

Sentencias Fundamentales del DDL

1. Sentencias Create.

Las operaciones más importantes que se pueden realizar con la sentencia **Create** son las siguientes:

- **Create database.**

Permite crear una base de datos con un nombre dado.

Requiere privilegio CREATE en la base de datos. Create schema es sinónimo de Create database.

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
    [create_option] ...

create_option: [DEFAULT] {
    CHARACTER SET [=] charset_name
    | COLLATE [=] collation_name
    | ENCRYPTION [=] {'Y' | 'N'}
}
```

Sentencias Fundamentales del DDL

1. Sentencias Create.

- **Create table.**

Permite crear una tabla con un nombre dado.

Se crea por defecto con el motor de almacenamiento InnoDB y debe existir una base de datos sobre la que se crea dicha tabla.

MySQL no establece ningún límite en el número de tablas.

Requiere privilegio CREATE en la tabla.

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...)
    [AUTO_INCREMENT [=] value]

create_definition: {
    col_name data_type [NOT NULL | NULL] [DEFAULT {literal | (expr)}] [AUTO_INCREMENT] [UNIQUE [KEY]] [[PRIMARY] KEY]

| {INDEX | KEY} [index_name] [index_type] (key_part,...) [index_option] ...
| {FULLTEXT | SPATIAL} [INDEX | KEY] [index_name] (key_part,...) [index_option] ...
| [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (key_part,...) [index_option] ...
| [CONSTRAINT [symbol]] UNIQUE [INDEX | KEY] [index_name] [index_type] (key_part,...) [index_option] ...
| [CONSTRAINT [symbol]] FOREIGN KEY [index_name] (col_name,...) reference_definition
| [CONSTRAINT [symbol]] CHECK (expr) [[NOT] ENFORCED]
}
```

2. Sentencias Alter.

Las operaciones más importantes que se pueden realizar con la sentencia Alter son las siguientes:

- **Alter database.**

Permite cambiar las características generales de la base de datos.

Requiere privilegios de ALTER (modificación) en la base de datos.

Alter Schema es sinónimo de Alter database.

```
ALTER {DATABASE | SCHEMA} [db_name]
    alter_option ...

alter_option: {
    [DEFAULT] CHARACTER SET [=] charset_name
| [DEFAULT] COLLATE [=] collation_name
| [DEFAULT] ENCRYPTION [=] {'Y' | 'N'}
| READ ONLY [=] {DEFAULT | 0 | 1}
}
```

Sentencias Fundamentales del DDL

2. Sentencias Alter.

- **Alter table.**

Permite cambiar la estructura de una tabla.

```
ALTER TABLE tbl_name
    [alter_option [, alter_option] ...]

alter_option: {
    table_options
| ADD [COLUMN] col_name column_definition [FIRST | AFTER col_name]
| ADD [COLUMN] (col_name column_definition,...)
| ADD {INDEX | KEY} [index_name] [index_type] (key_part,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (key_part,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] UNIQUE [INDEX | KEY] [index_name] [index_type] (key_part,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] FOREIGN KEY [index_name] (col_name,...) reference_definition
| ADD [CONSTRAINT [symbol]] CHECK (expr) [[NOT] ENFORCED]

| CHANGE [COLUMN] old_col_name new_col_name column_definition [FIRST | AFTER col_name]
| MODIFY [COLUMN] col_name column_definition [FIRST | AFTER col_name]
| RENAME COLUMN old_col_name TO new_col_name

| DROP {CHECK | CONSTRAINT} symbol
| DROP [COLUMN] col_name
| DROP {INDEX | KEY} index_name
| DROP PRIMARY KEY
| DROP FOREIGN KEY fk_symbol

| RENAME {INDEX | KEY} old_index_name TO new_index_name
| RENAME [TO | AS] new_tbl_name
```

Sentencias Fundamentales del DDL

3. Sentencias Drop.

Las operaciones más importantes que se pueden realizar con la sentencia **Drop** son las siguientes:

- **Drop database.**

Permite borrar todas las tablas de la base de datos y elimina también la propia base de datos.

Hay que ser muy cuidadoso con esta sentencia.

Requiere el privilegio DROP en la base de datos. Drop schema es similar a Drop database.

```
DROP {DATABASE | SCHEMA} [IF EXISTS] db_name
```

- **Drop table.**

Permite borrar una o más tablas.

Se requiere el privilegio DROP en cada tabla.

El borrado de una tabla también incluye la eliminación de sus triggers asociados.

```
DROP [TEMPORARY] TABLE [IF EXISTS]  
tbl_name [, tbl_name] ...  
[RESTRICT | CASCADE]
```

Sentencias Fundamentales del DDL

4. Sentencia Rename.

La sentencia RENAME TABLE renombra una o más tablas.

Requiere los privilegios de ALTER y DROP en la tabla original y CREATE e INSERT en la nueva tabla.

```
RENAME TABLE  
  tbl_name TO new_tbl_name  
[, tbl_name2 TO new_tbl_name2] ...
```

Sentencias Fundamentales del DDL

4. Sentencia Rename.

Los usos más importantes de esta sentencia son los siguientes:

- Renombramiento de una única tabla.

```
RENAME TABLE old_table TO new_table;
```

- Renombramiento de varias tablas.

```
RENAME TABLE old_table1 TO new_table1,  
               old_table2 TO new_table2,  
               old_table3 TO new_table3;
```

- Intercambio de contenido entre las tablas old_table y new_table:

```
RENAME TABLE old_table TO tmp_table,  
               new_table TO old_table,  
               tmp_table TO new_table;
```

- Sentencias similares.

```
RENAME TABLE old_table TO new_table;
```

```
ALTER TABLE old_table RENAME new_table;
```

- Mover una tabla de una base de datos a otra, dejando la inicial sin dicha tabla:

```
RENAME TABLE current_db.tbl_name TO other_db.tbl_name;
```

5. Sentencia Truncate.

La sentencia TRUNCATE limpia completamente una tabla.

Requiere el privilegio **DROP** (borrado).

Lógicamente, la sentencia **Truncate table** es similar a una sentencia **Delete** que borre todas las filas, o a una secuencia de sentencias **Drop table** y **Create table**.

```
TRUNCATE [TABLE] tbl_name
```