**-- FINAL CODE**

**-- Create 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)

);

**-- 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);

**-- Insert some statements**

**-- 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);

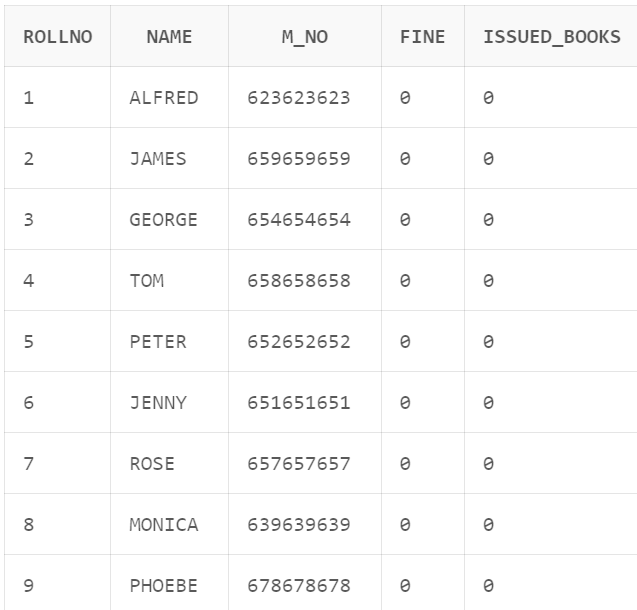
INSERT INTO student VALUES (12, 'CHIRAG', 8736482648,0, 0);

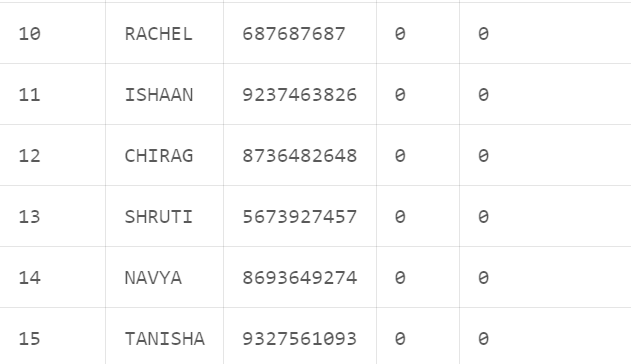
INSERT INTO student VALUES (13, 'SHRUTI', 5673927457,0, 0);

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

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

**Select \* from student;**

****

****

**-- 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 India Publications', 5, 200, 1);

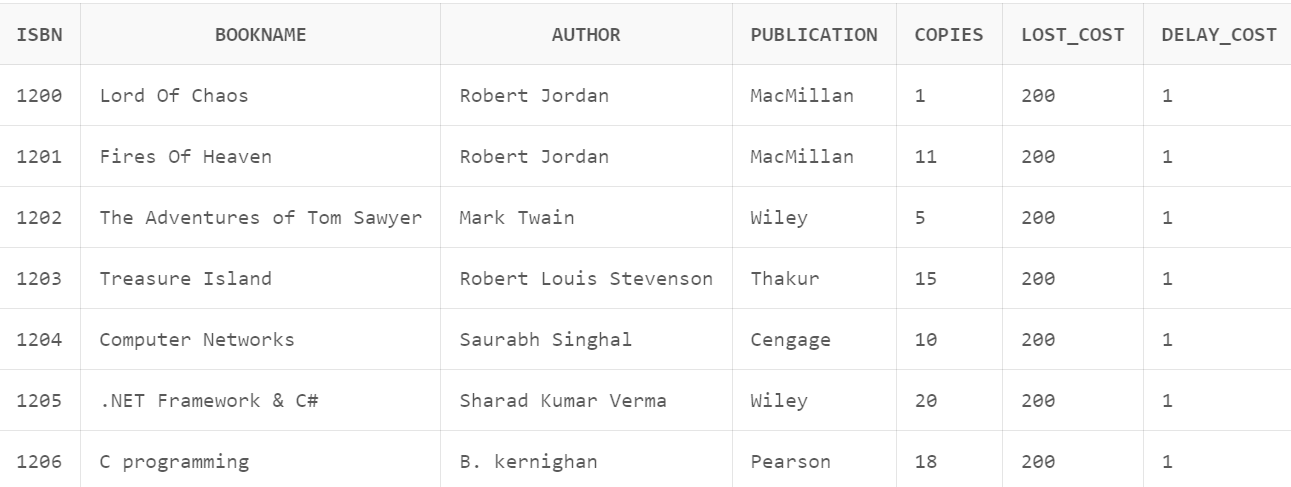
INSERT INTO lib VALUES(1203,'Treasure Island', 'Robert Louis Stevenson','Thakur Publications', 15, 200, 1);

