Problem Statement 1

Consider Tables:

1. Borrower (Roll_no, Name, Dateofissue, NameofBook, Satus)
2. Fine (Roll no, Date, Amt)

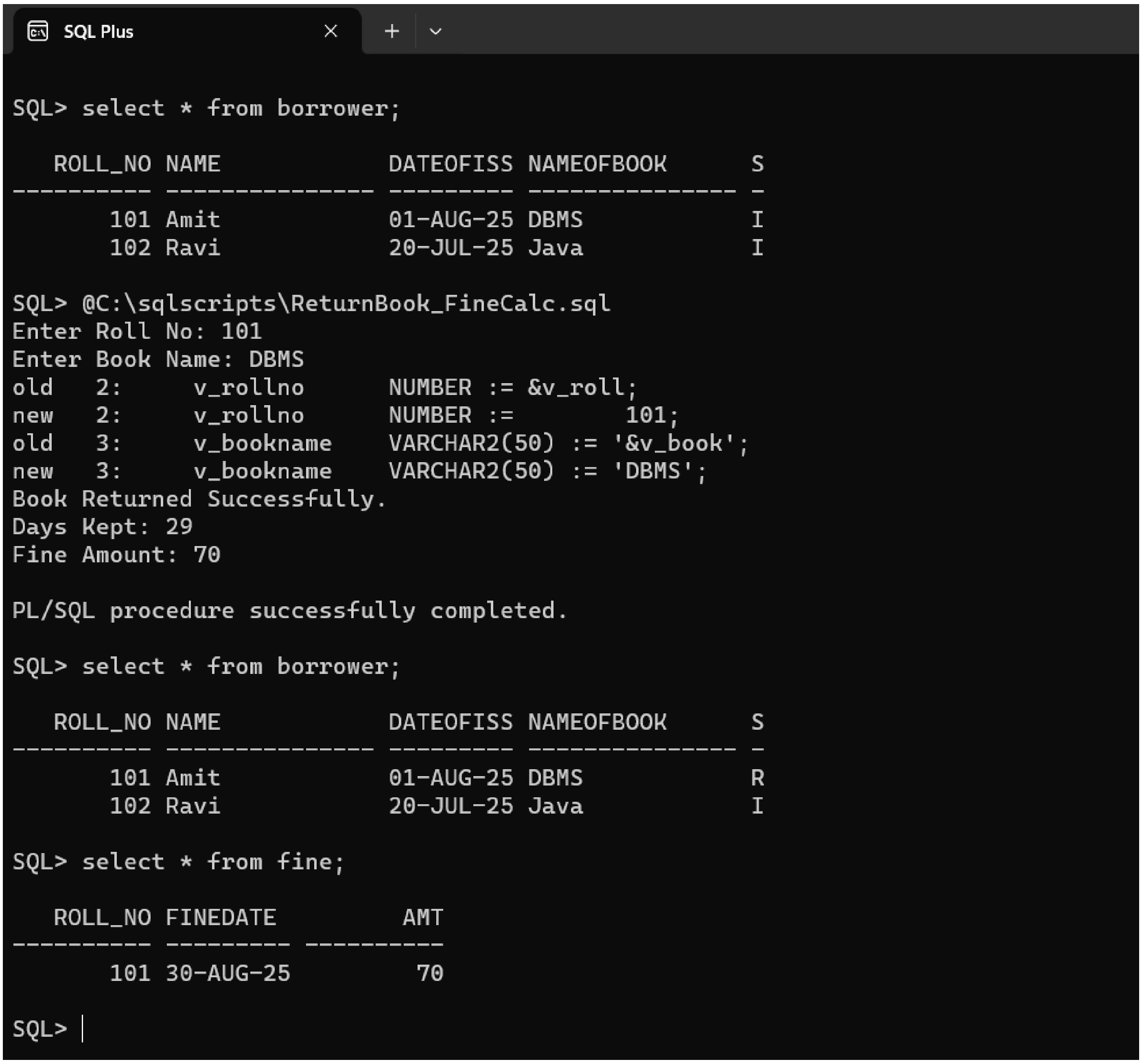
- Accept Roll_no and NameofBook from user.
- Check the number of days (from date of issue).
- If days are between 15 to 30 then fine amount will be Rs 5 per day.
- If no. of days 30, per day fine will be Rs 50 per day and for days less than 30, Rs. 5 per day.
- After submitting the book, status will change from 1 to R.
- If condition of fine is true, then details will be stored into fine table.
- Also handles the exception by named exception handler or user define exception handler.

