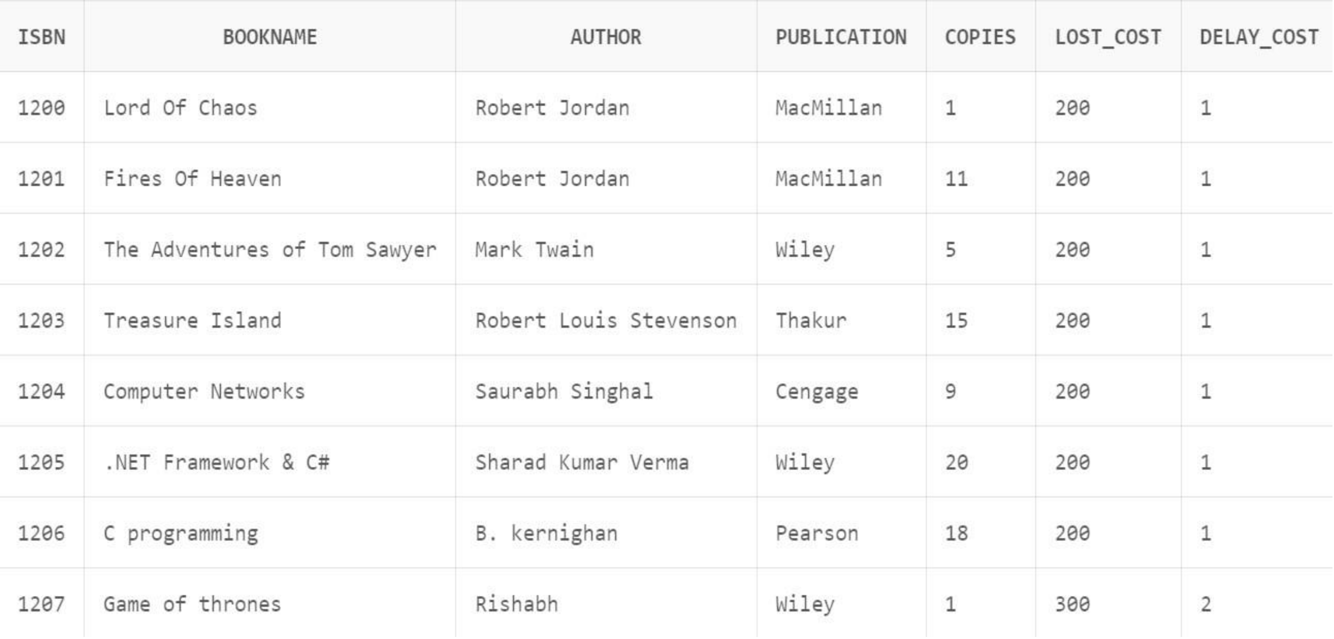
SELECT * FROM book;



BOOKID	ROLLNO	ISSUE_DATE	RETURN_DATE	ACTUAL_RETURN_DATE
1	12	01-APR-24	27-APR-24	01-MAY-24
1	13	01-FEB-24	27-MAR-24	01-MAR-24
1	14	27-APR-24	28-MAY-24	27-MAY-24
5	16	28-APR-24	28-MAY-24	-



BOOKID	ISBN	AVAILABILITY
1	1200	A
2	1201	A
3	1202	A
4	1203	A
5	1204	O
6	1205	A
21	1207	A



ISBN	BOOKNAME	AUTHOR	PUBLICATION	COPIES	LOST_COST	DELAY_COST
1200	Lord Of Chaos	Robert Jordan	MacMillan	1	200	1
1201	Fires Of Heaven	Robert Jordan	MacMillan	11	200	1
1202	The Adventures of Tom Sawyer	Mark Twain	Wiley	5	200	1
1203	Treasure Island	Robert Louis Stevenson	Thakur	15	200	1
1204	Computer Networks	Saurabh Singhal	Cengage	9	200	1
1205	.NET Framework & C#	Sharad Kumar Verma	Wiley	20	200	1
1206	C programming	B. kernighan	Pearson	18	200	1
1207	Game of thrones	Rishabh	Wiley	1	300	2

❖ ***Procedure 4***

-- return book --

```
CREATE OR REPLACE PROCEDURE return_book(
  book_id number, roll_no number) IS
i_date date;
r_date date;
ar_date date;
isbn_no number;
d_cost number;
no_of_copies number;
```

