

Bases de Datos

4.- Modelo Físico. DDL MySQL / MariaDB

IES La Encantá



Índice



- Modelo Físico
- SQL
 - DDL
- MariaDB
 - Instalación
 - Sentencias
 - Creando tablas

-
- El diagrama de flujo ilustra el ciclo de vida de un sistema de bases de datos, comenzando con el 'UNIVERSO DEL DISCURSO' (representado por una nube y un icono de personas). Este se convierte en la 'Especificación de Requisitos', lo que da lugar al 'MODELO CONCEPTUAL'. Un ejemplo de este modelo es el 'Modelo Entidad-Relación'. El modelo conceptual se transforma en un 'Esquema Conceptual' (ejemplo: 'Diagrama Entidad-Relación'), el cual se convierte en el 'MODELO LÓGICO' (ejemplo: 'Modelo Relacional'). El modelo lógico se transforma en un 'Esquema Lógico Preliminar' (ejemplo: 'Grafo Relacional Inicial'), el cual sufre un proceso de 'VALIDACION' (ejemplo: 'Normalización') para convertirse en un 'Esquema Lógico Definitivo' (ejemplo: 'Grafo Relacional Final'). Este esquema definitivo se convierte en el 'MODELO FISICO' (ejemplo: 'SGBD-Oracle, Tipos de datos, etc.'). El modelo físico se transforma en un 'Esquema de Implementación' (ejemplo: 'Script de creación'), el cual se utiliza para crear la 'BD' (Base de Datos).
- ```

graph TD
 A[UNIVERSO DEL DISCURSO] --> B[Especificación de Requisitos]
 B --> C[MODELO CONCEPTUAL]
 C --> D[Esquema Conceptual]
 D --> E[MODELO LÓGICO]
 E --> F[Esquema Lógico Preliminar]
 F --> G[VALIDACION]
 G --> F
 F --> H[Esquema Lógico Definitivo]
 H --> I[MODELO FISICO]
 I --> J[Esquema de Implementación]
 J --> K[(BD)]

```

# SQL



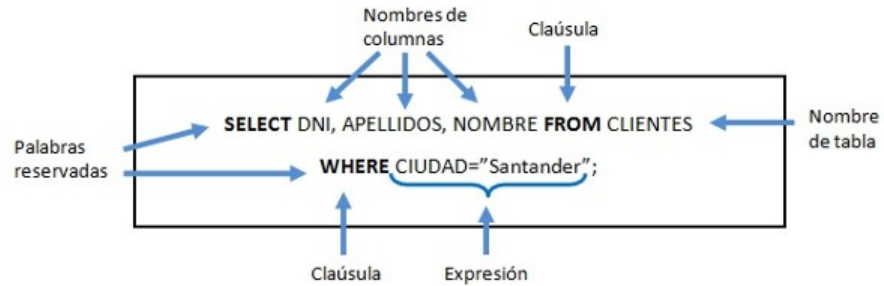
- *Standard Query Language*
- DDL : *Data Definition Language*
  - Permite crear y manipular la estructura de una BD
- DML: *Data Manipulation Language*
  - Permite recuperar, almacenar, modificar, eliminar, insertar y actualizar datos de una BD
- DCL: *Data Control Language*
  - Permite crear roles, permisos e integridad referencial, así como el control de acceso

# Versiones SQL



- SQL-86: funcionalidad mínima para que un lenguaje se considere SQL.
- SQL-89: añade instrucciones para gestionar las claves ajenas (reglas de integridad referencial).
- SQL-92: estándar base.
- SQL:1999: se añaden extensiones hacia la programación orientada objetos + triggers / procedimientos
- SQL:2003. añade características de SQL/ XML.

# Sintaxis



| Símbolo    | Descripción                                         |
|------------|-----------------------------------------------------|
| Mayúsculas | palabra reservada de SQL                            |
| Minúsculas | variable que hay que sustituir por un dato concreto |
| []         | Opcional                                            |
|            | Separa opciones alternativas                        |
| [ ]        | Se elige uno de los valores                         |
| { }        | Obliga elegir uno de los valores                    |
| ...        | Número variable de datos                            |

```
SELECT [ALL | DISTINCT] columna1 [columna2, columna3,] | *
FROM tabla1 [tabla2, tabla3,]
[WHERE condición]
[ORDER BY expr1 [DESC | ASC] [, expr2 [DESC | ASC]]
```

# Instrucciones DDL



- **CREATE:** Permite para crear nuevas tablas, campos e índices.
- **ALTER:** Permite modificar las tablas agregando campos o cambiando la definición de los campos.
- **DROP:** Empleado para eliminar tablas e índices.
- **TRUNCATE:** Empleado para eliminar todos los registros de una tabla.
- **COMMENT:** Utilizado para agregar comentarios al diccionario de datos.
- **RENAME:** Permite renombrar objetos.

# MySQL - MariaDB



- <https://mariadb.com/>
- 100% Open Source
- V10.5 (v10 Marzo14)



- 2009 *Oracle* compra *Sun* (propietaria de MySQL)
- 2012 Se crea *Maria foundation*



# Instalación



```
> sudo apt install mariadb-server
> sudo apt install mysql-server
```

- **Tras instalar:**

```
> sudo mysql_secure_installation
> sudo systemctl status mariadb
```

- **Opción B: Instalar XAMPP**

- Apache + MySQL + PHP + Perl

# Cliente



- Una vez instalado, nos conectamos con el cliente

```
> sudo mariadb
```

```
> sudo mysql -h localhost -u root -p
```

```
ubuntu@ubuntu-VirtualBox:~/Descargas$ sudo mariadb
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 48
Server version: 10.3.25-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> █
```

-h: host

-u: usuario

-p: password (la pide al entrar)

# Dentro de la consola



- Todas las sentencias acaban en ;
  - > SHOW DATABASES;
  - > USE miBaseDeDatos;
  - > SHOW TABLES;
  - > DESCRIBE nombreTabla;

# Interactuando con la consola



- `->` Indica que espera la siguiente línea del comando
- `'>` Indica que espera que se cierre una comilla sencilla
- `">` Indica que espera que se cierre una comilla doble
- `mysql>` Indica que está listo para una nueva consulta

# Cargar archivos



- Poder almacenar las sentencias en un archivo y luego ejecutarlas.

- Desde dentro del cliente

```
mysql> source nombreArchivo.sql
```

- Desde un terminal

```
> mysql -u usuario -p database < nombreArchivo.sql
```

# Scripts



- Es más cómodo almacenar las sentencias SQL en un archivo de texto con extensión `.sql`
- `> source nombreArchivo.sql`
- A tener en cuenta:
  - Si es necesario, borrar y crear una nueva base de datos.
  - Crear los recursos.
  - Insertar datos para comprobar que los recursos se han creado correctamente.

# Creando el usuario



Es mejor crear un usuario y no usar root (fijaros que es un usuario local (localhost) si queremos un usuario que pueda entrar de manera remota debemos poner % : 'usuario'@'%' )

```
> CREATE USER
'usuario'@'localhost' IDENTIFIED
BY 'password';

> GRANT ALL PRIVILEGES ON *.* TO
'usuario'@'localhost' WITH GRANT
OPTION;

> FLUSH PRIVILEGES;
```

# Creando BBDD y borrándolas.



- <https://mariadb.com/kb/en/create-database/>
- El primer paso es crear una BBDD.
- > **CREATE DATABASE** [IF NOT EXISTS] nombreBD;
  - > create database prueba;
  - > create database if not exists prueba;
- Si necesitamos borrarla (ya no la necesitamos):
- > **DROP DATABASE** [IF EXISTS] nombreBD;
  - > drop database prueba;
  - > drop database if exists prueba;



# Creando tablas



<https://mariadb.com/kb/en/create-table/>

- > **CREATE TABLE** [IF NOT EXISTS]  
[basededatos.]nombreDeTabla (  
    columna1 tipoDato1, columna2  
    tipoDato2, ...  
    columnaN tipoDatoN);

> *CREATE TABLE USUARIO (nombre VARCHAR(25));*



# Nombrado de tablas

- No puede haber nombres de tablas repetidas.
- Deben comenzar por un carácter alfabético.
- Su longitud máxima es de 30 caracteres.
- Sólo se permiten letras del alfabeto inglés, dígitos o el signo de guión bajo.
- No puede coincidir con las palabras reservadas de SQL (por ejemplo, no podemos llamar a una tabla WHERE).
- No se distingue entre mayúsculas y minúsculas.
- *Convención: TABLAS en MAYÚSCULAS y columnas en minúsculas*