Program Code –

```
ACCEPT v_roll NUMBER PROMPT 'Enter Roll No: '
ACCEPT v_book CHAR PROMPT 'Enter Book Name: '
SET SERVEROUTPUT ON;

DECLARE
   v_rollno      NUMBER := &v_roll;
   v_bookname    VARCHAR2(50) := '&v_book';
   v_dateissue   DATE;
   v_days        NUMBER;
   v_fine        NUMBER := 0;

BEGIN
   -- Fetch Date of Issue
   SELECT DateofIssue INTO v_dateissue
   FROM Borrower
   WHERE Roll_no = v_rollno AND NameofBook = v_bookname AND Status = 'I';

   -- Days kept
   v_days := TRUNC(SYSDATE - v_dateissue);

   -- Fine Calculation
   IF v_days <= 15 THEN
```

```
        v_fine := 0;
    ELSIF v_days > 15 AND v_days <= 30 THEN
        v_fine := (v_days - 15) * 5;
    ELSE
        v_fine := (15 * 5) + ((v_days - 30) * 50);
    END IF;

    -- Update status
    UPDATE Borrower
    SET Status = 'R'
    WHERE Roll_no = v_rollno AND NameofBook = v_bookname;

    -- Insert fine if any
    IF v_fine > 0 THEN
        INSERT INTO Fine (Roll_no, FineDate, Amt)
        VALUES (v_rollno, SYSDATE, v_fine);
    END IF;

    COMMIT;

    DBMS_OUTPUT.PUT_LINE('Book Returned Successfully.');
    DBMS_OUTPUT.PUT_LINE('Days Kept: ' || v_days);
    DBMS_OUTPUT.PUT_LINE('Fine Amount: ' || v_fine);

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No record found for given Roll_no and Book.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/
```

Output –



```
SQL> select * from borrower;

   ROLL_NO NAME            DATEOFISS NAMEOFBOOK      S
---------- --------------- --------- --------------- -
       101 Amit            01-AUG-25 DBMS            I
       102 Ravi            20-JUL-25 Java            I

SQL> @C:\sqlscripts\ReturnBook_FineCalc.sql
Enter Roll No: 101
Enter Book Name: DBMS
old   2:     v_rollno      NUMBER := &v_roll;
new   2:     v_rollno      NUMBER :=        101;
old   3:     v_bookname    VARCHAR2(50) := '&v_book';
new   3:     v_bookname    VARCHAR2(50) := 'DBMS';
Book Returned Successfully.
Days Kept: 29
Fine Amount: 70

PL/SQL procedure successfully completed.

SQL> select * from borrower;

   ROLL_NO NAME            DATEOFISS NAMEOFBOOK      S
---------- --------------- --------- --------------- -
       101 Amit            01-AUG-25 DBMS            R
       102 Ravi            20-JUL-25 Java            I

SQL> select * from fine;

   ROLL_NO FINEDATE         AMT
---------- --------- ----------
       101 30-AUG-25         70

SQL> 
```