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;**

****

**-- 3.book**

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

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

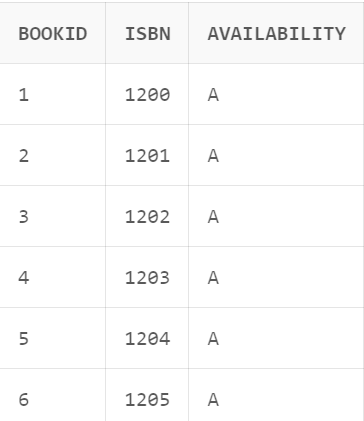
INSERT INTO book VALUES(NULL, 1202, 'A');

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

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

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

**SELECT \* FROM book;**

****

**-- 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'));

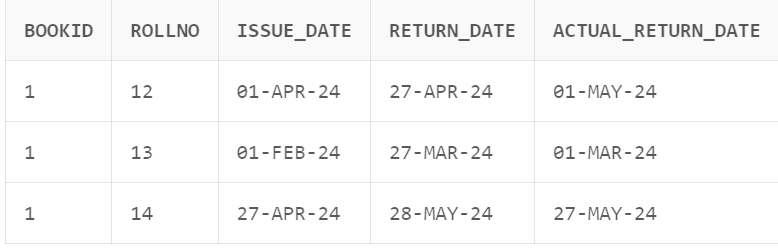
INSERT INTO subscription VALUES(1,14,

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;**

****

**-- Drop tables**

DROP TABLE student;

DROP TABLE subscription;

DROP TABLE lib;

DROP TABLE book;

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

**-- *Procedure 1* --**

**-- Add 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;

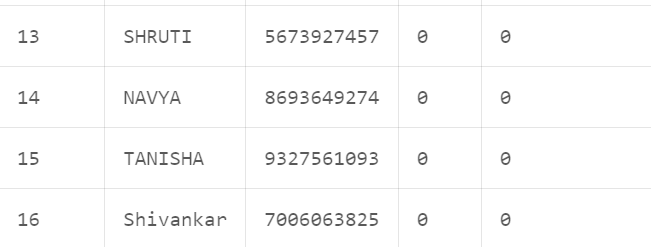
name := 'Shivankar';

m\_no := '7006063825';

add\_student(roll\_no,name,m\_no);

END;

**SELECT \* FROM student;**

** // 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;**

****

**// 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');

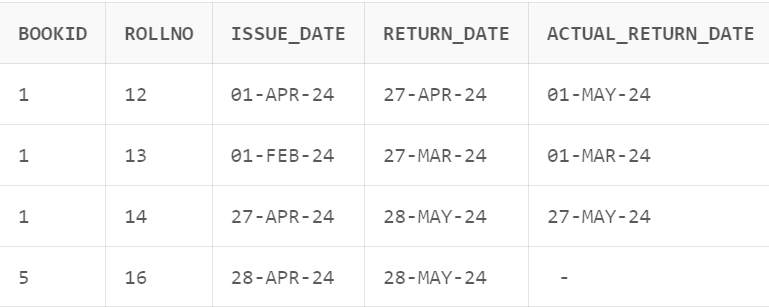
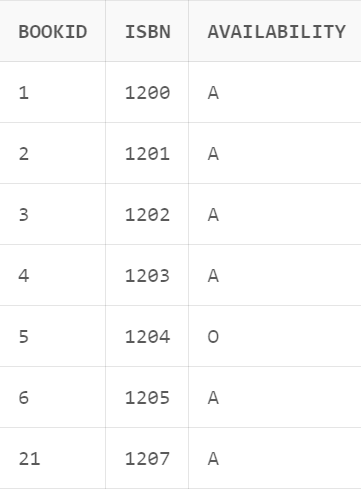
issue\_book(roll\_no,book\_id,issue\_date);

