# Assignment 5

**Q1.** Write a PL/SQL stored Procedure for following requirements and call the procedure in appropriate PL/SQL block.
1. Borrower(Rollno, 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 5 per 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 the condition of fine is true, then details will be stored into the fine table.

**Ans:-**

```
CREATE OR REPLACE PROCEDURE Fine_calculation (roll_no IN NUMBER, name_of_book
IN VARCHAR2)
AS
  days_overdue NUMBER(10) := 0;
  fine_amount NUMBER(10,2);
  date_of_issue DATE;
  status CHAR(1);
BEGIN
  SELECT DateofIssue, Status INTO date_of_issue, status
  FROM Borrower
  WHERE Roll_no =   roll_no AND NameofBook = name_of_book;
  days_overdue := SYSDATE - date_of_issue;
  IF days_overdue <= 15 THEN
     fine_amount := 0;
  ELSIF days_overdue <= 30 THEN
     fine_amount := 5 * days_overdue;
  ELSE
     fine_amount := 50 * days_overdue;
  END IF;
  UPDATE Borrower SET Status = 'R' WHERE Roll_no = roll_no AND
NameofBook=name_of_book;
  IF fine_amount > 0 THEN
      INSERT INTO Fine VALUES (roll_no, SYSDATE, fine_amount);
  END IF;
  DBMS_OUTPUT.PUT_LINE('Fine= ' || fine_amount);
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error: Could not calculate the fine amount for the book');
```

```
END;


DECLARE
  roll_no NUMBER(10);
  name_of_book VARCHAR2(100);
 BEGIN
  roll_no := &roll_no;
  name_of_book := '&name_of_book';
  Fine_calculation(roll_no,name_of_book);
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Error: Could not calculate the fine amount for the book');
END;




INSERT INTO Borrower VALUES (1, 'Amit',TO_DATE('01-APR-2023','DD-MON-YYYY'),
'Java', 'I');
INSERT INTO Borrower VALUES (2,'Alok',TO_DATE ('02-APR-2023','DD-MON-YYYY'),
'Python','I');
INSERT INTO Borrower VALUES (3, 'Sneha', TO_DATE('03-APR-2023',
'DD-MON-YYYY'), 'C++', 'I');
INSERT INTO Borrower VALUES (4, 'Neha', TO_DATE('04-APR-2023',
'DD-MON-YYYY'),'JavaScript','I');
INSERT INTO Borrower VALUES (5, 'Rahul', TO_DATE('05-APR-2023',
'DD-MON-YYYY'), 'SQL', 'I');

INSERT INTO Fine VALUES (2, SYSDATE, 75.00);

ROLL_NO  DAT            AMT
----------      ---------       ----------
        2       10-MAY-23    75
```

**OUTPUT:-**

```
Enter value for roll_no: 1
old   5:   roll_no := &roll_no;
new   5:   roll_no := 1;
Enter value for name_of_book: Java
```

```
old   6:   name_of_book := '&name_of_book';
new   6:   name_of_book := 'Java';

PL/SQL procedure successfully completed.
ROLL_NO  DAT            AMT
---------      ---------          ----------
        2       10-MAY-23         75
        1       10-MAY-23      2000
```

