College of Engineering, Trivandrum

Department of Computer Science and Engineering

CS333 APPLICATION SOFTWARE DEVELOPMENT LAB

LABORATORY REPORT 12

Procedures, Functions and Packages

Student Name	Student ID
1. Justine Biju(S5)	170445(Roll No:37)

Submission Date : 19/08/2019

1 Introduction

1. Function
   A function is a subprogram that computes a value. Functions and procedures are structured alike, except that functions have a RETURN clause. The syntax is as follows:

   ```
   FUNCTION name [(parameter[, parameter, ...])] RETURN
   datatype IS [local declarations]
   BEGIN
   executable
   statements
   [EXCEPTION
   exception
   handlers] END
   [name];
   ```

2. Procedure
   A procedure is a subprogram that performs a specific action.
   The syntax is as follows:

   ```
   PROCEDURE name [(parameter[,
   parameter, ...])] IS [local declarations]
   BEGIN
   executable
   statements
   [EXCEPTION
   exception
   handlers] END
   [name];
   ```

3. Procedure
   A package is a schema object that groups logically related PL/SQL types, items and subprograms.
   The syntax is as follows:

   ```
   CREATE [OR REPLACE] PACKAGE
   package_name [AUTHID {CURRENT_USER |
   DEFINER}] {IS | AS} [type_definition
   [type_definition] ...]
   [cursor_spec [cursor_spec] ...]
   [item_declaration
   [item_declaration] ...]
   [{subprogram_spec | call_spec} [{subprogram_spec |
   call_spec}]...] END [package_name];
   ```

All these are supported by PostgreSQL and can be implemented in a similar fashion.

2 Questions

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

```
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: Question 1

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

```
CREATE OR PROCEDURE FUNCTION 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;
```

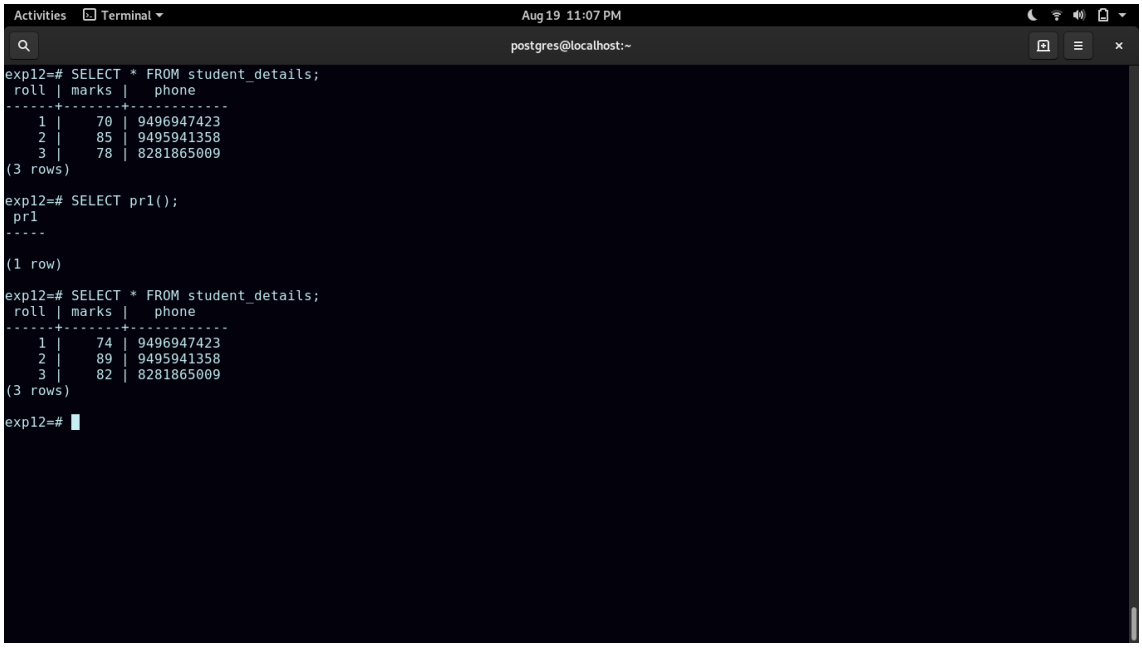

Figure 2: Question 2

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 PROCEDURE porf() AS $$
DECLARE
    sum INTEGER;
    c1 CURSOR FOR SELECT * FROM student;
    rec RECORD;
BEGIN
    OPEN c1;
    LOOP
        FETCH c1 INTO rec;
        EXIT WHEN NOT FOUND;
        sum = (rec.m1 + rec.m2 + rec.m3)/3;
        IF sum > 40 THEN
            UPDATE student
            SET total = m1+m2+m3, grade = 'P'
            WHERE CURRENT OF c1;
        ELSE
            UPDATE student
            SET total = m1+m2+m3, grade = 'F'
            WHERE CURRENT OF c1;
        END IF;
    END LOOP;
    CLOSE c1;
END;
$$ LANGUAGE plpgsql;
```

