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

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

**Assignment** **No.** **7**

**Title:** Stored Procedure Stored Function

**1.** **Write** **a** **PL/SQL** **stored** **Procedure** **for** **following** **requirements** **and** **call** **the** **procedure** **in** **appropriate** **PL/SQL** **block.**

1. **Borrower(Rollin,** **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** **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.**

**SQL> select \* from borrower;**

ROLL NAME DOI BOOKNAME STATUS

---------- ---------- --------- ---------- ----------101 abc 01-JUL-19 abc\_b i

102 pqr 20-JUN-19 pqr\_b i 103 lmn 20-JUL-19 lmn\_b i

**SQL> set serveroutput on; SQL>**

**SQL> create or replace procedure finecalculate(missue in date,mroll in number) is**

**2**

**3 mamt number(10); 4 mdiff number(10); 5**

**6 begin**

**7 mdiff:=sysdate-missue; 8**

**9 if mdiff>15 then 10**

**11 if mdiff<30 then 12**

**13 mamt:=mdiff\*5;**

**14 insert into fine values(mroll,sysdate,mamt); 15 dbms\_output.put\_line('fine is '||mamt);**

**16 update borrower set status='r' where roll=mroll; 17**

**18 else**

**19**

**20 mamt:=30\*5+(mdiff-30)\*50;**

**21 insert into fine values(mroll,sysdate,mamt); 22 dbms\_output.put\_line('fine is '||mamt);**

**23 update borrower set status='r' where roll=mroll; 24**

**25 end if; 26 else**

**27 mamt:=0;**

**28 insert into fine values(mroll,sysdate,mamt); 29 dbms\_output.put\_line('fine is '||mamt);**

**30 update borrower set status='r' where roll=mroll; 31**

**32**

**33 end if; 34 end;**

**35 /**

Procedure created.

**SQL> Declare**

**2 missue1 date;**

**3 mroll1 number(10); 4**

**5 Begin 6**

**7 mroll1:=&mroll1;**

**8 select doi into missue1 from borrower where roll=mroll1; 9 finecalculate(missue1,mroll1);**

**10**

**11 Exception**

**12 when no\_data\_found then**

**13 dbms\_output.put\_line(mroll1||' not found'); 14 End;**

**15 /**

Enter value for mroll1: 101 old 7: mroll1:=&mroll1; new 7: mroll1:=101;

fine is 120

PL/SQL procedure successfully completed.

SQL> /

Enter value for mroll1: 102 old 7: mroll1:=&mroll1; new 7: mroll1:=102;

fine is 400

PL/SQL procedure successfully completed.

SQL> /

Enter value for mroll1: 103

old 7: mroll1:=&mroll1; new 7: mroll1:=103; fine is 0

PL/SQL procedure successfully completed.

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

ROLL DOR AMT ---------- --------- ----------

101 24-JUL-19 120 102 24-JUL-19 400 103 24-JUL-19 0

**SQL> select \* from borrower;**

ROLL NAME DOI BOOKNAME STATUS

---------- ---------- --------- ---------- ----------101 abc 01-JUL-19 abc\_b r

102 pqr 20-JUN-19 pqr\_b r 103 lmn 20-JUL-19 lmn\_b r

2. **Write** **a** **stored** **function** **in** **PL/SQL** **for** **given** **requirement** **and** **use** **the** **same** **in** **PL/SQL** **block.** **Account** **no.** **and** **branch** **name** **will** **be** **accepted** **from** **user.** **The** **same** **will** **be** **searched** **in** **table** **acct\_details.** **If** **status** **of** **account** **is** **active** **then** **display** **appropriate** **message** **and** **also** **store** **the** **account** **details** **in** **active\_acc\_details** **table,** **otherwise** **display** **message** **on** **screen** **“account** **is** **inactive”.**

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

ACCNO BRANCH ST

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

101 akurdi I

102 nigdi A

103 akurdi A

104 akurdi A

105 nigdi I

**SQL> set serveroutput on;**

**SQL> create or replace function check\_account( maccno number,bname varchar)**

**2 return number is 3**

**4 mstatus varchar(2);**

**5**

**6 begin**

**7 select status into mstatus from acc\_details where accno=maccno and branch=bname;**

**8 if mstatus='A' then 9 return 1;**

**10 else**

**11 return 0; 12 end if; 13**

**14 end; 15 /**

Function created.

**SQL> Declare**

**2 macc number(10);**

**3 bname varchar(10);**

**4 f number(2);**

**5 begin**

**6 macc:=&account\_no;**

**7 bname:='&branch\_name';**

**8 f:=check\_account(macc,bname);**

**9 if f=1 then**

**10 dbms\_output.put\_line('Account is active');**

**11 insert into active\_acc\_details values(macc,bname);**

**12 end if;**

**13 if f=0 then**

**14 dbms\_output.put\_line('Account is inactive');**

**15 end if;**

**16 exception**

**17 when no\_data\_found then**

**18 dbms\_output.put\_line('Account no. not found');**

**19 end;**

**20 /**

Enter value for account\_no: 101

old 6: macc:=&account\_no;

new 6: macc:=101;

Enter value for branch\_name: akurdi

old 7: bname:='&branch\_name';

new 7: bname:='akurdi';

**Account is inactive**

PL/SQL procedure successfully completed.

