Subject: Database Management System & SQL Laboratory

EXPERIMENT NO.: 09

TITLE: Write a code to implement Stored Procedures on DB.

LEARNING OBJECTIVES:

1. To study the fundamental concepts of database management.
2. To learn the basic issues of transaction processing and concurrency control.
3. To learn a powerful, flexible and scalable general-purpose distributed database.

THEORY:

Procedure is a subprogram used to perform a specific action. A subprogram is a named block of PL/SQL. There are two types of subprograms in PL/SQL namely Procedures and Functions. Every subprogram will have a declarative part, an executable part or body, and an exception handling part, which is optional. Declarative part contains variable declarations. Body of a subprogram contains executable statements of SQL and PL/SQL. Statements to handle exceptions are written in exception part. Procedure specification begins with CREATE and ends with procedure name or parameters list.
Procedures that do not take parameters are written without a parenthesis. The body of the procedure starts after the keyword IS or AS and ends with keyword END.

Syntax for Procedure:

CREATE [OR REPLACE] PROCEDURE procedure_name [(parameter_name [IN|OUT|IN OUT] type [, ...])]

{ISAS}
BEGIN
procedure
body
EXCEPTION
Exception handling
END procedure_name;

Procedure is created using CREATE PROCEDURE statement.

OR REPLACE specifies the procedure is to replace an existing procedure if present. A procedure may be passed multiple parameters. IN OUT | IN OUT specifies the mode of the parameter. Type specifies the data type of the parameter.

IN -The parameter can be referenced by the procedure or function. The value of the parameter cannot be overwritten by the procedure or function.

OUT- The parameter cannot be referenced by the procedure or function, but the value of the parameter can be overwritten by the procedure or function.
IN OUT- The parameter can be referenced by the procedure or function and the value of the parameter can be overwritten by the procedure or function.

A function is a named PL/SQL Block which is similar to a procedure. The major difference between a procedure and a function is, a function must always return a value, but a procedure may or may not return a value.

Syntax for Function:

CREATE [OR REPLACE] FUNCTION function_name [parameters]
RETURN return_datatype; IS Declaration_section BEGIN Execution_section Return return_variable; EXCEPTION exception section Return return_variable; END;

Example :

- Write a PL/SQL Procedure using IN OUT type paramter

```
CREATE OR REPLACE PROCEDURE min (a IN number , b IN number , c OUTnumber)
IS
BEGIN
  IF a < b THEN      c:=a;
END IF;
END min;
```

Procedure call:

```
DECLARE
   x number:=40;     y
number:=50;     z
number;
BEGIN
   min(x,y,z);
    dbms_output.put_line('Minimum of (40,50): ' || z);
END;
```

· Write Function to Calculate Factorial of given number

Ans:

```
CREATE OR REPLACE FUNCTION fact (a IN number)
        i number;       fct number :=1;         BEGIN
            for I IN 1..a LOOP
            fct:=fct*1;
            END LOOP
            Return fct;
        END fct;
```

Function call:

```
DECLARE
a number ;
        b number;
BEGIN          a:=:a;
        b:=:fct(a);
    dbms_output.put_line('Factorial of '|| a || 'is '|| b);
END;
```

NOTE : Please ensure that you also add the Industrial Problem (2) in your submission/document along with the existing content.

References for Theory:

- Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", MGH
- Connally T, Begg C., "Database Systems", Pearson Education

- Raghurama Krishan, "Database Management Systems", McGrawHill
- S.K.Singh, "Database Systems : Concepts, Design and Application", Pearson

CONCLUSION: