**Name :** Rohit Bangar **Class :** TE-C **Roll No :** TECOC306 **Batch :** C1

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

**Assignment-5**

**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> create table studd(roll int , att int , status varchar(10));**

Table created.

**SQL> select \* from studd;**

ROLL ATT STATUS

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

1 80 -

2 93 -

3 75 -

4 72 -

5 87 -

6 67 -

**SQL> set serveroutput on;**

**SQL> declare**

**mroll number(10);**

**matt number(10);**

**begin**

**mroll := &mroll;**

**select att into matt from studd where roll = mroll;**

**if matt<75 then**

**dbms\_output.put\_line(mroll || ' Is detained');**

**update studd set status = 'D' where roll = matt;**

**else**

**dbms\_output.put\_line(mroll || ' Is not Detained');**

**update studd set status = 'ND' where roll = matt;**

**end if;**

**exception**

**when no\_data\_found then**

**dbms\_output.put\_line(mroll || ' not found');**

**end;**

**/**

Enter value for mroll: 4

old 5: mroll := &mroll;

new 5: mroll := 4;

4 Is detained

**PL/SQL procedure successfully completed**.

**SQL> select \* from studd;**

ROLL ATT STATUS

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

1 80 -

2 93 -

3 75 -

4 72 D

5 87 -

6 67 -

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2] 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**

->

**declare**

**bal\_notavail exception;**

**acc\_no number;**

**amount number;**

**op number;**

**baln number;**

**begin**

**acc\_no:=&acc\_no;**

**amount:=&amount;**

**dbms\_output.put\_line('Enter 1 : DEPOSIT');**

**dbms\_output.put\_line('Enter 2 : WITHDRAW');**

**op:=&op;**

**select bal into baln from acc\_master where acc\_no=acc\_num;**

**if op=1 then**

**update acc\_master set bal=bal+amount where acc\_no=acc\_num;**

**dbms\_output.put\_line(amount ||': DEPOSITED');**

**elsif op=2 and amount>baln then**

**raise bal\_notavail;**

**else**

**update acc\_master set bal=bal-amount where acc\_no=acc\_num;**

**dbms\_output.put\_line(amount ||': WITHDRAWN');**

**end if;**

**exception**

**when bal\_notavail then**

**dbms\_output.put\_line('Balance NOT available to withdraw');**

**end;**

==============================================================================

**SQL> select \* from acc\_master;**

ACC\_NUM BAL

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

101 1000

102 5500

103 13000

Enter value for acc\_no: 101

old 8: acc\_no:=&acc\_no;

new 8: acc\_no:=101;

Enter value for amount: 2500

old 9: amount:=&amount;

new 9: amount:=2500;

Enter value for op: 2

old 12: op:=&op;

new 12: op:=2;

Enter 1 : DEPOSIT

Enter 2 : WITHDRAW

**Balance NOT available to withdraw**

**PL/SQL procedure successfully completed.**

Enter value for acc\_no: 101

old 8: acc\_no:=&acc\_no;

new 8: acc\_no:=101;

Enter value for amount: 500

old 9: amount:=&amount;

new 9: amount:=500;

Enter value for op: 2

old 12: op:=&op;

new 12: op:=2;

Enter 1 : DEPOSIT

Enter 2 : WITHDRAW

500: WITHDRAWN

**PL/SQL procedure successfully completed**.

Enter value for acc\_no: 102

old 8: acc\_no:=&acc\_no;

new 8: acc\_no:=102;

Enter value for amount: 600

old 9: amount:=&amount;

new 9: amount:=600;

Enter value for op: 1

old 12: op:=&op;

new 12: op:=1;

Enter 1 : DEPOSIT

Enter 2 : WITHDRAW

600: DEPOSITED

**PL/SQL procedure successfully completed.**

**SQL> select \* from acc\_master;**

ACC\_NUM BAL

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

101 500

102 6100

103 13000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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.**

->

**declare**

**br exception;**

**c\_no number;**

**bal number;**

**begin**

**c\_no:=&c\_no;**

**select bal\_due into bal from client\_master where c\_no=cli\_num;**

**if bal<0 then**

**raise br;**

**else**

**dbms\_output.put\_line('Client\_no : '||c\_no||' Balance : '||bal);**

**end if;**

**exception**

**when br then**

**dbms\_output.put\_line('BR violated by client\_no : '||c\_no);**

**end;**

=============================================================

**SQL> select \* from client\_master;**

CLI\_NUM BAL\_DUE

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

701 5000

702 -5

703 0

Enter value for c\_no: 703

old 6: c\_no:=&c\_no;

new 6: c\_no:=703;

Client\_no : 703 Balance : 0

**PL/SQL procedure successfully completed**.

Enter value for c\_no: 702

old 6: c\_no:=&c\_no;

new 6: c\_no:=702;

