DATABASE MANAGEMENT SYSTEMS - PROJECT

LIBRARY MANAGEMENT SYSTEM

CREATION OF TABLES:

```
SQL> create table student_details (student_reg_no varchar(10) pr
imary key, student_name varchar(20), gender varchar(7), phone_no
 number(12));

Table created.
```

```
SQL> create table staff_details (staff_id number(5) primary key,
 staff_name varchar(20), gender varchar(7), phone_no number(12))
;

Table created.
```

```
SQL> create table book_details (book_id number(5) primary key, b
ook_name varchar(25), book_subject varchar(10), author varchar(2
0), publication_name varchar(10), publication_year number(5));

Table created.
```

```
SQL> create table book_position_details (book_id number(5), floo
r_no number(2), shelf_id varchar(5));

Table created.
```

```
SQL> create table student_transactions (student_reg_no varchar(1
0), book_id number(5), issue_date date, due_date date);

Table created.
```

```
SQL> create table staff_transactions (staff_id number(5), book_i
d number(5), issue_date date, due_date date);

Table created.
```

```
SQL> create table transaction_details_student (student_reg_no va
rchar(10), no_of_books_borrowed number(2), total_fine number(4))
;

Table created.
```

```
SQL> create table transaction_details_staff (staff_id number(5),
 no_of_books_borrowed number(2), total_fine number(4));

Table created.
```

CREATING FOREIGN KEYS:

```
SQL> alter table book_position_details add constraint fk1_book_p
ositon_details foreign key (book_id) references book_details (bo
ok_id);

Table altered.
```

```
SQL> alter table student_transactions add constraint fk1_student
_transactions foreign key (student_reg_no) references student_de
tails (student_reg_no);

Table altered.
```

```
SQL> alter table student_transactions add constraint fk2_student
_transactions foreign key (book_id) references book_details (boo
k_id);

Table altered.
```

```
SQL> alter table staff_transactions add constraint fk1_staff_tra
nsactions foreign key (staff_id) references staff_details (staff
_id);

Table altered.
```

```
SQL> alter table staff_transactions add constraint fk2_staff_tra
nsactions foreign key (book_id) references book_details (book_id
);

Table altered.
```

```
SQL> alter table transaction_details_student add constraint fk1_
transaction_details foreign key (student_reg_no) references stud
ent_details (student_reg_no);

Table altered.
```

```
SQL> alter table transaction_details_staff add constraint fk1_tr
ansaction_details_staff foreign key (staff_id) references staff_
details (staff_id);

Table altered.
```

QUERIES USED FOR INSERTING THE DATA:

```
SQL> insert into student_details values ('&student_reg_no', '&st
udent_name', '&gender', &phone_no);
```

```
SQL> insert into staff_details values (&staff_id, '&staff_name',
 '&gender', &number);
```

```
SQL> insert into book_details values (&book_id, '&book_name', '&book_
subject', '&author', '&publication_name', &publication_year);
```

```
SQL> insert into book_position_details values (&book_id, &floor_
no, '&shelf_id');
```

```
SQL> insert into student_transactions (student_reg_no, book_id,
issue_date) values ('&student_reg_no', &book_id, '&issue_date');
```

```
SQL> insert into staff_transactions (staff_id, book_id, issue_da
te) values ('&staff_id', &book_id, '&issue_date');
```

```
SQL> insert into transaction_details_student values ('&student_r
eg_no', &no_of_books_borrowed, &total_fine);
```

```
SQL> insert into transaction_details_staff values (&staff_id, &n
o_of_books_borrowed, &total_fine);
```