---

**Q2.** 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 the message on screen “account is inactive”.

**Ans:-**

```
CREATE OR REPLACE FUNCTION check_acc_status(acc_no IN NUMBER, Branch IN
VARCHAR2)
RETURN VARCHAR2
IS
  Stat VARCHAR(9);
BEGIN
  SELECT Status INTO Stat FROM acc_details WHERE acc=acc_no AND
branch=Branch;
  If stat='active' Then
    INSERT INTO active_acc_details VALUES (acc_no,Branch);
    RETURN  'Account is Active';
  Else
    RETURN  'Account is Inactive';
  END If;
EXCEPTION
  WHEN no_data_found THEN
    RETURN 'Account not found';
  WHEN others THEN
    RETURN 'ERROR!';
END;
```

```
DECLARE
  Acc NUMBER;
  Branch VARCHAR(8);
  Msg VARCHAR(50);
BEGIN
  Acc := &Account_number;
  Branch := '&Branch_name';
  Msg := check_acc_status(Acc,Branch);
  DBMS_OUTPUT.PUT_LINE(Msg);
END;

CREATE TABLE acc_details (Acc NUMBER, name VARCHAR(8), branch
VARCHAR(8), status VARCHAR(9));

INSERT INTO acc_details VALUES (101, 'John', 'Paris', 'active');
INSERT INTO acc_details VALUES (102, 'Jane', 'London', 'inactive');
INSERT INTO acc_details VALUES (103, 'Mark', 'Mumbai', 'active');
INSERT INTO acc_details VALUES (104, 'Lucas', 'Seoul', 'inactive');
INSERT INTO acc_details VALUES (105, 'Maria', 'Madrid', 'active');

CREATE TABLE active_acc_details(acc NUMBER, branch VARCHAR(8));

INSERT INTO active_acc_details VALUES(105, 'Madrid');
```

**OUTPUT:-**

```
Enter value for account_number: 101
old   6:   Acc := &Account_number;
new   6:   Acc := 101;
Enter value for branch_name: Paris
old   7:   Branch := '&Branch_name';
new   7:   Branch := 'Paris';
Account is Active

PL/SQL procedure successfully completed.
```

---

**Q3**. Write a Stored Procedure namely proc_Grade for the categorization of students. 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 and 900 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)

**Ans:-**

```
CREATE OR REPLACE PROCEDURE proc_Grade (roll_no IN NUMBER)
AS
  marks NUMBER;
BEGIN
  SELECT total_marks INTO marks FROM Stud_Marks WHERE name IN (SELECT
Name FROM Result WHERE Roll=roll_no);
  IF marks >= 990 AND marks <= 1500 THEN
    DBMS_OUTPUT.PUT_LINE( 'Roll number ' || roll_no ||  ' is in Distinction category');
  ElsIF marks >= 900 THEN
    DBMS_OUTPUT.PUT_LINE( 'Roll number ' || roll_no ||  ' is in First Class category');
  ElsIf marks >=  825 Then
    DBMS_OUTPUT.PUT_LINE( 'Roll number ' || roll_no ||  ' is in Higher Second Class
category');
  END IF;
EXCEPTION
  WHEN no_data_found THEN
    DBMS_OUTPUT.PUT_LINE('Enter a valid roll number');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('ERROR!');
END;

DECLARE
  roll NUMBER;
BEGIN
  roll := &Roll_number;
  proc_Grade (roll);
END;

CREATE TABLE Stud_Marks(name VARCHAR(9), total_marks NUMBER);
  INSERT INTO Stud_Marks VALUES('abc',889);
  INSERT INTO Stud_Marks VALUES('pqr',989);
  INSERT INTO Stud_Marks VALUES('xyz',775);
  INSERT INTO Stud_Marks VALUES('lmn',1225);
```

```
  INSERT INTO Stud_Marks VALUES('Xab',1105);

CREATE TABLE Result(Roll NUMBER, Name VARCHAR(9), Class VARCHAR(3));
  INSERT INTO Result VALUES(1,'abc','A');
  INSERT INTO Result VALUES(2,'xyz','A');
  INSERT INTO Result VALUES(3,'pql','A');
  INSERT INTO Result VALUES(4,'Xab','A');
  INSERT INTO Result VALUES(5,'lmn','A');
```

**OUTPUT:-**

```
Enter value for roll_number: 1
old   4:   roll := &Roll_number;
new   4:   roll := 1;
Roll number 1 is in Higher Second Class category

PL/SQL procedure successfully completed.
—----------------------------------------------------------------
Enter value for roll_number: 5
old   4:   roll := &Roll_number;
new   4:   roll := 5;
Roll number 5 is in Distinction category

PL/SQL procedure successfully completed.
```