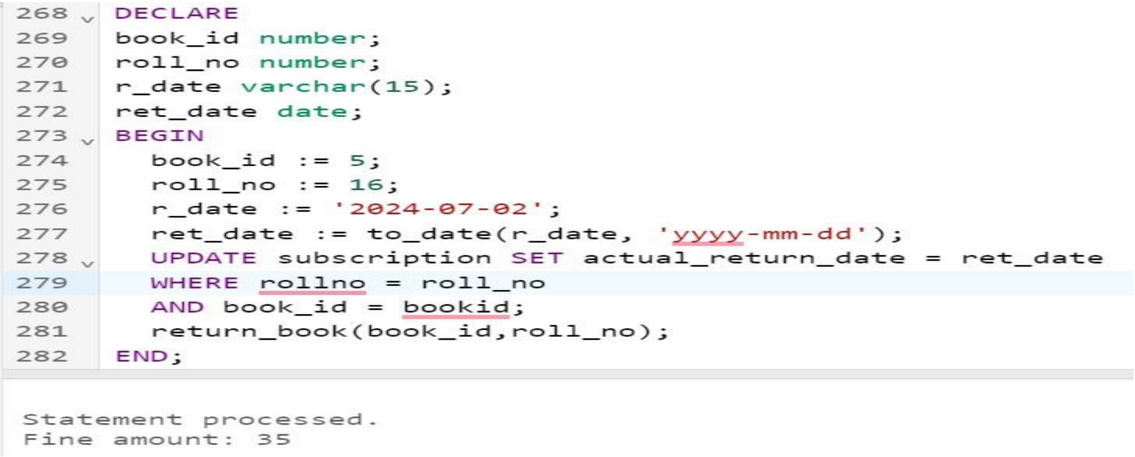
```
fine_amount number;
BEGIN
  SELECT isbn INTO isbn_no FROM book WHERE bookid = book_id;
  SELECT copies, delay_cost INTO no_of_copies, d_cost FROM lib
  WHERE isbn = isbn_no;
  no_of_copies := no_of_copies + 1;
  UPDATE book SET availability = 'A' WHERE bookid = book_id;
  UPDATE lib SET copies = no_of_copies WHERE isbn = isbn_no;
  SELECT issue_date, return_date, actual_return_date INTO i_date, r_date,
  ar_date FROM subscription WHERE bookid = book_id AND rollno = roll_no;
  IF ar_date > r_date THEN

    fine_amount := (ar_date - r_date) * d_cost;
    UPDATE student SET fine = fine_amount WHERE rollno = roll_no;
    dbms_output.put_line('Fine amount: ' || fine_amount);
  END IF;
  UPDATE student SET issued_books = issued_books - 1 WHERE rollno = roll_no;
  --DELETE FROM subscription WHERE bookid = book_id AND rollno = roll_no;
END;
DECLARE
book_id number;
roll_no number;
r_date varchar(15);
ret_date date;
BEGIN
  book_id := 4;
  roll_no := 16;
  r_date := '2024-07-02';
  ret_date := to_date(r_date, 'yyyy-mm-dd');
  UPDATE subscription SET actual_return_date = ret_date
  WHERE rollno = roll_no
  AND book_id = bookid;
return_book(book_id,roll_no);
END;
```

SELECT * FROM student;

SELECT * FROM subscription;

SELECT * FROM lib;

SELECT * FROM book;



```
DECLARE
book_id number;
roll_no number;
r_date varchar(15);
ret_date date;
BEGIN
  book_id := 5;
  roll_no := 16;
  r_date := '2024-07-02';
  ret_date := to_date(r_date, 'yyyy-mm-dd');
  UPDATE subscription SET actual_return_date = ret_date
  WHERE rollno = roll_no
  AND book_id = bookid;
  return_book(book_id,roll_no);
END;
```

```
Statement processed.
Fine amount: 35
```

// Fine generated due to late return of the book .

❖ ***Procedure 5***

-- pay fine –

```
CREATE OR REPLACE PROCEDURE pay_fine(roll_no number) IS
BEGIN
  UPDATE student SET fine = 0 WHERE rollno = roll_no;
END;


DECLARE
rollno number;
BEGIN
  rollno := 16;
pay_fine(rollno);
END;
```

SELECT * FROM student;

❖ ***Procedure 6***

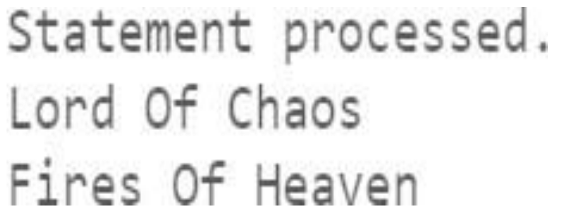
-- BOOK INFO –

```
  CREATE OR REPLACE PROCEDURE book_info(i_sbn IN NUMBER) IS
  c_opies lib.copies%TYPE;
  d_elay_cost lib.delay_cost%TYPE;
  b_ook_name lib.bookname%TYPE;
  l_ost_cost lib.lost_cost%TYPE;
  p_ublisher lib.publication%TYPE;
  a_uthor lib.author%TYPE;
BEGIN
  SELECT copies, delay_cost, bookname, lost_cost, publication, author
  INTO c_opies, d_elay_cost, b_ook_name, l_ost_cost, p_ublisher, a_uthor
  FROM lib
  WHERE lib.isbn = i_sbn;
  dbms_output.put_line('Copies : '||c_opies);
  dbms_output.put_line('delay_cost : '||d_elay_cost);
  dbms_output.put_line('book_name : '||b_ook_name);
  dbms_output.put_line('lost_cost : '||l_ost_cost);
  dbms_output.put_line('publisher : '||p_ublisher);
  dbms_output.put_line('author : '||a_uthor);
END;

DECLARE
    i_sbn number;
BEGIN
  dbms_output.put_line('Enter the book number');
  i_sbn:=1201;
  book_info(i_sbn);
END;
```

