

PL/SQL - EXIT STATEMENT

http://www.tutorialspoint.com/plsql/plsql_exit_statement.htm

Copyright © tutorialspoint.com

The **EXIT** statement in PL/SQL programming language has following two usages:

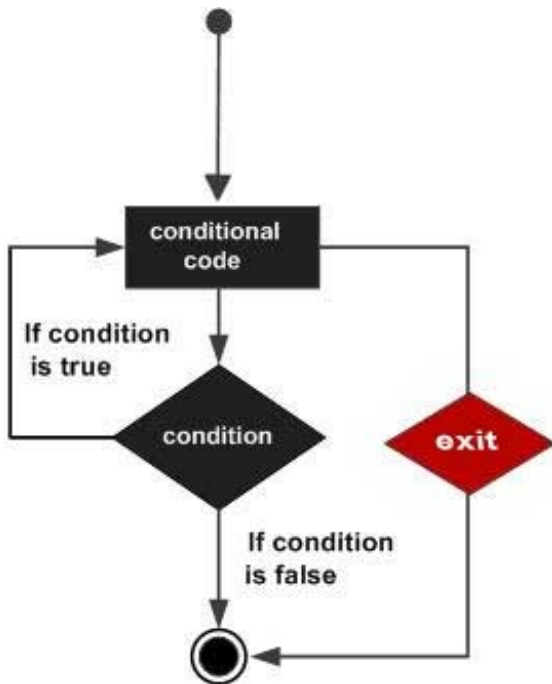
- When the EXIT statement is encountered inside a loop, the loop is immediately terminated and program control resumes at the next statement following the loop.
- If you are using nested loops (i.e. one loop inside another loop), the EXIT statement will stop the execution of the innermost loop and start executing the next line of code after the block.

Syntax:

The syntax for a EXIT statement in PL/SQL is as follows:

```
EXIT;
```

Flow Diagram:



Example:

```
DECLARE
    a number(2) := 10;
BEGIN
    -- while loop execution
    WHILE a < 20 LOOP
        dbms_output.put_line ('value of a: ' || a);
        a := a + 1;
        IF a > 15 THEN
            -- terminate the loop using the exit statement
            EXIT;
        END IF;
    END LOOP;
END;
```

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

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15

PL/SQL procedure successfully completed.
```

The EXIT WHEN Statement

The **EXIT-WHEN** statement allows the condition in the **WHEN** clause to be evaluated. If the condition is true, the loop completes and control passes to the statement immediately after **END LOOP**.

Following are two important aspects for the **EXIT WHEN** statement:

- Until the condition is true, the **EXIT-WHEN** statement acts like a **NULL** statement, except for evaluating the condition, and does not terminate the loop.
- A statement inside the loop must change the value of the condition.

Syntax:

The syntax for an **EXIT WHEN** statement in PL/SQL is as follows:

```
EXIT WHEN condition;
```

The **EXIT WHEN** statement **replaces a conditional statement like if-then** used with the **EXIT** statement.

Example:

```
DECLARE
    a number(2) := 10;
BEGIN
    -- while loop execution
    WHILE a < 20 LOOP
        dbms_output.put_line ('value of a: ' || a);
        a := a + 1;
        -- terminate the loop using the exit when statement
    EXIT WHEN a > 15;
    END LOOP;
END;
/
```

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

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 15

PL/SQL procedure successfully completed.
```