# Tipos de datos



| Tipo de Campo    | Tamaño de Almacenamiento               |
|------------------|----------------------------------------|
| TINYINT          | 1 byte                                 |
| SMALLINT         | 2 bytes                                |
| MEDIUMINT        | 3 bytes                                |
| INT              | 4 bytes                                |
| INTEGER          | 4 bytes                                |
| BIGINT           | 8 bytes                                |
| FLOAT(X)         | 4 ú 8 bytes                            |
| FLOAT            | 4 bytes                                |
| DOUBLE           | 8 bytes                                |
| DOUBLE PRECISION | 8 bytes                                |
| REAL             | 8 bytes                                |
| DECIMAL(M,D)     | M+2 bytes sí D > 0, M+1 bytes sí D = 0 |
| NUMERIC(M,D)     | M+2 bytes if D > 0, M+1 bytes if D = 0 |

| Tipo de Campo | Tamaño de Almacenamiento |
|---------------|--------------------------|
| DATE          | 3 bytes                  |
| DATETIME      | 8 bytes                  |
| TIMESTAMP     | 4 bytes                  |
| TIME          | 3 bytes                  |
| YEAR          | 1 byte                   |

| Tipo de campo               | Tamaño de Almacenamiento                                |
|-----------------------------|---------------------------------------------------------|
| CHAR(n)                     | n bytes                                                 |
| VARCHAR(n)                  | n + 1 bytes                                             |
| TINYBLOB, TINYTEXT          | Longitud+1 bytes                                        |
| BLOB, TEXT                  | Longitud +2 bytes                                       |
| MEDIUMBLOB, MEDIUMTEXT      | Longitud +3 bytes                                       |
| LONGBLOB, LONGTEXT          | Longitud +4 bytes                                       |
| ENUM('value1','value2',...) | 1 ó dos bytes dependiendo del número de valores         |
| SET('value1','value2',...)  | 1, 2, 3, 4 ó 8 bytes, dependiendo del número de valores |



# Tipos más utilizados

- <https://mariadb.com/kb/en/data-types/>
- CHAR: cadena de texto fijo (<256)
- VARCHAR: cadena de texto variable (<4000)
- TEXT: texto con restricciones (varios GB)
- BIT, TINYINT, INT (-2.147.483.648 a 2.147.483.647), BIGINT, DECIMAL (M, D) , FLOAT y DOUBLE, REAL.
  - id INT UNSIGNED (0 a 4.294.967.295)
- DATE, DATETIME, TIMESTAMP
  - fecha TIMESTAMP default CURRENT\_TIMESTAMP
- para booleanos: BOOLEAN o TINYINT (1)

# Crear tablas con restricciones



- **CREATE** [TEMPORARY] **TABLE** [db.]tabla  
(campo1 tipo [(tamaño)] [NOT NULL | NULL]  
[DEFAULT valor] [UNIQUE [KEY] | PRIMARY KEY]  
[REFERENCES tablaexterna [(campoexterno1,  
campoexterno2)]  
[ON DELETE {CASCADE | SET NULL | NO ACTION}]  
[ON UPDATE {CASCADE | SET NULL | NO ACTION}],  
... ,  
[CONSTRAINT de múltiples campos]) **ENGINE =**  
**InnoDB**;
- InnoDB → motor que soporta claves ajenas. Es el motor por defecto para MariaDB

# Ejemplos de CREATE TABLE I



- `CREATE TABLE PROVEEDOR(nombre VARCHAR(25));`
- `CREATE TABLE PROVEEDOR (  
    nombre VARCHAR(25),  
    localidad VARCHAR(30) DEFAULT 'Elche');`
- `CREATE TABLE PRESTAMO (  
    idPrestamo DECIMAL(8),  
    fechaPrestamo DATE DEFAULT (CURRENT_DATE)  
);`

# Ejemplos de CREATE TABLE II



- `CREATE TABLE CIUDAD (  
nombre CHAR(20) NOT NULL,  
poblacion INT NULL,  
codigoPostal DECIMAL(6) NOT NULL DEFAULT 28941);`
- `CREATE TABLE CIUDAD (  
nombre CHAR(20) NOT NULL,  
poblacion INT DEFAULT 5000);`

# Indicando la PK



- `CREATE TABLE CIUDAD (  
nombre CHAR(20) PRIMARY KEY,  
poblacion INT DEFAULT 5000);`
- `CREATE TABLE CIUDAD (  
nombre CHAR(20),  
poblacion INT DEFAULT 5000,  
PRIMARY KEY(nombre));`
- `CREATE TABLE EMPLEADO (  
codigo VARCHAR(9),  
departamento VARCHAR(15),  
nombre VARCHAR(40),  
PRIMARY KEY (codigo, departamento));`



# PK autoincrementable



- Campo de tipo numérico + `AUTO_INCREMENT`
  - En *Oracle* se utiliza una secuencia (`SEQUENCE`)
- Con cada inserción, se crea un nuevo valor.

```
CREATE TABLE PERSONA (
 id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
 nombre VARCHAR(40),
 fecha DATE);
```

