

EXPERIMENT 14

EXPERIMENT-14

Title: 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.

- 1) Implement the above experiments of PL/SQL using functions and procedures.

CODE:

```
-- 1) Write a PL/SQL code to accept the value of A, B & C display which is greater.
CREATE OR REPLACE FUNCTION find_greatest(a INTEGER, b INTEGER, c INTEGER)
RETURNS TEXT AS $$
DECLARE
    greatest_val INTEGER;
BEGIN
    IF a >= b AND a >= c THEN
        greatest_val := a;
    ELSIF b >= a AND b >= c THEN
        greatest_val := b;
    ELSE
        greatest_val := c;
    END IF;
    RETURN 'The greatest number is: ' || greatest_val;
END;
$$ LANGUAGE plpgsql;

SELECT find_greatest(5, 9, 3);
```

```
-- 2) Using PL/SQL Statements create a simple loop that displays the message "Welcome to PL/SQL Programming" 20 times.
CREATE OR REPLACE PROCEDURE display_message()
LANGUAGE plpgsql AS $$
DECLARE
    counter INTEGER := 1;
BEGIN
    WHILE counter <= 20 LOOP
        RAISE NOTICE 'Welcome to PL/SQL Programming';
        counter := counter + 1;
    END LOOP;
END;
$$;

CALL display_message();
```

```
CREATE OR REPLACE FUNCTION factorial(n INTEGER)
RETURNS INTEGER AS $$
DECLARE
    result INTEGER := 1;
BEGIN
    IF n < 0 THEN
        RAISE EXCEPTION 'Factorial is not defined for negative numbers';
    ELSIF n = 0 THEN
        RETURN 1;
    ELSE
        FOR i IN 1..n LOOP
            result := result * i;
        END LOOP;
    END IF;
    RETURN result;
END;
$$ LANGUAGE plpgsql;

SELECT factorial(5);
```

```
CREATE OR REPLACE FUNCTION generate_fibonacci(n INTEGER)
RETURNS TABLE(fibonacci INTEGER) AS $$
DECLARE
    a INTEGER := 0;
    b INTEGER := 1;
    temp INTEGER;
    i INTEGER := 0;
BEGIN
    RETURN QUERY SELECT a;
    RETURN QUERY SELECT b;


    FOR i IN 3..n LOOP
        temp := a + b;
        RETURN QUERY SELECT temp;
        a := b;
        b := temp;
    END LOOP;
END;
$$ LANGUAGE plpgsql;

SELECT * FROM generate_fibonacci(10);
```

```
CREATE OR REPLACE FUNCTION sum_of_first_n(n INTEGER)
RETURNS INTEGER AS $$
DECLARE
    sum INTEGER := 0;
BEGIN
    FOR i IN 1..n LOOP
        sum := sum + i;
    END LOOP;
    RETURN sum;
END;
$$ LANGUAGE plpgsql;

SELECT sum_of_first_n(10);
```

OUTPUT:


	find_greatest text	
1	The greatest number is: 9	


```

NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
NOTICE: Welcome to PL/SQL Programming
CALL


```

Query returned successfully in 53 msec.

	factorial integer 
1	120

	sum_of_first_n integer 
1	55

S

	fibonacci integer	
1	0	
2	1	
3	1	
4	2	
5	3	
6	5	
7	8	
8	13	
9	21	
10	34	