DATA IN THE TABLE AFTER INSERTION:

```
SQL> select * from student_details;

STUDENT_RE STUDENT_NAME         GENDER    PHONE_NO
---------- -------------------- ------- ----------
23mia1001  ram                  male    9876543210
23mia1002  olivia               female  1234567890
23mia1003  john                 male    8765432109
23mia1004  kiana                female  2345678901
23mia1005  james                male    5678901234
23mia1006  william              male    3456789012
23mia1007  ava                  female  4567890123
23mia1008  alexander            male    8901234567
23mia1009  mike                 male    6789012345
23mia1010  emma                 female  1209384756

10 rows selected.
```

```
SQL> select * from staff_details;

  STAFF_ID STAFF_NAME           GENDER    PHONE_NO
---------- -------------------- ------- ----------
      1001 charles              male    7895462135
      1002 andrew               male    9632587410
      1003 tina                 female  7539514682
      1004 adam                 male    8624793155
      1005 sophia               female  8462597138
      1006 larry                male    1023756984
      1007 amelia               female  7890456198
      1008 sasha                female  5972186245
      1009 victor               male    9875200878
      1010 daisy                female  4507893125

10 rows selected.
```

```
SQL> select * from book_details;

   BOOK_ID BOOK_NAME                BOOK_SUBJE AUTHOR              PUBLICATIO
---------- ------------------------ ---------- ------------------- ----------
PUBLICATION_YEAR
----------------
       101 introduction to dbms     dbms       chris               marico
           2021

       102 sql/plsql                dbms       korth               rupa
           2019

       103 into algorithms          dsa        cormen              jimmy
           2017


   BOOK_ID BOOK_NAME                BOOK_SUBJE AUTHOR              PUBLICATIO
---------- ------------------------ ---------- ------------------- ----------
PUBLICATION_YEAR
----------------
       104 data structures          dsa        stephen             peguin
           2015

       105 digital electronics      eee        anuradha            westland
           2022

       106 semiconductor electronics eee       stephen             peguin
           2019


   BOOK_ID BOOK_NAME                BOOK_SUBJE AUTHOR              PUBLICATIO
---------- ------------------------ ---------- ------------------- ----------
PUBLICATION_YEAR
----------------
       107 probabilty               math       stan                rupa
           2017

       108 statistics               math       jami                westland
           2021

       109 basic french             french     korth               rupa
           2020


   BOOK_ID BOOK_NAME                BOOK_SUBJE AUTHOR              PUBLICATIO
---------- ------------------------ ---------- ------------------- ----------
PUBLICATION_YEAR
----------------
       110 cliches in france        french     thomas              marico
           2021


10 rows selected.
```

```
SQL> select * from book_position_details;

   BOOK_ID   FLOOR_NO SHELF
---------- ---------- -----
       101          2 s205
       102          2 s209
       103          1 s102
       104          1 s108
       105          3 s309
       106          3 s305
       107          4 s408
       108          4 s403
       109          1 s104
       110          1 s108

10 rows selected.
```

```
SQL> select * from student_transactions;

STUDENT_RE    BOOK_ID ISSUE_DAT DUE_DATE
---------- ---------- --------- ---------
23mia1004         108 25-MAR-24
23mia1008         104 15-MAR-24
23mia1007         105 22-MAR-24
23mia1009         102 05-APR-24
```

Using the “update” command to insert the values of due_date, which is 14 from the issue_date.

```
SQL> update student_transactions set due_date = issue_date + 14;

4 rows updated.
```

```
SQL> select * from student_transactions;

STUDENT_RE    BOOK_ID ISSUE_DAT DUE_DATE
---------- ---------- --------- ---------
23mia1004         108 25-MAR-24 08-APR-24
23mia1008         104 15-MAR-24 29-MAR-24
23mia1007         105 22-MAR-24 05-APR-24
23mia1009         102 05-APR-24 19-APR-24
```

```
SQL> select * from staff_transactions;

  STAFF_ID    BOOK_ID ISSUE_DAT DUE_DATE
---------- ---------- --------- ---------
      1001         107 12-APR-24
      1007         109 14-APR-24
      1010         101 10-APR-24
      1002         106 29-MAR-24
```