❖ *Procedure 7*

-- similar_author_books

```
CREATE OR REPLACE PROCEDURE similar_author_books(auth in varchar)
AS
TEMP varchar(300);
CURSOR c1 IS SELECT bookname FROM lib WHERE author = auth;
rec varchar(300);
BEGIN
FOR rec in c1 LOOP
    dbms_output.put_line(rec.bookname);
END LOOP;
END;
-- Exec similar_author_books('BookName')
DECLARE
author varchar(40);
BEGIN
  author := 'Robert Jordan';
similar_author_books(author);
END;
```

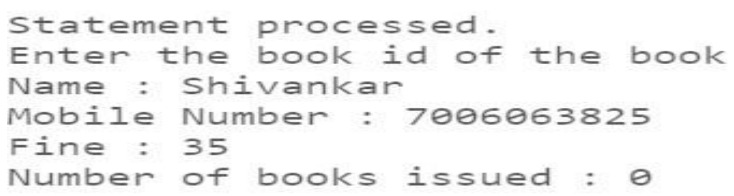


```
Statement processed.
Lord Of Chaos
Fires Of Heaven
```

❖ *Procedure 8*

-- Get student details from book id

```
CREATE OR REPLACE PROCEDURE student_details(book_id IN NUMBER) IS
  roll_no subscription.rollno%TYPE;
  n_ame student.name%TYPE;
  mobile_num student.m_no%TYPE;
  f_ine student.fine%TYPE;
  books_issued student.issued_books%TYPE;
BEGIN
  SELECT s.rollno
  INTO roll_no
  FROM subscription s
  WHERE s.bookid = book_id;

  SELECT st.name, st.m_no, st.fine, st.issued_books
  INTO n_ame, mobile_num, f_ine, books_issued
  FROM student st
  WHERE st.rollno = roll_no;
  dbms_output.put_line('Name : '||n_ame);
  dbms_output.put_line('Mobile Number : '||mobile_num);
  dbms_output.put_line('Fine : '||f_ine);
  dbms_output.put_line('Number of books issued : '||books_issued);
END;

DECLARE
book_id number;
BEGIN
dbms_output.put_line('Enter the book id of the book');
book_id:=5;
student_details(book_id);
END;
```



```
Statement processed.
Enter the book id of the book
Name : Shivankar
Mobile Number : 7006063825
Fine : 35
Number of books issued : 0
```

// Here info. Of user related to that bookid came . the fine here is Rs. 35 which can be paid using procedure 5.

--CODE ENDS--

9. Conclusion

"In conclusion, the development of this Library Resource Database Management System (LRDBMS) underscores the significance of efficient data management in facilitating streamlined library operations. Through the integration of SQL and PL/SQL, we have successfully designed a robust system capable of managing diverse library resources, including books, journals, and multimedia materials.

Our project not only focused on the organization and retrieval of library resources but also emphasized user accessibility and security. We ensure that only authorized individuals can perform specific actions within the system, safeguarding sensitive library data.

Moreover, the incorporation of SQL queries and PL/SQL procedures has enhanced the system's functionality by enabling complex data manipulations and automating routine tasks. This not only improves efficiency but also reduces the margin for error in manual processes.

Furthermore, the modular design of our LRDBMS allows for scalability and adaptability, ensuring that it can accommodate future expansions and modifications as per the evolving needs of the library environment.

In essence, this project demonstrates the potential of database management systems in optimizing library operations, improving resource utilization, and enhancing user experience. Moving forward, continuous refinement and updates will be essential to maintain the system's effectiveness and relevance in the dynamic landscape of library management."

GITHUB LINK :
https://github.com/Shivankar1710/DBMS_PROJECT/tree/main/PROJECT_DBMS

10. References

- **KnowledgeHut. (n.d.). Entity in DBMS. Retrieved from https://www.knowledgehut.com/blog/database/entity-in-dbms**

- **GeeksforGeeks. (n.d.). Entity in DBMS. Retrieved from https://www.geeksforgeeks.org/entity-in-dbms/**

- **JavaTpoint. (n.d.). Entity in DBMS. Retrieved from https://www.javatpoint.com/entity-in-dbms**

- **Oracle. (n.d.). LiveSQL. Retrieved from https://livesql.oracle.com/apex/f?p=590:1000**