**EXPERIMENT-10**

**Title: 10. To understand the concepts of function and procedure in PL/SQL.**

**Objective:** Students will be able to implement the Pl/SQL programs using function and procedure.

**Theory:**

In PL/SQL, functions and procedures are both named PL/SQL blocks that contain one or more SQL or PL/SQL statements. They are used to encapsulate a set of related tasks and can be invoked by other PL/SQL blocks, SQL statements, or external applications.

**Procedure**:

* A procedure is a named PL/SQL block that performs one or more specific tasks.
* It can have zero or more parameters that can be passed to it when it is called.
* Procedures do not return any value explicitly. They are typically used to perform actions like data manipulation, transaction control, or business logic implementation.
* Procedures are invoked using the **CALL** or **EXECUTE** statement, or by simply using their name in SQL\*Plus, SQL Developer, or other SQL client tools.

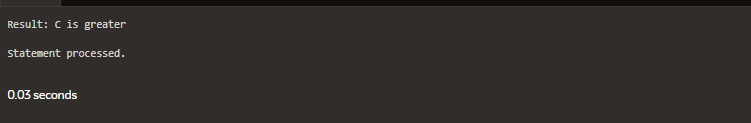
**Function**:

* A function is a named PL/SQL block that performs a specific task and returns a single value.
* It must return a value of a specific data type defined in its declaration.
* Functions can have zero or more parameters, and they can be passed both IN and OUT parameters.
* Functions are invoked within SQL statements or PL/SQL blocks where an expression can be evaluated, such as SELECT statements, assignments, or conditions.
* Example:

Implement the experiments of lab 9 using functions and procedures

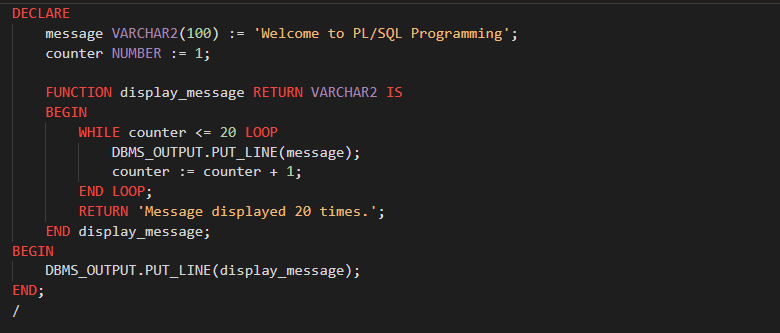
1. Write a PL/SQL code to accept the value of A, B & C display which is greater.

