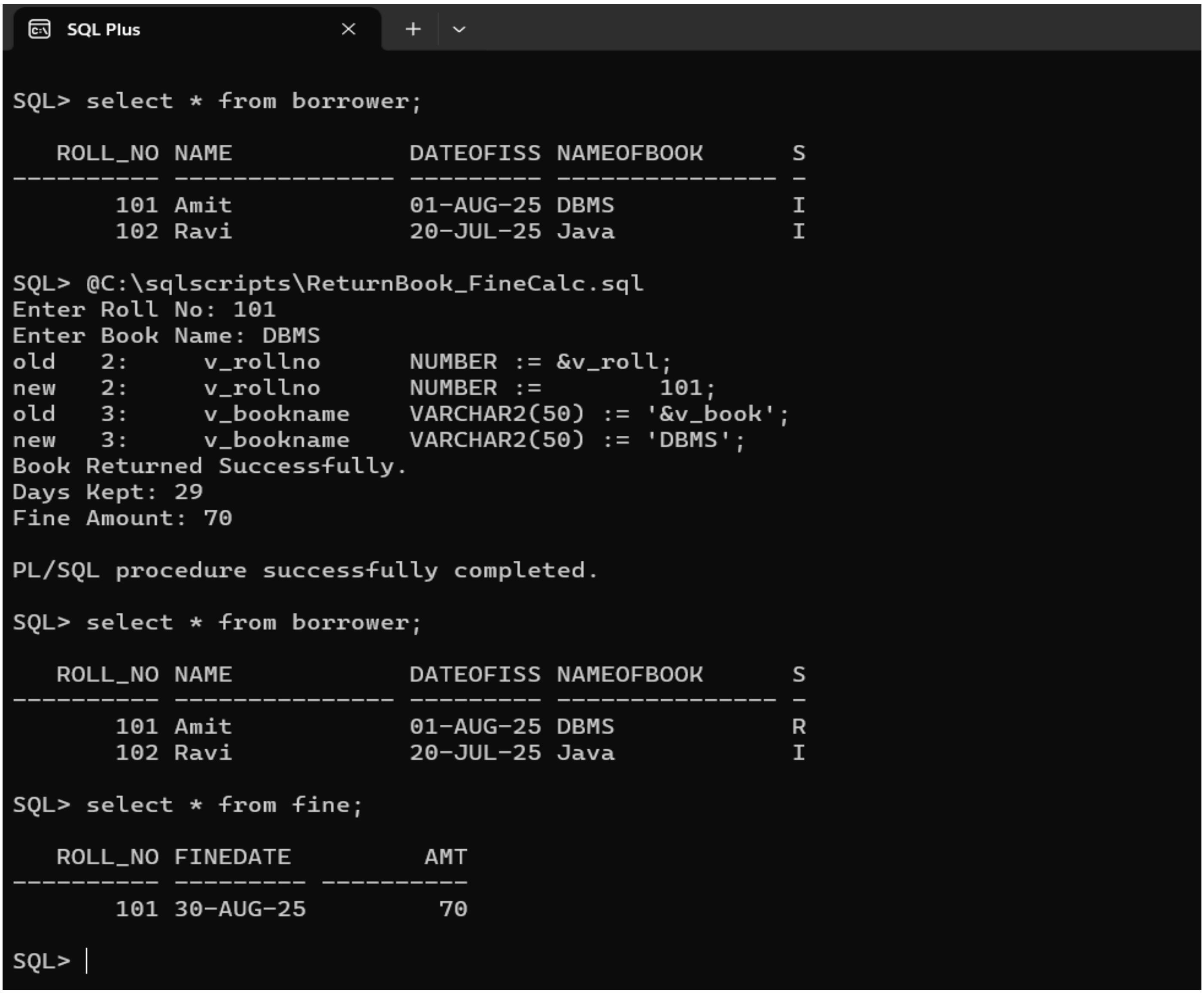
Program Code –

```
