

# PL/SQL - NESTED LOOPS

[http://www.tutorialspoint.com/plsql/plsql\\_nested\\_loops.htm](http://www.tutorialspoint.com/plsql/plsql_nested_loops.htm)

Copyright © tutorialspoint.com

PL/SQL allows using one loop inside another loop. Following section shows few examples to illustrate the concept.

The syntax for a nested basic LOOP statement in PL/SQL is as follows:

```
LOOP
    Sequence of statements1
    LOOP
        Sequence of statements2
    END LOOP;
END LOOP;
```

The syntax for a nested FOR LOOP statement in PL/SQL is as follows:

```
FOR counter1 IN initial_value1 .. final_value1 LOOP
    sequence_of_statements1
    FOR counter2 IN initial_value2 .. final_value2 LOOP
        sequence_of_statements2
    END LOOP;
END LOOP;
```

The syntax for a nested WHILE LOOP statement in Pascal is as follows:

```
WHILE condition1 LOOP
    sequence_of_statements1
    WHILE condition2 LOOP
        sequence_of_statements2
    END LOOP;
END LOOP;
```

## Example:

The following program uses a nested basic loop to find the prime numbers from 2 to 100:

```
DECLARE
    i number(3);
    j number(3);
BEGIN
    i := 2;
    LOOP
        j:= 2;
        LOOP
            exit WHEN ((mod(i, j) = 0) or (j = i));
            j := j +1;
        END LOOP;
        IF (j = i ) THEN
            dbms_output.put_line(i || ' is prime');
        END IF;
        i := i + 1;
        exit WHEN i = 50;
    END LOOP;
END;
```

When the above code is executed at SQL prompt, it produces the following result:

```
2 is prime
3 is prime
5 is prime
7 is prime
```

```
11 is prime  
13 is prime  
17 is prime  
19 is prime  
23 is prime  
29 is prime  
31 is prime  
37 is prime  
41 is prime  
43 is prime  
47 is prime
```

PL/SQL procedure successfully completed.