**SQL (Structured Query Language)**

**Poniendo en práctica la teoría en MySQL: DDL**

El lenguaje SQL tiene dos grandes rubros o dos, digamos, sublenguajes. Son DDL &

**DDL (Data Definition Language)**

Es un lenguaje que nos ayuda a crear la estructura de una base de datos. Más precisamente, nos ayuda a crear los cimientos: las relaciones, las entidades; todo eso que ya hemos venido hablando en la *teoría* con respecto a los diagramas; entonces, es este el lenguaje que nos va a permitir crear todas esas estructuras ya de facto en una base de datos real. **En DDL se utilizan tres grandes comandos: Create, Alter & Drop.**

**Create**

Como su nombre lo dice, nos ayuda a crear, ¿crear qué cosa? Pues, podría ser una base de datos (schema), también podría crear una tabla o una vista; al igual, podría ser un índice, etc.

**Alter**

Como su nombre también lo dice, nos va a permitir *alterar* o modificar algunas de esas entidades que hemos creado con *create;* al igual,podemos modificar una tabla, por ejemplo, agrengandole una nueva columna (o quitandosela); o, también, podría considerar cambiar un tipo de dato de alguna columna, por ejemplo.

**Drop**

*Soltar.* Tiene que ver con *dejar ir.* Es básicamente la sentencia que nos va a ayudar a **borrar;** por lo que, toca tener mucho cuidado con ella. Al hacer *drop,* por ejemplo, tú puedes borrar una columna, borrar una tabla completa o, incluso, toda tu base de datos.

**Los tres grandes grupos de objetos que vamos a manipular con el lenguaje DDL son: Database, Table & View.**

**Database:** Se entiende como el repositorio de datos que va a usar algún proyecto, generalmente grande, que se encuentre desarrollando.

**Table:** Son la traducción a SQL de cómo crear las entidades, la puesta en marcha de lo que ya hemos visto en la parte teórica de las bases de datos relacionales; es decir, con *table* es que se crean ya las entidades, pero haciendo uso de un lenguaje de programación como SQL. Es como se pone en práctica toda la teoría relacionada a la creación de entidades, tanto la parte de *entidad-relación* como los *diagramas físicos*; también, lo que tiene que ver con sus atributos, tipos de datos y restricciones, etc: la información que envuelve a toda *table* en sí.

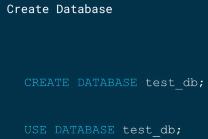
**View:** Hace referencia a la manera en cómo proyectamos toda la información de una base de datos de tal manera que se entienda muy bien, casi que para cualquier persona, lo que se transmite. Usted se recuerda que, como ya tendríamos *normalizada* toda la información; tendríamos, entonces, un pedacito de la información en una tabla, otro pedacito en otra tabla *,* y otro en otra *(table)* y así sucesivamente. Como la idea es que toda esa información pueda ser presentada de una manera clara y conjunta, de una manera coherente para cualquier persona y que sirva para comunicar una información concreta, interpretable y con sentido propio, se recurre al recurso de los *views* por medio de los *selects* (más adelante profundizamos este nuevo concepto).



***Entonces, tenemos que, los comandos (Create, Alter & Drop) son los que manipulan a los objetos (Database, Table & View). Echemos un vistazo a las primeras sentencias que podemos declarar al conjugar ambos factores: comandos & objetos.***

***Un vistazo a las primeras sentencias...***

***Ejemplo 1.***



Lo que estamos haciendo es, por medio del **comando** *create,* crear una base de datos. La base de datos, recuerdese, es un **objeto:** *Database (*o lo que es lo mismo: *Schema).* Posteriormente le hemos pasado un nombre a la reciente base de datos creada, el cual es: “test\_db”. Esto en cuanto a nuestra primera **sentencia.**

Luego, en la segunda sentencia, hemos dejado claro que quiero usar la base de datos “test\_db” al declarar la expresión *use.* Es preciso hacer uso del recurso *use* para que la consola interprete que tienes definido sobre qué objeto vas a ejercer una acción y/o manipular (hacer consultas o algún cambio).

Apliquemos la primera sentencia, del ejemplo 1, en una base de datos real por medio de MySQL Workbench; esto para que veas, ahora que ya conoces, cuáles son las sentencias (comandos) y cuáles son los objetos y, en consecuencia, cuál sería la estructura que nos va a quedar al final en una sentencia SQL.