2. A library tracks borrowed books and their return status. A fine of ₹2 is applied for each day after
the due date.
Schema:
● Borrowers(borrow_id INT PRIMARY KEY, student_name VARCHAR(50), due_date DATE, return_date DATE, fine INT DEFAULT 0)
Task:
Write a stored procedure using a cursor to:
● Loop through all records in Borrowers
● For each student who returned the book late, calculate the number of overdue days
● Multiply overdue days by ₹2 and update the fine column
● Show a message like: Fine of ₹20 updated for Rahul Singh

```
-- Stored Procedure: calculate_fines
DELIMITER $$

CREATE PROCEDURE calculate_fines()
BEGIN
    -- Declare variables
    DECLARE done INT DEFAULT FALSE;
    DECLARE borrowId INT;
    DECLARE studentName VARCHAR(50);
    DECLARE dueDate DATE;
    DECLARE returnDate DATE;
    DECLARE overdueDays INT;
    DECLARE fineAmount INT;

    -- Declare cursor for Borrowers table
    DECLARE borrower_cursor CURSOR FOR
        SELECT borrow_id, student_name, due_date, return_date
        FROM Borrowers;

    -- Handler to exit the loop
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    -- Open the cursor
    OPEN borrower_cursor;

    read_loop: LOOP
        FETCH borrower_cursor INTO borrowId, studentName, dueDate, returnDate;
        IF done THEN
            LEAVE read_loop;
        END IF;

        -- Check if the book was returned late
        IF returnDate > dueDate THEN
            SET overdueDays = DATEDIFF(returnDate, dueDate);
```

```
            SET fineAmount = overdueDays * 2;

            -- Update fine in the table
            UPDATE Borrowers
            SET fine = fineAmount
            WHERE borrow_id = borrowId;

            -- Show message
            SELECT CONCAT('Fine of ₹', fineAmount, ' updated for ', studentName) AS
Message;
        END IF;
    END LOOP;

    -- Close the cursor
    CLOSE borrower_cursor;
END$$

DELIMITER ;

-- Sample Table Setup (if needed):
CREATE TABLE Borrowers (
    borrow_id INT PRIMARY KEY,
    student_name VARCHAR(50),
    due_date DATE,
    return_date DATE,
    fine INT DEFAULT 0
);

INSERT INTO Borrowers (borrow_id, student_name, due_date, return_date)
VALUES
(1, 'Rahul Singh', '2024-04-10', '2024-04-15'),
(2, 'Anjali Mehta', '2024-04-12', '2024-04-12'),
(3, 'Vikram Rao', '2024-04-05', '2024-04-08');

-- To Execute the Procedure:
CALL calculate_fines();
```