Assignment 4

Q1) Consider table Stud(Roll, Att,Status)
Write a PL/SQL block for following requirement and handle the exceptions.
Roll no. of student will be entered by user. Attendance of roll no. entered by user will be checked in Stud table. If attendance is less than 75% then display the message "Term not granted" and set the status in stud table as "D". Otherwise display message "Term granted" and set the status in stud table as "ND"

```
SQL> create table stud176(rollno int, att int,status varchar(50));

Table created.

SQL> insert into stud176 values(1,26,'');

1 row created.

SQL> insert into stud176 values(2,65,'');

1 row created.

SQL> insert into stud176 values(3,95,'');

1 row created.

SQL> insert into stud176 values(4,75,'');

1 row created.

SQL> insert into stud176 values(5,87,'');

1 row created.

SQL> declare
  2  2  mroll int;
  3  /
SQL> declare
  2  mroll int;
  3  matt int;
  4  begin
  5  mroll:=&mroll;
  6  select att into matt from stud176 where rollno=mroll;
  7  if matt<75 then
  8  dbms_output.put_line(mroll || ' Term not granted');
  9  update stud176 set status='D' where rollno=mroll;
 10  else
 11  dbms_output.put_line(mroll || ' Term granted');
 12  update stud176 set status='ND' where rollno=mroll;
 13  end if;
 14  Exception
 15  when no_data_found then
```

```
 16  dbms_output.put_line(mroll||'Not found');
 17  end;
 18  /
Enter value for mroll: 7
old   5: mroll:=&mroll;
new   5: mroll:=7;

PL/SQL procedure successfully completed.

SQL> select * from stud176;

    ROLLNO        ATT STATUS
---------- ---------- --------------------------------------------------
         1         26
         2         65
         3         95
         4         75
         5         87

SQL> declare
  2  mroll int;
  3  matt int;
  4  begin
  5  mroll:=&mroll;
  6  select att into matt from stud176 where rollno=mroll;
  7  if matt<75 then
  8  dbms_output.put_line(mroll || ' Term not granted');
  9  update stud176 set status='D' where rollno=mroll;
 10  else
 11  dbms_output.put_line(mroll || ' Term granted');
 12  update stud176 set status='ND' where rollno=mroll;
 13  end if;
 14  Exception
 15  when no_data_found then
 16  dbms_output.put_line(mroll||'Not found');
 17  end;
 18  /
Enter value for mroll: 4
old   5: mroll:=&mroll;
new   5: mroll:=4;

PL/SQL procedure successfully completed.

SQL> select* from stud176;

    ROLLNO        ATT STATUS
---------- ---------- --------------------------------------------------
         1         26
         2         65
         3         95
         4         75 ND
         5         87
```

Q2) Write a PL/SQL block for following requirement using user defined exception handling.
The account_master table records the current balance for an account, which is updated whenever, any deposits or withdrawals takes place. If the withdrawal attempted is more than the current balance held in the account. The user defined exception is raised, displaying an appropriate message. Write a PL/SQL block for above requirement using user defined exception handling.

```
SQL> create table accmtr(bal int,accno int);

Table created.

SQL> insert int accmtr values(52000,21);
insert int accmtr values(52000,21)
       *
ERROR at line 1:
ORA-00925: missing INTO keyword


SQL> insert into accmtr values(52000,21);

1 row created.

SQL> insert into accmtr values(530000,22);

1 row created.

SQL> insert into accmtr values(80000,23);

1 row created.

SQL> insert into accmtr values(165000,24);

1 row created.


SQL> declare
  2  baln int;
  3  acno int;
  4  ch int;
  5  trns int;
  6  insufficient exception;
  7  begin
  8  dbms_output.put_line('Enter the Acc no. : ');
  9  acno:=&acno;
 10  trns:=&trns;
 11  select bal into baln from accmtr where accno = acno;
 12  ch:=&ch;
 13  if ch=1 then
 14  update accmtr set bal=bal+trns where accno=acno;
 15  else
 16  if (trns<=baln) then
 17  update accmtr set bal=(bal-trns) where accno=acno;
```

```
 18  else
 19  raise insufficient;
 20  end if;
 21  end if;
 22  Exception
 23  When insufficient then
 24  dbms_output.put_line('Sufficint balance is not available in account');
 25  end;
 26  /
Enter value for acno: 21
old   9: acno:=&acno;
new   9: acno:=21;
Enter value for trns: 25000
old  10: trns:=&trns;
new  10: trns:=25000;
Enter value for ch: 2
old  12: ch:=&ch;
new  12: ch:=2;

PL/SQL procedure successfully completed.

SQL> select * from accmtr;

      BAL      ACCNO
---------- ----------
    27000         21
   530000         22
    80000         23
   165000         24
```

