SQL LIKE

The 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

|  |  |
| --- | --- |
| **LIKE Operator** | **Description** |
| WHERE CustomerName LIKE 'a%' | Finds any values that start with "a" |
| WHERE CustomerName LIKE '%a' | Finds any values that end with "a" |
| WHERE CustomerName LIKE '%or%' | Finds any values that have "or" in any position |
| WHERE CustomerName LIKE '\_r%' | Finds any values that have "r" in the second position |
| WHERE CustomerName LIKE 'a\_\_%' | Finds any values that start with "a" and are at least 3 characters in length |
| WHERE ContactName LIKE 'a%o' | Finds any values that start with "a" and ends with "o" |

selects all customers with a CustomerName starting with "a":

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

selects all customers with a CustomerName ending with "a":

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

selects all customers with a CustomerName that have "or" in any position:

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

selects all customers with a CustomerName that have "r" in the second position:

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

selects all customers with a CustomerName that does NOT start with "a":

SELECT \* FROM Customers WHERE CustomerName NOT LIKE 'a%';

# SQL IN Operator

## The 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.

**selects all customers that are located in "Germany", "France" or "UK":**

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

selects all customers that are NOT located in "Germany", "France" or "UK":

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

selects all customers that are from the same countries as the suppliers:

SELECT \* FROM Customers WHERE Country IN (SELECT Country FROM Suppliers);

## 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.

**selects all products with a price BETWEEN 10 and 20:**

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

## NOT BETWEEN Example

To display the products outside the range of the previous example, use NOT BETWEEN:

SELECT \* FROM Products WHERE Price NOT BETWEEN 10 AND 20;

## BETWEEN with IN Example

SQL statement selects all products with a price BETWEEN 10 and 20. In addition; do not show products with a CategoryID of 1,2, or 3:

SELECT \* FROM Products WHERE Price BETWEEN 10 AND 20

AND NOT CategoryID IN (1,2,3);

## BETWEEN Text Values Example

## SQL statement selects all products with a ProductName BETWEEN Carnarvon Tigers and Mozzarella di Giovanni:

SELECT \* FROM Products WHERE ProductName BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni' ORDER BY ProductName;

SQL statement selects all products with a ProductName BETWEEN Carnarvon Tigers and Chef Anton's Cajun Seasoning:

SELECT \* FROM Products WHERE ProductName BETWEEN "Carnarvon Tigers" AND "Chef Anton's Cajun Seasoning" ORDER BY ProductName;

SQL statement selects all products with a ProductName NOT BETWEEN Carnarvon Tigers and Mozzarella di Giovanni:

SELECT \* FROM Products WHERE ProductName NOT BETWEEN 'Carnarvon Tigers' AND 'Mozzarella di Giovanni'ORDER BY ProductName;

## BETWEEN Dates Example

SQL statement selects all orders with an OrderDate BETWEEN '01-July-1996' and '31-July-1996':

SELECT \* FROM Orders WHERE OrderDate BETWEEN #01/07/1996# AND #31/07/1996#;

OR:

SELECT \* FROM Orders  
WHERE OrderDate BETWEEN '1996-07-01' AND '1996-07-31';

# SQL Aliases

SQL aliases are used to give a table, or a column in a table, a temporary name.

Aliases are often used to make column names more readable.

An alias only exists for the duration of the query.

SQL statement creates two aliases, one for the CustomerID column and one for the CustomerName column:

SELECT CustomerID AS ID, CustomerName AS Customer  
FROM Customers;