**What is SQL?**

SQL is a standard language for storing, manipulating and retrieving data in databases.

* SQL stands for Structured Query Language..
* SQL lets you access and manipulate databases…
* SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987..

**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 stands for Relational Database Management System.**

RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

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.

## 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
* 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
  1. CREATE DATABASE statement is used to create a new SQL database. **Syntax :- CREATE DATABASE** databasename;
  2. DROP DATABASE statement is used to drop an existing SQL database.

**Syntax**:- **DROP DATABASE** *databasename*;

**Note:-** Be careful before dropping a database. Deleting a database will result in loss of complete information stored in the database!..

* 1. The BACKUP DATABASE statement is used in SQL Server to create a full back up of an existing SQL database.

### **Syntax :- BACKUP DATABASE**databasename **TO DISK =** 'filepath';

**Syntax:- BACKUP DATABASE** databasename

**TO DISK** = 'filepath'

**WITH** DIFFERENTIAL;

**Tip:-** Always back up the database to a different drive than the actual database. Then, if you get a disk crash, you will not lose your backup file along with the database.

* 1. The CREATE TABLE statement is used to create a new table in a database.

### **Syntax:- CREATE TABLE** table\_name (     column1 datatype,     column2 datatype,     column3 datatype,    .... );

The column parameters specify the names of the columns of the table.The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

Create Table Using Another Table A copy of an existing table can also be created using CREATE TABLE. If you create a new table using an existing table, the new table will be filled with the existing values from the old table.

**Syntax:-**

**CREATE TABLE** new\_table\_name **AS**

**SELECT** column1, column2,...

**FROM** existing\_table\_name

**WHERE ....;**

* 1. The DROP TABLE statement is used to drop an existing table in a database.

### **Syntax:- DROP TABLE** table\_name;

**Note:-** Be careful before dropping a table. Deleting a table will result in loss of complete information stored in the table!

The TRUNCATE TABLE statement is used to delete the data inside a table, but not the table itself.

### **Syntax:- TRUNCATE TABLE** table\_name;

* 1. The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

To add a column in a table, use the following **syntax:-ALTER TABLE**  *table\_name*  
**ADD**  *column\_name datatype*;

To delete a column in a table, use the following syntax (notice that some database systems don't allow deleting a column):- **ALTER TABLE**  *table\_name*  
**DROP COLUMN**  *column\_name*;

To rename a column in a table, use the following **syntax:- ALTER TABLE** *table\_name*  
RENAME **COLUMN** *old\_name***to***new\_name*;

**To change the data type of a column in a table, use the following syntax:-**

1. SQL Server / MS Access:-**ALTER TABLE** *table\_name*  
   **ALTER COLUMN** *column\_name datatype*;
2. My SQL / Oracle (prior version 10G):- **ALTER TABLE** *table\_name*  
    **MODIFY COLUMN***column\_name datatype*;
3. Oracle 10G and later:-**ALTER TABLE** *table\_name*  
   **MODIFY** *column\_name datatype*;
   1. SQL constraints are used to specify rules for data in a table.

**SQL Create Constraints** Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement. **Syntax:-CREATE TABLE** table\_name (  
    column1 datatype ***constraint***,  
    column2 datatype ***constraint***,  
    column3 datatype ***constraint***,  
    ....  
);

SQL constraints are used to specify rules for the data in a table. Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted. Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:-

* [**NOT NULL**](https://www.w3schools.com/sql/sql_notnull.asp) - Ensures that a column cannot have a NULL value
* [**UNIQUE**](https://www.w3schools.com/sql/sql_unique.asp) - Ensures that all values in a column are different
* [**PRIMARY KEY**](https://www.w3schools.com/sql/sql_primarykey.asp) - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
* [**FOREIGN KEY**](https://www.w3schools.com/sql/sql_foreignkey.asp) - Prevents actions that would destroy links between tables..
* [**CHECK**](https://www.w3schools.com/sql/sql_check.asp)- Ensures that the values in a column satisfies a specific condition..
* [**DEFAULT**](https://www.w3schools.com/sql/sql_default.asp)- Sets a default value for a column if no value is specified..
* [**CREATE INDEX**](https://www.w3schools.com/sql/sql_create_index.asp) - Used to create and retrieve data from the database very quickly..

**SQL NOT NULL on ALTER TABLE**

To create a NOT NULL constraint on the "Age" column when the "Persons" table is already created, use the following SQL:-

SQL Server / MS Access:-

**ALTER TABLE** Persons  
**ALTER COLUMN** Age int **NOT NULL**;

My SQL / Oracle (prior version 10G):-

**ALTER TABLE** Persons  
**MODIFY COLUMN** Age int **NOT NULL**;

Oracle 10G and later:-

**ALTER TABLE** Persons  
**MODIFY** Age int **NOT NULL**;

* 1. The UNIQUE constraint ensures that all values in a column are different.

Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.

A PRIMARY KEY constraint automatically has a UNIQUE constraint.

However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

SQL UNIQUE Constraint on CREATE TABLE

The following SQL creates a UNIQUE constraint on the "ID" column when the "Persons" table is created:

**SQL Server / Oracle / MS Access:**

CREATE TABLE Persons (  
    ID int NOT NULL UNIQUE,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int  
);

**MySQL:**

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    UNIQUE (ID)  
);

To name a UNIQUE constraint, and to define a UNIQUE constraint on multiple columns, use the following SQL syntax:

**MySQL / SQL Server / Oracle / MS Access:**

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CONSTRAINT UC\_Person UNIQUE (ID,LastName)  
);

ADVERTISEMENT

SQL UNIQUE Constraint on ALTER TABLE

To create a UNIQUE constraint on the "ID" column when the table is already created, use the following SQL:

**MySQL / SQL Server / Oracle / MS Access:**

ALTER TABLE Persons  
ADD UNIQUE (ID);

To name a UNIQUE constraint, and to define a UNIQUE constraint on multiple columns, use the following SQL syntax:

**MySQL / SQL Server / Oracle / MS Access:**

ALTER TABLE Persons  
ADD CONSTRAINT UC\_Person UNIQUE (ID,LastName);

DROP a UNIQUE Constraint

To drop a UNIQUE constraint, use the following SQL:

**MySQL:**

ALTER TABLE Persons  
DROP INDEX UC\_Person;

**SQL Server / Oracle / MS Access:**

ALTER TABLE Persons  
DROP CONSTRAINT UC\_Person;