# Indicando la AK → UK



- Atributo **UNIQUE**
  - No permite repetidos
  - Implica la creación de un índice

```
CREATE TABLE EMPLEADO (
dni VARCHAR(9) PRIMARY KEY,
nSegSocial VARCHAR(15) UNIQUE,
nombre VARCHAR(40));
```



# Nombrado de las restricciones

- Tres letras para el nombre de la tabla + \_
- Tres letras con la columna (o columnas) afectadas por la restricción + \_
- Dos letras con la abreviatura del tipo de restricción. La abreviatura puede ser:
  - PK. PRIMARY KEY
  - UK. UNIQUE
  - FK. FOREIGN KEY
  - CK. CHECK (validación)

# Claves ajenas como atributos



- <https://mariadb.com/kb/en/constraint/>
- **CONSTRAINT** nombre **FOREIGN KEY** col  
**REFERENCES** tabla (clave1)

```
CREATE TABLE ALQUILER (
 dni VARCHAR(9),
 codPelicula INT UNSIGNED,
 CONSTRAINT PK_ALQUILER PRIMARY KEY (dni,
 codPelicula),
 CONSTRAINT FK_ALQ_CLI FOREIGN KEY (dni)
REFERENCES CLIENTE(dni),
 CONSTRAINT FK_ALQ_PEL FOREIGN KEY (codPelicula)
REFERENCES PELICULA(cod)
);
```

# Clave ajena compuesta



```
CREATE TABLE EXISTENCIA (
 tipo CHAR(9),
 modelo DECIMAL(3),
 numAlmacen DECIMAL(1),
 cantidad DECIMAL(7),
 CONSTRAINT FK_EXI_PIE FOREIGN KEY (tipo,
modelo) REFERENCES PIEZA(tipo,modelo),
 CONSTRAINT FK_EXI_ALM FOREIGN KEY (numAlmacen)
REFERENCES ALMACEN(codigo),
 CONSTRAINT PK_EXISTENCIAS PRIMARY KEY(tipo,
modelo, numAlmacen)
);
```

# Restricciones de propagación



- ON DELETE, ON UPDATE
  - NO ACTION: se impide la operación
    - Comportamiento por defecto
  - CASCADE: la operación se propaga
  - SET NULL: la clave ajena se pone a NULL
- ```
CREATE TABLE TELEFONO (  
  numero CHAR(12) PRIMARY KEY,  
  id VARCHAR(9),  
  CONSTRAINT FK_TEL_PER FOREIGN KEY (id)  
  REFERENCES CLIENTE (dni)  
  ON DELETE CASCADE ON UPDATE SET NULL);
```

Restricciones de validación



- Atributo CHECK

```
CREATE TABLE INGRESO (  
    cod DECIMAL(5) PRIMARY KEY,  
    concepto VARCHAR(40) NOT NULL,  
    importe DECIMAL(11,2),  
    CONSTRAINT CK_INGRESO CHECK (importe>0 AND  
importe<8000));
```

```
CREATE TABLE INGRESO2 (  
    cod DECIMAL(5) PRIMARY KEY,  
    concepto VARCHAR(40) NOT NULL,  
    importeMax DECIMAL(11,2),  
    importe DECIMAL(11,2),  
    CONSTRAINT CK_INGRESO CHECK (importe<importeMax));31
```



Enumeraciones

- <https://mariadb.com/kb/en/enum/>
- Permiten restringir el conjunto de valores de un determinado campo.

```
curso    ENUM ('0', '1', '2'),
```

```
horario  ENUM ('mañana', 'tarde', 'noche')
```

Ejemplo:

```
CREATE TABLE fruits (
```

```
    id INT NOT NULL auto_increment PRIMARY KEY,
```

```
    fruit ENUM('apple','orange','pear'),
```

```
    weight INT
```

```
);
```

```
> DESCRIBE fruits;
```


Otras operaciones con tablas



- Borrar una tabla
 - **DROP** TABLE nombreTabla;
- Vaciar una tabla:
 - **TRUNCATE** [TABLE] nombreTabla;
- Renombrar:
 - **ALTER TABLE** nombreTablaViejo **RENAME TO** nombreNuevo;
 - **RENAME TABLE [IF EXISTS]** nombreTablaViejo **TO** nombreNuevo;



Operaciones con columnas

- <https://mariadb.com/kb/en/alter-table/>
- Para añadir columnas
 - **ALTER TABLE** nombreTabla **ADD** (nombreColumna TipoDatos [Propiedades] [,columnaSiguiente tipoDatos [propiedades]...])
- Borrar columnas
 - **ALTER TABLE** nombreTabla **DROP** (columna [,columnaSiguiente, ...]) ;
- Modificar columnas:
 - **ALTER TABLE** nombreTabla **MODIFY** (columna tipo [propiedades] [,columnaSiguiente tipo [propiedades] ...])



