

Bases de datos

Tema 4

Diseño físico relacional



1º DAW / DAM

Índice

Tema 4 – Diseño físico relacional

1. Introducción
 1. Gestores de bases de datos
 2. Herramientas de bases de datos
 3. Conexión
2. SQL: DDL
3. Operaciones con bases de datos
 1. Creación
 2. Modificación
 3. Borrado
4. Operaciones con tablas
 1. Creación
 2. Tipos de datos
 3. Características
 4. Consulta de las tablas
 5. Modificaciones
 6. Borrado
 7. Renombrado

1. Introducción

1.1 Gestores de Bases de Datos

Un **Sistema Gestor de Base de Datos (SGBD)** o DataBase Managenent System (**DBMS**) es un sistema que permite la **creación, gestión y administración de bases de datos**, así como la **elección y manejo de las estructuras necesarias** para el almacenamiento y búsqueda de información del modo más eficiente posible.

También se les conoce como “motores de bases de datos”, aunque el término no es del todo correcto.

open source



comerciales



1. Introducción

1.2 Herramientas de Bases de Datos

Las **herramientas** de gestión de **bases de datos** proporcionan un interfaz web que permite administrar las **bases de datos**.

Estas **herramientas** también permiten ejecutar **sentencias SQL** desde este interfaz de usuario. También se les conocen como clientes de bases de datos



HeidiSQL



Sequel Pro



1. Introducción

1.3 Conexión a una base de datos

Las conexiones a una base de datos siguen un esquema cliente-servidor, siendo el cliente la propia herramienta que se utilice para interactuar con la base de datos; y el servidor, la propia base de datos.

Un **servidor de base de datos**, también conocido como database server o SGBD en caso de bases de datos relacionales, es un **tipo de software de servidor que permiten la organización de la información mediante el uso de tablas, índices y registros.**

1. Introducción

1.3 Conexión a una base de datos



Para levantar un servidor de bases de datos, se va a instalar el XAMPP, un programa que permite generar un servidor web apache, un servidor MySQL e incluye php instalado, aunque sólo vamos a usar de momento el servidor MySQL



[Link de descarga](#)

[Tutorial instalación](#) (Windows)



Creación de la primera base de datos



Instalar el XAMPP en el equipo

Activar el servidor MySQL

Acceder al localhost/phpmyadmin

Crear vuestra primera base de datos llamada pruebas

Crear una tabla



HeidiSQL



Sequel Pro

Cientes de bases de datos

Descargar [HeidiSQL](#) para Windows
o [SequelPro](#) para MacOS X

Realizar conexión al localhost

Añadir otra tabla a la base de
datos "pruebas"



Actividad en clase

Descargar [HeidiSQL](#) para Windows o [SequelPro](#) para MacOS X

Realizar conexión al localhost

Añadir otra tabla a la base de datos "pruebas"

Índice

Tema 4 – Diseño físico relacional

1. Introducción

1. Gestores de bases de datos
2. Herramientas de bases de datos
3. Conexión

2. **SQL: DDL**

3. Operaciones con bases de datos

1. Creación
2. Modificación
3. Borrado

4. Operaciones con tablas

1. Creación
2. Tipos de datos
3. Características
4. Consulta de las tablas
5. Modificaciones
6. Borrado
7. Renombrado

2. SQL: DDL



SQL: Structured Query Language

SQL es un lenguaje de consultas estructurado que permite crear, manipular y descargar datos de una base de datos relacional. Como ya se ha visto con anterioridad, SQL se puede subdividir en los siguientes apartados o sublenguajes



- **DDL:** Data Definition Language, para la definición de los datos
- **DML:** Data Manipulation Language, para manipular datos de la base de datos
- **DCL:** Data Control Language, para controlar los accesos a los datos de la base de datos

Este tema se centrará sobre el apartado de **DDL** utilizando **MySQL**




2. SQL: DDL

DDL: Data Definition Language

Este sublenguaje del SQL permite definir datos, y sus funciones principales son:

- Crear tablas, índices y otros objetos de la base de datos
- Definir las estructuras físicas donde se almacenarán los objetos de la base de datos como espacios de tablas (tablespaces) o ficheros de datos (datafiles).

