-- Step 1: Create the database

CREATE DATABASE IF NOT EXISTS seven;

-- Step 2: Use the created database

USE seven;

-- Step 3: Create the Borrower and Fine tables

CREATE TABLE Borrower (

Roll\_no INT PRIMARY KEY,

Name VARCHAR(100),

Date\_of\_Issue DATE,

Name\_of\_Book VARCHAR(100),

Status CHAR(1) DEFAULT 'I' -- 'I' for Issued, 'R' for Returned

);

CREATE TABLE Fine (

Roll\_no INT,

Date DATE,

Amt INT,

FOREIGN KEY (Roll\_no) REFERENCES Borrower(Roll\_no)

);

-- Step 4: Insert sample data into Borrower table

INSERT INTO Borrower (Roll\_no, Name, Date\_of\_Issue, Name\_of\_Book) VALUES

(1, 'Alice', '2024-07-01', 'Introduction to Algorithms'),

(2, 'Bob', '2024-07-05', 'Machine Learning Yearning'),

(3, 'Charlie', '2024-07-10', 'Deep Learning'),

(4, 'David', '2024-07-15', 'Data Science for Business');

-- Step 5: Create a PL/SQL block to calculate and apply fines

DELIMITER //

CREATE PROCEDURE CalculateAndApplyFines(p\_Roll\_no INT, p\_Name\_of\_Book VARCHAR(100))

BEGIN

DECLARE v\_days\_overdue INT;

DECLARE v\_fine\_amount INT;

DECLARE v\_date\_of\_issue DATE;

DECLARE v\_status CHAR(1);

DECLARE v\_current\_date DATE;

SET v\_current\_date = CURDATE();

-- Check if the book is issued

SELECT Date\_of\_Issue, Status INTO v\_date\_of\_issue, v\_status

FROM Borrower

WHERE Roll\_no = p\_Roll\_no AND Name\_of\_Book = p\_Name\_of\_Book;

-- Handle if no records are found

IF v\_status IS NULL THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'No such book issued to this borrower';

ELSE

-- Calculate the number of overdue days

SET v\_days\_overdue = DATEDIFF(v\_current\_date, DATE\_ADD(v\_date\_of\_issue, INTERVAL 15 DAY));

-- Determine fine amount

IF v\_days\_overdue > 0 THEN

IF v\_days\_overdue <= 30 THEN

SET v\_fine\_amount = v\_days\_overdue \* 5; -- Rs 5 per day for days between 15 and 30

ELSE

SET v\_fine\_amount = (30 \* 5) + ((v\_days\_overdue - 30) \* 50); -- Rs 50 for days after 30

END IF;

-- Update the Borrower status to 'R' for returned

UPDATE Borrower SET Status = 'R' WHERE Roll\_no = p\_Roll\_no AND Name\_of\_Book = p\_Name\_of\_Book;

-- Insert fine details into the Fine table

INSERT INTO Fine (Roll\_no, Date, Amt) VALUES (p\_Roll\_no, v\_current\_date, v\_fine\_amount);

END IF;

END IF;

END //

DELIMITER ;

-- Step 6: Call the procedure to calculate and apply fines

CALL CalculateAndApplyFines(1, 'Introduction to Algorithms'); -- Example to check for Roll\_no 1 and book

-- Step 7: Select from the Fine table to view applied fines

SELECT \* FROM Fine;

-- Explanation of the Code

-- Database Creation: It starts by creating a database named assignment4 if it does not exist.

-- Tables Creation: Two tables, Borrower and Fine, are created to manage borrower information and fine records, respectively.

-- Sample Data Insertion: Sample records are inserted into the Borrower table to simulate book issues.

-- Procedure Creation:

-- A stored procedure CalculateAndApplyFines is created to calculate the fine for a specified Roll\_no and Name\_of\_Book.

-- The procedure checks if the book has been issued and calculates the number of overdue days.

-- Fines are calculated based on the overdue days:

-- Rs 5 per day for days between 15 and 30.

-- Rs 50 per day for days after 30.

-- The status of the book is updated to 'R' (returned) in the Borrower table.

-- If a fine is applicable, it is inserted into the Fine table.

-- Calling the Procedure: An example call is made to apply fines for a specific roll number and book name.

-- Viewing Results: The last SELECT statement retrieves the fine records to see the results.