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 –

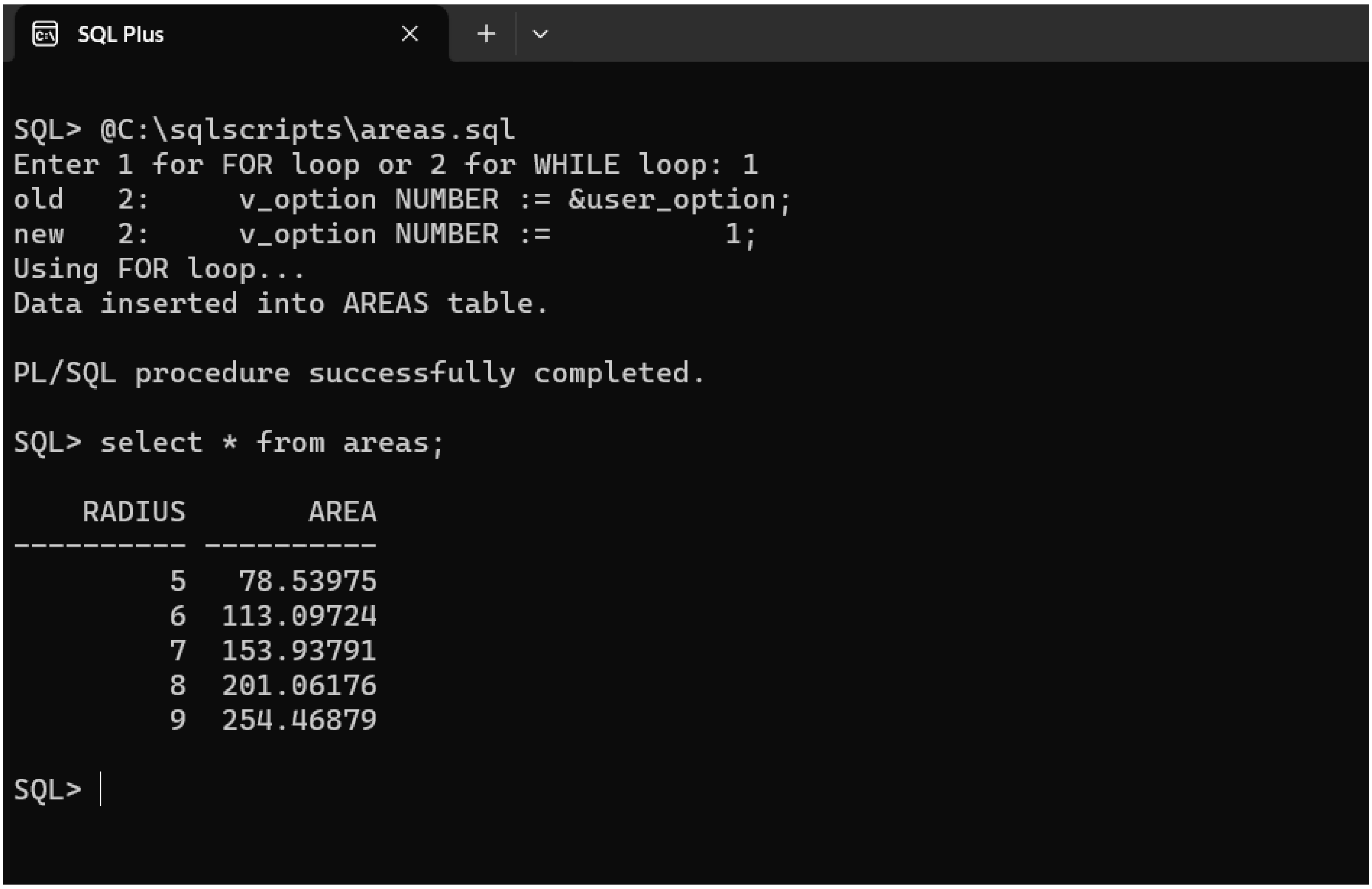


