**Assignment No : 5**

**Write a PL/SQL Block of code for:**

**a) Borrower(Rollin, Name, DateofIssue, NameofBook, Status)**

**b) Fine(Roll\_no, Date, Amt)**

**Calculate the fine for a book by accessing the borrower’s information.**

**Note:** Create Borrower and fine tables on sql prompt first, insert records in borrower

then execute commit;

**Note:** After executing PL/SQL block to see changes on sql prompt press commit in sql developer;

SQL> desc borrower;

Name Null? Type

----------------------------------------- -------- --------------------

ROLLNO NOT NULL NUMBER(38)

NAME VARCHAR2(20)

DOI DATE

NAMEOFBOOK VARCHAR2(20)

STATUS CHAR(1)

SQL> desc fine;

Name Null? Type

----------------------------------------- -------- --------------------

ROLLNO NUMBER(38)

FDATE DATE

AMT NUMBER(38)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PL/SQL Block\*\*\*\*\*\*\*\*/

DECLARE

var\_roll number(6);

var\_name varchar2(25);

var\_fine number(6):=0;

var\_days number(6):=0;

BEGIN

var\_roll := &var\_roll;

dbms\_output.put\_line('Rollno' || var\_roll);

SELECT abs(doi - sysdate)

INTO var\_days

FROM borrower

WHERE rollno=var\_roll and name='&var\_name' and status='I';

if(var\_days > 15 and var\_days<=30)then

var\_fine:=(var\_days-15)\*5;

elsif(var\_days>30)then

var\_days:=var\_days-15;

var\_fine:=var\_days\*5;

var\_days:=var\_days-15;

var\_fine:=var\_fine + (var\_days)\*50;

end if;

update borrower set status = 'R' where rollno=var\_roll;

if(var\_fine > 0)then

insert into fine values(var\_roll,sysdate,var\_fine);

end if;

dbms\_output.put\_line('The student '|| var\_name || ' has fine ' || var\_fine);

END;

/\* procedure

Write a Stored Procedure namely proc\_Grade for the categorization of student. If marks scored by students in examination is <=1500 and marks>=990 then student will be placed in distinction category if marks scored are between 989 and900 category is first class, if marks 899 and 825 category is Higher Second Class Write a PL/SQL block for using procedure created with above requirement. Stud\_Marks(Rollno,name, total\_marks) Result(Roll,Name, Class)

\*/

Step 1: Create tables as Stud\_Marks(Rollno,name, total\_marks), Result(Roll,Name, Class)

Step 2: Insert records in Stud\_Marks , execute commit.

step 3: Execute procedure using sql developer

step 4: Display and check contents in Result table.(Result table must inculde records of students in stud\_marks and their class)

create or replace PROCEDURE PROC\_GRADE AS

CURSOR MARKS\_UPDATE IS SELECT s.rollno,s.name,s.totalmarks from stud\_marks s;

BEGIN

FOR myrecord IN MARKS\_UPDATE LOOP

if myrecord.totalmarks >990 and myrecord.totalmarks<=1500 then

INSERT INTO result values(myrecord.rollno,myrecord.name,'distinction');

elsif myrecord.totalmarks >989 and myrecord.totalmarks<= 900 then

INSERT INTO result values(myrecord.rollno,myrecord.name,'firstclass');

elsif myrecord.totalmarks >899 and myrecord.totalmarks<= 825 then

INSERT INTO result values(myrecord.rollno,myrecord.name,'Higher 2nd');

else

INSERT INTO result values(myrecord.rollno,myrecord.name,'2nd Class');

end if;

END LOOP;

COMMIT;

END PROC\_GRADE;

/\*

Assignment no 8 Trigger

Write a database trigger on library table. The System should keep track of the records that are being updated or deleted. The old value of updated or deleted records should be added in Library\_Audit Table.

\*/

SQL> connect system

Enter password:

Connected.

SQL> create table library(id number,book varchar(20),author varchar(20));

Table created.

SQL> insert into library values(1,'C++','kanetkar');

1 row created.

SQL> insert into library values(2,'C','Das');

1 row created.

SQL> insert into library values(3,'DBMS','Korth');

1 row created.

CREATE OR REPLACE TRIGGER BOOKS\_AUDIT

BEFORE DELETE OR UPDATE ON library

REFERENCING OLD AS OLD NEW AS NEW

FOR EACH ROW

BEGIN

INSERT INTO library\_audit

VALUES

( :old.id,

:old.book,

:old.author,

sysdate);

END;

SQL> alter table library\_audit add sdate date;

Table altered.

SQL> select \* from library\_audit;

no rows selected

SQL> delete from library where book = 'C';

1 row deleted.

SQL> select \* from library\_audit;

no rows selected

SQL> select \* from library;

ID BOOK AUTHOR

---------- -------------------- --------------------

1 C++ kanetkar

3 DBMS Korth

4 SE Pressman

SQL> delete from library where book = 'C++';

1 row deleted.

SQL> select \* from library\_audit;

ID BOOK AUTHOR SDATE

---------- -------------------- -------------------- ---------

1 C++ kanetkar 10-OCT-17

SQL> update library set book = 'C' where book = 'C++';

0 rows updated.

SQL> update library set book as 'C' where book='C++';

update library set book as 'C' where book='C++'

\*

ERROR at line 1:

ORA-00927: missing equal sign

SQL> update library set book = 'C' where book='C++';

0 rows updated.

SQL> select \* from library;

ID BOOK AUTHOR

---------- -------------------- --------------------

3 DBMS Korth

4 SE Pressman

SQL> update library set author= 'Coreman' where book = 'SE';

1 row updated.

SQL> select \* from library\_audit;

ID BOOK AUTHOR SDATE

---------- -------------------- -------------------- ---------

1 C++ kanetkar 10-OCT-17

4 SE Pressman 10-OCT-17

/\*Assignment No : 06

Write a PL/SQL block of code using cursor that will merge the data available in newly created table N\_RollCall with the data available in the O\_RollCall. If the data in the first table already exists in the second table then that data should be skipped.\*/

/\*Step 1: Note: First create table O\_RollCall(rollno int,name varchar2(10)) & N\_RollCall

Step 2: Write cursor in SQL developer & run \*/

DECLARE

CURSOR ROLL\_CALL\_UPDATE IS

SELECT rollno,name from N\_RollCall;

temp O\_RollCall.rollno%TYPE := NULL;

begin

FOR myrecord IN ROLL\_CALL\_UPDATE LOOP

select count(\*) into temp from O\_RollCall where rollno = myrecord.rollno;

dbms\_output.put\_line('test');

if temp = 0 then

dbms\_output.put\_line('NOT Updated - If NOT Found');

INSERT INTO O\_RollCall values myrecord;

end if;

END LOOP;

COMMIT;

end;