**SQL Introduction – 5**

**DML (Data Manipulation Language)**

## The SQL AND & OR Operators

The AND operator displays a record if both the first condition AND the second condition are true.

The OR operator displays a record if either the first condition OR the second condition is true.

## AND Operator Example

The following SQL statement selects all customers from the country "Germany" AND the city "Berlin", in the "Customers" table:

## Example

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

## OR Operator Example

The following SQL statement selects all customers from the city "Berlin" OR "München", in the "Customers" table:

## Example

SELECT \* FROM Customers
WHERE City='Berlin'
OR City='München';

## Combining AND & OR

You can also combine AND and OR (use parenthesis to form complex expressions).

The following SQL statement selects all customers from the country "Germany" AND the city must be equal to "Berlin" OR "München", in the "Customers" table:

## Example

SELECT \* FROM Customers
WHERE Country='Germany'
AND (City='Berlin' OR City='München');

## The SQL ORDER BY Keyword

The ORDER BY keyword is used to sort the result-set by one or more columns.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in a descending order, you can use the DESC keyword.

### SQL ORDER BY Syntax

SELECT column\_name,column\_name
FROM table\_name
ORDER BY column\_name,column\_name ASC|DESC;

## ORDER BY Example

The following SQL statement selects all customers from the "Customers" table, sorted by the "Country" column:

## Example

SELECT \* FROM Customers
ORDER BY Country;

## ORDER BY DESC Example

The following SQL statement selects all customers from the "Customers" table, sorted DESCENDING by the "Country" column:

## Example

SELECT \* FROM Customers
ORDER BY Country DESC;

## ORDER BY Several Columns Example

The following SQL statement selects all customers from the "Customers" table, sorted by the "Country" and the "CustomerName" column:

## Example

SELECT \* FROM Customers
ORDER BY Country,CustomerName;

## The SQL LIKE Operator

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

### SQL LIKE Syntax

SELECT column\_name(s)
FROM table\_name
WHERE column\_name LIKE pattern;

## SQL LIKE Operator Examples

The following SQL statement selects all customers with a City starting with the letter "s":

## Example

SELECT \* FROM Customers
WHERE City LIKE 's%';

**Tip:** The "%" sign is used to define wildcards (missing letters) both before and after the pattern. You will learn more about wildcards in the next chapter.

The following SQL statement selects all customers with a City ending with the letter "s":

## Example

SELECT \* FROM Customers
WHERE City LIKE '%s';

The following SQL statement selects all customers with a Country containing the pattern "land":

## Example

SELECT \* FROM Customers
WHERE Country LIKE '%land%';

Using the NOT keyword allows you to select records that does NOT match the pattern.

The following SQL statement selects all customers with a Country NOT containing the pattern "land":

## Example

SELECT \* FROM Customers
WHERE Country NOT LIKE '%land%';

## SQL Wildcard Characters

In SQL, wildcard characters are used with the SQL LIKE operator.

SQL wildcards are used to search for data within a table.

With SQL, the wildcards are:

|  |  |
| --- | --- |
| **Wildcard** | **Description** |
| % | A substitute for zero or more characters |
| \_ | A substitute for a single character |
| [charlist] | Sets and ranges of characters to match |
| [^charlist]or[!charlist] | Matches only a character NOT specified within the brackets |

## Using the SQL % Wildcard

The following SQL statement selects all customers with a City starting with "ber":

## Example

SELECT \* FROM Customers
WHERE City LIKE 'ber%';

The following SQL statement selects all customers with a City containing the pattern "es":

## Example

SELECT \* FROM Customers
WHERE City LIKE '%es%';

## Using the SQL \_ Wildcard

The following SQL statement selects all customers with a City starting with any character, followed by "erlin":

## Example

SELECT \* FROM Customers
WHERE City LIKE '\_erlin';

The following SQL statement selects all customers with a City starting with "L", followed by any character, followed by "n", followed by any character, followed by "on":

## Example

SELECT \* FROM Customers
WHERE City LIKE 'L\_n\_on';

## Using the SQL [charlist] Wildcard

The following SQL statement selects all customers with a City starting with "b", "s", or "p":

## Example

SELECT \* FROM Customers
WHERE City LIKE '[bsp]%';

The following SQL statement selects all customers with a City starting with "a", "b", or "c":

## Example

SELECT \* FROM Customers
WHERE City LIKE '[a-c]%';

The following SQL statement selects all customers with a City NOT starting with "b", "s", or "p":

## Example

SELECT \* FROM Customers
WHERE City LIKE '[!bsp]%';