MySQL

MySQL SQL

MySQL SQL

- **What is SQL?**
- SQL is the standard language for dealing with Relational Databases.
- SQL is used to insert, search, update, and delete database records.

- **How to Use SQL**
- The following SQL statement selects all the records in the "Customers" table:

```
SELECT * FROM Customers;
```

MySQL SQL

- SQL keywords are NOT case sensitive: select is the same as SELECT
- Some database systems require a semicolon at the end of each SQL statement.
- Semicolon is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.

MySQL SQL

- **Some of The Most Important SQL Commands:**
- SELECT - extracts data from a database
- UPDATE - updates data in a database
- DELETE - deletes data from a database
- INSERT INTO - inserts new data into a database
- CREATE DATABASE - creates a new database

MySQL SQL

- **Some of The Most Important SQL Commands:**
- ALTER DATABASE - modifies a database
- CREATE TABLE - creates a new table
- ALTER TABLE - modifies a table
- DROP TABLE - deletes a table
- CREATE INDEX - creates an index (search key)
- DROP INDEX - deletes an index

MySQL SELECT Statement

- **SELECT Syntax**

```
SELECT column1, column2, ...
FROM table_name;
```

- Here, column1, column2, ... are the field names of the table you want to select data from.
- If you want to select all the fields available in the table, use the following syntax:

```
SELECT * FROM table_name;
```

MySQL SELECT Statement

- **Demo Database**
- Below is a selection from the "Customers" table in the Northwind sample database:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

MySQL SELECT Statement

- **SELECT Columns Example**
- The following SQL statement selects the "CustomerName", "City", and "Country" columns from the "Customers" table:

```
SELECT CustomerName, City, Country FROM Customers;
```

- **SELECT * Example**
- The following SQL statement selects ALL the columns from the "Customers" table:

```
SELECT * FROM Customers;
```

MySQL SELECT Statement

- **The MySQL SELECT DISTINCT Statement**
- The SELECT DISTINCT statement is used to return only distinct (different) values.
- Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

- **SELECT DISTINCT Syntax**

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

MySQL SELECT Statement

- **SELECT Example Without DISTINCT**
- The following SQL statement selects all (including the duplicates) values from the "Country" column in the "Customers" table:

```
SELECT Country FROM Customers;
```

- Now, let us use the **SELECT DISTINCT** statement and see the result.

MySQL SELECT Statement

- **SELECT DISTINCT Examples**
- The following SQL statement selects only the DISTINCT values from the "Country" column in the "Customers" table:

```
SELECT DISTINCT Country FROM Customers;
```

- The following SQL statement counts and returns the number of different (distinct) countries in the "Customers" table:

```
SELECT COUNT(DISTINCT Country) FROM Customers;
```

MySQL WHERE Clause

- **The MySQL WHERE Clause**
- The WHERE clause is used to filter records.
- It is used to extract only those records that fulfill a specified condition.

- **WHERE Syntax**

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

MySQL WHERE Clause

Note: The `WHERE` clause is not only used in `SELECT` statements, it is also used in `UPDATE`, `DELETE`, etc.!

MySQL WHERE Clause

- **Demo Database**
- Below is a selection from the "Customers" table in the Northwind sample database:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

MySQL WHERE Clause

- **WHERE Clause Example**
- The following SQL statement selects all the customers from "Mexico":

```
SELECT * FROM Customers
WHERE Country = 'Mexico';
```

- **Text Fields vs. Numeric Fields**
- SQL requires single quotes around text values (most database systems will also allow double quotes).
- However, numeric fields should not be enclosed in quotes:

MySQL WHERE Clause

- **Operators in The WHERE Clause**
- The following operators can be used in the WHERE clause:

Operator	Description	Example
=	Equal	Try it
>	Greater than	Try it
<	Less than	Try it
>=	Greater than or equal	Try it
<=	Less than or equal	Try it

MySQL WHERE Clause

- **Operators in The WHERE Clause**
- The following operators can be used in the WHERE clause:

Operator	Description	Example
=	Equal	Try it
<>	Not equal. **Note:** In some versions of SQL this operator may be written as !=	Try it
BETWEEN	Between a certain range	Try it
LIKE	Search for a pattern	Try it
IN	To specify multiple possible values for a column	Try it

MySQL AND, OR and NOT Operators

- **The MySQL 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.

MySQL AND, OR and NOT Operators

- **AND Syntax**

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

- **OR Syntax**

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

MySQL AND, OR and NOT Operators

- **NOT Syntax**

```
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;
```

MySQL AND, OR and NOT Operators

- **Demo Database**
- Below is a selection from the "Customers" table in the Northwind sample database:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

MySQL AND, OR and NOT Operators

- **AND Example**
- The following SQL statement selects all fields from "Customers" where country is "Germany" AND city is "Berlin":

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

- **OR Example**
- The following SQL statement selects all fields from "Customers" where city is "Berlin" OR "Stuttgart":

```
SELECT * FROM Customers
WHERE City = 'Berlin' OR City = 'Stuttgart';
```

MySQL AND, OR and NOT Operators

- **OR Example**
- The following SQL statement selects all fields from "Customers" where country is "Germany" OR "Spain":

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

- **NOT Example**
- The following SQL statement selects all fields from "Customers" where country is NOT "Germany":

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

MySQL AND, OR and NOT Operators

- **Combining AND, OR and NOT**
- You can also combine the AND, OR and NOT operators.
- The following SQL statement selects all fields from "Customers" where country is "Germany" AND city must be "Berlin" OR "Stuttgart" (use parenthesis to form complex expressions):

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

MySQL AND, OR and NOT Operators

- **Combining AND, OR and NOT**
- The following SQL statement selects all fields from "Customers" where country is NOT "Germany" and NOT "USA":

```
SELECT * FROM Customers
WHERE NOT Country = 'Germany' AND NOT Country = 'USA';
```

MySQL ORDER BY Keyword

- **The MySQL ORDER BY Keyword**
- The ORDER BY keyword is used to sort the result-set in ascending or descending order.
- The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

- **ORDER BY Syntax**

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

MySQL ORDER BY Keyword

- **Demo Database**
- Below is a selection from the "Customers" table in the Northwind sample database:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

MySQL ORDER BY Keyword

- **ORDER BY Example**
- The following SQL statement selects all customers from the "Customers" table, sorted by the "Country" column:

```
SELECT * FROM Customers
ORDER BY Country;
```

MySQL ORDER BY Keyword

- **ORDER BY DESC Example**
- The following SQL statement selects all customers from the "Customers" table, sorted DESCENDING by the "Country" column:

```
SELECT * FROM Customers
ORDER BY Country DESC;
```

MySQL ORDER BY Keyword

- **ORDER BY Several Columns Example**
- The following SQL statement selects all customers from the "Customers" table, sorted by the "Country" and the "CustomerName" column. This means that it orders by Country, but if some rows have the same Country, it orders them by CustomerName:

```
SELECT * FROM Customers
ORDER BY Country, CustomerName;
```

MySQL ORDER BY Keyword

- **ORDER BY Several Columns Example 2**
- The following SQL statement selects all customers from the "Customers" table, sorted ascending by the "Country" and descending by the "CustomerName" column:

```
SELECT * FROM Customers
ORDER BY Country ASC, CustomerName DESC;
```