Q3

```
SQL> create table Book_Borrower_176(Roll_no number, name varchar(20), Dateofiss date, Name_of_Book
varchar(20), status varchar(10));
Table created.
SQL> create table fine_176(roll number, Date_of_Return date, Amount number);
Table created.
SQL> insert into Book_Borrower_176 values(1, 'Ayush', '16-Apr-2023', 'DBMS', 'I');
1 row created.
SQL> insert into Book_Borrower_176 values(2, 'Pratik', '15-Mar-2023', 'CN', 'I');
1 row created.
SQL> insert into Book_Borrower_176values(3, 'Yash', '20-Feb-2023', 'MA', 'I');
1 row created.
SQL> insert into Book_Borrower_176 values(4, 'Ritesh', '10-Apr-2023', 'SE', 'I');
1 row created.
SQL> select * from Book_Borrower_176;
  ROLL_NO NAME                 DATEOFISS NAME_OF_BOOK        STATUS
---------- -------------------- --------- -------------------- ----------
         1 Ayush                16-APR-23 DBMS                 I
         2 Pratik               15-MAR-23 CN                   I
         3 Yash                 20-FEB-23 MA                   I
         4 Ritesh               10-APR-23 SE                   I
```

```
SQL> Declare
  2  mroll number;
  3  mdateofissue date;
  4  sdate date;
  5  noofdays int;
  6  fine number;
  7  Begin
  8  mroll:=&mroll;
  9  select dateofiss into mdateofissue from Book_Borrower_176 where Roll_no=mroll;
 10  select sysdate into sdate from dual;
 11  noofdays:=to_date(sdate)-to_date(mdateofissue);
 12  if noofdays>15 and noofdays<30 then
 13  fine:=(noofdays-15)*5;
 14  insert into fine_176 values(mroll, sdate, fine);
 15  update Book_Borrower_176 set status='R' where Roll_no=mroll;
 16  elsif noofdays>30 then
 17  fine:=(noofdays-15)*50;
 18  insert into fine_176 values(mroll, current_date, fine);
 19  update Book_Borrower_176 set status='R' where Roll_no=mroll;
 20  else
 21  update Book_Borrower_176 set status='R' where Roll_no=mroll;
 22  end if;
 23  Exception
 24  when no_data_found then
 25  dbms_output.put_line('Record Not Found');
 26  End;
 27  /
Enter value for mroll: 1
old   8: mroll:=&mroll;
new   8: mroll:=1;

PL/SQL procedure successfully completed.

SQL> Declare
  2  mroll number;
  3  mdateofissue date;
  4  sdate date;
  5  noofdays int;
6  fine number;
  7  Begin
  8  mroll:=&mroll;
  9  select dateofiss into mdateofissue from Book_Borrower_176 where Roll_no=mroll;
 10  select sysdate into sdate from dual;
 11  noofdays:=to_date(sdate)-to_date(mdateofissue);
 12  if noofdays>15 and noofdays<30 then
 13  fine:=(noofdays-15)*5;
 14  insert into fine_176 values(mroll, sdate, fine);
 15  update Book_Borrower_176 set status='R' where Roll_no=mroll;
 16  elsif noofdays>30 then
 17  fine:=(noofdays-15)*50;
 18  insert into fine_176 values(mroll, current_date, fine);
 19  update Book_Borrower_176 set status='R' where Roll_no=mroll;
```

```
 20  else
 21  update Book_Borrower_176 set status='R' where Roll_no=mroll;
 22  end if;
 23  Exception
 24  when no_data_found then
 25  dbms_output.put_line('Record Not Found');
 26  End;
 27  /
Enter value for mroll: 2
old   8: mroll:=&mroll;
new   8: mroll:=2;

SQL> Declare
  2  mroll number;
  3  mdateofissue date;
  4  sdate date;
  5  noofdays int;
  6  fine number;
  7  Begin
  8  mroll:=&mroll;
  9  select dateofiss into mdateofissue from Book_Borrower_176 where Roll_no=mroll;
 10  select sysdate into sdate from dual;
 11  noofdays:=to_date(sdate)-to_date(mdateofissue);
 12  if noofdays>15 and noofdays<30 then
 13  fine:=(noofdays-15)*5;
 14  insert into fine_176 values(mroll, sdate, fine);
 15  update Book_Borrower_176 set status='R' where Roll_no=mroll;
 16  elsif noofdays>30 then
 17  fine:=(noofdays-15)*50;
 18  insert into fine_176 values(mroll, current_date, fine);
 19  update Book_Borrower_176 set status='R' where Roll_no=mroll;
 20  else
 21  update Book_Borrower_176 set status='R' where Roll_no=mroll;
 22  end if;
 23  Exception
 24  when no_data_found then
 25  dbms_output.put_line('Record Not Found');
 26  End;
 27  /
Enter value for mroll: 5
old   8: mroll:=&mroll;
new   8: mroll:=5;
Record Not Found

PL/SQL procedure successfully completed.

SQL> select * from fine_176;

     ROLL DATE_OF_R     AMOUNT
---------- --------- ----------
         2 25-APR-23       1300
         3 25-APR-23       2450
```