

PL/SQL - COMPARISON OPERATORS

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

Copyright © tutorialspoint.com

Comparison operators are used for comparing one expression to another. The result is always either TRUE, FALSE OR NULL.

Operator	Description	Example
LIKE	The LIKE operator compares a character, string, or CLOB value to a pattern and returns TRUE if the value matches the pattern and FALSE if it does not.	If 'Zara Ali' like 'Z% A_i' returns a Boolean true, whereas, 'Nuha Ali' like 'Z% A_i' returns a Boolean false.
BETWEEN	The BETWEEN operator tests whether a value lies in a specified range. x BETWEEN a AND b means that x >= a and x <= b.	If x = 10 then, x between 5 and 20 returns true, x between 5 and 10 returns true, but x between 11 and 20 returns false.
IN	The IN operator tests set membership. x IN <i>set</i> means that x is equal to any member of set.	If x = 'm' then, x in 'a','b','c' returns boolean false but x in 'm','n','o' returns Boolean true.
IS NULL	The IS NULL operator returns the BOOLEAN value TRUE if its operand is NULL or FALSE if it is not NULL. Comparisons involving NULL values always yield NULL.	If x = 'm', then 'x is null' returns Boolean false.

LIKE Operator:

This program tests the LIKE operator, though you will learn how to write procedure in PL/SQL, but I'm going to use a small *procedure* to show the functionality of LIKE operator:

```
DECLARE
PROCEDURE compare (value varchar2, pattern varchar2 ) is
BEGIN
    IF value LIKE pattern THEN
        dbms_output.put_line ('True');
    ELSE
        dbms_output.put_line ('False');
    END IF;
END;

BEGIN
    compare('Zara Ali', 'Z%A_i');
    compare('Nuha Ali', 'Z%A_i');
END;
/
```

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

```
True
False
```

PL/SQL procedure successfully completed.

BETWEEN Operator:

The following program shows the usage of the BETWEEN operator:

```

DECLARE
    x number(2) := 10;
BEGIN
    IF (x between 5 and 20) THEN
        dbms_output.put_line('True');
    ELSE
        dbms_output.put_line('False');
    END IF;

    IF (x BETWEEN 5 AND 10) THEN
        dbms_output.put_line('True');
    ELSE
        dbms_output.put_line('False');
    END IF;

    IF (x BETWEEN 11 AND 20) THEN
        dbms_output.put_line('True');
    ELSE
        dbms_output.put_line('False');
    END IF;
END;
/

```

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

```

True
True
False

```

PL/SQL procedure successfully completed.

IN and IS NULL Operators:

The following program shows the usage of IN and IS NULL operators:

```

DECLARE
    letter varchar2(1) := 'm';
BEGIN
    IF (letter in ('a', 'b', 'c')) THEN
        dbms_output.put_line('True');
    ELSE
        dbms_output.put_line('False');
    END IF;

    IF (letter in ('m', 'n', 'o')) THEN
        dbms_output.put_line('True');
    ELSE
        dbms_output.put_line('False');
    END IF;

    IF (letter is null) THEN
        dbms_output.put_line('True');
    ELSE
        dbms_output.put_line('False');
    END IF;
END;
/

```

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

```

False
True
False

```

PL/SQL procedure successfully completed.