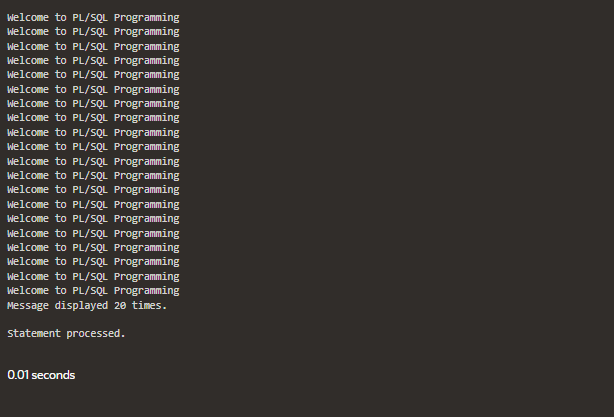
**Using Function:**



1. Using PL/SQL Statements create a simple loop that display message “Welcome to PL/SQL Programming” 20 times.

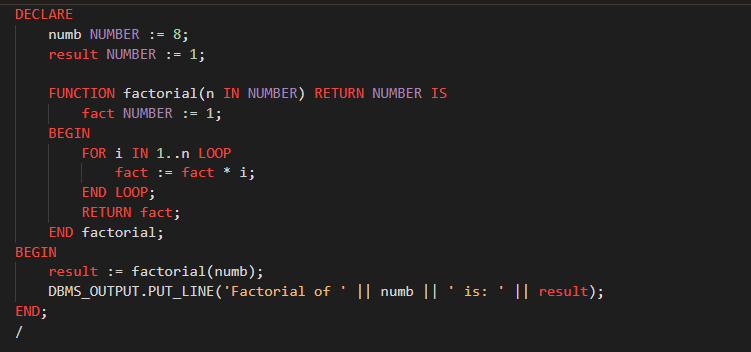
**Using Function:**





1. Write a PL/SQL code block to find the factorial of a number.

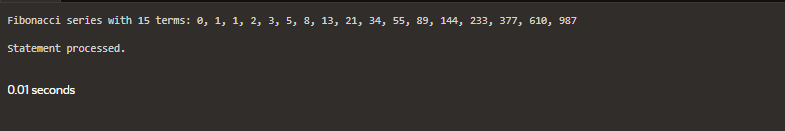
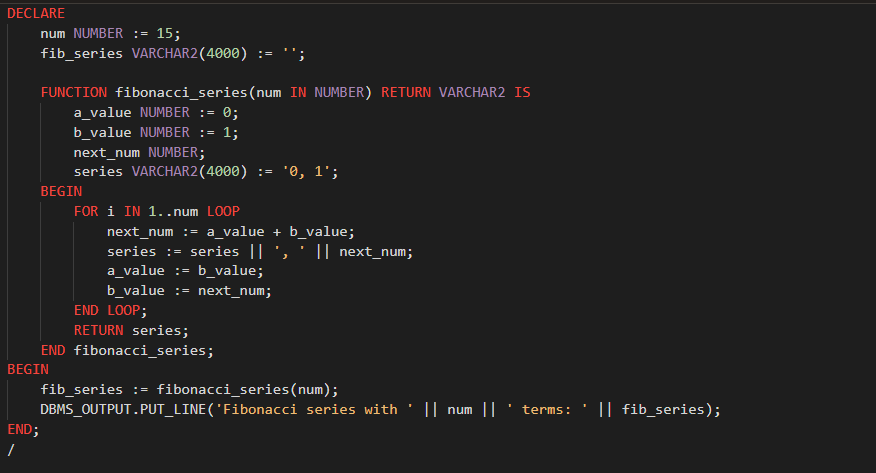
**Using Function:**





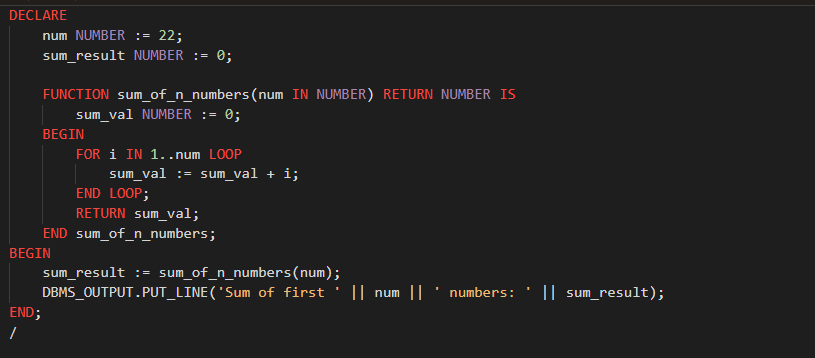
1. Write a PL/SQL program to generate Fibonacci series.

**Using Function:**



1. Write a PL/SQL code to find the sum of first N numbers.

**Using Function:**



**EXPERIMENT-11**

**Title: 11. To understand the concepts of implicit and explicit cursor.**

**Objective:** Students will be able to implement the concept of implicit and explicit cursor.

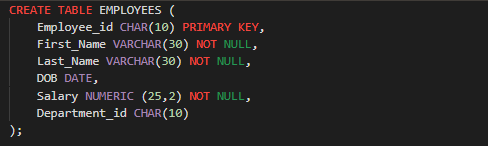
**Theory:**

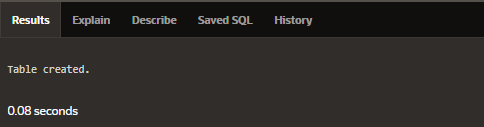
In PL/SQL, a cursor is a named control structure used to retrieve data row by row from the result set of a SELECT statement. Cursors are particularly useful when dealing with queries that return multiple rows of data.

