Database Queries for Assignment-A4

Creating tables

```
CREATE TABLE Borrower (
  roll_no INT,
  issuer_name VARCHAR(255),
  issue_date DATE,
  book_name VARCHAR(255),
  status VARCHAR(1),
  PRIMARY KEY (roll_no)
);

CREATE TABLE Fine (
  roll_no INT,
  return_date DATE,
  amt INT,
  FOREIGN KEY (roll_no) REFERENCES Borrower (roll_no)
);
```

Inserting data

```
INSERT INTO Borrower VALUES (1, 'Kalas', TO_DATE('2024-10-19', 'YYYY-MM-DD'), 'DBMS', 'I');
INSERT INTO Borrower VALUES (2, 'Himanshu', TO_DATE('2024-10-01', 'YYYY-MM-DD'), 'TOC', 'I');
INSERT INTO Borrower VALUES (3, 'MEPA', TO_DATE('2024-10-25', 'YYYY-MM-DD'), 'IoT', 'I');
INSERT INTO Borrower VALUES (4, 'Kshitij', TO_DATE('2024-10-29', 'YYYY-MM-DD'), '1984', 'I');
```

Procedure

```
DECLARE
  p_roll NUMBER; -- specify roll number here since Live SQL cannot take input from user
  -- Eg. p_roll NUMBER := 1; will take roll number 1 as input
  p_book VARCHAR2(255); -- specify book name here since Live SQL cannot take input from user
  -- Eg. p_book VARCHAR2(255) := 'DBMS';
  p_issueDate DATE;
  totalDays NUMBER;
  currentDate DATE;
  fineAmt NUMBER;
  nodata EXCEPTION;

BEGIN
  -- Check if roll number is valid
  IF (p_roll <= 0) THEN
    RAISE nodata;
  END IF;

  -- Storing values from table in variables
  SELECT issue_date INTO p_issueDate FROM Borrower WHERE roll_no = p_roll AND book_name = p_book;

  -- Getting the total days since book issue
  SELECT TRUNC(SYSDATE) - p_issueDate INTO totalDays FROM dual;

  -- Calculating fine
  IF (totalDays > 30) THEN
    fineAmt := totalDays * 50; -- Rs. 50 per day for total days greater than 30
  ELSIF (totalDays BETWEEN 15 AND 30) THEN
    fineAmt := totalDays * 5; -- Rs. 5 per day for total days between 15 and 30
  ELSE
    fineAmt := 0;
  END IF;

  -- Inserting data into Fine table
  IF fineAmt > 0 THEN
    DBMS_OUTPUT.PUT_LINE('Roll no. ' || p_roll || ' has been fined Rs. ' || fineAmt || ' for being ' || totalDays || ' days late.');
    INSERT INTO Fine VALUES (p_roll, SYSDATE, fineAmt);
  ELSE
    DBMS_OUTPUT.PUT_LINE('Roll no. ' || p_roll || ' does not have to pay any fine.');
  END IF;
  UPDATE Borrower SET status = 'R' WHERE roll_no = p_roll AND book_name = p_book;

EXCEPTION
  WHEN nodata THEN
    DBMS_OUTPUT.PUT_LINE('Roll number' || p_roll || ' not found.');
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occured. Error: ' || SQLERRM);

END;
/
```