

Practical 4: Unnamed PL/SQL Code Block

Objective:

To demonstrate the use of Control Structures and Exception Handling in MySQL Stored Procedures.

Problem Statement:

Consider the following 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. Calculate fine as per conditions:

- Days <= 15 -> No fine
- Days between 16-30 -> Rs.5 per day
- Days > 30 -> Rs.50 per day

After book return, status changes from 'I' to 'R'. If fine > 0, insert into Fine table.

```
-- Step 1: Create Database
CREATE DATABASE LibraryDB;
USE LibraryDB;

-- Step 2: Create Borrower Table
CREATE TABLE Borrower (
    Roll_no INT,
    Name VARCHAR(100),
    Date_of_Issue DATE,
    Name_of_Book VARCHAR(100),
    Status CHAR(1)
);
```

```
-- Step 3: Create Fine Table
CREATE TABLE Fine (
    Roll_no INT,
```

```

Date DATE,
Amt DECIMAL(10,2)
);

-- Step 4: Insert Sample Data
INSERT INTO Borrower VALUES
(1, 'Shruti', '2025-08-01', 'DBMS Book', 'I'),
(2, 'Riya', '2025-10-15', 'Java Programming', 'I'),
(3, 'Amit', '2025-10-25', 'Python Book', 'I');

-- Step 5: Change Delimiter
DELIMITER //

-- Step 6: Create Stored Procedure
CREATE PROCEDURE ReturnBook(IN p_rollno INT, IN p_book VARCHAR(100))
BEGIN
    DECLARE v_date_issue DATE;
    DECLARE v_days INT;
    DECLARE v_fine INT DEFAULT 0;
    DECLARE v_not_found CONDITION FOR SQLSTATE '02000';

    DECLARE CONTINUE HANDLER FOR v_not_found
    BEGIN
        SELECT 'Error: No record found for given Roll_no and Book.' AS message;
    END;

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        SELECT 'An unexpected error occurred. Please check input values.' AS message;
    END;

    SELECT Date_of_Issue INTO v_date_issue
    FROM Borrower
    WHERE Roll_no = p_rollno
        AND Name_of_Book = p_book
        AND Status = 'I'
    LIMIT 1;

    SET v_days = DATEDIFF(CURDATE(), v_date_issue);

    IF v_days <= 15 THEN
        SET v_fine = 0;
    ELSEIF v_days BETWEEN 16 AND 30 THEN
        SET v_fine = v_days * 5;
    ELSE

```

```

        SET v_fine = v_days * 50;
END IF;

UPDATE Borrower
SET Status = 'R'
WHERE Roll_no = p_rollno
AND Name_of_Book = p_book;

IF v_fine > 0 THEN
    INSERT INTO Fine (Roll_no, Date, Amt)
    VALUES (p_rollno, CURDATE(), v_fine);
END IF;

SELECT CONCAT('Book returned. Days: ', v_days, ', Fine: ', v_fine) AS message;
END //
DELIMITER ;

-- Step 7: Execute Procedure
CALL ReturnBook(1, 'DBMS Book');

-- Step 8: View Fine Table
SELECT * FROM Fine;

```