CREATE DATABASE

CREATE DATABASE testDB;

## DROP DATABASE

## DROP DATABASE testDB;

## The MySQL CREATE TABLE Statement

## CREATE TABLE table\_name (     column1 datatype,     column2 datatype,     column3 datatype,    .... );

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

## MySQL DROP TABLE Example

## DROP TABLE Shippers;

## MySQL TRUNCATE TABLE

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.

## ADD Column

ALTER TABLE table\_name  
ADD column\_name datatype;

## DROP COLUMN

ALTER TABLE table\_name  
DROP COLUMN column\_name;

## MODIFY COLUMN

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

## MySQL Constraints

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.

* [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

## 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

ALTER TABLE Persons  
MODIFY Age int NOT NULL;

## UNIQUE Constraint on CREATE TABLE

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.

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

UNIQUE constraint on multiple columns

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

ALTER TABLE Persons  
ADD UNIQUE (ID);

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

ALTER TABLE Persons  
DROP INDEX UC\_Person;

## MySQL PRIMARY KEY Constraint

The PRIMARY KEY constraint uniquely identifies each record in a table.

Primary keys must contain UNIQUE values, and cannot contain NULL values.

A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

## PRIMARY KEY on CREATE TABLE

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

CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    CONSTRAINT PK\_Person PRIMARY KEY (ID,LastName)  
);

## PRIMARY KEY on ALTER TABLE

ALTER TABLE Persons  
ADD PRIMARY KEY (ID);

ALTER TABLE Persons  
ADD CONSTRAINT PK\_Person PRIMARY KEY (ID,LastName);

## DROP a PRIMARY KEY Constraint

ALTER TABLE Persons  
DROP PRIMARY KEY;

## MySQL FOREIGN KEY Constraint

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the [PRIMARY KEY](https://www.w3schools.com/MySQL/mysql_primarykey.asp) in another table.

The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

## FOREIGN KEY on CREATE TABLE

CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  
);

REATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    CONSTRAINT FK\_PersonOrder FOREIGN KEY (PersonID)  
    REFERENCES Persons(PersonID)  
);

## FOREIGN KEY on ALTER TABLE

## ALTER TABLE Orders ADD CONSTRAINT FK\_PersonOrder FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);

## DROP a FOREIGN KEY Constraint

## ALTER TABLE Orders DROP FOREIGN KEY FK\_PersonOrder;

## MySQL CHECK Constraint

## CHECK on CREATE TABLE

## CREATE TABLE Persons (     ID int NOT NULL,     LastName varchar(255) NOT NULL,     FirstName varchar(255),     Age int,     CHECK (Age>=18) );

## CREATE TABLE Persons (     ID int NOT NULL,     LastName varchar(255) NOT NULL,     FirstName varchar(255),     Age int,     City varchar(255),     CONSTRAINT CHK\_Person CHECK (Age>=18 AND City='Sandnes') );

## CHECK on ALTER TABLE

## ALTER TABLE Persons ADD CHECK (Age>=18);

## ALTER TABLE Persons ADD CONSTRAINT CHK\_PersonAge CHECK (Age>=18 AND City='Sandnes');

## DROP a CHECK Constraint

## ALTER TABLE Persons DROP CHECK CHK\_PersonAge;

## MySQL DEFAULT Constraint

## DEFAULT on CREATE TABLE

## CREATE TABLE Persons (     ID int NOT NULL,     LastName varchar(255) NOT NULL,     FirstName varchar(255),     Age int,     City varchar(255) DEFAULT 'Sandnes'

## );

## CREATE TABLE Orders (     ID int NOT NULL,     OrderNumber int NOT NULL,     OrderDate date DEFAULT CURRENT\_DATE() );

## DEFAULT on ALTER TABLE

## ALTER TABLE Persons ALTER City SET DEFAULT 'Sandnes';

## DROP a DEFAULT Constraint

## ALTER TABLE Persons ALTER City DROP DEFAULT;

## What is an AUTO INCREMENT Field?

Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table.

Often this is the primary key field that we would like to be created automatically every time a new record is inserted.

## CREATE TABLE Persons (     Personid int NOT NULL AUTO\_INCREMENT,     LastName varchar(255) NOT NULL,     FirstName varchar(255),     Age int,     PRIMARY KEY (Personid) );

## ALTER TABLE Persons AUTO\_INCREMENT=100;

## The MySQL INSERT INTO Statement

## INSERT INTO table\_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);

## INSERT INTO table\_name VALUES (value1, value2, value3, ...);

## The MySQL UPDATE Statement

## UPDATE table\_name SET column1 = value1, column2 = value2, ... WHERE condition;

## UPDATE Customers SET ContactName = 'Alfred Schmidt', City = 'Frankfurt' WHERE CustomerID = 1;

## The MySQL DELETE Statement

## DELETE FROM table\_name WHERE condition;

## The MySQL LIMIT Clause

The LIMIT clause is used to specify the number of records to return.

The LIMIT clause is useful on large tables with thousands of records. Returning a large number of records can impact performance

SELECT column\_name(s)  
FROM table\_nameWHERE condition  
LIMIT number;

* 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.

DDL commands

|  |  |
| --- | --- |
| CREATE DATABASE | Creates a new database. |
| DROP DATABASE | Deletes a database. |
| CREATE TABLE | Creates a new table in a database. |
| ALTER TABLE | Alters the structure of an existing table. |
| DROP TABLE | Removes a table from a database. |

Rename

1. **Rename a Table**

RENAME TABLE current\_table\_name TO new\_table\_name;

1. **Rename a Column:**

ALTER TABLE table\_name

CHANGE COLUMN old\_column\_name new\_column\_name new\_data\_type;

DML Commands

DML stands for **Data Manipulation Language**. It deals with data manipulation and includes the most common SQL statements such as SELECT, INSERT, UPDATE, DELETE,

|  |  |
| --- | --- |
| SELECT | Retrieves data from a table. |
| INSERT | Inserts new data into a table. |
| UPDATE | Updates existing data in a table. |
| DELETE | Deletes data from a table. |

INSERT

INSERT INTO table\_name (column1, column2, column3, ...)

VALUES (value1, value2, value3, ...);

Update

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

Delete

DELETE FROM table\_name WHERE condition;