# **SQL UNIQUE Constraint**

**SQL UNIQUE Constraint:**

1. The UNIQUE constraint ensures that all values in a column are different.
2. Both the UNIQUE and PRIMARY KEY constraints provide a guarantee for uniqueness for a column or set of columns.
3. A PRIMARY KEY constraint automatically has a UNIQUE constraint.
4. However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

**SQL UNIQUE Constraint on CREATE TABLE:**

1. 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)  ); |

**SQL UNIQUE Constraint on ALTER TABLE**

1. 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**

1. 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; |