```
SQL> select * from borrower;

   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> |
```

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 -



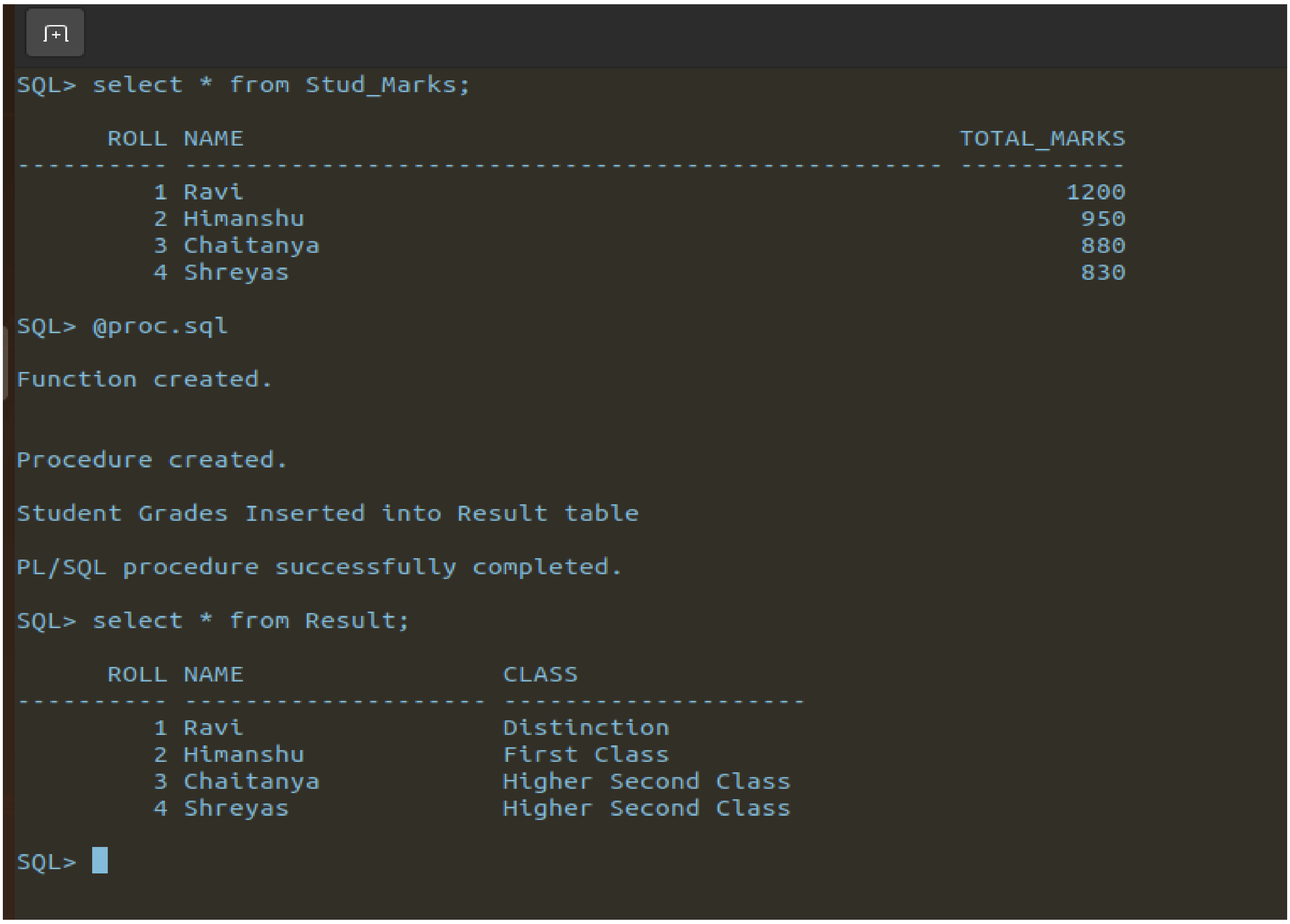
```
SQL> @C:\sqlscripts\areas.sql
Enter 1 for FOR loop or 2 for WHILE loop: 1
old   2:     v_option NUMBER := &user_option;
new   2:     v_option NUMBER :=           1;
Using FOR loop...
Data inserted into AREAS table.

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> 
```

Program Code -

```
SET SERVEROUTPUT ON;
CREATE OR REPLACE FUNCTION fn_GetGrade(p_marks NUMBER)
RETURN VARCHAR2 IS
   v_class VARCHAR2(30);
BEGIN
   IF p_marks BETWEEN 990 AND 1500 THEN
      v_class := 'Distinction';
   ELSIF p_marks BETWEEN 900 AND 989 THEN
      v_class := 'First Class';
   ELSIF p_marks BETWEEN 825 AND 899 THEN
      v_class := 'Higher Second Class';
   ELSE
      v_class := 'Not Categorized';
   END IF;
   RETURN v_class;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      RETURN 'No Data Found';
   WHEN OTHERS THEN
      RETURN 'Error Occurred';
END;
/

CREATE OR REPLACE PROCEDURE proc_Grade IS
BEGIN
   DELETE FROM Result;
   INSERT INTO Result (roll, name, class)
   SELECT roll,
          name,
          fn_GetGrade(total_marks)
   FROM Stud_Marks;
   COMMIT;
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No Data Found in Stud_Marks');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END;
/


BEGIN
    proc_Grade;
    DBMS_OUTPUT.PUT_LINE('Student Grades Inserted into Result table');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error in main block: ' || SQLERRM);
END;
/
```

Output –



```
SQL> select * from Stud_Marks;

      ROLL NAME                                                TOTAL_MARKS
---------- -------------------------------------------------- -----------
         1 Ravi                                                      1200
         2 Himanshu                                                   950
         3 Chaitanya                                                  880
         4 Shreyas                                                    830

SQL> @proc.sql

Function created.


Procedure created.

Student Grades Inserted into Result table

PL/SQL procedure successfully completed.

SQL> select * from Result;

      ROLL NAME                 CLASS
---------- -------------------- --------------------
         1 Ravi                 Distinction
         2 Himanshu             First Class
         3 Chaitanya            Higher Second Class
         4 Shreyas              Higher Second Class

SQL> 
```