 # SYCOA302 Ritish Shelke

Problem Statement:

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”

PL/SQL BLOCK

```
declare
roll_no number(5);
att number(3);

begin
roll_no:=&roll_no;
select attendance into att from stud where roll=roll_no;

IF att<75 THEN
        dbms_output.put_line('Term Not Granted');
        update stud set status='D' where roll=roll_no;
ELSE
        dbms_output.put_line('Term Granted');
        update stud set status='ND' where roll=roll_no;
END IF;

Exception
WHEN no_data_found THEN
dbms_output.put_line(roll_no||'Student not found');
End;
```

```
SQL> declare
  2  roll_no number(5);
  3  att number(3);
  4
  5  begin
  6  roll_no:=&roll_no;
  7  select attendance into att from stud where roll=roll_no;
  8
  9  IF att<75 THEN
 10     dbms_output.put_line('Term Not Granted');
 11     update stud set status='D' where roll=roll_no;
 12  ELSE
 13     dbms_output.put_line('Term Granted');
 14     update stud set status='ND' where roll=roll_no;
 15  END IF;
 16
 17  Exception
 18  WHEN no_data_found THEN
 19  dbms_output.put_line(roll_no||'Student not found');
 20  End;
 21  /
Enter value for roll_no: 101
old   6: roll_no:=&roll_no;
new   6: roll_no:=101;
Term Granted
```

```
SQL> select * from stud;

      ROLL ATTENDANCE STATU
---------- ---------- -----
       101         90
       102         74
       103         80
       104         50
```

```
SQL> select * from stud;

      ROLL ATTENDANCE STATU
---------- ---------- -----
       101         90 ND
       102         74 D
       103         80
       104         50
```

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.

PL/SQL BLOCK

```
declare
macc_no number(5);
mbalance float;
mamount float;
insuffiecient_balance exception;

begin
macc_no:=&macc_no;
mamount:=&mamount;
select balance into mbalance from account_master where acc_no=macc_no;

if (mamount<=mbalance) then
        Dbms_output.put_line('Withdraw Success | Balance updated');
        update account_master set balance=(balance-mamount) where
acc_no=macc_no;
else
        raise insuffiecient_balance;
end if;

exception
when insuffiecient_balance then
dbms_output.put_line('Withdraw failed,Insuffiecient balance');
when no_data_found then
dbms_output.put_line('Account does not exist');


end;
```

```
SQL> declare
  2  macc_no number(5);
  3  mbalance float;
  4  mamount float;
  5  insuffiecient_balance exception;
  6
  7  begin
  8  macc_no:=&macc_no;
  9  mamount:=&mamount;
 10  select balance into mbalance from account_master where acc_no=macc_no;
 11
 12  if (mamount<=mbalance) then
 13     Dbms_output.put_line('Withdraw Success | Balance updated');
 14     update account_master set balance=(balance-mamount) where acc_no=macc_no;
 15  else
 16     raise insuffiecient_balance;
 17  end if;
 18
 19  exception
 20  when insuffiecient_balance then
 21  dbms_output.put_line('Withdraw failed,Insuffiecient balance');
 22  when no_data_found then
 23  dbms_output.put_line('Account does not exist');
 24
 25
 26  end;
 27  /
Enter value for macc_no: 202
old   8: macc_no:=&macc_no;
new   8: macc_no:=202;
Enter value for mamount: 1600
old   9: mamount:=&mamount;
new   9: mamount:=1600;
Withdraw Success | Balance updated

PL/SQL procedure successfully completed.
```

```
SQL> select * from account_master;

    ACC_NO    BALANCE
---------- ----------
       201         89
       202      11600
       203        100
       204        500
       205        200
```

```
SQL> select * from account_master
  2  ;

    ACC_NO    BALANCE
---------- ----------
       201         89
       202      10000
       203        100
       204        500
       205        200
```

3. 1. Borrower(Roll_no, Name, Date_of_Issue, Name_of_Book, 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 fine amount will be Rs 5per day.

 If no. of days>30, fine will be Rs 50 per day & for days less than 30, Rs. 5 per day.

 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.

PL/SQL BLOCK

```
declare
mroll number(5);
name_of_book varchar(50);
issue_date date;
no_of_days number(5);
today_date date;
fine_amt number(5);

begin
-- step 1 : accept input
mroll:=&mroll;
name_of_book:='&name_of_book';

select date_of_issue into issue_date from borrower where roll_no=mroll and
book_name=name_of_book;
dbms_output.put_line('issue date is  '||issue_date);

select sysdate into today_date from dual;
dbms_output.put_line('Today date is  '||today_date);

no_of_days:=trunc(today_date)-issue_date;
dbms_output.put_line('No of days is  '||no_of_days);

-- step 2 calculate fine amount
```

```
   IF no_of_days > 30 THEN
      fine_amt := 75 + 50 * (no_of_days - 30);
   ELSIF no_of_days > 15 THEN
      fine_amt := 5 * (no_of_days - 15);
   ELSE
      fine_amt := 0;
   END IF;
dbms_output.put_line('Fine  '||fine_amt);
-- step 3 change status
update borrower set status='R' where roll_no=mroll;

-- step 4  if fine applicable insert into fine table
IF fine_amt > 0 THEN
        insert into fine (roll_no,f_date,amt) values (mroll,today_date,fine_amt);
END IF;
end;
```

```
Enter value for mroll: 302
old  11: mroll:=&mroll;
new  11: mroll:=302;
Enter value for name_of_book: REACT.JS
old  12: name_of_book:='&name_of_book';
new  12: name_of_book:='REACT.JS';
issue date is  01-MAR-23
Today date is  28-APR-23
No of days is  58
Fine  1475

PL/SQL procedure successfully completed.
```

ROLL_NO	NAME	DATE_OF_ISSUE	BOOK_NAME	STATUS
302	Ritish Shelke	14-APR-23	DSA	R
276	Saurabh Asnare	10-APR-23	JAVA	R
288	Pranav Kulkarni	01-APR-23	PYTHON	I
302	Ritish Shelke	01-MAR-23	REACT.JS	R

4 rows returned in 0.00 seconds CSV Export

ROLL_NO	F_DATE	AMT
276	28-APR-23	15
302	28-APR-23	1475

2 rows returned in 0.00 seconds CSV Export