

Base de Datos I

Introducción a SQL - DDL

Herramientas

- MySQL

- Es un Sistema de Gestión de Bases de Datos (SGBD)
- Tiene una versión gratuita y otra paga
- Hay que descargar el [Server](#) y el [Workbench \(IDE\)](#)



- SQLFiddle (opcional)

- Es una web que genera una Base de datos para que podamos trabajar directamente
- Se puede acceder en sqlfiddle.com

Conceptos

- SQL (Structured Query Language)
 - DDL (Data Definition Language)
 - Se utiliza para crear y modificar la estructura de la Base de datos
 - Algunas palabras reservadas son: CREATE, ALTER, DROP, TRUNCATE
 - DML (Data Manipulation Language)
 - Son sentencias utilizadas para la manipulación (crear, eliminar, modificar, consultar) de los datos de una base de datos.
 - Algunas palabras reservadas: SELECT, INSERT, UPDATE, DELETE,

Tipos de datos

Tipos de datos

- INT

- Ocupa 4 bytes
- Puede alojar un número del -2147483648 al 2147483647
- Opción UNSIGNED
- Otras variantes: TINYINT, SMALLINT, MEDIUMINT, BIGINT

Tipos de datos

- **DECIMAL(M,D)**

- Longitud no requerida. M=Longitud total, incluyendo decimales (Default 10), D=Decimales (Default 0).
- Es equivalente a NUMERIC
- Otras variantes: FLOAT, DOUBLE

Tipos de datos

- BOOL, BOOLEAN
 - Es equivalente a TINYINT

Tipos de datos

- DATE

- Formato Default: 'yyyy-mm-dd'
- La base de datos almacena la fecha en un formato desconocido por nosotros
- Para enviar y obtener fechas a la base de datos utilizamos una cadena de caracteres ej 'yyyy-mm-dd', 'yyyymmdd', 'yyyy/mm/dd'

Tipos de datos

- TIME
 - Formato Default: 'hh:mi:ss'

Tipos de datos

- DATETIME
 - Formato Default: 'yyyy-mm-dd hh:mi:ss'

Tipos de datos

- CHAR(M)
 - Soporta una cantidad fija de caracteres
 - Longitud no requerida (Default 1)
 - Completa con espacios hasta la longitud
 - Hasta 255 caracteres

Tipos de datos

- **VARCHAR(M)**

- Sirve para guardar una cadena de caracteres variable
- Longitud requerida
- Longitud + 1 byte (cuando longitud es ≤ 255) o 2 bytes (>255)
- Hasta 65532 caracteres

Tipos de datos

- TEXT

- Otras variantes: TINYTEXT, MEDIUMTEXT, LONGTEXT

Sentencias DDL

Nomenclatura

PALABRA RESERVADA OPCION1 | OPCION2 [OPCIONAL] <nombre>;

Creación de una base de datos

```
CREATE DATABASE | SCHEMA [IF NOT EXISTS] <NOMBRE>;
```

```
DROP DATABASE | SCHEMA [IF EXISTS] <NOMBRE>;
```

```
USE <NOMBRE>;
```




```
CREATE SCHEMA IF NOT EXISTS universidad;  
DROP DATABASE universidad;  
USE universidad;
```

Creación de tablas

CREATE TABLE <TABLA>


(<CAMPO_1> <TIPO_DATO_1> [RESTRICCIONES_CAMPO_1],
<CAMPO_2> <TIPO_DATO_2> [RESTRICCIONES_CAMPO_2],
...
<CAMPO_N> <TIPO_DATO_N> [RESTRICCIONES_CAMPO_N],
[RESTRICCIONES_TABLA]);



```
CREATE TABLE alumnos (legajo SMALLINT,  
    nombre VARCHAR(40),  
    apellido VARCHAR(40),  
    fecha_nacimiento DATE);
```

Restricciones de campo

- NOT NULL
- PRIMARY KEY (Claves primarias simples)
- UNIQUE (Claves únicas simples)
- AUTO_INCREMENT (Tipos numéricos. Debe ser PK. Sólo puede haber una)
- DEFAULT <VALOR>



```
CREATE TABLE alumno (legajo SMALLINT PRIMARY KEY AUTO_INCREMENT,  
                        nombre VARCHAR(40) NOT NULL,  
                        apellido VARCHAR(40) NOT NULL,  
                        fecha_nacimiento DATE);
```

Restricciones de tabla

- [CONSTRAINT <NOMBRE>] PRIMARY KEY(<LISTA_CAMPOS>)
- [CONSTRAINT <NOMBRE>] FOREIGN KEY(<LISTA_CAMPOS>) REFERENCES <TABLA_REF>(<LISTA_CAMPOS_REF>)
- [CONSTRAINT <NOMBRE>] UNIQUE(<LISTA_CAMPOS>)
- [CONSTRAINT <NOMBRE>] INDEX(<LISTA_CAMPOS>)

