Name : Mayuresh Rajendrraa Deshmukh

Class : TE A

Roll No : TECOA124

Batch : A2

Assignment No. 4

Title: PL/SQL Block and Exception Handling

1. 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> select * from stud;

     ROLL       ATT ST
---------- ---------- --
        1         90
        2         75
        3         66
        4      82.55
        5      70.89
```

```
  SQL> set serveroutput on
  SQL> declare
    2    mroll number(2);
    3    matt number(4,2);
    4  begin
    5    mroll:=&mroll;
    6    select att into matt from stud where roll=mroll;
    7    if matt<75 then
    8            update stud set status='D' where roll=mroll;
    9            dbms_output.put_line('term not granted');
   10     else
   11             update stud set status='ND' where roll=mroll;
   12             dbms_output.put_line('term granted');
   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: 2
  old   5:        mroll:=&mroll;
  new   5:        mroll:=2;
  term granted

  PL/SQL procedure successfully completed.

  SQL> /
  Enter value for mroll: 5
  old   5:        mroll:=&mroll;
  new   5:        mroll:=5;
  term not granted

  PL/SQL procedure successfully completed.

  SQL> /
  Enter value for mroll: 6
  old   5:        mroll:=&mroll;
  new   5:        mroll:=6;
  6 not found


Enter value for mroll: 1
old   5:        mroll:=&mroll;
new   5:        mroll:=1;
term granted

PL/SQL procedure successfully completed.


SQL> select * from stud;

      ROLL        ATT ST
---------- ---------- --
         1         90 ND
         2         75 ND
         3         66
         4      82.55
         5      70.89 D
```

---

2. **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> select * from account_master;

       ACCNO    BALANCE
  ---------- ----------
         101      50000
         102      10000
         103       2000
         104       5000
         105      75001

  SQL> set serveroutput on
  SQL> declare
  2    maccno number(3);
  3    withdraw number(10,2);
  4    macbal number(10,2);
  5    less_bal exception;
  6
  7  begin
  8    maccno:=&maccno;
  9    select balance into macbal from account_master where accno=maccno;
 10     withdraw:=&withdraw;
 11     if macbal<withdraw then
 12             raise less_bal;
 13     else
 14             macbal:=macbal-withdraw;
 15             update account_master set balance=macbal where accno=maccno;
 16             dbms_output.put_line('Money withdrawn');
 17     end if;
 18  exception
 19     when less_bal then
 20             dbms_output.put_line('Insufficient balance');
 21  end;
 22  /
Enter value for maccno: 101
old   8:        maccno:=&maccno;
new   8:        maccno:=101;
Enter value for withdraw: 2000
old  10:        withdraw:=&withdraw;
new  10:        withdraw:=2000;
Money withdrawn

PL/SQL procedure successfully completed.

SQL> /
Enter value for maccno: 103
old   8:        maccno:=&maccno;
new   8:        maccno:=103;
Enter value for withdraw: 5000
old  10:        withdraw:=&withdraw;
new  10:        withdraw:=5000;
Insufficient balance

PL/SQL procedure successfully completed.

SQL> select * from account_master;

     ACCNO    BALANCE
---------- ----------
       101      48000
       102      10000
       103       2000
       104       5000
       105      75001
```

3.Write an SQL code block these raise a user defined exception where business rule is voilated. BR for client_master table specifies when the value of bal_due field is less than 0 handle the exception.

```
SQL> select * from client_master;

      ACNO    BAL_DUE
---------- ----------
       101        200
       102         -1
       103        999
       104        -12

SQL> declare
  2     macc number(3);
  3     mbal number(3);
  4     br exception;
  5  begin
  6     macc:=&macc;
  7     select bal_due into mbal from client_master where acno=macc;
  8     if mbal<0 then
  9             raise br;
 10     else
 11             dbms_output.put_line('Balance due OK');
 12     end if;
 13  exception
 14     when br then
 15             dbms_output.put_line('balance due invalid');
 16     when no_data_found then
 17             dbms_output.put_line(macc||' not found');
 18
 19  end;
 20  /
Enter value for macc: 101
old   6:        macc:=&macc;
new   6:        macc:=101;
Balance due OK

PL/SQL procedure successfully completed.

SQL> /
Enter value for macc: 102
old   6:        macc:=&macc;
new   6:        macc:=102;
balance due invalid

PL/SQL procedure successfully completed.

SQL> /
Enter value for macc: 103
old   6:        macc:=&macc;
new   6:        macc:=103;
Balance due OK

PL/SQL procedure successfully completed.

SQL> /
Enter value for macc: 104
old   6:        macc:=&macc;
new   6:        macc:=104;
balance due invalid
```

```
PL/SQL procedure successfully completed.

SQL> /
Enter value for macc: 105
old   6:        macc:=&macc;
new   6:        macc:=105;
105 not found

PL/SQL procedure successfully completed.
```

4.

1. Borrower(Roll_no, Name, DateofIssue, NameofBook, Status)

2. Fine(Roll_no,Date,Amt)

- **Accept roll_no & name of book from user.**
- **Check the number of days (from date of issue), if days are between 15 to 30 then fineamount will be Rs 5per day.**
- **If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 perday.**
- **After submitting the book, status will change from I to R.**
- **If condition of fine is true, then details will be stored into fine table.**

Also handles the exception by named exception handler or user define exception handler.

```
SQL> select * from borrower;

      ROLL NAME           DOI       BOOK                 S
---------- -------------- --------- -------------------- -
       101 ashwin         03-AUG-19 toc                  I
       102 hemangi        05-SEP-19 mis                  I
       103 rutuj          20-AUG-19 CN                   I

SQL> select * from fine;

no rows selected

SQL> set serveroutput on
SQL> declare
  2    mroll number(3);
  3    nmbk varchar2(20);
  4    mdoi date;
  5    days number(3);
  6    mfine number(3);
  7
  8  begin
  9    mroll:=&mroll;
 10
 11     select doi into mdoi from borrower where roll=mroll;
 12     days:=sysdate-mdoi;
 13     if days>=15 and days<=30 then
 14            mfine:=days*5;
 15             insert into fine values(mroll,mfine);
 16             update borrower set status='R' where roll=mroll;
 17     elsif days>30 then
 18             mfine:=150+(days-30)*50;
 19             insert into fine values(mroll,mfine);
 20             update borrower set status='R' where roll=mroll;
 21     else
 22             update borrower set status='R' where roll=mroll;
 23     end if;
 24  exception
 25     when no_data_found then
 26             dbms_output.put_line(mroll||' not found');
 27  end;
 28  /
Enter value for mroll: 101
old   9:        mroll:=&mroll;
new   9:        mroll:=101;

PL/SQL procedure successfully completed.

SQL> /
Enter value for mroll: 102
old   9:        mroll:=&mroll;
new   9:        mroll:=102;

PL/SQL procedure successfully completed.

SQL> /
Enter value for mroll: 103
old   9:        mroll:=&mroll;
new   9:        mroll:=103;

PL/SQL procedure successfully completed.

SQL> /014
Enter value for mroll: 104
old   9:        mroll:=&mroll;
new   9:        mroll:=104;
104 not found

PL/SQL procedure successfully completed.

SQL> select * from borrower;

      ROLL NAME            DOI       BOOK                 S
---------- --------------- --------- -------------------- -
       101 ashwin          03-AUG-19 toc                  R
       102 hemangi         05-SEP-19 mis                  R
       103 rutuj           20-AUG-19 CN                   R

SQL> select * from fine;

      ROLL        AMT
---------- ----------
       101        800
       103        130
```