

SQL

¿Qué es SQL?

Es un lenguaje para manipular una BD. Se utiliza para crear tablas, modificar tablas, consultar datos, etc....

El lenguaje SQL se compone de tres sublenguajes: LDD, LMD y LCD.

LDD → Crear y borrar BD y tablas. Modificar la estructura de una tabla.

LMD → Consultar datos de una BD, insertar, borrar y modificar registros.

LCD → Permite crear y borrar usuarios y otorgar y eliminar privilegios de acceso a los usuarios.

Antes de ver los diferentes lenguajes voy a indicar algunas especificaciones a tener en cuenta.

1. Símbolos utilizados en la sintaxis y su significado.
 - Las llaves {} indican obligación, es decir que debe ir una sentencia no puede estar vacío.
 - Los corchetes [] indican opcionalidad.
 - La barra | indica que se puede elegir entre una u otra opción.
2. Todas las sentencias de mysql terminan con punto y coma ;
3. Aunque en el programa que vamos a utilizar se puede crear todo de forma gráfica NO lo utilizaremos. Siempre vamos a utilizar código SQL, para ello tendremos que ir a la pestaña SQL.
4. Los tipos de datos de los campos siempre van a ser varchar. Solo serán int o integer si se van a realizar operaciones con

ellos, es decir si es un precio o una cantidad. Las fechas serán de tipo date. En aules tenéis un PDF con los tipos de datos.

LDD

Veamos las instrucciones del LDD:

- Creación de una base de datos

- CREATE DATABASE nombre_bd;

Ejemplo: create database instituto;

el ejemplo crea una BD llamada instituto.

- Borrar una base de datos

- DROP DATABASE nombre_bd;

Ejemplo: drop database instituto;

El ejemplo borra la base de datos instituto. El SGBD siempre preguntara si está seguro a la hora de borrar.

- Creación de una tabla

- CREATE TABLE nombre_tabla

(

Nombre_columna tipo_dato [NOT NULL] [UNIQUE]
[DEFAULT valor] [PRIMARY KEY],

....

PRIMARY KEY (lista claves primarias),

FOREIGN KEY (lista claves ajenas) REFERENCES
tabla_existente(campoalquesereferencia)

)

[NOT NULL] → indicara que ese campo no puede contener valores nulos

[UNIQUE] → indicara que los valores de ese campo no se pueden repetir. Serán las claves alternativas.

[DEFAULT valor] → Se puede indicar un valor por defecto para ese campo.

La diferencia de establecer la clave primaria en el nombre de la columna o después, es que en el primer caso no se pueden definir claves primarias compuestas y en el segundo caso sí. Los campos en la lista de claves tanto primarias como ajenas van separados por comas.

Ejemplo: create table cliente(

dni varchar(9) primary key,

nombre varchar(50),

dirección varchar(50),

telefono varchar(9)

);

create table pedidos(

codigo varchar(5) primary key,

fecha date,

dni varchar(9),

foreign key (dni) references cliente (dni)

);

Todas las instrucciones del create table terminan con una coma , excepto la última.

- Borrar una tabla de la base de datos

- DROP TABLE nombre_tabla;

Ejemplo: drop table cliente;

- Renombrar tablas

- RENAME TABLE nombre_tabla_existente TO nombre_nuevo;

Ejemplo: rename table cliente to cliente1;

- Modificar estructura de una tabla

- ALTER TABLE nombre_tabla

- {

- ADD nombre_comlumna tipo [NOT NULL][UNIQUE] [FIRST | AFTER nombre_columna_existente] *añade una columna a la tabla. Si no indicamos la posición se añadirá al final de la tabla.*

- | DROP COLUMN nombre_columna_existente *borra una columna de la tabla*

- | DROP PRIMARY KEY *borra la clave primaria*

- | DROP FOREIGN KEY nombre_clave_ajena *borra la clave ajena indicada*

- | MODIFY nombre_columna_existente nuevo_tipo_datos [NOT NULL][,...] *cambia el tipo de datos del campo NO su nombre*

|CHANGE COLUMN nombre_columna_existente
nombre_nueva_columna nombre_tipo *cambia el nombre de
la columna*

| ADD UNIQUE (nombre_columna_existente, ...) *añade un
campo único*

| ADD PRIMARY KEY (nombre_columna_existente, ...) *añade una clave primaria*

| ADD FOREIGN KEY (nombre_columna_existente)
REFERENCES nombre_tabla_referenciada
(nombre_columna_referenciada) *añade una clave ajena*

};

Las llaves no se ponen, indican que después del alter table
debe ir una sentencia.

Ejemplos:

Añadir un campo a una tabla	ALTER TABLE empleado ADD nombre VARCHAR(50);
Podemos añadir varios campos a la vez	ALTER TABLE empleado ADD nombre VARCHAR(50), apellidos VARCHAR(40);
Borrar columna de una tabla	ALTER TABLE empleado DROP COLUMN nombre;
Cambia tipo de datos de un campo	ALTER TABLE cliente MODIFY categoría integer;
Cambia el nombre de la columna	ALTER TABLE pedido CHANGE cant cantidad integer;