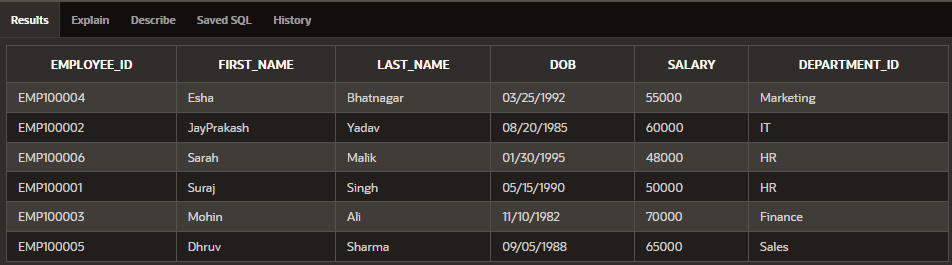
There are two types of cursors in PL/SQL:

1. **Implicit Cursor**:
   * Implicit cursors are automatically created by Oracle to handle the result sets of SQL queries that are not explicitly assigned to a cursor variable.
   * They are mainly used for SELECT, INSERT, UPDATE, and DELETE statements.
   * The attributes of an implicit cursor, such as **%ROWCOUNT**, **%FOUND**, **%NOTFOUND**, and **%ISOPEN**, provide information about the execution status of the SQL statement.
2. **Explicit Cursor:**

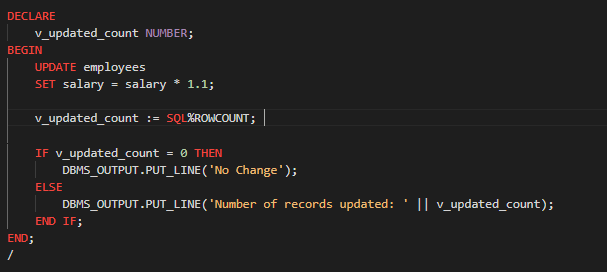
* Explicit cursors are user-defined cursors that allow for more control over the result set processing.
* They are explicitly declared, opened, fetched, and closed by the programmer.
* Explicit cursors are useful when you need to process query results row by row or when you want to execute a dynamic SQL statement.
* They are typically used in scenarios where more control or flexibility is required, such as iterating through query results, bulk processing, or handling exceptions.
* Example of an explicit cursor:

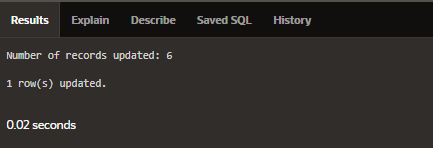


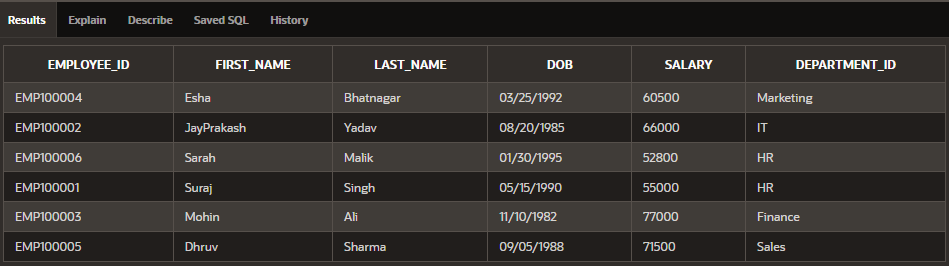




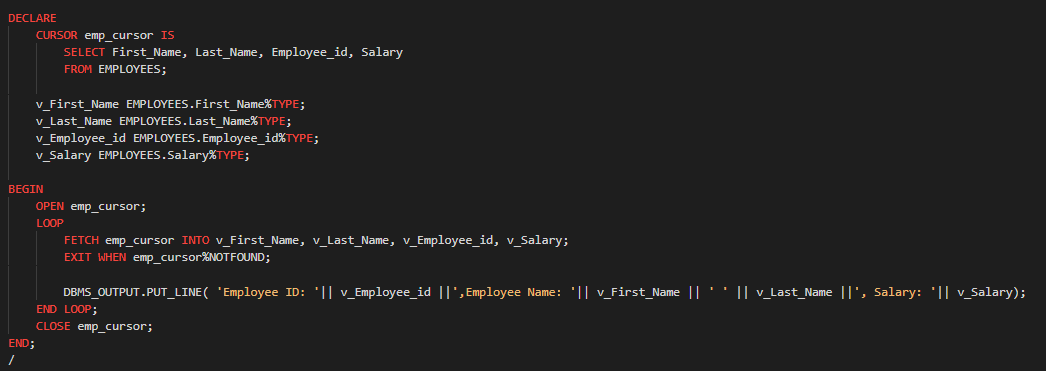
1. Using implicit cursor update the salary by an increase of 10% for all the records in EMPLOYEES table, and finally display how many records have been updated. If no records exist display the message “**No Change**”.

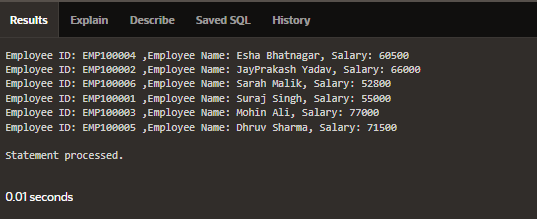




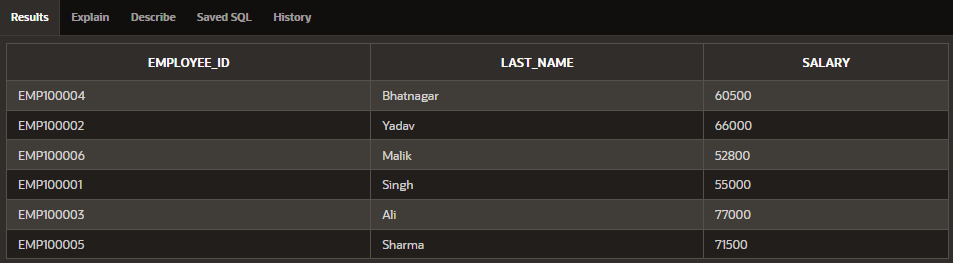
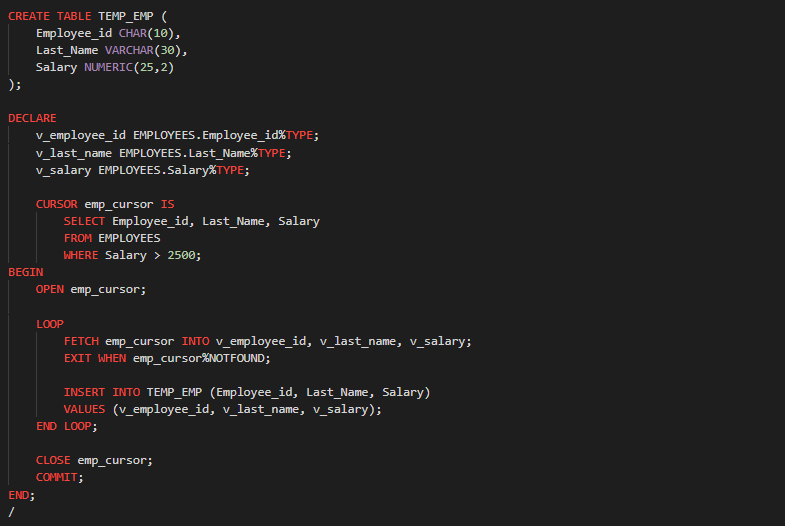


1. Using explicit cursor fetch the employee name, employee\_id and salary of all the records from EMPLOYEES table.





1. Using explicit cursor Insert the records from EMPLOYEES table for the columns employee\_id, Last\_Name and salary for those records whose salary exceeds 2500 into a new table TEMP\_EMP.

****