**IMPLEMENTATION OF CONTROL STATEMENTS USING PL/SQL**

**PL/SQL program to find factorials of a number**

SQL> set serveroutput on;

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;

/

**OUTPUT**

Enter value for n: 5

old 3: n number: = &n;

new 3: n number: = 5;

factorial is 120

PL/SQL procedure successfully completed.

**PL/SQL program to find the greatest of three numbers**

SQL> declare

a number: =&n;

b number: =&n;

c number: =&n;

begin

if a>b and a>c then

dbms\_output.put\_line (a||' is greater than '||b||' and '||c);

elsif b>a and b>c then

dbms\_output.put\_line (b||' is greater than '||a||' and '||c);

else

dbms\_output.put\_line (c||' is greater than '||a||' and '||b);

end if;

end;

/

**OUTPUT**

Enter value for a: 5

old 2: a NUMBER: = &a;

new 2: a NUMBER: = 5;

Enter value for b: 4

old 3: b NUMBER: = &b;

new 3: b NUMBER: = 4;

Enter value for c: 7

old 4: c NUMBER: = &c;

new 4: c NUMBER: = 7;

7 is greater than 5 and 4

PL/SQL procedure successfully completed.

**PL/SQL program to implement a calculator**

SQL> declare

num1 NUMBER: =&num1;

num2 NUMBER: =&num2;

operator CHAR (1): ='&operator';

result number;

BEGIN

IF operator = '+' THEN

result: = num1 + num2;

ELSIF operator = '-' THEN

result: = num1 - num2;

ELSIF operator = '\*' THEN

result: = num1 \* num2;

ELSIF operator = '/' THEN

IF num2! = 0 THEN

result: = num1 / num2;

ELSE

dbms\_output.put\_line ('Error: Division by zero is not allowed.');

END IF;

ELSE

dbms\_output.put\_line ('Error: Invalid operator.');

END IF;

dbms\_output.put\_line ('Result: ' || result);

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line ('An error occurred: ' || SQLERRM);

end;

/

**OUTPUT**

Enter value for num1: 5

old 2: num1 NUMBER: =&num1;

new 2: num1 NUMBER: =5;

Enter value for num2: 6

old 3: num2 NUMBER: =&num2;

new 3: num2 NUMBER: =6;

Enter value for operator: +

old 4: operator CHAR (1): ='&operator';

new 4: operator CHAR (1): ='+';

Result: 11

PL/SQL procedure successfully completed.

Enter value for num1: 5

old 2: num1 NUMBER: =&num1;

new 2: num1 NUMBER: =5;

Enter value for num2: 6

old 3: num2 NUMBER: =&num2;

new 3: num2 NUMBER: =6;

Enter value for operator: -

old 4: operator CHAR (1): ='&operator';

new 4: operator CHAR (1): ='-';

Result: -1

PL/SQL procedure successfully completed.

Enter value for num1: 5

old 2: num1 NUMBER: =&num1;

new 2: num1 NUMBER: =5;

Enter value for num2: 6

old 3: num2 NUMBER: =&num2;

new 3: num2 NUMBER: =6;

Enter value for operator: \*

old 4: operator CHAR (1): ='&operator';

new 4: operator CHAR (1): ='\*';

Result: 30

PL/SQL procedure successfully completed.

Enter value for num1: 60

old 2: num1 NUMBER: =&num1;

new 2: num1 NUMBER: =60;

Enter value for num2: 5

old 3: num2 NUMBER: =&num2;

new 3: num2 NUMBER: =5;

Enter value for operator: /

old 4: operator CHAR (1): ='&operator';

new 4: operator CHAR (1): ='/';

Result: 12

PL/SQL procedure successfully completed.

Enter value for num1: 5

old 2: num1 NUMBER: =&num1;

new 2: num1 NUMBER: =5;

Enter value for num2: 0

old 3: num2 NUMBER: =&num2;

new 3: num2 NUMBER: =0;

Enter value for operator: /

old 4: operator CHAR (1): ='&operator';

new 4: operator CHAR (1): ='/';

Error: Division by zero is not allowed.

Result:

PL/SQL procedure successfully completed.

**PL/SQL program to generate Fibonacci series**

SQL> declare

numofterms NUMBER:= &numofterms;

a NUMBER: =0;

b NUMBER: =1;

nextterm NUMBER;

BEGIN

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

FOR i IN 1..numofterms LOOP

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

nextterm: = a + b;

a: = b;

b: = nextterm;

END LOOP;

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

EXCEPTION

WHEN OTHERS THEN

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

END;

/

**OUTPUT**

Enter value for numofterms: 5

old 2: numofterms NUMBER: = &numofterms;

new 2: numofterms NUMBER: = 5;

Fibonacci Series:

0 1 1 2 3

PL/SQL procedure successfully completed.