College Of Engineering Trivandrum

Application Software Development Lab

Saurav S B
S5 CSE Roll No:56

TVE17CS057

Department of Computer Science

September 1, 2019

Cycle 2

Exp No 11

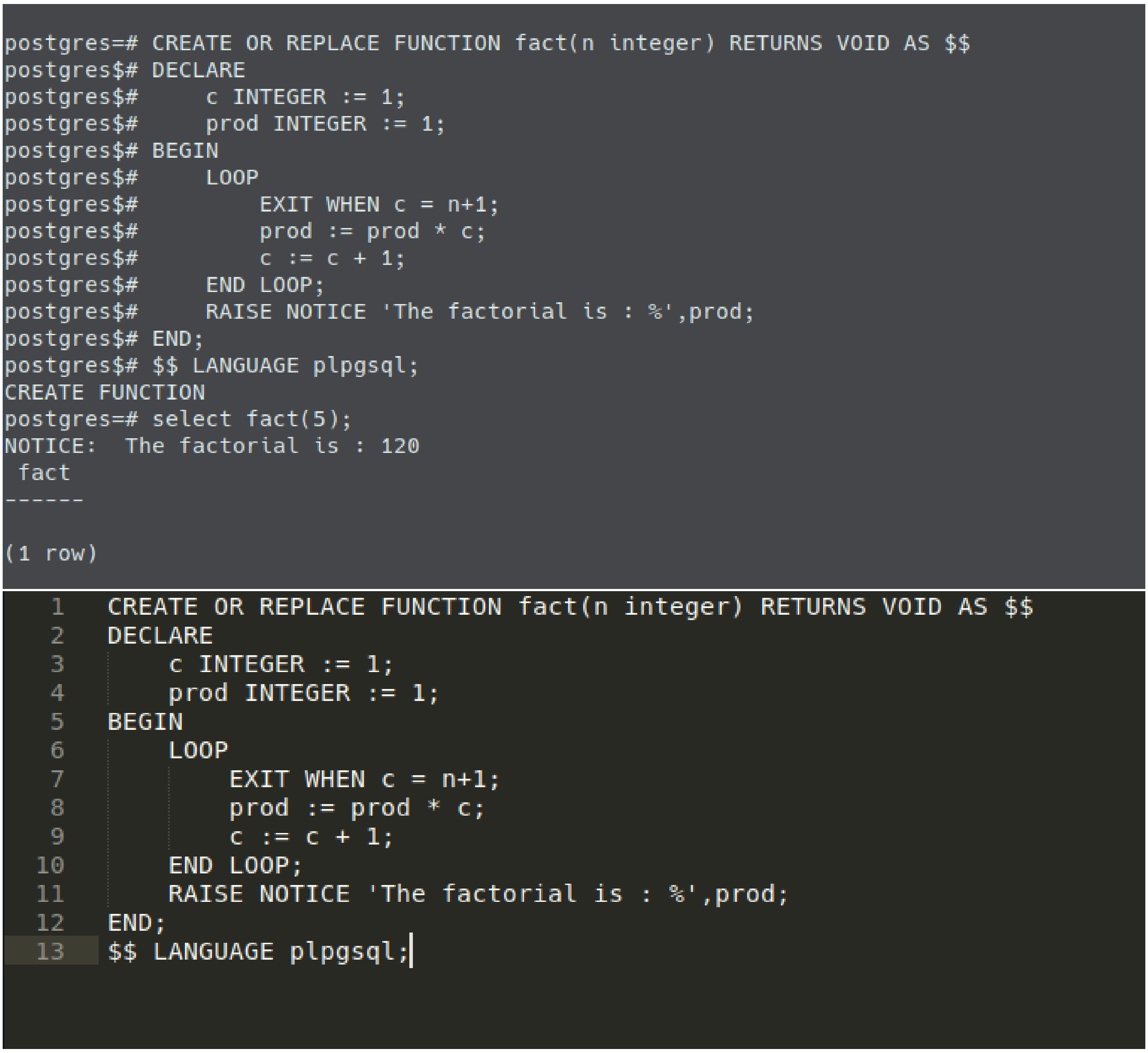
PROCEDURES, FUNCTIONS AND PACKAGES

1 Aim

To study PL/SQL Procedures , Functions and Packages.