Restricciones de campo (Aplicación FK).

```
CREATE TABLE CIUDAD (
```

```
    CodCiudad INT(2) PRIMARY KEY,  
    Nombre VARCHAR(100) NOT NULL);
```

```
CREATE TABLE PROVEEDOR(
```

```
    CodProv INT (5) PRIMARY KEY,  
    Nombre VARCHAR (20) NOT NULL,  
    Domicilio VARCHAR (100),  
    CodCiudad_FK INT(2) NOT NULL,  
    FOREIGN KEY (CodCiudad_FK) REFERENCES Ciudad (CodCiudad));
```

Restricciones de campo (Aplicación FK).

```
CREATE TABLE PRODUCTO (
```

```
    CodProducto INT(2) PRIMARY KEY,  
    Nombre VARCHAR(100) NOT NULL);
```

```
CREATE TABLE CLIENTE (
```

```
    CodCliente INT(2) PRIMARY KEY,  
    Nombre VARCHAR(100) NOT NULL,  
    Apellido VARCHAR(100) NOT NULL,  
    Fecha_Nac DATE NOT NULL);
```

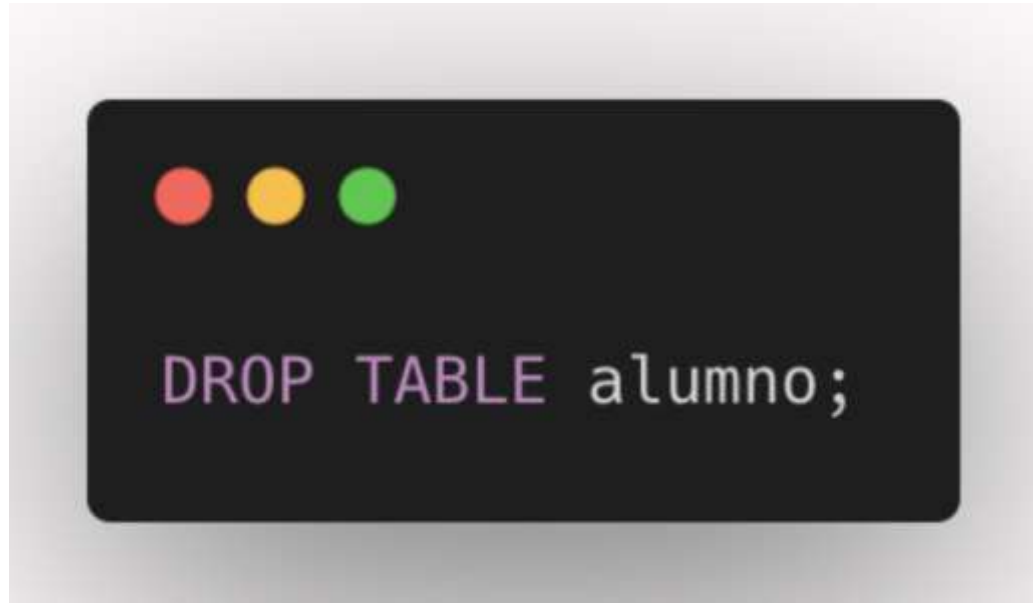
```
CREATE TABLE VENTA(
```

```
    CodVenta INT (5) PRIMARY KEY,  
    Fecha_Venta VARCHAR (20) NOT NULL,  
    CodCliente_FK INT(2) NOT NULL,  
    CodProducto_FK INT(2) NOT NULL,
```

```
    FOREIGN KEY (CodCliente_FK) REFERENCES CLIENTE (CodCliente)),  
    FOREIGN KEY (CodProducto_FK) REFERENCES PRODUCTO (CodProducto)),
```


Borrado de tablas

`DROP TABLE <TABLA>;`



Modificación de tablas

- `ALTER TABLE <TABLA> ADD COLUMN <CAMPO> <TIPO_DATO> [RESTRICCIONES_CAMPO];`
- `ALTER TABLE <TABLA> DROP COLUMN <CAMPO>;`
- `ALTER TABLE <TABLA> CHANGE <NOMBRE_CAMPO_VIEJO> <NOMBRE_CAMPO_NUEVO> <TIPO_DATO> [RESTRICCIONES_CAMPO];`



```
ALTER TABLE alumno ADD COLUMN domicilio VARCHAR(100) NOT NULL;
```

```
ALTER TABLE alumno DROP COLUMN nombre;
```

```
ALTER TABLE alumno CHANGE apellido nombre_apellido VARCHAR(100) NOT NULL;
```

Modificación de restricciones

- `ALTER TABLE <TABLA> ADD PRIMARY KEY(<LISTA_CAMPOS>);`
- `ALTER TABLE <TABLA> DROP PRIMARY KEY;`
- `ALTER TABLE <TABLA> ADD [CONSTRAINT <NOMBRE>] FOREIGN KEY(<LISTA_CAMPOS>) REFERENCES <TABLA_REF>(<LISTA_CAMPOS_REF>);`
- `ALTER TABLE <TABLA> DROP FOREIGN KEY <NOMBRE>;`



```
ALTER TABLE alumno ADD PRIMARY KEY (tipo_documento, numero_documento);  
ALTER TABLE alumno ADD FOREIGN KEY (tipo_documento) REFERENCES tipo_documento(tipo);
```