DDL tiene 3 instrucciones básicas:

- **CREATE** tipo_objeto Nombre Definición  crea un objeto de un tipo determinado (DATABASE, TABLE, INDEX...), con un nombre(usuarios) y una definición (id, nombre, email, password,...)
- **ALTER** tipo_objeto Nombre Definición  modifica un objeto de un tipo determinado a través de su nombre.
- **DROP** tipo_objeto Nombre  elimina un objeto de un tipo determinado a través de su nombre. DROP TABLE usuarios borraría la tabla de usuarios

2. SQL: DDL

Notación de los apuntes

La notación empleada por estos apuntes incluye palabras clave del propio lenguaje, definiciones de otros elementos, expresiones, variables, etc. Como resumen, se tendría lo siguiente:

- **Palabras en mayúsculas:** palabras reservadas del lenguaje. Ej: SELECT, DROP, CREATE,...
- **Palabras en minúsculas:** por lo general variables, expresiones del usuario o parámetros. Ej: el nombre de una tabla o de un campo.
- **Paréntesis:** agrupa varios valores relacionados. Ej: INSERT ('id', 'email', ...)
- **Corchetes:** indican opcionalidad. Ej: CREATE [TEMPORARY] TABLE
- **Llaves:** indican alternativa obligatoria
- **Puntos suspensivos:** indican continuidad en un ejemplo.

Cada sentencia SQL tiene que ir separada de la anterior por un **punto y coma**.

Índice

Tema 4 – Diseño físico relacional

1. Introducción
 1. Gestores de bases de datos
 2. Herramientas de bases de datos
 3. Conexión
2. SQL: DDL
3. **Operaciones con bases de datos**
 1. Creación
 2. Modificación
 3. Borrado
4. Operaciones con tablas
 1. Creación
 2. Tipos de datos
 3. Opciones de los campos
 4. Consulta de las tablas
 5. Modificaciones
 6. Borrado
 7. Renombrado

3. Operaciones con bases de datos

Creación de Bases de datos

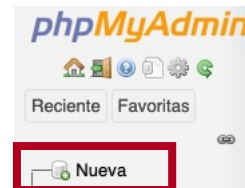
En MySQL, para crear una base de datos se usa el comando CREATE DATABASE:

CREATE DATABASE [IF NOT EXISTS] nombre_bd

Además, se le puede añadir una serie de especificaciones para definir el CHARSET (conjunto de caracteres válidos, p. ej. utf8, ascii, latin1,...) y el tipo de COLLATION (conjunto de reglas que definen cómo comparar y buscar cadenas de caracteres, p.ej. utf8_general_ci, ascii_general_ci, latin1_swedish_ci, etc). [más info](#) y [cuadro resumen](#)

CREATE DATABASE IF NOT EXISTS usuarios **CHARACTER SET** utf8 **COLLATE** utf8_general_ci

Para poder utilizar una base de datos, primero hay que establecer que el resto de operaciones que se vayan a realizar, van a ser sobre dicha base de datos. Para ello, se emplea el comando **USE** nombre_bd.



CREATE DATABASE es equivalente a hacer click en este apartado del phpMyAdmin

3. Operaciones con bases de datos

Modificación de Bases de datos

El comando ALTER DATABASE permite cambiar las características de funcionamiento de una base de datos. No obstante, en MySQL, sólo se puede cambiar el juego de caracteres y su colación

ALTER DATABASE nombre_bd Parametro_a_cambiar Nuevo_valor;

Ej: **ALTER DATABASE** usuarios CHARSET latin1;

Borrado de Bases de datos

El comando DROP DATABASE sirve para eliminar una base de datos específica. En MySQL es necesario indicar el nombre de la base de datos que se desea borrar.

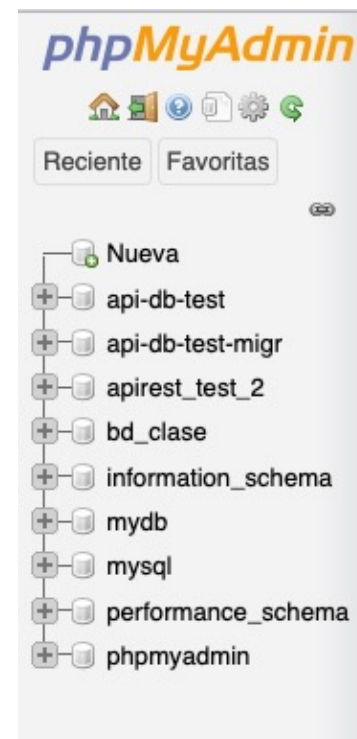
DROP DATABASE nombre_bd;

Ej: **DROP DATABASE** usuarios;

3. Operaciones con bases de datos

Ver Bases de datos del servidor

Para ver las bdd existentes en el servidor, se puede utilizar el comando **SHOW DATABASES**



Índice

Tema 4 – Diseño físico relacional

1. Introducción

1. Gestores de bases de datos
2. Herramientas de bases de datos
3. Conexión

2. SQL: DDL

3. Operaciones con bases de datos

1. Creación
2. Modificación
3. Borrado

4. Operaciones con tablas

1. Creación
2. Tipos de datos
3. Opciones de campo, tabla y referencia
4. Consulta de las tablas
5. Modificaciones
6. Borrado
7. Renombrado

4. Operaciones con tablas

Creación de tablas

En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

```
CREATE [TEMPORARY] TABLE [nombre_bd.]nombre_tabla [(definición_creación,...)] [opciones_tabla]  
[] = Opcional
```

4. Operaciones con tablas

Creación de tablas

En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

```
CREATE [TEMPORARY] TABLE [nombre_bd.]nombre_tabla [(definición_creación,...)] [opciones_tabla]
```

[] = Opcional

Crear tabla temporal

4. Operaciones con tablas

Creación de tablas

En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

CREATE [TEMPORARY] **TABLE** [nombre_bd.]nombre_tabla [(definición_creación,...)] [opciones_tabla]

[] = Opcional

Crear tabla temporal

Especifica el nombre de otra bd

4. Operaciones con tablas

Creación de tablas



En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

CREATE [TEMPORARY] **TABLE** [nombre_bd.]nombre_tabla [(definición_creación,...)] [opciones_tabla]

[]= Opcional

Crear tabla temporal

Especifica el nombre de otra bd

nombre_columna tipo_datos [opciones_campo] [opciones_referencia]

4. Operaciones con tablas

Creación de tablas: tipos de datos

En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

CREATE [TEMPORARY] **TABLE** [nombre_bd.]nombre_tabla [(definición_creación,...)] [opciones_tabla]

[]= Opcional

Crear tabla temporal

Especifica el nombre de otra bd

nombre_columna **tipo_datos** [opciones_campo] [opciones_referencia]

Los tipos de datos que pueden usarse para MySQL son los siguientes:

Enteros



UNSIGNED= se pone cuando no va a tener parte negativa

Tipo de dato	Tamaño	Descripción
TINYINT [UNSIGNED]	1 byte	Entero muy pequeño (-128 a 127)
SMALLINT [UNSIGNED]	2 bytes	Entero pequeño (-32768 a 32767)
MEDIUMINT [UNSIGNED]	3 bytes	Entero mediano (-8388608 a 8388607)
INT [UNSIGNED]	4 bytes	Entero por defecto (-2147483648 to 2147483647)
INTEGER [UNSIGNED]	4 bytes	Igual que INT
BIGINT [UNSIGNED]	8 bytes	Entero grande (-9223372036854775808 a 9223372036854775807)

El uso de paréntesis p.ej. INT(3) no afecta a la cantidad de bytes. [Más info](#)

4. Operaciones con tablas

Creación de tablas: tipos de datos

En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

CREATE [TEMPORARY] **TABLE** [nombre_bd.] nombre_tabla [(definición_creación,...)] [opciones_tabla]

[]= Opcional

Crear tabla temporal

Especifica el nombre de otra bd

nombre_columna **tipo_datos** [opciones_campo] [opciones_referencia]

Los tipos de datos que pueden usarse para MySQL son los siguientes:

Reales



UNSIGNED= se pone cuando no va a tener parte negativa

Tipo de dato	Tamaño	Descripción
FLOAT [UNSIGNED]	4 bytes	Decimales de coma flotante. Representan un valor de forma aproximada. Se les puede especificar la precisión entre paréntesis.
DOUBLE [UNSIGNED]	8 bytes	Igual que FLOAT, pero de mayor tamaño
DECIMAL (longitud, decimales)	variable	Decimales de tipo fijo exacto. Se define el número de dígitos entre paréntesis y el número de decimales
NUMERIC (longitud, decimales)	variable	Igual que DECIMAL

4. Operaciones con tablas

Creación de tablas: tipos de datos

En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

CREATE [TEMPORARY] **TABLE** [nombre_bd.]nombre_tabla [(definición_creación,...)] [opciones_tabla]

[]= Opcional

Crear tabla temporal

Especifica el nombre de otra bd

nombre_columna **tipo_datos** [opciones_campo] [opciones_referencia]

Los tipos de datos que pueden usarse para MySQL son los siguientes:

Caracteres →

Tipo de dato	Tamaño	Descripción
CHAR (longitud)	fijo	Caracteres de longitud fija, de más rápido acceso que VARCHAR. Siempre ocupan lo mismo independientemente del contenido del campo.
VARCHAR (longitud máx)	variable	Cadena de longitud variable, cuya longitud máxima se define entre paréntesis
TEXT	variable	Sus valores son tratados como strings no binarios
BLOB	variable	Valores tratados como strings binarios
ENUM(valor1, valor2, ...)	lista	Sirve para seleccionar un valor de un conjunto de valores fijos
SET(valor1, valor2, ...)	conjuntos	Puede tener 0 o más valores

4. Operaciones con tablas

Creación de tablas: tipos de datos

En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

CREATE [TEMPORARY] **TABLE** [nombre_bd.]nombre_tabla [(definición_creación,...)] [opciones_tabla]

[]= Opcional

Crear tabla temporal

Especifica el nombre de otra bd

nombre_columna **tipo_datos** [opciones_campo] [opciones_referencia]

Los tipos de datos que pueden usarse para MySQL son los siguientes:

Tiempo



Tipo de dato	Formato	Descripción
DATE	aaaa-mm-dd	
TIME	hh:mm:ss	
TIMESTAMP	aaaa-mm-dd hh:mm:ss	
DATETIME	aaaa-mm-dd hh:mm:ss	Mayor rango que TIMESTAMP

[más info](#)

4. Operaciones con tablas

Creación de tablas: opciones de campo

En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

```
CREATE [TEMPORARY] TABLE [nombre_bd.]nombre_tabla [(definición_creación,...)] [opciones_tabla]
```

[]= Opcional

Crear tabla temporal

Especifica el nombre de otra bd

nombre_columna tipo_datos **[opciones_campo]** [opciones_referencia]

Un campo puede tener o no múltiples opciones. Las que pueden usarse para MySQL son las siguientes:

- **NULL | NOT NULL:** especifican si la columna admite o no valores nulos (campo vacío o valores desconocidos)
- **DEFAULT valor:** indica el valor por defecto que tendrá un campo en caso de no especificarse
- **UNIQUE KEY | PRIMARY KEY:** especifica si ese campo es clave primaria o única. La gran diferencia entre ambas es que una clave primaria no puede ser nula, mientras que una única sí.

4. Operaciones con tablas



Creación de tablas: opciones de referencia

En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

CREATE [TEMPORARY] **TABLE** [nombre_bd.]nombre_tabla [(definición_creación,...)] [opciones_tabla]

[]= Opcional

Crear tabla temporal

Especifica el nombre de otra bd

nombre_columna tipo_datos [opciones_campo] **[opciones_referencia]**

Sirve para crear claves foráneas, y sigue la siguiente sintaxis:

nombre_columna tipo_datos **REFERENCES** nombre_tabla_foranea(nombre_columna) **[opciones de comportamiento]**

Hay 2 opciones de comportamiento: **ON DELETE** y **ON UPDATE** y que pueden tomar 3 valores:

- **NO ACTION:** impide la actualización o borrado de la tabla hija si hay una de estas operaciones en la tabla padre
- **CASCADE:** la operación realizada en la tabla padre se propaga a la tabla hija
- **SET NULL:** se establece a NULL la clave foránea ante el borrado o actualización de la tabla padre

4. Operaciones con tablas



Creación de tablas: opciones de tabla

En MySQL, para crear una tabla, se usa el comando CREATE TABLE:

CREATE [TEMPORARY] **TABLE** [nombre_bd.]nombre_tabla [(definición_creación,...)] **[opciones_tabla]**

[]= Opcional

Crear tabla temporal

Especifica el nombre de otra bd

nombre_columna tipo_datos [opciones_campo] [opciones_referencia]



- **ENGINE** = nombre_motor (por lo general Inno_DB)
- **AUTO_INCREMENT** = valor (valor en el que empieza el auto_increment de la tabla)
- **[DEFAULT] CHARACTER SET** juego_caracteres **[COLLATE]** colación
- **CHECKSUM** = {0 | 1}
- **COMMENT** = 'string con comentario'
- **MAX_ROWS** = valor (valor máximo de filas)
- **MIN_ROWS** = valor (valor mínimo de filas)
- **IF NOT EXISTS** (la tabla se crea si no existe)

4. Operaciones con tablas

Creación de tablas: ejemplo

CREATE [TEMPORARY] TABLE [nombre_bd.]nombre_tabla [(definición_creación,...)] [opciones_tabla]



nombre_columna tipo_datos [opciones_campo] [opciones_referencia]

```
CREATE TABLE pruebas_db.usuarios(  
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(20),  
  apellidos VARCHAR(20),  
  nombre_usuario CHAR(10),  
  f_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

también vale



```
id INT UNSIGNED NOT NULL AUTO_INCREMENT,  
PRIMARY KEY (id),
```

```
CREATE TABLE pruebas_db.posts(  
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  titulo VARCHAR(20) NOT NULL,  
  contenido TEXT,  
  usuario INT UNSIGNED REFERENCES usuarios(id),  
);
```

también vale



```
usuario INT UNSIGNED,  
FOREIGN KEY (usuario) REFERENCES usuarios(id)
```

4. Operaciones con tablas

Consulta de tablas



Para consultar las tablas disponibles en una base de datos MySQL, se utiliza el comando **SHOW TABLES**

```
[mysql> show tables;
+-----+
| Tables_in_pruebas_db |
+-----+
| usuarios              |
+-----+
1 row in set (0,00 sec)
```

Por otra parte, para consultar la estructura de una tabla en concreto, se utiliza el comando

DESCRIBE [nombre_bd.]nombre_tabla



```
[mysql> describe usuarios;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default        | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int unsigned  | NO   | PRI | NULL           | auto_increment |
| nombre     | varchar(20)   | YES  |     | NULL           |                |
| apellidos  | varchar(20)   | YES  |     | NULL           |                |
| nombre_usuario | char(15)      | YES  |     | NULL           |                |
| f_creacion | timestamp     | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0,01 sec)
```

4. Operaciones con tablas

Modificación de tablas

El comando para modificar una tabla en MySQL es **ALTER TABLE**

ALTER TABLE nombre_tabla especificación_alter, otra_especificación, ...

Las especificaciones alter pueden ser las siguientes:

- **ADD:** permite añadir una columna (y permite especificar dónde a través de FIRST | AFTER nombre_columna). También sirve para añadir claves (constraints)
- **DROP:** permite eliminar restricciones de claves foráneas o primarias, dejando el tipo de dato y su contenido intacto
- **MODIFY:** cambia el tipo de datos de una columna.
- **CHANGE:** permite cambiar el nombre de una columna.
- Opciones de la tabla que varían con cada SGBD...

Ej. ALTER TABLE usuarios ADD telefono CHAR(12) AFTER email;

[más info](#)

4. Operaciones con tablas

Borrado de tablas

En MySQL se pueden borrar una o varias tablas con el comando **DROP TABLE**

```
DROP TABLE nombre_tabla, nombre_otra_tabla, ...
```

Renombrado de tablas

Para renombrar una tabla, se usa el comando **RENAME TABLE**

```
RENAME TABLE nombre_tabla TO nuevo_nombre
```