2 Questions

1. Create a function factorial to find the factorial of a number. Use this function in a PL/SQL Program to display the factorial of a number read from the user.



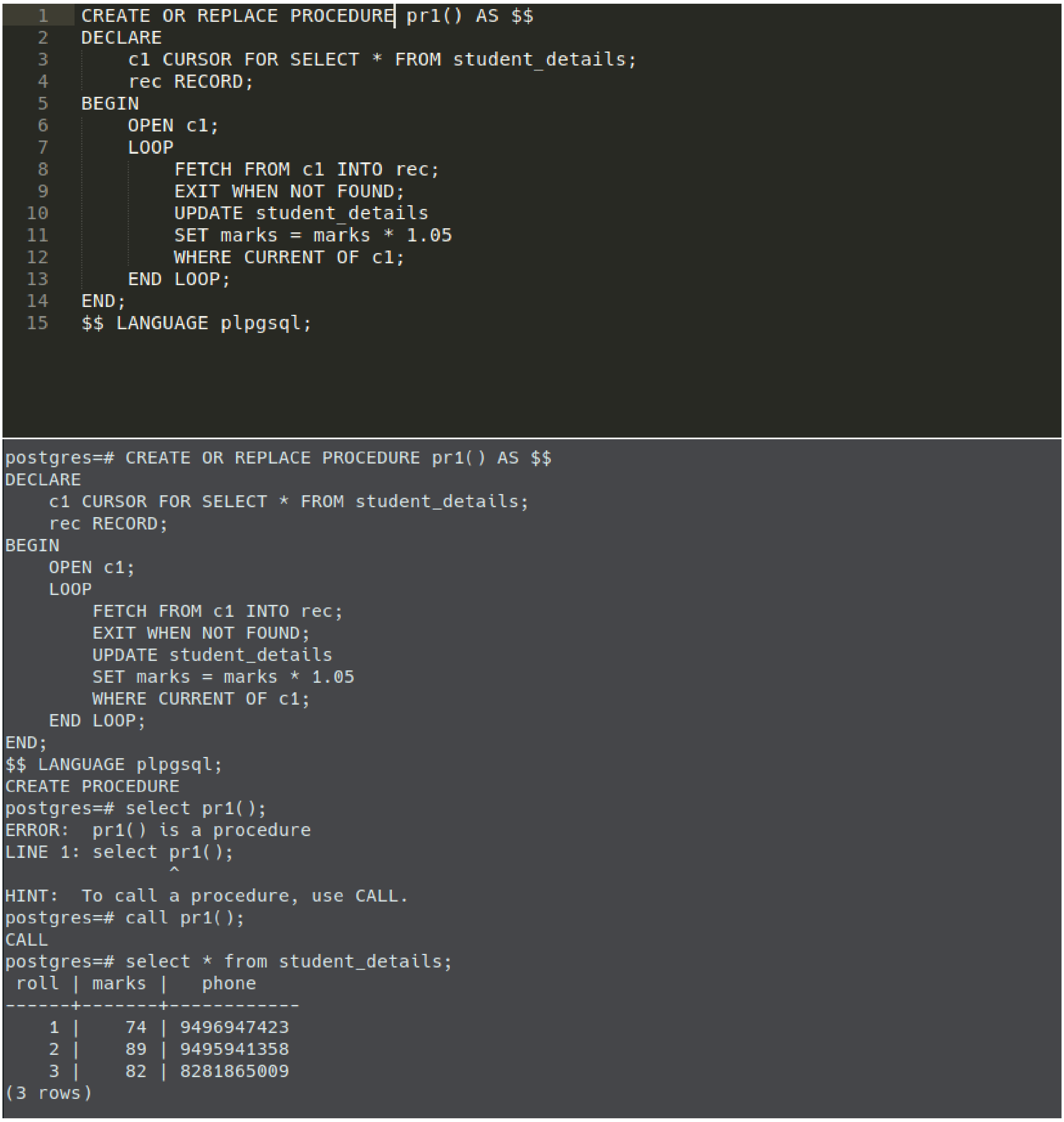
```
postgres=# CREATE OR REPLACE FUNCTION fact(n integer) RETURNS VOID AS $$
postgres$# DECLARE
postgres$#     c INTEGER := 1;
postgres$#     prod INTEGER := 1;
postgres$# BEGIN
postgres$#     LOOP
postgres$#         EXIT WHEN c = n+1;
postgres$#         prod := prod * c;
postgres$#         c := c + 1;
postgres$#     END LOOP;
postgres$#     RAISE NOTICE 'The factorial is : %',prod;
postgres$# END;
postgres$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
postgres=# select fact(5);
NOTICE:  The factorial is : 120
 fact
------

(1 row)
```

```
CREATE OR REPLACE FUNCTION fact(n integer) RETURNS VOID AS $$
DECLARE
    c INTEGER := 1;
    prod INTEGER := 1;
BEGIN
    LOOP
        EXIT WHEN c = n+1;
        prod := prod * c;
        c := c + 1;
    END LOOP;
    RAISE NOTICE 'The factorial is : %',prod;
END;
$$ LANGUAGE plpgsql;
```

