DDL Commands for MySQL

DDL (Data Definition Language) commands in MySQL are used to define and manage database structures such as databases, tables, and indexes. Here's a comprehensive guide to using all DDL commands, with examples based on creating your own database and table.

DDL (Data Definition Language) commands in MySQL are used to define and manage database structures such as databases, tables, and indexes. Here's a comprehensive guide to using all DDL commands, with examples based on creating your own database and table.

# CREATE DATABASE

This command is used to create a new database.

**CREATE** DATABASE my\_database;

sql

# USE DATABASE

This command selects the database to use.

USE my\_database;

sql

# CREATE TABLE

This command creates a new table in the selected database.

**CREATE TABLE** my\_table (

id INT AUTO\_INCREMENT **PRIMARY** KEY, name VARCHAR(100) **NOT NULL**,

age INT **NOT NULL**,

created\_at TIMESTAMP **DEFAULT** CURRENT\_TIMESTAMP

);

sql

# ALTER TABLE

This command is used to modify an existing table structure.

Add a new column

Modify an existing column

**ALTER TABLE** my\_table **ADD COLUMN** email VARCHAR(100);

sql

Change a column name and type

**ALTER TABLE** my\_table MODIFY **COLUMN** name VARCHAR(150) **NOT NULL**;

sql

Drop a column

**ALTER TABLE** my\_table CHANGE **COLUMN** email email\_address VARCHAR(200);

sql

Add a primary key (if not already defined)

**ALTER TABLE** my\_table **DROP COLUMN** age;

sql

**ALTER TABLE** my\_table **ADD PRIMARY** KEY (id);

sql

 Add a foreign key

Assuming there is another table **`another\_table`** with a column **`another\_id`**:

**ALTER TABLE** my\_table **ADD COLUMN** another\_id INT;

**ALTER TABLE** my\_table **ADD CONSTRAINT** fk\_another\_id **FOREIGN** KEY (another\_id) **REFERENCES**

another\_table(another\_id);

sql

# RENAME TABLE

This command renames an existing table.

RENAME **TABLE** my\_table **TO** my\_new\_table;

sql

# DROP TABLE

This command deletes an existing table and all of its data.

**DROP TABLE** my\_new\_table;

sql

# TRUNCATE TABLE

This command removes all rows from a table but does not delete the table structure.

**TRUNCATE TABLE** my\_table;

sql

# CREATE INDEX

This command creates an index on one or more columns of a table to improve query performance.

 Single-column index

Multi-column index

**CREATE** INDEX idx\_name **ON** my\_table (name);

sql

**CREATE** INDEX idx\_name\_created\_at **ON** my\_table (name, created\_at);

sql

# DROP INDEX

This command removes an existing index.

**DROP** INDEX idx\_name **ON** my\_table;

sql

# CREATE VIEW

This command creates a virtual table based on the result set of an SQL query.

**CREATE VIEW** my\_view **AS SELECT** name, created\_at **FROM** my\_table

**WHERE** created\_at > '2024-01-01';

sql

# DROP VIEW

This command removes an existing view.

**DROP VIEW** my\_view;

sql

# Complete Example Sequence

-- Create a new database

**CREATE** DATABASE my\_database;

-- Use the created database USE my\_database;

-- Create a new table

**CREATE TABLE** my\_table (

id INT AUTO\_INCREMENT **PRIMARY** KEY, name VARCHAR(100) **NOT NULL**,

age INT **NOT NULL**,

created\_at TIMESTAMP **DEFAULT** CURRENT\_TIMESTAMP

);

-- Alter the table to add a new column

**ALTER TABLE** my\_table **ADD COLUMN** email VARCHAR(100);

-- Modify an existing column

**ALTER TABLE** my\_table MODIFY **COLUMN** name VARCHAR(150) **NOT NULL**;

-- Change a column name and type

**ALTER TABLE** my\_table CHANGE **COLUMN** email email\_address VARCHAR(200);

-- Drop a column

**ALTER TABLE** my\_table **DROP COLUMN** age;

-- Rename the table

RENAME **TABLE** my\_table **TO** my\_new\_table;

sql

-- Create an index

**CREATE** INDEX idx\_name **ON** my\_new\_table (name);

-- Drop the index

**DROP** INDEX idx\_name **ON** my\_new\_table;

-- Create a view

**CREATE VIEW** my\_view **AS SELECT** name, created\_at **FROM** my\_new\_table

**WHERE** created\_at > '2024-01-01';

-- Drop the view

**DROP VIEW** my\_view;

-- Truncate the table

**TRUNCATE TABLE** my\_new\_table;

-- Drop the table

**DROP TABLE** my\_new\_table;

-- Drop the database

**DROP** DATABASE my\_database;

This sequence of commands demonstrates the use of all DDL operations to create, modify, manage, and delete database structures in MySQL.