|  |  |  |  |
| --- | --- | --- | --- |
| TE Comp Roll number : | | | |
| Experiment no. : 8 Date of Implementation : | | | |
| Aim : To implement PL/pgSQL | | | |
| Tool Used : PostgreSQL | | | |
| Related Course outcome : At the end of the course, Students will be able to Use  SQL : Standard language of relational database | | | |
| **Rubrics for assessment of Experiment:**   |  |  |  |  | | --- | --- | --- | --- | | Indicator | Poor | Average | Good | | Timeliness   * Maintains assignment deadline (3) | Assignment not done (0) | One or More than One week late (1-2) | Maintains deadline (3) | | Completeness and neatness   * Complete all parts of assignment(3) | N/A | < 80% complete (1-2) | 100% complete (3) | | Originality   * Extent of plagiarism(2) | Copied it from someone else(0) | At least few questions have been done without copying(1) | Assignment has been solved completely without copying (2) | | Knowledge   * In depth knowledge of the assignment(2) | Unable to answer 2 questions(0) | Unable to answer 1 question (1) | Able to answer 2 questions (2) | | | | |
| **Assessment Marks :**   |  |  | | --- | --- | | Timeliness |  | | Completeness and neatness |  | | Originality |  | | Knowledge |  | | Total |  | | | | |
| **Total : (Out of 10)** | | | |
| **Teacher's Sign :** | | | |
| EXPERIMENT 8 | PL/pgSQL |
| Aim | To implement PL/pgSQL |
| Tools | PostgreSQL  <http://www.postgresqltutorial.com/postgresql-stored-procedures/> |
| Procedure | PL/pgSQL is a loadable procedural language for the Postgres database system. This package was originally written by Jan Wieck. The design goals of PL/pgSQL were to create a loadable procedural language that can be used to create functions and trigger procedures, adds control structures to the SQL language.  Structure of PL/pgSQL  PL/pgSQL is a block-structured language. The complete text of a function definition must be a block. A block is defined as:  [<<label>>]  [ DECLARE Declarations ]  BEGIN  statements  END [label];  Each declaration and each statement within a block is terminated by a semicolon. A block that appears within another block must have a semicolon after END , as shown above; however the final END that concludes a function body does not require a semicolon   |  |  | | --- | --- | | IF boolean-expression THEN  statements  END IF; | IF boolean-expression  THEN  statements  ELSE statements  END IF; |  |  |  | | --- | --- | | WHILE boolean-expression  LOOP  statements  END LOOP [label]; | FOR name IN [ REVERSE ] expression..expression  [ BY expression] LOOP  statements  END LOOP [label];  FOR i IN 1..10 LOOP  -- i will take on the values 1,2,3,4,5,6,7,8,9,10 within the loop  END LOOP;  FOR i IN REVERSE 10..1 LOOP  -- i will take on the values 10,9,8,7,6,5,4,3,2,1 within the loop  END LOOP;  FOR i IN REVERSE 10..1 BY 2 LOOP  -- i will take on the values 10,8,6,4,2 within the loop  END LOOP; | |
|  |
| **Procedure** | 1. Write a block to display sum of digits of a three digit number 2. Write a block to display square of 1 to 10 3. Write a block to display Fibonacci series upto 8th term (start with 0,1) |
| **Post Lab Questions:** | 1. Give advantages of PLSQL vs SQL 2. Explain data types of PgSQL |