Figure 1:

2. Create a table student_details(roll int,marksint, phone int).Create a procedure pr1 to update all rows in the database. Boost the marks of all students by 5%.

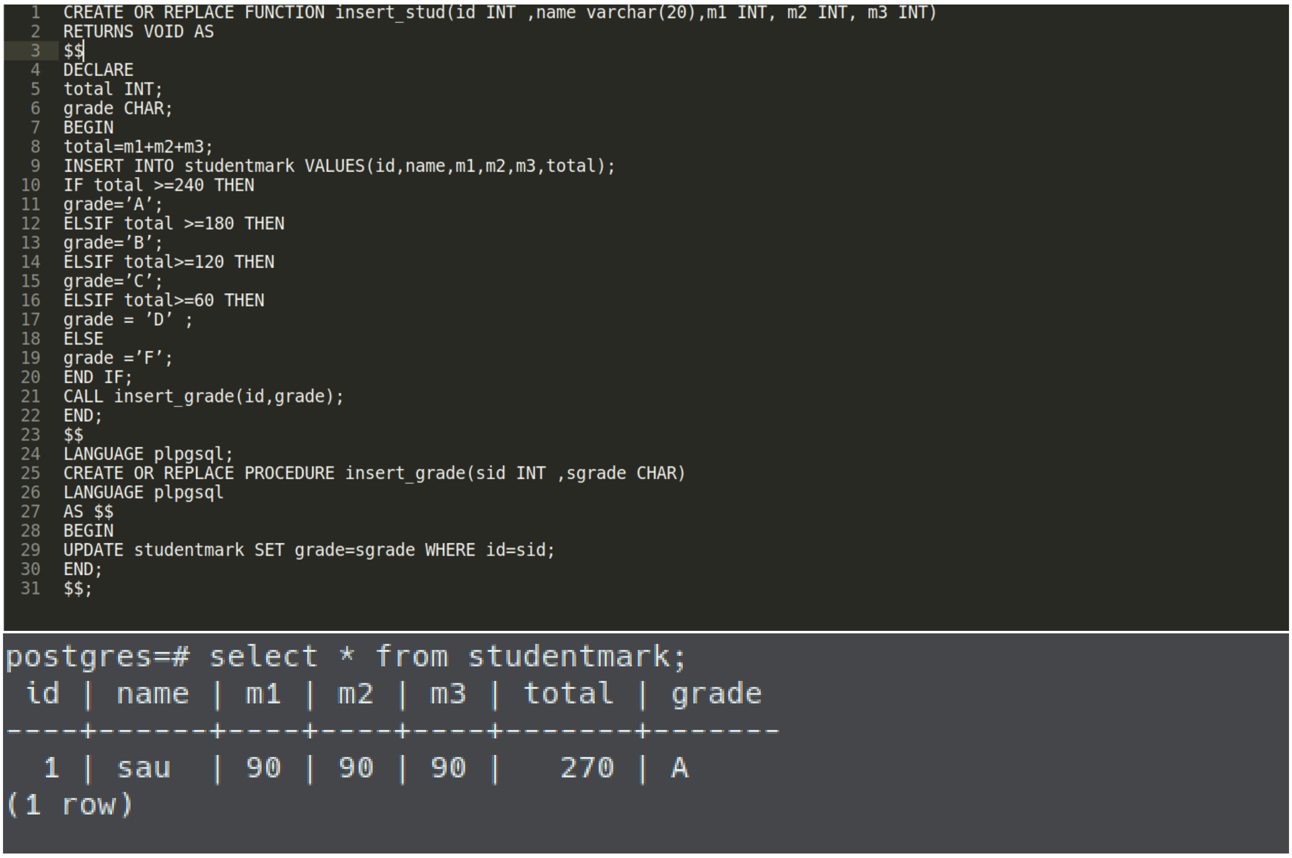


```
CREATE OR REPLACE PROCEDURE pr1() AS $$
DECLARE
    c1 CURSOR FOR SELECT * FROM student_details;
    rec RECORD;
BEGIN
    OPEN c1;
    LOOP
        FETCH FROM c1 INTO rec;
        EXIT WHEN NOT FOUND;
        UPDATE student_details
        SET marks = marks * 1.05
        WHERE CURRENT OF c1;
    END LOOP;
END;
$$ LANGUAGE plpgsql;
```

```
postgres=# CREATE OR REPLACE PROCEDURE pr1() AS $$
DECLARE
    c1 CURSOR FOR SELECT * FROM student_details;
    rec RECORD;
BEGIN
    OPEN c1;
    LOOP
        FETCH FROM c1 INTO rec;
        EXIT WHEN NOT FOUND;
        UPDATE student_details
        SET marks = marks * 1.05
        WHERE CURRENT OF c1;
    END LOOP;
END;
$$ LANGUAGE plpgsql;
CREATE PROCEDURE
postgres=# select pr1();
ERROR:  pr1() is a procedure
LINE 1: select pr1();
               ^
HINT:  To call a procedure, use CALL.
postgres=# call pr1();
CALL
postgres=# select * from student_details;
 roll | marks |   phone
------+-------+------------
    1 |    74 | 9496947423
    2 |    89 | 9495941358
    3 |    82 | 8281865009
(3 rows)
```

Figure 2:

3. Create table student (id, name, m1, m2, m3, total,grade).Create a function f1 to calculate grade. Create a procedure p1 to update the total and grade.

