

Consider Tables:

1. Borrower (Roll_no, Name, Date of Issue, Name of Book, Status)

2. Fine (Roll_no, Date, Amt)

- Accept Roll_no and 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.
- Also handles the exception by named exception handler or user define exception handler.

PL/SQL Block: -

```
CREATE TABLE Borrower (Rollno NUMBER(4), Name VARCHAR2(20), DateofIssue DATE,
NameofBook VARCHAR2(30), Status VARCHAR2(10));
```

```
INSERT INTO Borrower VALUES (14, 'Ram', TO_DATE('2024-09-10', 'YYYY-MM-DD'), 'DBMS',
'T');
```

```
INSERT INTO Borrower VALUES (27, 'Soham', TO_DATE('2024-09-10', 'YYYY-MM-DD'), 'Theory
of Computation', 'I');
```

```
INSERT INTO Borrower VALUES (34, 'Mohan', TO_DATE('2024-09-10', 'YYYY-MM-DD'),
'Computer Networks', 'I');
```

```
INSERT INTO Borrower VALUES (48, 'Om', TO_DATE('2024-09-10', 'YYYY-MM-DD'), 'SPOS', 'I');
```

```
CREATE TABLE Fine (Rollno NUMBER(4), Dates DATE, Amount NUMBER(10));
```

```
CREATE OR REPLACE PROCEDURE calc_Fine( r IN NUMBER, b IN VARCHAR2)
IS
```

```
    doi Borrower.DateofIssue%TYPE;
```

```
    diff NUMBER;
```

```
    fine_amount NUMBER := 0;
```

```
BEGIN
```

```
    SELECT DateofIssue INTO doi FROM Borrower WHERE Rollno = r AND NameofBook = b;
    diff := TRUNC(SYSDATE) - TRUNC(doi);
```

```
    IF diff BETWEEN 15 AND 30 THEN
```

```
        fine_amount := diff * 5;
```

```
    ELSIF diff > 30 THEN
```

```
        fine_amount := 30 * 5 + (diff - 30) * 50;
```

```
    END IF;
```

```
    IF fine_amount > 0 THEN
```

```
        INSERT INTO Fine (Rollno, Dates, Amount) VALUES (r, SYSDATE, fine_amount);
```

```
    END IF;
```

```
    COMMIT;
```

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        DBMS_OUTPUT.PUT_LINE('No such borrower or book found.');
```

```
    WHEN OTHERS THEN
```

```
DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/

CREATE OR REPLACE PROCEDURE submit( r IN NUMBER)
IS
BEGIN
    UPDATE Borrower SET Status = 'R' WHERE Rollno = r;

    DELETE FROM Fine WHERE Rollno = r;
    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
```

```
BEGIN
    calc_Fine(14, 'DBMS');
    calc_Fine(27, 'Theory of Computation');
    calc_Fine(34, 'Computer Networks');
    calc_Fine(48, 'SPOS');
END;
/
```

```
SELECT * FROM Borrower;
```

ROLLNO	NAME	DATEOFISSUE	NAMEOFBOOK	STATUS
14	Ram	10-SEP-24	DBMS	I
27	Soham	10-SEP-24	Theory of Computation	I
34	Mohan	10-SEP-24	Computer Networks	I
48	Om	10-SEP-24	SPOS	I

```
SELECT * FROM Fine;
```

ROLLNO	DATES	AMOUNT
14	02-OCT-24	110
27	02-OCT-24	110
34	02-OCT-24	110
48	02-OCT-24	110

```
BEGIN
  submit(14);
  submit(27);
  submit(34);
  submit(48);
END;
/
```

```
SELECT * FROM Borrower;
```

ROLLNO	NAME	DATEOFISSUE	NAMEOFBOOK	STATUS
14	Ram	10-SEP-24	DBMS	R
27	Soham	10-SEP-24	Theory of Computation	R
34	Mohan	10-SEP-24	Computer Networks	R
48	Om	10-SEP-24	SPOS	R

```
SELECT * FROM Fine;
```

output:-

no data found

Q2. Write a **PL/SQL code block** to calculate the area of a circle for a value of radius varying from 5 to 9. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns, radius and area.

PL/SQL Block: -

```
CREATE TABLE areas (radius NUMBER(5),area NUMBER(10, 2));
```

```
DECLARE
```

```
  r NUMBER(5);
```

```
  a NUMBER(10, 2);
```

```
  pi CONSTANT NUMBER := 3.14159;
```

```
BEGIN
```

```
  FOR r IN 5..9 LOOP
```

```
    a := pi * r * r;
```

```
    INSERT INTO areas (radius, area) VALUES (r, a);
```

```
  END LOOP;
```

```
END;
```

```
/
```

```
SELECT * FROM areas;
```

RADIUS	AREA
5	78.54
6	113.1
7	153.94
8	201.06
9	254.47