

PL/Sql-2 Write an Unnamed PL/SQL of code for the following requirements: - Schema:
Borrower (Rollin, Name, DateofIssue, NameofBook, Status) Fine (**Roll_no,Date,Amt**)
Accept roll_no & name of book from user. Check the number of days (from date of issue).

- 1. If days are between 15 to 30 then fine amounts will be Rs 5 per day.**
- 2. If no. of days>30, per day fine will be Rs 50 per day & for days less than 30, Rs. 5 per day.**
- 3. After submitting the book, status will change from I to R.**
- 4. If condition of fine is true, then details will be stored into fine table.**

```
CREATE TABLE Borrower (
```

```
    Rollin INT,  
    Name VARCHAR(50),  
    DateOfIssue DATE,  
    NameOfBook VARCHAR(100),  
    Status CHAR(1)
```

```
) ;
```

```
CREATE TABLE Fine (
```

```
    Roll_no INT,  
    Date DATE,  
    Amt DECIMAL(10,2)
```

```
) ;
```

```
DELIMITER //
```

```
CREATE PROCEDURE Return_Book(
```

```
    IN p_roll INT,  
    IN p_book VARCHAR(100)
```

```
)
```

```
BEGIN
```

```
    DECLARE v_issue DATE;
```

```
    DECLARE v_days INT;
```

```
    DECLARE v_fine DECIMAL(10,2) DEFAULT 0;
```

```
-- Get the issue date for the given roll and book
```

```
SELECT DateOfIssue INTO v_issue  
FROM Borrower
```

```
WHERE Rollin = p_roll AND NameOfBook = p_book;
```

```
-- Calculate total days between issue and return
```

```
SET v_days = DATEDIFF(CURDATE(), v_issue);
```

```
-- Calculate fine based on number of days
```

```
IF v_days BETWEEN 15 AND 30 THEN
```

```

        SET v_fine = v_days * 5;
ELSEIF v_days > 30 THEN
    SET v_fine = (30 * 5) + ((v_days - 30) * 50);
ELSE
    SET v_fine = 0;
END IF;

-- Update the status from 'I' (Issued) to 'R' (Returned)
UPDATE Borrower
SET Status = 'R'
WHERE Rollin = p_roll AND NameOfBook = p_book;

-- If fine applicable, insert into Fine table
IF v_fine > 0 THEN
    INSERT INTO Fine(Roll_no, Date, Amt)
    VALUES (p_roll, CURDATE(), v_fine);
END IF;

-- Display message output
SELECT CONCAT('Book Returned Successfully! Days: ', v_days, ', Fine:
Rs ', v_fine) AS Message;

END //

INSERT INTO Borrower VALUES
(101, 'Ravi', '2025-10-01', 'DBMS', 'I'),
(102, 'Anita', '2025-10-20', 'OS', 'I'),
(103, 'Meena', '2025-10-25', 'Python', 'I');

CALL Return_Book(101, 'DBMS');
//
```

Output Example:

Book Returned Successfully! Days: 35, Fine: Rs 400

Check Tables

```
SELECT * FROM Borrower;
//
```

```
SELECT * FROM Fine;
//
```