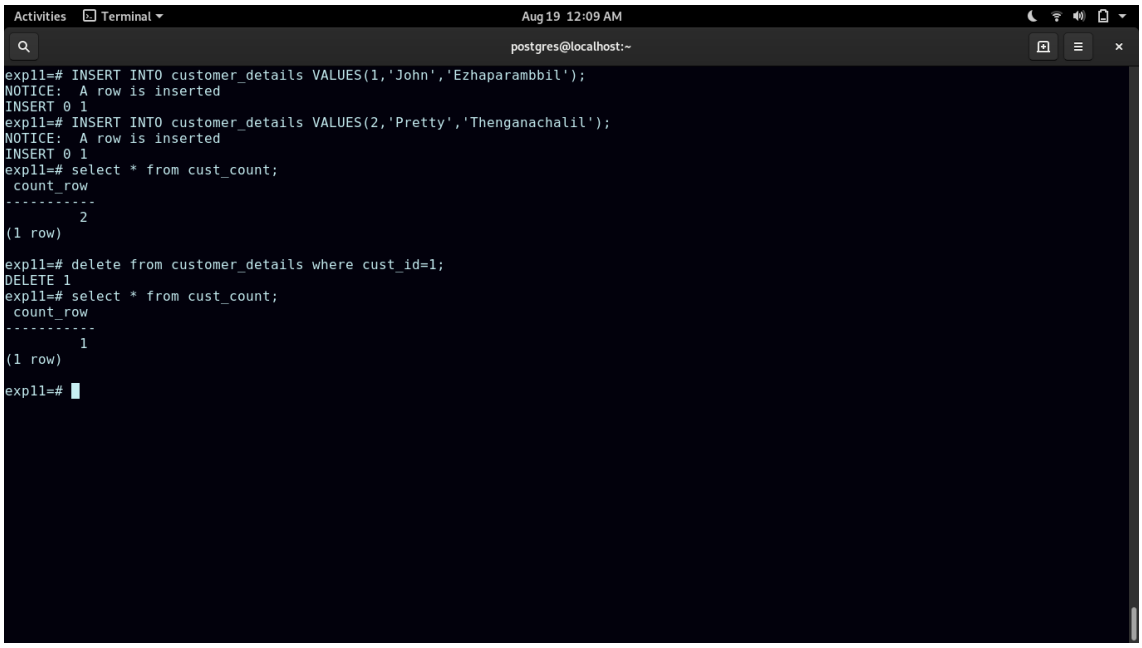

Figure 3: Question 3

4. 4.Create a package pk1 consisting of the following functions and procedures proc1, proc2, fn11, fn22.