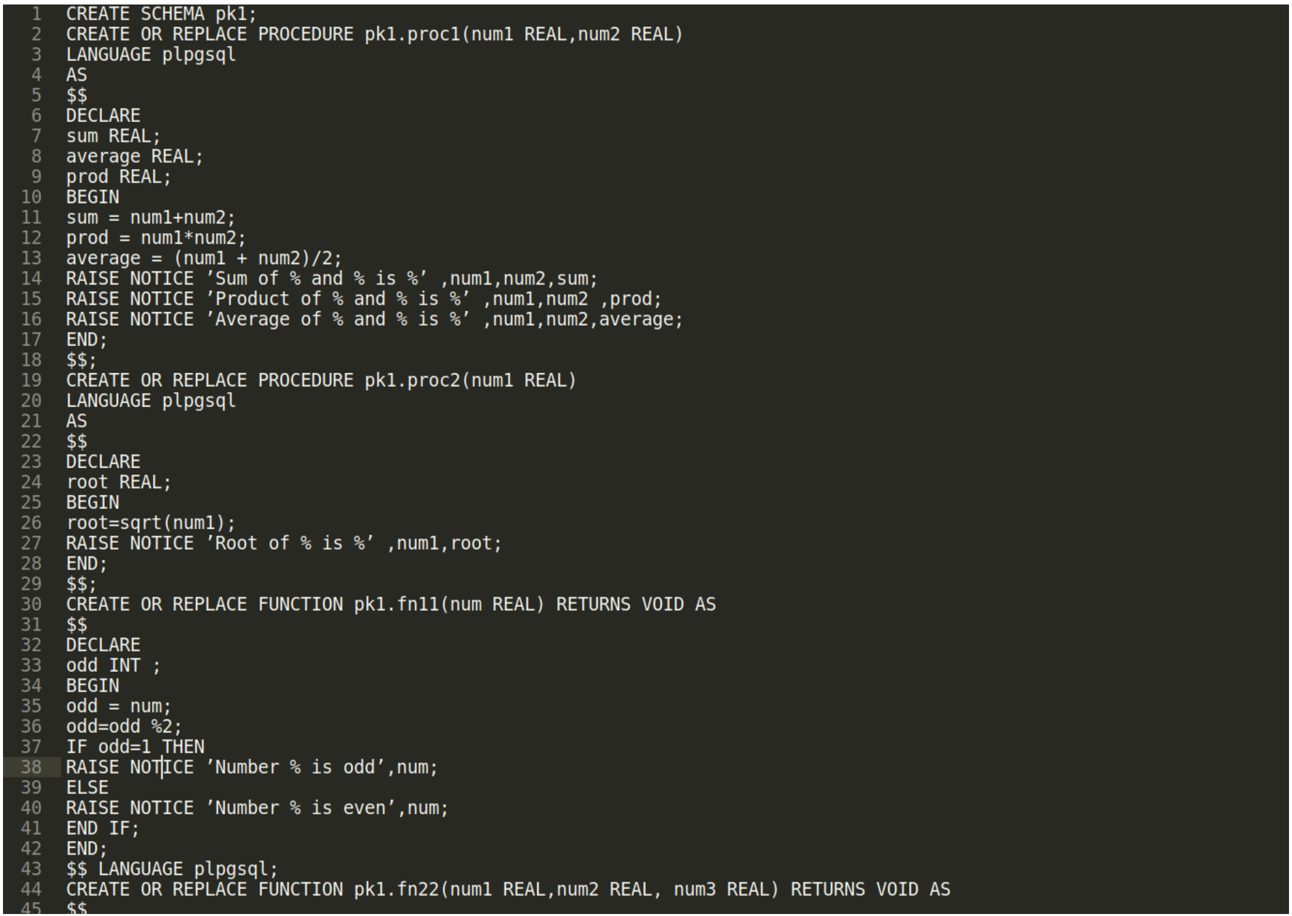


```
CREATE OR REPLACE FUNCTION insert_stud(id INT ,name varchar(20),m1 INT, m2 INT, m3 INT)
RETURNS VOID AS
$$
DECLARE
total INT;
grade CHAR;
BEGIN
total=m1+m2+m3;
INSERT INTO studentmark VALUES(id,name,m1,m2,m3,total);
IF total >=240 THEN
grade='A';
ELSIF total >=180 THEN
grade='B';
ELSIF total>=120 THEN
grade='C';
ELSIF total>=60 THEN
grade = 'D' ;
ELSE
grade ='F';
END IF;
CALL insert_grade(id,grade);
END;
$$
LANGUAGE plpgsql;
CREATE OR REPLACE PROCEDURE insert_grade(sid INT ,sgrade CHAR)
LANGUAGE plpgsql
AS $$
BEGIN
UPDATE studentmark SET grade=sgrade WHERE id=sid;
END;
$$;
```

