

Conditional Logic : IF « PL SQL Statements « Oracle PL/SQL Tutorial

[Home](#)[Oracle PL/SQL Tutorial](#)

1. Introduction
2. Query Select
3. Set
4. Insert Update Delete
5. Sequences
6. Table
7. Table Joins
8. View
9. Index
10. SQL Data Types
11. Character String Functions
12. Aggregate Functions
13. Date Timestamp Functions
14. Numerical Math Functions
15. Conversion Functions
16. Analytical Functions
17. Miscellaneous Functions
18. Regular Expressions Functions
19. Statistical Functions
20. Linear Regression Functions
21. PL SQL Data Types
22. **PL SQL Statements**
23. PL SQL Operators
24. PL SQL Programming
25. Cursor
26. Collections
27. Function Procedure Packages
28. Trigger
29. SQL PLUS Session Environment
30. System Tables Data Dictionary
31. System Packages
32. Object Oriented
33. XML
34. Large Objects
35. Transaction
36. User Privilege

[Oracle PL/SQL Tutorial » PL SQL Statements » IF](#) Search

[International SMS Gateway](#)

[www.clickatell.com](#)

Connect via HTTP, FTP & many more. 10 Free test SMS when you register!

AdChoices

22.1.1. Conditional Logic

You may use the IF, THEN, ELSE, ELSIF, and END IF keywords in PL/SQL for performing conditional logic.

The following syntax illustrates the use of conditional logic:

```
IF condition1 THEN
  statements1
ELSIF condition2 THEN
  statements2
ELSE
  statements3
END IF;
```

where

1. condition1 and condition2 are Boolean expressions that evaluate to true or false.
2. statements1, statements2, and statements3 are PL/SQL statements.

This conditional logic flows as follows:

1. If condition1 is true, then statements1 is executed.
2. If condition1 is false but condition2 is true, then statements2 is executed.
3. If neither condition1 nor condition2 are true, then statements3 is executed.

```
SQL> CREATE OR REPLACE PROCEDURE cant_go_there
2 AS
3   l_salary NUMBER := 10000;
4 BEGIN
5   IF l_salary > 20000
6   THEN
7     dbms_output.put_line ('Executive');
8   ELSE
9     dbms_output.put_line ('Drone');
10  END IF;
11 END cant_go_there;
12 /
```

Procedure created.

```
SQL>
SQL> SHOW ERRORS
No errors.
SQL>
```

```
SQL> CREATE OR REPLACE PROCEDURE cant_go_there
2 AS
3   l_name varchar2(100) := 'steven';
4 BEGIN
5   IF l_name = 'STEVEN'
6   THEN
7     dbms_output.put_line ('Impossible');
8   ELSE
9     dbms_output.put_line ('Guaranteed');
10  END IF;
11 END cant_go_there;
12 /
```

Procedure created.

```
SQL>
SQL> SHOW ERRORS
No errors.
SQL>
```

[PL Sql Triggers](#)

www.allroundautomations.com

Download the 30 day trail version for
PL/SQL IDE!



[PLC Simulators](#)

www.plcdev.com

Programming made easy for all types of
PLCs



AdChoices

AdChoices

22.1.IF

- 22.1.1. [Conditional Logic](#)
- 22.1.2. [Handling conditions](#)
- 22.1.3. [A Simple Condition Statement](#)
- 22.1.4. [A Simple Condition Statement with BOOLEAN variable](#)
- 22.1.5. [The IF...THEN...ELSE Statement](#)
- 22.1.6. [Use IF THEN ELSE IF](#)
- 22.1.7. [IF...ELSE statements](#)
- 22.1.8. [Using an ELSIF Statement](#)
- 22.1.9. [IF..ELSIF ladder](#)
- 22.1.10. [Block IF statement](#)
- 22.1.11. [IF with ELSE](#)
- 22.1.12. [The Syntax for IF...ELSIF](#)
- 22.1.13. [ELSIF Ladder](#)
- 22.1.14. [The Syntax for Nested IF Statements](#)
- 22.1.15. [Use if with 'IN'](#)
- 22.1.16. [Three valued comparison](#)
- 22.1.17. [JUMP out of a IF statement with goto](#)
- 22.1.18. [Comparing with NULL](#)
- 22.1.19. [If block statement](#)
- 22.1.20. [Create a function and call it in an if statement](#)
- 22.1.21. [PLW-06002: Unreachable code](#)

java2s.com | [Contact Us](#) | [Privacy Policy](#)

Copyright 2009 - 12 Demo Source and Support. All rights reserved.

All other trademarks are property of their respective owners.