Using the “update” command to insert the values of due_date, which is 14 from the issue_date.

```
SQL>  update staff_transactions set due_date = issue_date + 14;

4 rows updated.
```

```
SQL> select * from staff_transactions;

  STAFF_ID    BOOK_ID ISSUE_DAT DUE_DATE
---------- ---------- --------- ---------
      1001        107 12-APR-24 26-APR-24
      1007        109 14-APR-24 28-APR-24
      1010        101 10-APR-24 24-APR-24
      1002        106 29-MAR-24 12-APR-24
```

```
SQL> select * from transaction_details_student;

STUDENT_RE NO_OF_BOOKS_BORROWED TOTAL_FINE
---------- -------------------- ----------
23mia1001                     0        120
23mia1002                     0          0
23mia1003                     0         12
23mia1004                     1         19
23mia1005                     0          8
23mia1006                     0          7
23mia1007                     1         90
23mia1008                     1         19
23mia1009                     1          0
23mia1010                     0          0

10 rows selected.
```

```
SQL> select * from transaction_details_staff;

  STAFF_ID NO_OF_BOOKS_BORROWED TOTAL_FINE
---------- -------------------- ----------
      1001                    1         10
      1002                    1          0
      1003                    0         90
      1004                    0         19
      1005                    0         25
      1006                    0          0
      1007                    1         89
      1008                    0         54
      1009                    0         72
      1010                    1         27

10 rows selected.
```

QUERIES:

1. List all the student details who have borrowed a book after 01st of april in this year.

```
SQL> select student_details.student_reg_no, student_details.stud
ent_name, student_details.gender, student_transactions.issue_dat
e from student_details join student_transactions on student_deta
ils.student_reg_no = student_transactions.student_reg_no where s
tudent_transactions.issue_date > '01-apr-24';

STUDENT_RE STUDENT_NAME         GENDER  ISSUE_DAT
---------- -------------------- ------- ---------
23mia1009  mike                 male    05-APR-24
```

2. Finding the number of students who have not borrowed any book.

```
SQL> select count(student_reg_no) from transaction_details_stude
nt where no_of_books_borrowed = 0;

COUNT(STUDENT_REG_NO)
---------------------
                    6
```

3. Finding the staff_id who don't have any fine.

```
SQL> select staff_id from transaction_details_staff where total_
fine = 0;

  STAFF_ID
----------
      1002
      1006
```

4. Displaying the student details who have paid fine more than the average fine paid by the students.

```
SQL> select avg(total_fine) from transaction_details_student;

AVG(TOTAL_FINE)
---------------
           27.5
```

```
SQL> select student_reg_no, total_fine from transaction_details_
student where total_fine > (select avg(total_fine) from transact
ion_details_student);

STUDENT_RE TOTAL_FINE
---------- ----------
23mia1001         120
23mia1007          90
```

5. Prompting the user to enter the book_id and finding the position of the book in the library.

```
SQL> ed
Wrote file afiedt.buf

  1  declare
  2  b_id book_position_details.book_id%type;
  3  b_f_n book_position_details.floor_no%type;
  4  b_s_id book_position_details.shelf_id%type;
  5  begin
  6  b_id := '&b_id';
  7  select book_id, floor_no, shelf_id into b_id, b_f_n, b_s_id
 from book_position_details where book_id = b_id;
  8  dbms_output.put_line('book id: '||b_id);
  9  dbms_output.put_line('floor no: '||b_f_n);
 10  dbms_output.put_line('shelf id: '||b_s_id);
 11* end;
SQL> /
Enter value for b_id: 107
old   6: b_id := '&b_id';
new   6: b_id := '107';
book id: 107
floor no: 4
shelf id: s408

PL/SQL procedure successfully completed.
```

6. Prompting the user to enter the book_id and getting the book details.

```
  1  declare
  2  b_id book_details.book_id%type;
  3  b_n book_details.book_name%type;
  4  b_s book_details.book_subject%type;
  5  b_a book_details.author%type;
  6  b_p book_details.publication_name%type;
  7  b_p_y book_details.publication_year%type;
  8  begin
  9  b_id := '&b_id';
 10  select book_id, book_name, book_subject, author, publication_nam
e, publication_year into b_id, b_n, b_s, b_a, b_p, b_p_y from book_de
tails where book_id = b_id;
 11  dbms_output.put_line('book id: '||b_id);
 12  dbms_output.put_line('book name: '||b_n);
 13  dbms_output.put_line('book subject: '||b_s);
 14  dbms_output.put_line('author: '||b_a);
 15  dbms_output.put_line('publication: '||b_p);
 16  dbms_output.put_line('publication_year: '||b_p_y);
 17* end;
SQL> /
Enter value for b_id: 108
old   9: b_id := '&b_id';
new   9: b_id := '108';
book id: 108
book name: statistics
book subject: math
author: jami
publication: westland
publication_year: 2021

PL/SQL procedure successfully completed.
```

