CREATE DATABASE testDB;

The DROP DATABASE statement is used to drop an existing SQL database.

DROP DATABASE testDB;

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

CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);

Create Table Using Another Table

A copy of an existing table can also be created using CREATE TABLE.

The new table gets the same column definitions. All columns or specific columns can be selected.

If you create a new table using an existing table, the new table will be filled with the existing values from the old table.

CREATE TABLE TestTable AS  
SELECT customername, contactname  
FROM customers(existing\_table\_name);

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

DROP TABLE Shippers;

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

TRUNCATE TABLE table\_name;

## MySQL ALTER TABLE Statement

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.

ALTER TABLE table\_name  
ADD column\_name datatype;

ALTER TABLE Customers  
ADD Email varchar(255);

To delete a column in a table.

ALTER TABLE table\_name  
DROP COLUMN column\_name;

ALTER TABLE Customers  
DROP COLUMN Email;

To change the data type of a column in a table.

ALTER TABLE table\_name  
MODIFY COLUMN column\_name datatype;

we want to add a column named "DateOfBirth" in the "Persons" table.

ALTER TABLE Persons  
ADD DateOfBirth date;

we want to change the data type of the column named "DateOfBirth" in the "Persons" table.

ALTER TABLE Persons  
MODIFY COLUMN DateOfBirth year;

we want to delete the column named "DateOfBirth" in the "Persons" table.

ALTER TABLE Persons  
DROP COLUMN DateOfBirth;

# **MySQL Constraints**

SQL constraints are used to specify rules for data in a table.

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.

CREATE TABLE table\_name (  
    column1 datatype *constraint*,  
    column2 datatype *constraint*,  
    column3 datatype *constraint*  
);

## MySQL Constraints

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

**NOT NULL**

By default, a column can hold NULL values.

The NOT NULL constraint enforces a column to NOT accept NULL values.

## NOT NULL on CREATE TABLE

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

## 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:

ALTER TABLE Persons  
MODIFY Age int NOT NULL;

# **MySQL UNIQUE Constraint**

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.

## UNIQUE Constraint on CREATE TABLE

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

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.

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

## UNIQUE Constraint on ALTER TABLE

To create a UNIQUE constraint on the "ID" column when the table is already created.

ALTER TABLE Persons  
ADD UNIQUE (ID);

To name a UNIQUE constraint, and to define a UNIQUE constraint on multiple columns.

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

## DROP a UNIQUE Constraint

ALTER TABLE Persons  
DROP INDEX UC\_Person;

# **MySQL PRIMARY KEY Constraint**