SQL AND, OR and NOT Operators

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

* The AND operator displays a record if all the conditions separated by AND are TRUE.
* The OR operator displays a record if any of the conditions separated by OR is TRUE.

The NOT operator displays a record if the condition(s) is NOT TRUE.

**Example**

SELECT \* FROM Customers WHERE Country='Germany' AND City='Berlin';

SELECT \* FROM Customers WHERE Country='Germany' OR Country='Spain';

SELECT \* FROM Customers WHERE NOT Country='Germany';

SQL MIN() and MAX() Functions

The MIN() function returns the smallest value of the selected column.

The MAX() function returns the largest value of the selected column.

**Example**

SELECT MIN(Price) AS SmallestPrice FROM Products;

SELECT MAX(Price) AS LargestPrice FROM Products;

**SQL COUNT(), AVG() and SUM()** Functions

**The COUNT()** function returns the number of rows that matches a specified criterion.

**Example**

SELECT COUNT(ProductID) FROM Products;

**The AVG()** function returns the average value of a numeric column.

**Example**

SELECT AVG(Price)FROM Products;

**The SUM()** function returns the total sum of a numeric column.

**Example**

SELECT SUM(Quantity) FROM OrderDetails;

SQL BETWEEN Operator

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

The BETWEEN operator is inclusive: begin and end values are included.

**Example**

SELECT \* FROM Products WHERE Price BETWEEN 10 AND 20;

SQL IN Operator

The IN operator allows you to specify multiple values in a WHERE clause.

The IN operator is a shorthand for multiple OR conditions.

**Example**

SELECT \* FROM Customers WHERE Country IN ('Germany', 'France', 'UK');

SQL NOT IN Operator

**Examples**

SELECT \* FROM Customers WHERE Country NOT IN ('Germany', 'France', 'UK');

SQL LIKE Operator

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

* % - The percent sign represents zero, one, or multiple characters
* \_ - The underscore represents a single character

**Example**

SELECT \* FROM Customers WHERE CustomerName LIKE 'a%';

IS NULL operator

the IS NULL operator is used to test for empty values (NULL values).

The following SQL lists all customers with a NULL value in the "Address" field:

**Example**

SELECT CustomerName, ContactName, Address FROM Customers WHERE Address IS NULL;

The IS NOT NULL Operator

The IS NOT NULL operator is used to test for non-empty values (NOT NULL values).

The following SQL lists all customers with a value in the "Address" field:

**Examples**

SELECT CustomerName, ContactName, Address FROM Customers

WHERE Address IS NOT NULL;

The SQL ANY and ALL Operators

The ANY and ALL operators are used with a WHERE or HAVING clause.

The ANY operator returns true if any of the subquery values meet the condition.

The ALL operator returns true if all of the subquery values meet the condition.

SQL ANY Examples

The ANY operator returns TRUE if any of the subquery values meet the condition.

The following SQL statement returns TRUE and lists the product names if it finds ANY records in the OrderDetails table that quantity = 10:

**Examples**

SELECT ProductName FROM Products WHERE ProductID = ANY (SELECT ProductID FROM OrderDetails WHERE Quantity = 10);

SQL ALL Example

The ALL operator returns TRUE if all of the subquery values meet the condition.

The following SQL statement returns TRUE and lists the product names if ALL the records in the OrderDetails table has quantity = 10 (so, this example will return FALSE, because not ALL records in the OrderDetails table has quantity = 10):

**Examples**

SELECT ProductName

FROM Products

WHERE ProductID = ALL (SELECT ProductID FROM OrderDetails WHERE Quantity = 10);