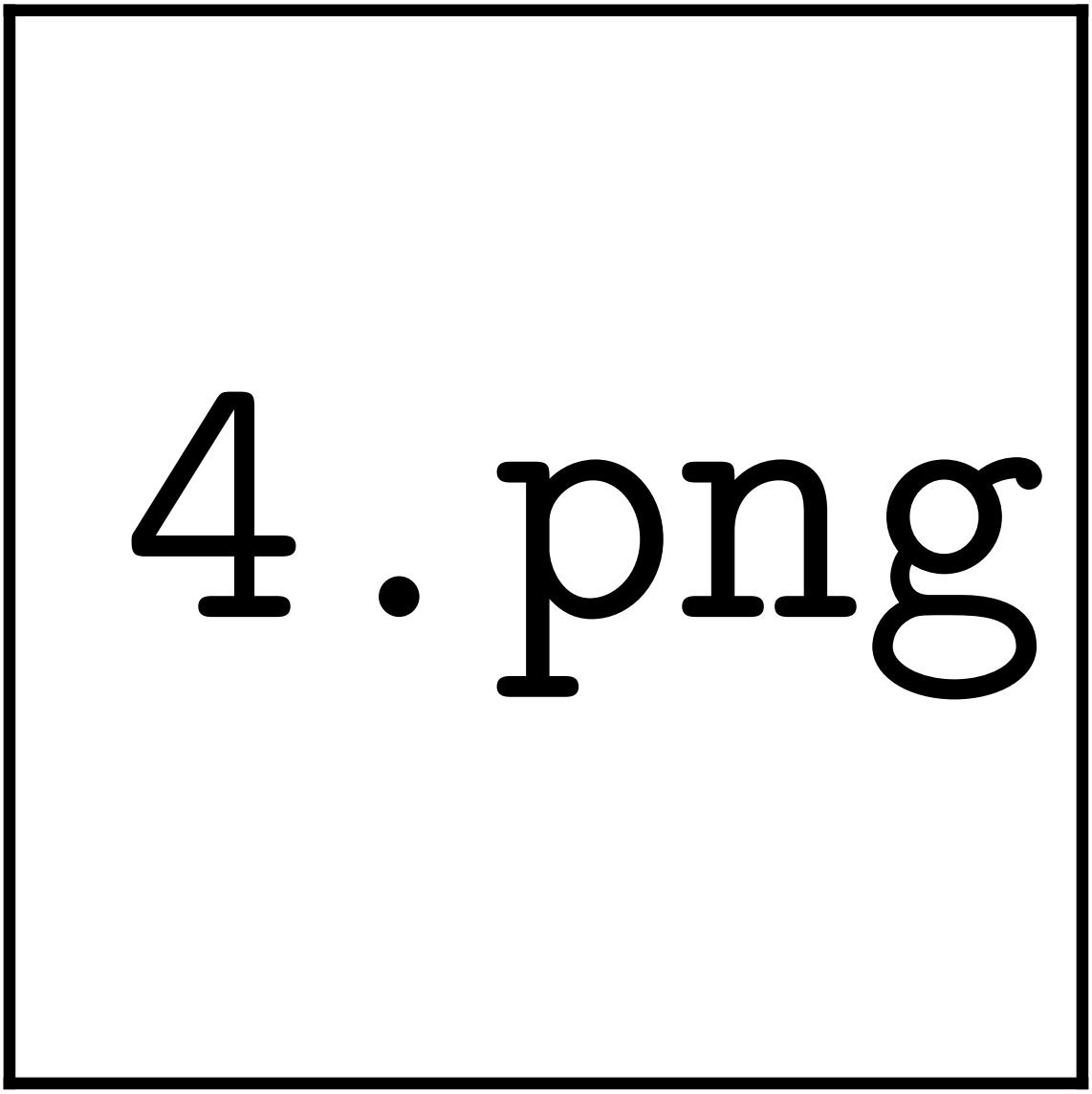

Figure 4: Question 4

3 Result

- Successfully implemented Functions, Procedures and Packages in PostgreSQL.