SQL> /

Enter value for account\_no: 102

old 6: macc:=&account\_no;

new 6: macc:=102;

Enter value for branch\_name: nigdi

old 7: bname:='&branch\_name';

new 7: bname:='nigdi';

**Account is active**

PL/SQL procedure successfully completed.

SQL> /

Enter value for account\_no: 103

old 6: macc:=&account\_no;

new 6: macc:=103;

Enter value for branch\_name: akurdi

old 7: bname:='&branch\_name';

new 7: bname:='akurdi';

**Account is active**

PL/SQL procedure successfully completed.

SQL> /

Enter value for account\_no: 104

old 6: macc:=&account\_no;

new 6: macc:=104;

Enter value for branch\_name: nigdi

old 7: bname:='&branch\_name';

new 7: bname:='nigdi';

**Account no. not found**

SQL> /

Enter value for account\_no: 104

old 6: macc:=&account\_no;

new 6: macc:=104;

Enter value for branch\_name: akurdi

old 7: bname:='&branch\_name';

new 7: bname:='akurdi';

**Account is active**

PL/SQL procedure successfully completed.

SQL> /

Enter value for account\_no: 105

old 6: macc:=&account\_no;

new 6: macc:=105;

Enter value for branch\_name: nigdi

old 7: bname:='&branch\_name';

new 7: bname:='nigdi';

**Account is inactive**

PL/SQL procedure successfully completed.

3. **Write** **a** **Stored** **Procedure** **namely** **proc\_Grade** **for** **the** **categorization** **of** **student.** **If** **marks** **scored** **by** **students** **in** **examination** **is** **<=1500** **and** **marks>=990** **then** **student** **will** **be** **placed** **in** **distinction** **category** **if** **marks** **scored** **are** **between** **989** **and900** **category** **is** **first** **class,** **if** **marks**

**899** **and** **825** **category** **is** **Higher** **Second** **Class**

**Write** **a** **PL/SQL** **block** **for** **using** **procedure** **created** **with** **above** **requirement.** **Stud\_Marks(name,** **total\_marks)**

**Result(Roll,Name,** **Class)**

**SQL> select \* from stud\_marks;**

ROLL NAME TOTAL\_MARKS ---------- ---------- -----------

101 xyz 846 102 abc 1000 103 l,n 950 104 ghj 1400

**SQL> set serveroutput on;**

**SQL> create or replace procedure proc\_grade(mroll in number) is 2**

**3 mmarks number(10); 4 mname varchar(10); 5 begin**

**6 select total\_marks into mmarks from stud\_marks where roll=mroll; 7 select name into mname from stud\_marks where roll=mroll;**

**8**

**9 if mmarks<=1500 and mmarks>=990 then**

**10 dbms\_output.put\_line('Status of roll '||mroll||' is distinction.'); 11 insert into result values(mroll,mname,'Dist');**

**12**

**13 elsif mmarks<=989 and mmarks>=900 then**

**14 dbms\_output.put\_line('Status of roll '||mroll||' is first class.'); 15 insert into result values(mroll,mname,'FC');**

**16**

**17 elsif mmarks<=899 and mmarks>=825 then**

**18 dbms\_output.put\_line('Status of roll '||mroll||' is Higher second class.');**

**19 insert into result values(mroll,mname,'HSC'); 20**

**21 end if; 22 end;**

**23 /**

Procedure created.

**SQL> declare**

**2 mroll number(10); 3**

**4 begin**

**5 mroll:=&roll;**

**6 proc\_grade(mroll); 7**

**8 Exception**

**9 when no\_data\_found then**

**10 dbms\_output.put\_line(mroll||' not found'); 11 End;**

**12 /**

Enter value for roll: 101 old 5: mroll:=&roll; new 5: mroll:=101;

**Status of roll 101 is Higher second class**.

PL/SQL procedure successfully completed.

SQL> /

Enter value for roll: 102 old 5: mroll:=&roll; new 5: mroll:=102;

**Status of roll 102 is distinction**.

PL/SQL procedure successfully completed.

SQL> /

Enter value for roll: 103 old 5: mroll:=&roll; new 5: mroll:=103;

**Status of roll 103 is first class.**

PL/SQL procedure successfully completed.

SQL> /

Enter value for roll: 104 old 5: mroll:=&roll; new 5: mroll:=104;

**Status of roll 104 is distinction.**

PL/SQL procedure successfully completed.

SQL> /

Enter value for roll: 105 old 5: mroll:=&roll; new 5: mroll:=105;

**105 not found**

PL/SQL procedure successfully completed.

**SQL> select \* from result;**

ROLL NAME CLASS ---------- ---------- -----

101 xyz HSC 102 abc Dist 103 l,n FC

104 ghj Dist