BR violated by client\_no : 702

**PL/SQL procedure successfully completed.**

Enter value for c\_no: 701

old 6: c\_no:=&c\_no;

new 6: c\_no:=701;

Client\_no : 701 Balance : 5000

**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 fine amount 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.**

->

**declare**

**neg\_days exception;**

**rn number;**

**bname varchar(20);**

**is\_date date;**

**diff int:=0;**

**fineamt int:=0;**

**begin**

**rn:=&rn;**

**bname:=&bname;**

**select i\_date into is\_date from borro where rn=roll and bname=book\_name;**

**diff:=sysdate()-is\_date;**

**if diff <0 then**

**raise neg\_days;**

**end if;**

**if (diff>15 and diff<30) then**

**fineamt:=diff\*5;**

**dbms\_output.put\_line('FINE CALCULATED FOR '||rn||': '||fineamt);**

**elsif diff>30 then**

**fineamt:=(diff-30)\*50+(15\*5);**

**dbms\_output.put\_line('FINE CALCULATED FOR '||rn||': '||fineamt);**

**else**

**dbms\_output.put\_line('NO FINE FOR : '||rn);**

**end if;**

**update borro set status='R' where rn=roll;**

**insert into fine values(rn,sysdate(),fineamt);**

**exception**

**when neg\_days then**

**dbms\_output.put\_line('NEGATIVE date');**

**when no\_data\_found then**

**dbms\_output.put\_line('ENTERED DATA NOT FOUND');**

**end;**

================================================================================================

**SQL> select \* from borro;**

ROLL NAME I\_DATE BOOK\_NAME STATUS

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

1 ABC 12-JUL-19 TOC -

2 DEF 20-JUL-19 CN -

3 XYZ 20-JUN-19 DBMS -

Enter value for rn: 3

old 9: rn:=&rn;

new 9: rn:=3;

Enter value for bname: 'DBMS'

old 10: bname:=&bname;

new 10: bname:='DBMS';

FINE CALCULATED FOR 3: 825

**PL/SQL procedure successfully completed.**

**SQL> select \* from borro;**

ROLL NAME I\_DATE BOOK\_NAME STATUS

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

1 ABC 12-JUL-19 TOC -

2 DEF 20-JUL-19 CN -

3 XYZ 20-JUN-19 DBMS R

**SQL> select \* from fine;**

ROLL RET\_DATE AMT

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

3 03-AUG-19 825

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter value for rn: 1

old 9: rn:=&rn;

new 9: rn:=1;

Enter value for bname: 'TOC'

old 10: bname:=&bname;

new 10: bname:='TOC';

FINE CALCULATED FOR 1: 115

**PL/SQL procedure successfully completed.**

**SQL> select \* from borro;**

ROLL NAME I\_DATE BOOK\_NAME STATUS

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

1 ABC 12-JUL-19 TOC R

2 DEF 20-JUL-19 CN -

3 XYZ 20-JUN-19 DBMS R

**SQL> select \* from fine;**

ROLL RET\_DATE AMT

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

3 03-AUG-19 825

1 03-AUG-19 115

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter value for rn: 2

old 9: rn:=&rn;

new 9: rn:=2;

Enter value for bname: 'CN'

old 10: bname:=&bname;

new 10: bname:='CN';

NO FINE FOR : 2

**PL/SQL procedure successfully completed.**

**SQL> select \* from borro;**

ROLL NAME I\_DATE BOOK\_NAME STATUS

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

1 ABC 12-JUL-19 TOC R

2 DEF 20-JUL-19 CN R

3 XYZ 20-JUN-19 DBMS R

**SQL> select \* from fine;**

ROLL RET\_DATE AMT

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

3 03-AUG-19 825

1 03-AUG-19 115

2 03-AUG-19 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**SQL> insert into borro values(4,'LMN','5-AUG-2019','SEPM',' ');**

1 row created.

Enter value for rn: 4

old 9: rn:=&rn;

new 9: rn:=4;

Enter value for bname: 'SEPM'

old 10: bname:=&bname;

new 10: bname:='SEPM';

NEGATIVE date

**PL/SQL procedure successfully completed.**

**SQL> select \* from borro;**

ROLL NAME I\_DATE BOOK\_NAME STATUS

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

1 ABC 12-JUL-19 TOC R

2 DEF 20-JUL-19 CN R

3 XYZ 20-JUN-19 DBMS R

4 LMN 05-AUG-19 SEPM -

**SQL> select \* from fine;**

ROLL RET\_DATE AMT

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

3 03-AUG-19 825

1 03-AUG-19 115

2 03-AUG-19 0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter value for rn: 5

old 9: rn:=&rn;

new 9: rn:=5;

Enter value for bname: 'iii'

old 10: bname:=&bname;

new 10: bname:='iii';

ENTERED DATA NOT FOUND

**PL/SQL procedure successfully completed.**