Boolean Logical Operations

A Boolean logical operator is an operator used to perform Boolean logic upon two Boolean expressions. Boolean logical operators return Boolean results (true or false) and take Boolean values as operands. While performing Boolean logic, the expression on the left is evaluated, followed by the expression on the right. The two expressions are finally evaluated in the context of the Boolean logical operator between them.

Boolean logical operators can be used to test or adjust the value of a Boolean variable. The result of the expression using these operators can be used in conditional statements to control the program flow through the code.

Boolean Operators are used to connect and define the relationship between your search terms. When searching electronic databases, you can use Boolean operators to either narrow or broaden your record sets. The three Boolean operators are **AND**, **OR** and **NOT**.

AND (&)

Use **AND** to narrow your search: all of your search terms will present in the retrieved records.

OR (|)

Use **OR** to broaden your search by connecting two or more synonyms.

NOT

Use **NOT** to exclude term(s) from your search results.

It is vital to understand the difference between the & and && operators when they are used in an expression where two conditions have to be evaluated. While the & operator always executes both conditions, && does not execute the second on failure of the first.

The || operator works similar to && by skipping the conditions after the first, if the first condition is true. Hence, && and || (referred to as conditional logical operators) are called short circuit operators. The | and || operators (and & and && operators) are not interchangeable since they operate differently.

The main operations of Boolean algebra are the conjunction *and* denoted as ∧, the disjunction *or* denoted as ∨, and the negation *not* denoted as ¬ . Boolean algebra has been fundamental in the development of digital electronics, and is provided for in all modern programming languages. It is also used in set theory and statistics.

The basic operations of Boolean algebra are as follows:

- AND (conjunction), denoted $x \wedge y$ (sometimes x AND y), satisfies $x \wedge y = 1$ if $x = y = 1$ and $x \wedge y = 0$ otherwise.
- OR (disjunction), denoted $x \vee y$ (sometimes x OR y), satisfies $x \vee y = 0$ if $x = y = 0$ and $x \vee y = 1$ otherwise.
- NOT (negation), denoted $\neg x$ (sometimes NOT x,$!x$), satisfies $\neg x = 0$ if $x = 1$ and $\neg x = 1$ if $x = 0$.

Alternatively the values of $x \wedge y$, $x \vee y$, and $\neg x$ can be expressed by tabulating their values with truth tables as follows.

x	y	$x \wedge y$	$x \vee y$	x	$\neg x$
0	**0**	0	0	**0**	1
1	**0**	0	1	**1**	0
0	**1**	0	1		
1	**1**	1	1		

Logical expressions, like comparison expressions, return `1` (true) or `0` (false) when processed. Logical operators combine two comparisons and return `1` or `0` depending on the results of the comparisons.

The logical operators are:

	Meaning

&	AND
	Returns 1 if both comparisons are true. For example:
	`(4 > 2) & (a = a)  /* true, so result is 1  */`
	`(2 > 4) & (a = a)  /* false, so result is 0 */`

|	Inclusive OR
	Returns 1 if at least one comparison is true. For example:
	`(4 > 2) | (5 = 3)  /* at least one is true, so result is 1 */`
	`(2 > 4) | (5 = 3)  /* neither one is true, so result is 0  */`

Truth table for OR , AND, and NOT operations
True = 1, False = 0

A	!A	B	!B	A || B	A&&B
0	1	0	1	0	0
1	0	0	1	1	0
0	1	1	0	1	0
1	0	1	0	1	1

illustrates the applications of Boolean operators

A	B	A AND B	A OR B	NOT A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Basic Logic Operations

Table 1.8
Truth Tables of Logical Operations

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Relational Operators

Operator	Meaning
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	equal to
!=	not equal to