7. To display all the details of female students using cursors in PL/SQL.

```
SQL> declare
  2  cursor c1 is select * from student_details where gender = '
female';
  3  v1 c1%rowtype;
  4  begin
  5  open c1;
  6  loop
  7  fetch c1 into v1;
  8  exit when c1%notfound;
  9  dbms_output.put_line(v1.student_reg_no||' '||v1.student_nam
e||' '||v1.gender||' '||v1.phone_no);
 10  end loop;
 11  end;
 12  /
23mia1002 olivia female 1234567890
23mia1004 kiana female 2345678901
23mia1007 ava female 4567890123
23mia1010 emma female 1209384756

PL/SQL procedure successfully completed.
```

8. Finding the books which are present in 2nd floor.

```
SQL> select book_id, shelf_id from book_position_details where f
loor_no = 2;

   BOOK_ID SHELF
---------- -----
       101 s205
       102 s209
```

9. Finding all the staff who have to return their book in the month of april.

```
SQL> select staff_id, due_date from staff_transactions where due
_date > '01-apr-24';

  STAFF_ID DUE_DATE
---------- ---------
      1001 26-APR-24
      1007 28-APR-24
      1010 24-APR-24
      1002 12-APR-24
```

10. Finding the book details of the book which was borrowed by the student with reg no ‘23mia1007’.

```
SQL> select student_transactions.student_reg_no, book_details.bo
ok_id, book_details.book_name, book_details.book_subject, book_d
etails.author, book_details.publication_name, book_details.publi
cation_year from student_transactions join book_details on stude
nt_transactions.book_id = book_details.book_id where student_tra
nsactions.student_reg_no = '23mia1007';

STUDENT_RE    BOOK_ID BOOK_NAME                 BOOK_SUBJE AUTHO
R
---------- ---------- ------------------------ ---------- -----
--------------
PUBLICATIO PUBLICATION_YEAR
---------- ----------------
23mia1007         105 digital electronics       eee        anura
dha
westland               2022
```

Updated tables for the given queries:

```
SQL> insert into student_transactions values('23mia1004', 103, '
17-apr-24', '01-may-24');

1 row created.
```

```
SQL> select * from student_transactions;

STUDENT_RE    BOOK_ID ISSUE_DAT DUE_DATE
---------- ---------- --------- ---------
23mia1004         108 25-MAR-24 08-APR-24
23mia1008         104 15-MAR-24 29-MAR-24
23mia1007         105 22-MAR-24 05-APR-24
23mia1009         102 05-APR-24 19-APR-24
23mia1004         103 17-APR-24 01-MAY-24
```

```
SQL> update transaction_details_student set no_of_books_borrowed
 = 2 where student_reg_no = '23mia1004';

1 row updated.
```

```
SQL> select * from transaction_details_student;

STUDENT_RE NO_OF_BOOKS_BORROWED TOTAL_FINE
---------- ------------------- ----------
23mia1001                    0        120
23mia1002                    0          0
23mia1003                    0         12
23mia1004                    2         19
23mia1005                    0          8
23mia1006                    0          7
23mia1007                    1         90
23mia1008                    1         19
23mia1009                    1          0
23mia1010                    0          0

10 rows selected.
```

Queries given:

1. To find the student who has borrowed the maximum number of books.

```
SQL> select student_reg_no from transaction_details_student wher
e no_of_books_borrowed = (select max(no_of_books_borrowed) from
transaction_details_student);

STUDENT_RE
----------
23mia1004
```

2. To find the total fine paid by each student.

```
SQL> select student_reg_no, total_fine from transaction_details_
student;

STUDENT_RE TOTAL_FINE
---------- ----------
23mia1001         120
23mia1002           0
23mia1003          12
23mia1004          19
23mia1005           8
23mia1006           7
23mia1007          90
23mia1008          19
23mia1009           0
23mia1010           0

10 rows selected.
```