

SQL

Structured Query Language

Lenguaje de consulta estructurado

¿Qué es SQL?

- **Structured Query Language** (Lenguaje de Consulta Estructurado)
- Comentarios en SQL
 - Dos guiones medios y **espacio**: `-- esto es un comentario`
 - Signo de número y espacio: `# esto es otro comentario`
 - Comentarios multilínea:
`/* esto es un comentario`

`en varias`

`Líneas */`

- **Las instrucciones (sentencias) SQL terminan con ;
(punto y coma)**

¿Qué es SQL?

- **Structured Query Language** (Lenguaje de Consulta Estructurado)
- **DDL** (Lenguaje de Definición de Datos): Es el lenguaje encargado de la definición, alteración y eliminación de objetos en la base de datos. (CREATE, ALTER, DROP)
- **DML** (Lenguaje de Manipulación de Datos): Es el lenguaje encargado de la manipulación de los datos. (SELECT, INSERT, UPDATE, DELETE)
- **DCL** (Lenguaje de Control de Datos): Es el lenguaje que controla el acceso a las bases de datos. (GRANT, REVOKE)
- **TCL** (Lenguaje de Control de Transacciones): Es el lenguaje que controla la ejecución de comandos DML; indispensable para la consistencia e integridad de los datos.

DDL

Lenguaje de Definición de Datos

- CREATE
- DROP

Lenguaje de Definición de Datos (DDL)

- **DDL** (Lenguaje de Definición de Datos): Es el lenguaje encargado de la definición, alteración y eliminación de objetos en la base de datos. (CREATE, ALTER, DROP)
- Ver las bases de datos que tenemos:
 - **SHOW DATABASES;**
- Crear una base de datos:
 - **CREATE DATABASE nombre_DB;**
 - *(se recomiendan nombres en minúsculas, sin espacios ni caracteres especiales)*
- Borrar una base de datos:
 - **DROP DATABASE nombre_DB;**

Ejemplos

- Crear 3 bases de datos
 - `create database ejemplo1;`
 - `create database ejemplo2;`
 - `create database ejemplo3;`
- Mostrar las bases de datos creadas
 - `show databases;`
- Habilitar la base de datos *ejemplo1*
 - `use ejemplo1`
- Borrar la base de datos *ejemplo2*
 - `drop database ejemplo2;`
- Mostrar las bases de datos
 - `show databases;`
- Cambiarse a la base de datos *ejemplo3*
 - `use ejemplo3`
- Observar en el prompt el cambio de base

Lenguaje de Definición de Datos (DDL)

- Mostrar cómo fue creada una BD
 - `SHOW CREATE DATABASE ejemplo1;`
- Mostrar listado de codificaciones
 - `SHOW CHARACTER SET;`
- Establecer la codificación de la BD:
 - `CREATE DATABASE nombre_DB DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;`

Ejercicios

- Crear la base de datos ejemplo2 asignando la codificación utf8
 - `CREATE DATABASE ejemplo2 DEFAULT CHARACTER SET utf8
DEFAULT COLLATE utf8_general_ci;`
- Mostrar cómo fueron creadas las BD
 - `SHOW CREATE DATABASE ejemplo2;`
 - `SHOW CREATE DATABASE ejemplo1;`

Creación de tablas

- Una base de datos almacena la información en tablas
- **¿Qué es una tabla?**
 - Es una estructura que organiza los datos en filas y columnas.
 - Cada columna es un campo o atributo.
 - Cada fila es un registro
 - Cada campo debe tener un nombre único que hace referencia a la información que almacenará
 - Cada columna tiene asociado un **tipo de dato** (integridad de dominio)
 - Esta información se usa al momento de crear la tabla

Tipos de datos

- **Numéricos:**

- **tinyint** [unsigned] → 1 byte (con signo: -128 a 127, sin signo: 0 a 255)
- **smallint** [unsigned] → 2 bytes (con signo: -32,768 a 32,767, sin signo: 0 a 65,535)
- **mediumint** [unsigned] → 3 bytes (con signo: -8,388,608 a 8,388,607, sin signo: 0 a 16,777,215)
- **int** [unsigned] → 4 bytes (con signo: -2,147,483,648 a 2,147,483,647, sin signo: 0 a 4,294,967,295)
- **bigint** [unsigned] → 8 bytes
(con signo: -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807,
sin signo: 0 a 18,446,744,073,709,551,615)

Tipos de datos

- **Numéricos de tipo flotante:**

- **float (e,d)** total de dígitos de **e** enteros y **d** decimales (4 bytes)
- **double (e,d)** igual a float pero de doble precisión (8 bytes)
- **decimal (e,d)** igual a float, se utiliza para manejar cantidades de dinero (4 bytes)

Tipos de datos

- **Texto:**

- **char(n)**: texto fijo (n byte)
- **varchar(n)** : texto variable (n bytes)
- **tinytext** (16 kb)
- **tinyblob** (16kb)
- **text** (64kb)
- **blob** (64kb)
- **enum(valores)**

Tipos de datos

- **Tiempo:**

- **Date:** Tipo fecha 'YYYY-MM-DD' ó 'YY-MM-DD' ó 'YYMMDD'
- **Time:** Tipo hora 'HH:MM:SS' ó 'HHMMSS'
- **DateTime:** 'YYYY-MM-DD HH-MM-SS'
- **Year:** Tipo año 'YYYY'
- **Timestamp:** Tipo instante 'YYYYMMDDhhmmss'

Sintaxis básica para crear tablas

```
CREATE TABLE nombre_tabla(  
nombre_campo1 TIPO_DATO,  
nombre_campo2 TIPO_DATO,  
....  
nombre_campoN TIPO_DATO);
```

Convención:
nombres de tablas
en minúsculas, sin
espacios y sin
caracteres
especiales como
acentos.

Ejemplos:

- Habilitar la base de datos a usar: **use ejemplo1**
- Crear la tabla usuario

```
CREATE TABLE usuario(  
id int, nombre varchar(30), rfc char(13));
```

Comandos para manejo de tablas

- Mostrar las tablas de la BD:
 - **SHOW TABLES;**
- Mostrar la **estructura** de las tablas:
 - **SHOW CREATE TABLE** nombre_tabla;
- Ejemplo: show create table usuario;
- Mostrar la **descripción** de las tablas:
 - **DESCRIBE** nombre_tabla;
 - **DESC** nombre_tabla;
- Ejemplo: desc usuario;
- **Borrar** una tabla de la BD:
 - **DROP TABLE** nombre_tabla;
- Ejemplo: drop table usuario;

Ejercicios

1. Crear una base de datos llamada *almacen*
2. Mostrar las bases de datos
3. Habilitar la base de datos *almacen*
4. Generar las siguientes tablas:
 - **fabricante**
 - Atributos:
 - clave_fabricante: cadena de longitud de 10 caracteres,
 - nombre: cadena de hasta 30 caracteres de longitud
 - **articulo**
 - Atributos:
 - clave_articulo: cadena de longitud de 5 caracteres,
 - nombre: cadena de hasta 30 caracteres de longitud
 - precio: valor decimal

¿Preguntas?