

Library Database Management System

Hitesh Aggarwal: 102103596

Rishabh Goyal: 102103585

Ishaan Sharma: 102103583

Aaryan Sood: 102103574

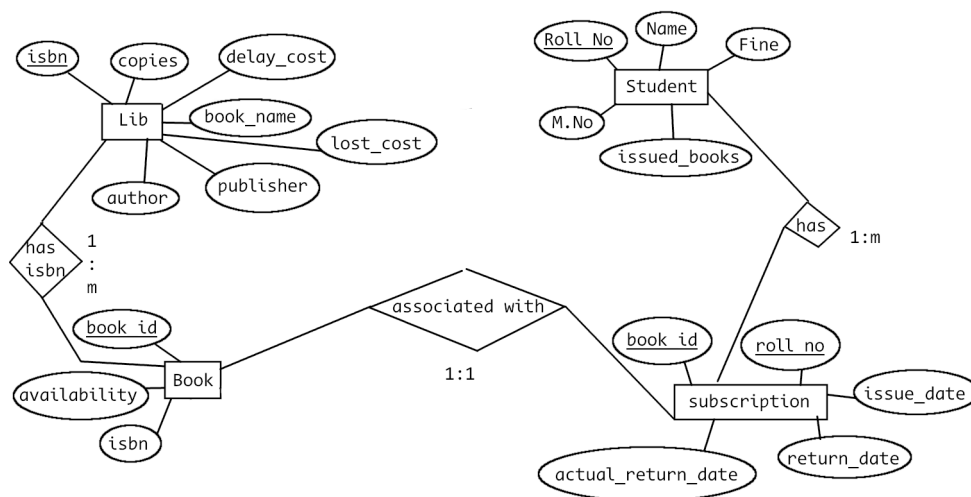
Ishaan Gupta: 102103572

Group: CO21

Problem Statement

Library Management System is an application designed to keep record of the data used by the library (specifically Library of a University). It is used by librarian to manage the library using a computerized system where he/she can add new books, students, issue books etc. Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non computerized system is used. All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

ER diagram



Tables

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

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

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

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

Normalization

Table: Lib

- **1st Normal Form:** There is no multi-valued attribute in the table, 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 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.
- **4th Normal Form:** The table is in BCNF and has no multi-valued dependency, so it is in 4th Normal form.
- **5th Normal Form:** The table cannot be decomposed into smaller tables, so it is in 5th Normal form.

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.
- **4th Normal Form:** The table is in BCNF and has no multi-valued dependency, so it is in 4th Normal form.

- **5th Normal Form:** The table cannot be decomposed into smaller tables, so it is in 5th Normal form.

Table: Book

- **1st Normal Form:** There is no multi-valued attribute in the table, 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 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.
- **4th Normal Form:** The table is in BCNF and has no multi-valued dependency, so it is in 4th Normal form.
- **5th Normal Form:** The table cannot be decomposed into smaller tables, so it is in 5th Normal form.

Table: Subscription

- **1st Normal Form:** There is no multi-valued attribute in the table, so it is in 1st 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.

- **4th Normal Form:** The table is in BCNF and has no multi-valued dependency, so it is in 4th Normal form.
- **5th Normal Form:** The table cannot be decomposed into smaller tables, so it is in 5th Normal form.

PL-SQL statements

Repository Link: <https://www.github.com/hitesh-aggarwal/dbms>

-- Crate tables

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)
    );

-- Drop tables
DROP TABLE student;

DROP TABLE subscription;

DROP TABLE lib;

DROP TABLE book;

-- select statements
SELECT * FROM student;

SELECT * FROM subscription;

SELECT * FROM lib;

SELECT * FROM book;

-- Add foreign keys now.
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);

-- some insert statements

-- 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 lib VALUES(1234,'Lord Of Chaos', 'Robert Jordan',
    'MacMillan', 10, 100, 1);
INSERT INTO lib VALUES(1235,'Fires Of Heaven', 'Robert Jordan',
    'MacMillan', 11, 100, 1);

