**EXPERIMENT 13**

1. **FACTORIAL**

SQL> declare

f number:=1;

n number:=&n;

begin

while n>0

loop

f:=n\*f;

n:=n-1;

end loop;

dbms\_output.put\_line('factorial is '||f);

end;

/

SQL> /

Enter value for n: 5

old 3: n number:=&n;

new 3: n number:=5;

Factorial equal to 120

PL/SQL procedure successfully completed.

SQL> /

Enter value for n: 6

old 3: n number:=&n;

new 3: n number:=6;

Factorial equal to720

PL/SQL procedure successfully completed.

1. **GREATEST AMONG 3 NUMBERS**

DECLARE

    a NUMBER :=&a;

    b NUMBER :=&b;

    c NUMBER :=&c;

BEGIN

    IF a > b

       AND a > c THEN

      dbms\_output.Put\_line('Greatest number is '

                           ||a);

    ELSIF b > a

          AND b > c THEN

      dbms\_output.Put\_line('Greatest number is '

                           ||b);

    ELSE

      dbms\_output.Put\_line('Greatest number is '

                           ||c);

    END IF;

END;

/

Enter value for a: 10

old 2: a NUMBER :=&a;

new 2: a NUMBER :=10;

Enter value for b: 5

old 3: b NUMBER :=&b;

new 3: b NUMBER :=5;

Enter value for c: 20

old 4: c NUMBER :=&c;

new 4: c NUMBER :=20;

Greatest number is 20

PL/SQL procedure successfully completed.

1. **CALCULATOR**

DECLARE

num1 NUMBER;

num2 NUMBER;

operator CHAR(1);

result NUMBER;

division\_by\_zero EXCEPTION;

PRAGMA EXCEPTION\_INIT(division\_by\_zero, -1476);

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Enter first number:');

num1 := &num1;

DBMS\_OUTPUT.PUT\_LINE('Enter second number:');

num2 := &num2;

DBMS\_OUTPUT.PUT\_LINE('Enter operator (+, -, \*, /):');

operator := '&operator';

CASE operator

WHEN '+' THEN

result := num1 + num2;

WHEN '-' THEN

result := num1 - num2;

WHEN '\*' THEN

result := num1 \* num2;

WHEN '/' THEN

IF num2 = 0 THEN

RAISE division\_by\_zero;

ELSE

result := num1 / num2;

END IF;

ELSE

DBMS\_OUTPUT.PUT\_LINE('Invalid operator!');

RETURN;

END CASE;

DBMS\_OUTPUT.PUT\_LINE('Result: ' || result);

EXCEPTION

WHEN division\_by\_zero THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Division by zero is not allowed.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

END;

/

/

Enter value for num1: 5

old 12: num1 := &num1;

new 12: num1 := 5;

Enter value for num2: 4

old 14: num2 := &num2;

new 14: num2 := 4;

Enter value for operator: +

old 16: operator := '&operator';

new 16: operator := '+';

Enter first number:

Enter second number:

Enter operator (+, -, \*, /):

Result: 9

PL/SQL procedure successfully completed.

SQL> /

Enter value for num1: 6

old 12: num1 := &num1;

new 12: num1 := 6;

Enter value for num2: 0

old 14: num2 := &num2;

new 14: num2 := 0;

Enter value for operator: /

old 16: operator := '&operator';

new 16: operator := '/';

Enter first number:

Enter second number:

Enter operator (+, -, \*, /):

Error: Division by zero is not allowed.

PL/SQL procedure successfully completed.

SQL> /

Enter value for num1: 10

old 12: num1 := &num1;

new 12: num1 := 10;

Enter value for num2: 5

old 14: num2 := &num2;

new 14: num2 := 5;

Enter value for operator: -

old 16: operator := '&operator';

new 16: operator := '-';

Enter first number:

Enter second number:

Enter operator (+, -, \*, /):

Result: 5

PL/SQL procedure successfully completed.

SQL> /

Enter value for num1: 6

old 12: num1 := &num1;

new 12: num1 := 6;

Enter value for num2: 5

old 14: num2 := &num2;

new 14: num2 := 5;

Enter value for operator: \*

old 16: operator := '&operator';

new 16: operator := '\*';

Enter first number:

Enter second number:

Enter operator (+, -, \*, /):

Result: 30

PL/SQL procedure successfully completed.

1. **FIBONACCI SERIES**

DECLARE

a NUMBER := &a;

fib1 NUMBER := 0;

fib2 NUMBER := 1;

next\_fib NUMBER;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Fibonacci Series:');

FOR i IN 1..a LOOP

DBMS\_OUTPUT.PUT(fib1 || ' ');

next\_fib := fib1 + fib2;

fib1 := fib2;

fib2 := next\_fib;

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('');

END;

/

Enter value for a: 10

old 2: a NUMBER := &a;

new 2: a NUMBER := 10;

Fibonacci Series:

0 1 1 2 3 5 8 13 21 34

PL/SQL procedure successfully completed.