

# SQL BASICS AND ADVANCED

---

## **INTRODUCTION**

SQL is a standard language for accessing databases.

how to use SQL to access and manipulate data in:

MySQL, SQL Server, Access, Oracle, Sybase, DB2, and other database systems.

### **SQL Syntax:**

```
SELECT Company, Country FROM Customers WHERE Country <> 'USA'
```

### **SQL Result:**

Company	Country
Island Trading	UK
Galería del gastrónomo	Spain
Laughing Bacchus Wine Cellars	Canada
Paris spécialités	France
Simons bistro	Denmark
Wolski Zajazd	Poland

SQL is a “standard language for accessing and manipulating databases”.

### **What is SQL?**

- ❖ SQL stands for Structured Query Language
- ❖ SQL lets you access and manipulate databases
- ❖ SQL is an ANSI (American National Standards Institute) standard

### **What Can SQL do?**

- ❖ SQL can execute queries against a database
- ❖ SQL can retrieve data from a database
- ❖ SQL can insert records in a database
- ❖ SQL can update records in a database
- ❖ SQL can delete records from a database
- ❖ SQL can create new databases
- ❖ SQL can create new tables in a database
- ❖ SQL can create stored procedures in a database
- ❖ SQL can create views in a database
- ❖ SQL can set permissions on tables, procedures, and views

## RDBMS

- ❖ RDBMS stands for Relational Database Management System.
- ❖ RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.
- ❖ The data in RDBMS is stored in database objects called tables.
- ❖ A table is a collections of related data entries and it consists of columns and rows.

## SQL BASICS

### Database Tables

A database most often contains one or more tables. Each table is identified by a name (e.g. "Customers" or "Orders"). Tables contain records (rows) with data.

Below is an example of a table called "Persons":

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

The table above contains three records (one for each person) and five columns (P\_Id, LastName, FirstName, Address, and City).

### SQL Statements

Most of the actions you need to perform on a database are done with SQL statements.

The following SQL statement will select all the records in the "Persons" table:

```
SELECT * FROM Persons
```

#### Note:

- SQL is not case sensitive
- Semicolon after SQL Statements?
- 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.
- We are using MS Access and SQL Server 2000 and we do not have to put a semicolon after each SQL statement, but some database programs force you to use it.

## SQL DML and DDL

SQL can be divided into two parts: The Data Manipulation Language (DML) and the Data Definition Language (DDL).

The query and update commands form the **DML part of SQL**:

**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

The **DDL part of SQL** permits database tables to be created or deleted. It also define indexes (keys), specify links between tables, and impose constraints between tables. The most important DDL statements in SQL are:

**CREATE DATABASE** - creates a new database

**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

## The SQL SELECT Statement

- The SELECT statement is used to select data from a database.
- The result is stored in a result table, called the result-set.
- SQL SELECT Syntax

```
SELECT column_name(s)
FROM table_name
And
SELECT * FROM table_name
```

**Note:** SQL is not case sensitive. SELECT is the same as select.

## An SQL SELECT Example

The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

Now we want to select the content of the columns named "LastName" and "FirstName" from the table above.

We use the following SELECT statement:

```
SELECT LastName,FirstName FROM Persons
```

The result-set will look like this:

LastName	FirstName
Hansen	Ola
Svendson	Tove
Pettersen	Kari

## SELECT \* Example

Now we want to select all the columns from the "Persons" table.  
We use the following SELECT statement:

```
SELECT * From Persons
```

## The SQL SELECT DISTINCT Statement

- In a table, some of the columns may contain duplicate values. This is not a problem, however, sometimes you will want to list only the different (distinct) values in a table.
- The DISTINCT keyword can be used to return only distinct (different) values.
- SQL SELECT DISTINCT Syntax

```
SELECT DISTINCT column_name(s)
FROM table_name
```

## SELECT DISTINCT Example

The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes

2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

Now we want to select only the distinct values from the column named "City" from the table above. We use the following SELECT statement:

```
SELECT DISTINCT City FROM Persons
```

The result-set will look like this:

City
Sandnes
Stavanger

## The WHERE Clause

- The WHERE clause is used to filter records.
- The WHERE clause is used to extract only those records that fulfill a specified criterion.
- SQL WHERE Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator value
```

## WHERE Clause Example

The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

Now we want to select only the persons living in the city "Sandnes" from the table above. We use the following SELECT statement:

```
SELECT * FROM Persons
WHERE City='Sandnes'
```



The result-set will look like this:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes

## Quotes Around Text Fields

SQL uses single quotes around text values (most database systems will also accept double quotes). Although, numeric values should not be enclosed in quotes.

For text values:

This is correct:

```
SELECT * FROM Persons WHERE FirstName='Tove'
```

This is wrong:

```
SELECT * FROM Persons WHERE FirstName=Tove
```

For numeric values:

This is correct:

```
SELECT * FROM Persons WHERE Year=1965
```

This is wrong:

```
SELECT * FROM Persons WHERE Year='1965'
```

## Operators Allowed in the WHERE Clause

With the WHERE clause, the following operators can be used:

Operator	Description
=	Equal
<>	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	If you know the exact value you want to return for at least one of the columns

**Note:** In some versions of SQL the <> operator may be written as !=

## The AND & OR Operators

- The AND & OR operators are used to filter records based on more than one condition.
- The AND operator displays a record if both the first condition and the second condition is true.
- The OR operator displays a record if either the first condition or the second condition is true.

### AND Operator Example

The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

Now we want to select only the persons with the first name equal to "Tove" AND the last name equal to "Svendson":

We use the following SELECT statement:

```
SELECT * FROM Persons  
WHERE FirstName='Tove' AND LastName='Svendson'
```

The result-set will look like this:

P_Id	LastName	FirstName	Address	City
2	Svendson	Tove	Borgvn 23	Sandnes

### OR Operator Example

Now we want to select only the persons with the first name equal to "Tove" OR the first name equal to "Ola":

We use the following SELECT statement:

```
SELECT * FROM Persons  
WHERE FirstName='Tove' OR FirstName='Ola'
```

The result-set will look like this:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes



2	Svendson	Tove	Borgvn 23	Sandnes
---	----------	------	-----------	---------

## Combining AND & OR

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

Now we want to select only the persons with the last name equal to "Svendson" AND the first name equal to "Tove" OR to "Ola":

We use the following SELECT statement:

```
SELECT * FROM Persons
WHERE LastName='Svendson' AND (FirstName='Tove' OR FirstName='Ola')
```

The result-set will look like this:

P_Id	LastName	FirstName	Address	City
2	Svendson	Tove	Borgvn 23	Sandnes

## The ORDER BY Keyword

- The ORDER BY keyword is used to sort the result-set.
- The ORDER BY keyword is used to sort the result-set by a specified column.
- The ORDER BY keyword sort the records in ascending order by default.
- If you want to sort the records in a descending order, you can use the DESC keyword.
- SQL ORDER BY Syntax

```
SELECT column_name(s)
FROM table_name ORDER BY column_name(s) ASC|DESC
```

## ORDER BY Example

The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Tom	Vingvn 23	Stavanger

Now we want to select all the persons from the table above, however, we want to sort the persons by their last name.

We use the following SELECT statement:

```
SELECT * FROM Persons ORDER BY LastName
```

The result-set will look like this:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
4	Nilsen	Tom	Vingvn 23	Stavanger
3	Pettersen	Kari	Storgt 20	Stavanger
2	Svendson	Tove	Borgvn 23	Sandnes

## ORDER BY DESC Example

Now we want to select all the persons from the table above, however, we want to sort the persons descending by their last name.

We use the following SELECT statement:

```
SELECT * FROM Persons ORDER BY LastName DESC
```

The result-set will look like this:

P_Id	LastName	FirstName	Address	City
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Tom	Vingvn 23	Stavanger
1	Hansen	Ola	Timoteivn 10	Sandnes

## The INSERT INTO Statement

- The INSERT INTO statement is used to insert new records in a table.
- The INSERT INTO statement is used to insert a new row in a table.
- SQL INSERT INTO Syntax
- It is possible to write the INSERT INTO statement in two forms.
- The first form doesn't specify the column names where the data will be inserted, only their values:

```
INSERT INTO table_name VALUES (value1, value2, value3,...)
```

- The second form specifies both the column names and the values to be inserted:

```
INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)
```