INSERT INTO book VALUES(NULL, 1234, 'A');
INSERT INTO book VALUES(NULL, 1235, 'A');

INSERT INTO subscription VALUES(1,1,
    to_date('01-02-2023','dd-mm-yyyy'),
    to_date('27-02-2023','dd-mm-yyyy'),
    to_date('01-03-2023','dd-mm-yyyy'));
INSERT INTO subscription VALUES(1,2,
    to_date('01-02-2023','dd-mm-yyyy'),
    to_date('27-02-2023','dd-mm-yyyy'),
    to_date('25-02-2023','dd-mm-yyyy'));

----- plsql starts here -----

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 := 1;
    name := 'abs';
    m_no := '1234567890';
    add_student(roll_no,name,m_no);
END;

```

```

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 := 1234;
    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 := 'ganja';
        lost_cost := 300;
        delay_cost := 2;
        add_first_book(isbn_no,bookname,author,publication,
            lost_cost,delay_cost);
    END IF;
END;

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 := 1;
    roll_no := 1;
    r_date := '2023-03-21';
    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;

```

```

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 := &rollno;
    pay_fine(rollno);
END;

```

```

-- procedure 6 --
-- BOOK INFO --
DECLARE
    i_sbn number;
    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_publisher lib.publication%TYPE;
a_uthor lib.author%TYPE;
PROCEDURE book_info(i_sbn in number) IS
BEGIN
    SELECT copies,delay_cost,bookname,lost_cost,publisher,author
    INTO c_opies,delay_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 : '||delay_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;
BEGIN
    dbms_output.put_line('Enter the book number');
    i_sbn:=&i_sbn;
    book_info(i_sbn);
END;

-- Get the total fine of a student
CREATE OR REPLACE PROCEDURE retrieve_pending_fine(roll in number,
    fine in OUT number, cur_date in varchar) AS
    del_cost number;
    isb_no number;
    days number;
    ret_date date:= to_date(cur_date,'yyyy-mm-dd');
    CURSOR c IS SELECT * FROM subscription WHERE rollno=roll;
BEGIN
    FOR rec in c LOOP
        IF ret_date > rec.return_date THEN
            SELECT isbn INTO isb_no FROM book WHERE bookid=rec.bookid;
            SELECT delay_cost INTO del_cost FROM lib WHERE isbn=isb_no;
            days := ret_date - rec.return_date;
            fine := fine + days*del_cost;
        END IF;
    END LOOP;
END;

DECLARE

```

```

    roll number;
    fine number;
BEGIN
    roll := 1;
    fine := 0;
    retrieve_pending_fine(roll,fine,'2023-07-25');
    dbms_output.put_line('Fine = ' || fine);
END;

```

```

-- 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 := 'rishabh';
    similar_author_books(author);
END;

```

```

-- procedure 8 --
-- Get student details from book id
DECLARE
    book_id subscription.bookid%TYPE;
    roll_no subscription.rollno%TYPE;
    n_ame student.name%TYPE;
    mobile_num student.m_no%TYPE;
    f_in student.fine%TYPE;
    books_issued student.issued_books%TYPE;

```

```

PROCEDURE student_details(book_id in number) IS
BEGIN
SELECT subscription.rollno INTO roll_no FROM subscription
WHERE bookid=book_id;
SELECT student.name,student.m_no,student.fine,student.issued_books
INTO n_ame, mobile_num,f_ine,books_issued FROM student
WHERE student.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;
BEGIN
dbms_output.put_line('Enter the book id of the book');
book_id:=&book_id;
student_details(book_id);
END;

```

```

-- Issue a book (procedure 1)
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 = '0' 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 := 1;
book_id := 1;
i_date := '2023-01-01';
issue_date := to_date(i_date, 'yyyy-mm-dd');
issue_book(roll_no,book_id,issue_date);
END;
```