10. PL/SQL BLOCK FOR TRANSACTION OPERATIONS OF A LIBRARY USING PACKAGE

AIM

To write a PL / SQL block for transaction operations of a library application using package.

ALGORITHM

Step – I : Create tables lib_master and lib_trans as follows:

```
SQL> create table lib_master (accno number(4), bookname
     varchar2(30),   author   varchar2(30),   publisher
     varchar2(30), status date, constraint pk_ano primary
     key(accno));
SQL> create table lib_trans (tr_dt date, accno number(4),
     tr_md char(1), constraint fk_ano foreign key(accno)
     references lib_master(accno) on delete cascade);
```

Step – II : Insert records in the lib_master table by using the Insert command as given below:

```
SQL> insert into lib_master (accno, bookname, author,
     publisher) values (&accno, '&bookname', '&author',
     '&publisher');
```

Step – III : Type ed in the SQL Editor. Key in the package specification and save it in a file (say E:\SKS\MC9217\Program_10\Pkg_lib.txt).

Step – IV : The package specification declares two subprograms issue and return and a variable sts of lib_master.status type. Execute the package specification as shown below:

```
SQL> @ E:\SKS\MC9217\Program_10\Pkg_lib.txt
```

Step – V : In the notepad window, define the package body and save it in a file (say E:\SKS\MC1657\Program_10\Pkgbody_lib.txt).

Step – VI : The package body defines the two subprograms issue and return with formal variable of lib_master.accno type.

i) The issue procedure checks the status of a particular book. If status is null, the procedure updates the status in the lib_master table to today's date and inserts a new record in the lib_trans table as a result of the issue transaction.

ii) The return procedure checks the status of a particular book. If status is null, the procedure updates the status value in the lib_master table to null and inserts a new record in the lib_trans table as a result of the return transaction. It also calculates the fine amount if the return date is more than 10 days of the issue date.

Execute the package body as shown below:

```
SQL> @ E:\SKS\MC9217\Program_10\Pkgbody_lib.txt
```

Step – VII : The PL / SQL procedure (E:\SKS\MC1657\Program_10\PLSQL_lib.txt) invokes the subprogram defined in the package body. Execute the PL / SQL procedure as shown below:

```
SQL> @ E:\SKS\MC9217\Program_10\PLSQL_lib.txt
```

Step – VIII: Input the Access No. and Transaction Mode values to execute a transaction.

Step – IX : Execute the following command for the dbms_output.put_line (a function to display a line of text) used in the PL / SQL procedure to take effect.

```
SQL> set serveroutput on
```

PL / SQL PROCEDURE

Pkg_lib.txt

```
create or replace package lib_pack is
 procedure issue(a lib_master.accno%type);
 procedure return(a lib_master.accno%type);
 sts lib_master.status%type;
end lib_pack;
/
```

Pkgbody_lib.txt

```
create or replace package body lib_pack as
 procedure issue(a lib_master.accno%type) is
 begin
  select status into sts from lib_master where accno = a;
  if sts is null then
   update lib_master set status = sysdate where accno = a;
   insert into lib_trans values(sysdate, a, 'I');
   dbms_output.put_line('Book issued successfully...');
  else
   dbms_output.put_line('Book already issued to someone...');
  end if;
 end issue;
 procedure return(a lib_master.accno%type) is
 begin
  select status into sts from lib_master where accno = a;
  if sts is null then
   dbms_output.put_line('Book is in library...');
  else
   if sysdate - sts > 10 then
    dbms_output.put_line('Collect      Fine      :      '      ||
    to_char(round((sysdate - sts)*2,2)));
   end if;
   update lib_master set status = null where accno = a;
   insert into lib_trans values(sysdate, a, 'R');
   dbms_output.put_line('Book returned sucessfully...');
  end if;
 end return;
end lib_pack;
/
```

PLSQL_lib.txt

```
declare
 a lib_master.accno%type;
 m char(1);
begin
 a := &access_no;
 m := '&mode';
 if upper(m) = 'I' then
  lib_pack.issue(a);
 elsif upper(m) = 'R' then
  lib_pack.return(a);
 else
  dbms_output.put_line('Transaction  mode  should  be  either  I  or
   R');
  end if;
end;
/
```

OUTPUT

```
SQL> @ E:\SKS\MC9217\Program_10\PLSQL_lib.txt

      Enter value for access_no: 1234
      old   5:  a := &access_no;
      new   5:  a := 1234;
      Enter value for mode: i
      old   6:  m := '&mode';
      new   6:  m := 'i';
      Book issued successfully...

      PL/SQL procedure successfully completed.

SQL> /

      Enter value for access_no: 1234
      old   5:  a := &access_no;
      new   5:  a := 1234;
      Enter value for mode: i
      old   6:  m := '&mode';
      new   6:  m := 'r';
      Book already issued to someone...

      PL/SQL procedure successfully completed.

SQL> /

      Enter value for access_no: 1234
      old   5:  a := &access_no;
      new   5:  a := 1234;
      Enter value for mode: r
      old   6:  m := '&mode';
      new   6:  m := 'r';
      Book returned sucessfully...

      PL/SQL procedure successfully completed.

SQL> /

      Enter value for access_no: 1234
      old   5:  a := &access_no;
      new   5:  a := 1234;
      Enter value for mode: r
      old   6:  m := '&mode';
      new   6:  m := 'r';
      Book is in library...

      PL/SQL procedure successfully completed.

SQL> /

      Enter value for access_no: 1234
      old   5:  a := &access_no;
      new   5:  a := 1234;
      Enter value for mode: g
      old   6:  m := '&mode';
      new   6:  m := 'g';
      Transaction mode should be either I or R

      PL/SQL procedure successfully completed.
```

RESULT: Thus, the above program was executed successfully.