Problem Statement 2

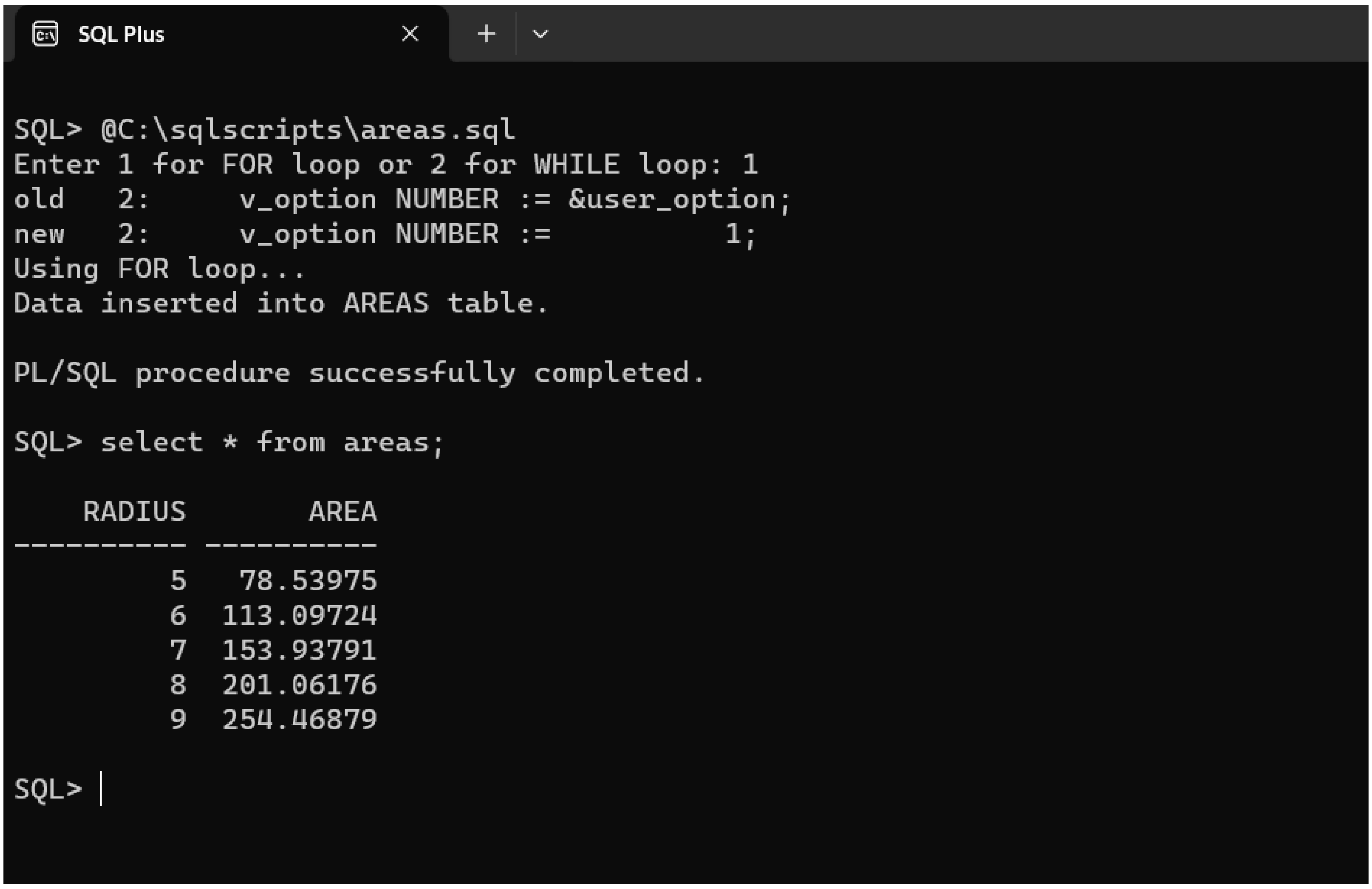
Write a PL/SQL code block to calculate the area of a circle for a value of radius varying from 5 to 9.

Store the radius and the corresponding values of calculated area in an empty table named areas consisting of two columns, radius and area.

Program Code -

```
ACCEPT user_option NUMBER PROMPT 'Enter 1 for FOR loop or 2 for WHILE loop: '
SET SERVEROUTPUT ON;

DECLARE
   v_option NUMBER := &user_option;
   v_radius NUMBER;
   v_area   NUMBER;
BEGIN
   CASE v_option
      WHEN 1 THEN
         DBMS_OUTPUT.PUT_LINE('Using FOR loop...');
         FOR r IN 5..9 LOOP
            v_area := 3.14159 * r * r;
            INSERT INTO areas VALUES (r, v_area);
         END LOOP;

      WHEN 2 THEN
         DBMS_OUTPUT.PUT_LINE('Using WHILE loop...');
         v_radius := 5;
         WHILE v_radius <= 9 LOOP
            v_area := 3.14159 * v_radius * v_radius;
            INSERT INTO areas VALUES (v_radius, v_area);
            v_radius := v_radius + 1;
         END LOOP;

      ELSE
         DBMS_OUTPUT.PUT_LINE('Invalid option! Enter 1 or 2.');
   END CASE;

   COMMIT;
   DBMS_OUTPUT.PUT_LINE('Data inserted into AREAS table.');
END;
/
```

Output -



```
SQL> @C:\sqlscripts\areas.sql
Enter 1 for FOR loop or 2 for WHILE loop: 1
old   2:     v_option NUMBER := &user_option;
new   2:     v_option NUMBER :=           1;
Using FOR loop...
Data inserted into AREAS table.

PL/SQL procedure successfully completed.

SQL> select * from areas;

    RADIUS       AREA
---------- ----------
         5   78.53975
         6  113.09724
         7  153.93791
         8  201.06176
         9  254.46879

SQL>
```