

Las operaciones que nos ofrece DDL para trabajar con la estructura de la base de datos son:

CREATE
USE
ALTER
DROP
SHOW
TRUNCATE
RENAME.

Podemos ver las bases de datos en nuestro servidor con la orden:
SHOW DATABASES

Borrado de una base de datos

La orden para borrar una base de datos es:
DROP DATABASE nombre_bdd

Drop tipo_objeto nombre_objeto

La orden DROP borra de forma irreversible una base de datos, por tanto debemos usarla cuidadosamente.

Al borrar la base de datos es obvio que perderemos todos los datos que se guardaban en ella.

Ejemplo:

```
CREATE TABLE Alumno (  
    Matricula INT AUTO_INCREMENT PRIMARY KEY,  
    dni        varchar(9) NOT NULL UNIQUE,  
    nombre     varchar (50),  
    apellido   varchar (50),  
    edad       INT UNSIGNED,
```

```
nota_media int DEFAULT 5,  
  
tutor INT NOT NULL,  
  
CONSTRAINT FK_Alumno_Tutor FOREIGN KEY (tutor) REFERENCES  
Tutor(codigo));
```

Después de haber creado una tabla podemos querer verla, revisar su estructura o borrarla.

Para ver las tablas de una base de datos:

```
SHOW TABLES
```

Para ver la estructura de una tabla:

```
SHOW COLUMNS FROM nombre_tabla
```

Para borrar una tabla:

```
DROP TABLE nombre_tabla
```

Modificación de tablas.

Después de haber creado una tabla, podremos cambiarle el nombre, agregar, eliminar o renombrar una columna, o cambiar el tipo de datos y cualquier otro atributo de la columna.

La estructura básica de la orden para modificar una tabla es:

```
ALTER TABLE nombre_tabla
```

Añadir una columna a una tabla:

```
ALTER TABLE Alumno ADD campo INT
```

Añadir una descripción a una columna:

```
ALTER TABLE Alumno ALTER campo SET DEFAULT 5
```

Eliminar el valor predeterminado de una columna:

ALTER TABLE Alumno ALTER campo DROP DEFAULT

Cambiar el nombre y la definición de una columna:

ALTER TABLE Alumno CHANGE campo campo2 float(8,2)

Modificar la definición de una columna conservando su nombre:

ALTER TABLE Alumno MODIFY campo2 dec(10,2)

Eliminar una columna de una tabla:

ALTER TABLE Alumno DROP campo2

Modificar el nombre de una tabla:

ALTER TABLE Alumno RENAME Estudiante

Añadir una restricción a una tabla:

ALTER TABLE Alumno ADD CONSTRAINT FK_Alumno_Delegado FOREIGN KEY (cod_delegado) REFERENCES Alumno(codigo_alumno);

Eliminar una restricción de una tabla:

ALTER TABLE Alumno DROP FOREIGN KEY FK_Alumno_Delegado;

Eliminar llaves primarias de una tabla:

ALTER TABLE Alumno DROP PRIMARY KEY;

Sentencia RENAME

Hemos visto que con la sentencia ALTER podemos modificar el nombre de una tabla de nuestra base de datos.

MySQL proporciona además la sentencia RENAME para el mismo propósito. Su sintaxis es la siguiente:

```
RENAME nombre_actual TO nombre_nuevo;
```

Podemos usar la sentencia RENAME para renombrar varias tablas:

```
RENAME table_1 TO table_1.0, table_2 TO table_2.0;
```

Sentencia TRUNCATE

La sentencia TRUNCATE elimina todos los datos de una tabla. La sintaxis es la siguiente:

```
TRUNCATE TABLE nombre_tabla;
```

delete

La sentencia TRUNCATE puede parecer equivalente a realizar un DELETE de todos los datos de una tabla, pero hay diferencias:

TRUNCATE es más rápido que DELETE. Se destruye la tabla y se crea de nuevo.
El campo AUTO_INCREMENT se reinicia (la próxima fila que se añada a la tabla tomará el primer valor del campo AUTO_INCREMENT).
TRUNCATE es una operación atómica (no transaccional).