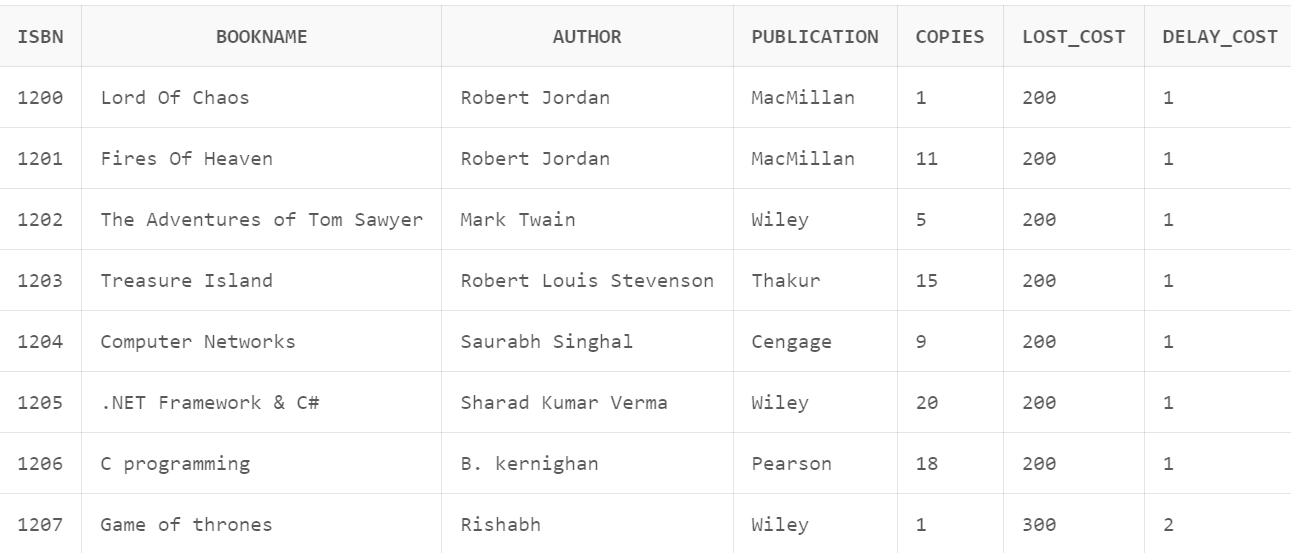
END;

**SELECT \* FROM subscription;**

**SELECT \* FROM lib;**

**SELECT \* FROM book;**

** **

****

**-- *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);

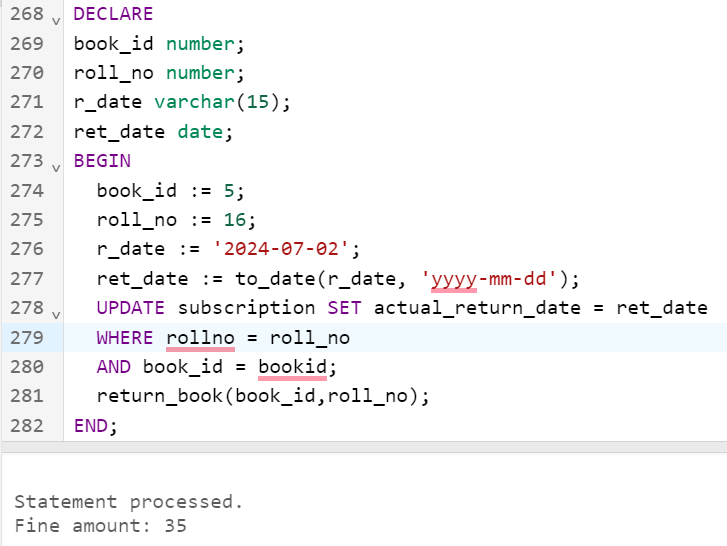
END;

**SELECT \* FROM student;**

**SELECT \* FROM subscription;**

**SELECT \* FROM lib;**

**SELECT \* FROM book;**

** // 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

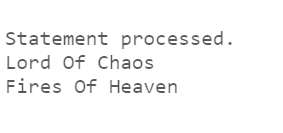
author varchar(40);

BEGIN

author := 'Robert Jordan';

similar\_author\_books(author);

END;



**-- *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;

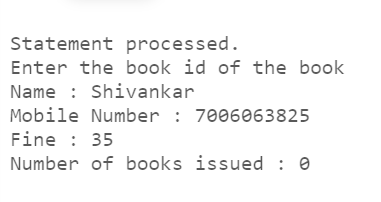
BEGIN

dbms\_output.put\_line('Enter the book id of the book');

book\_id:=5;

student\_details(book\_id);

END;

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

**THE END**

<https://github.com/Shivankar1710/DBMS_PROJECT/tree/main/PROJECT_DBMS>