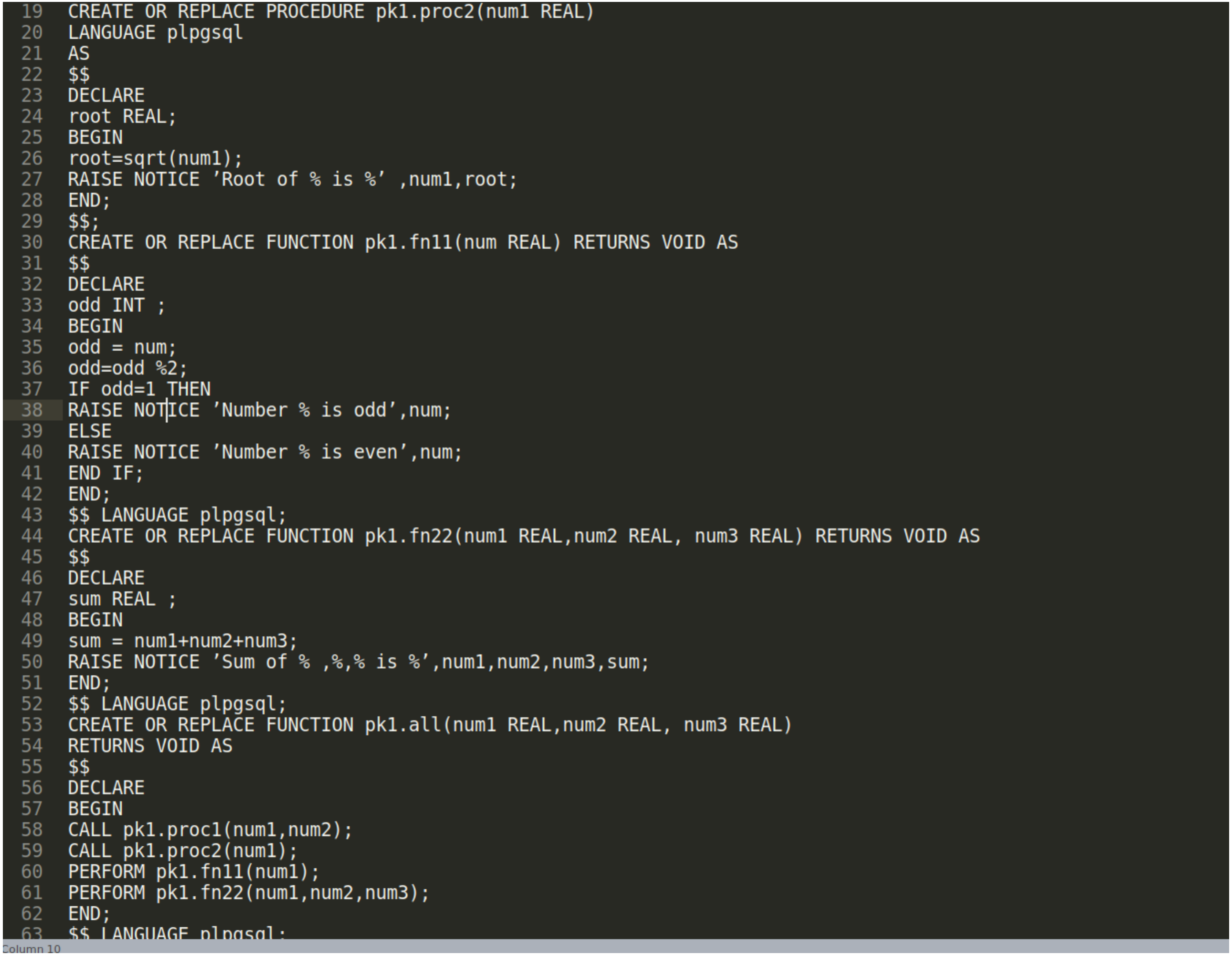
```
postgres=# select * from studentmark;
 id | name | m1 | m2 | m3 | total | grade
----+------+----+----+----+-------+-------
  1 | sau  | 90 | 90 | 90 |   270 | A
(1 row)
```

Figure 3:

4. Create a package pk1 consisting of the following functions and procedures Procedure proc1 to find the sum, average and product of two numbers Procedure proc2 to find the square root of a number Function named fn11 to check whether a number is even or not A function named fn22 to find the sum of 3 numbers Use this package in a PL/SQL program. Call the functions f11, f22 and procedures pro1, pro2 within the program and display their results

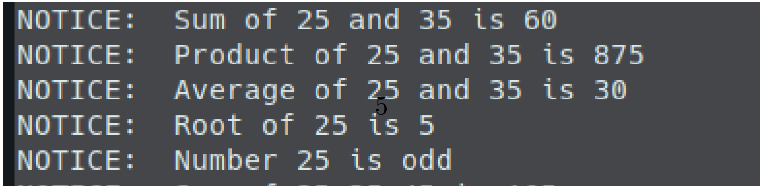


```
CREATE SCHEMA pk1;
CREATE OR REPLACE PROCEDURE pk1.proc1(num1 REAL,num2 REAL)
LANGUAGE plpgsql
AS
$$
DECLARE
sum REAL;
average REAL;
prod REAL;
BEGIN
sum = num1+num2;
prod = num1*num2;
average = (num1 + num2)/2;
RAISE NOTICE 'Sum of % and % is %' ,num1,num2,sum;
RAISE NOTICE 'Product of % and % is %' ,num1,num2 ,prod;
RAISE NOTICE 'Average of % and % is %' ,num1,num2,average;
END;
$$;
CREATE OR REPLACE PROCEDURE pk1.proc2(num1 REAL)
LANGUAGE plpgsql
AS
$$
DECLARE
root REAL;
BEGIN
root=sqrt(num1);
RAISE NOTICE 'Root of % is %' ,num1,root;
END;
$$;
CREATE OR REPLACE FUNCTION pk1.fn11(num REAL) RETURNS VOID AS
$$
DECLARE
odd INT ;
BEGIN
odd = num;
odd=odd %2;
IF odd=1 THEN
RAISE NOTICE 'Number % is odd',num;
ELSE
RAISE NOTICE 'Number % is even',num;
END IF;
END;
$$ LANGUAGE plpgsql;
CREATE OR REPLACE FUNCTION pk1.fn22(num1 REAL,num2 REAL, num3 REAL) RETURNS VOID AS
$$
```



```
CREATE OR REPLACE PROCEDURE pk1.proc2(num1 REAL)
LANGUAGE plpgsql
AS
$$
DECLARE
root REAL;
BEGIN
root=sqrt(num1);
RAISE NOTICE 'Root of % is %' ,num1,root;
END;
$$;
CREATE OR REPLACE FUNCTION pk1.fn11(num REAL) RETURNS VOID AS
$$
DECLARE
odd INT ;
BEGIN
odd = num;
odd=odd %2;
IF odd=1 THEN
RAISE NOTICE 'Number % is odd',num;
ELSE
RAISE NOTICE 'Number % is even',num;
END IF;
END;
$$ LANGUAGE plpgsql;
CREATE OR REPLACE FUNCTION pk1.fn22(num1 REAL,num2 REAL, num3 REAL) RETURNS VOID AS
$$
```

```
DECLARE
sum REAL ;
BEGIN
sum = num1+num2+num3;
RAISE NOTICE 'Sum of % ,%,% is %',num1,num2,num3,sum;
END;
$$ LANGUAGE plpgsql;
CREATE OR REPLACE FUNCTION pk1.all(num1 REAL,num2 REAL, num3 REAL)
RETURNS VOID AS
$$
DECLARE
BEGIN
CALL pk1.proc1(num1,num2);
CALL pk1.proc2(num1);
PERFORM pk1.fn11(num1);
PERFORM pk1.fn22(num1,num2,num3);
END;
$$ LANGUAGE plpgsql;
```



```
NOTICE:  Sum of 25 and 35 is 60
NOTICE:  Product of 25 and 35 is 875
NOTICE:  Average of 25 and 35 is 30
NOTICE:  Root of 25 is 5
NOTICE:  Number 25 is odd
```

3 Result

The PL/SQL program was executed successfully and the output was obtained.