Gestionando restricciones

- **Añadiendo:**

```
ALTER TABLE tabla ADD [CONSTRAINT nombre]  
tipoDeRestricción (columnas);
```

- **Borrando:**

```
ALTER TABLE tabla DROP {PRIMARY KEY |  
CONSTRAINT nombreRestricción} [CASCADE]
```

- **Habilitando/ deshabilitando claves**

```
ALTER TABLE tabla ENABLE KEYS;
```

```
ALTER TABLE tabla DISABLE KEYS;
```

Ejemplos ALTER TABLE I



- ```
CREATE TABLE CLIENTE (
 dni VARCHAR(9) PRIMARY KEY,
 cnombre VARCHAR(50),
 direccion VARCHAR(60)
);
```
- ```
CREATE TABLE MASCOTA (  
    codigo INTEGER PRIMARY KEY,  
    nombre VARCHAR(50),  
    raza VARCHAR(50)  
);
```

Ejemplos ALTER TABLE II

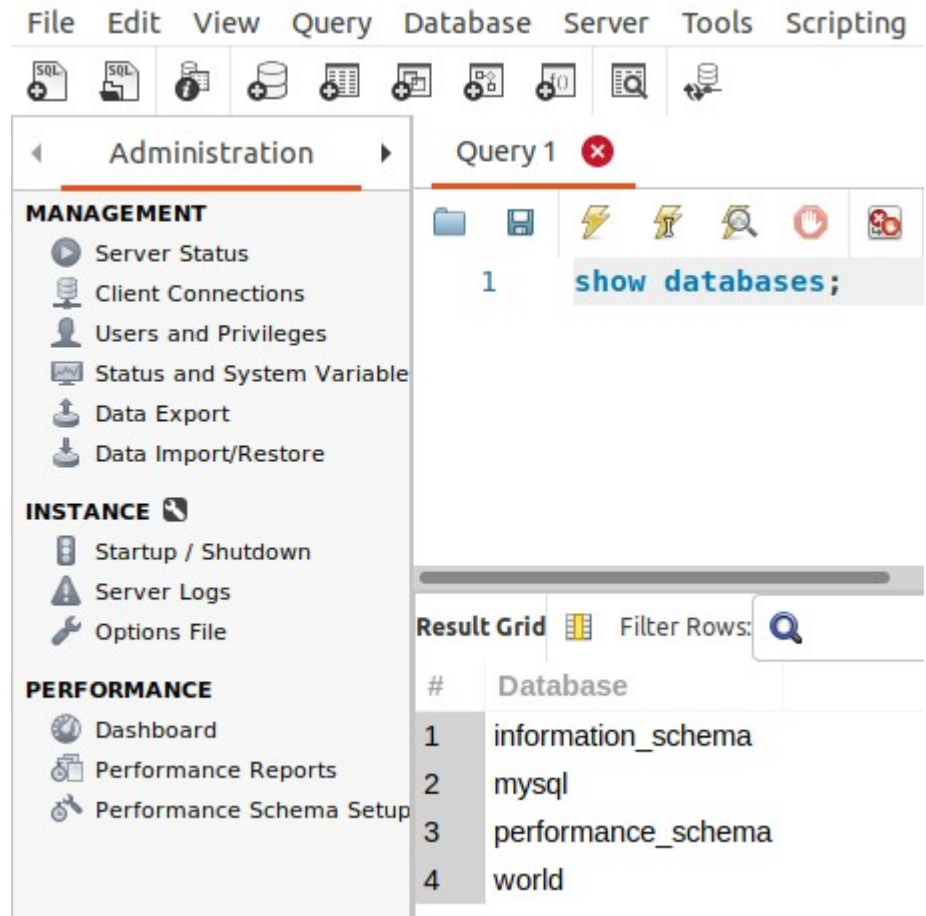


- `ALTER TABLE MASCOTA ADD especie
VARCHAR(10) AFTER raza;`
- `ALTER TABLE MASCOTA ADD cliente
VARCHAR(9) AFTER nombre;`
- `ALTER TABLE MASCOTA ADD CONSTRAINT
fk_duenyo FOREIGN KEY (cliente)
REFERENCES CLIENTE(dni);`
- `ALTER TABLE MASCOTA MODIFY codigo
INT(3) AUTO_INCREMENT;`
- `ALTER TABLE cliente CHANGE cnombre
nomMascosta VARCHAR(50);`

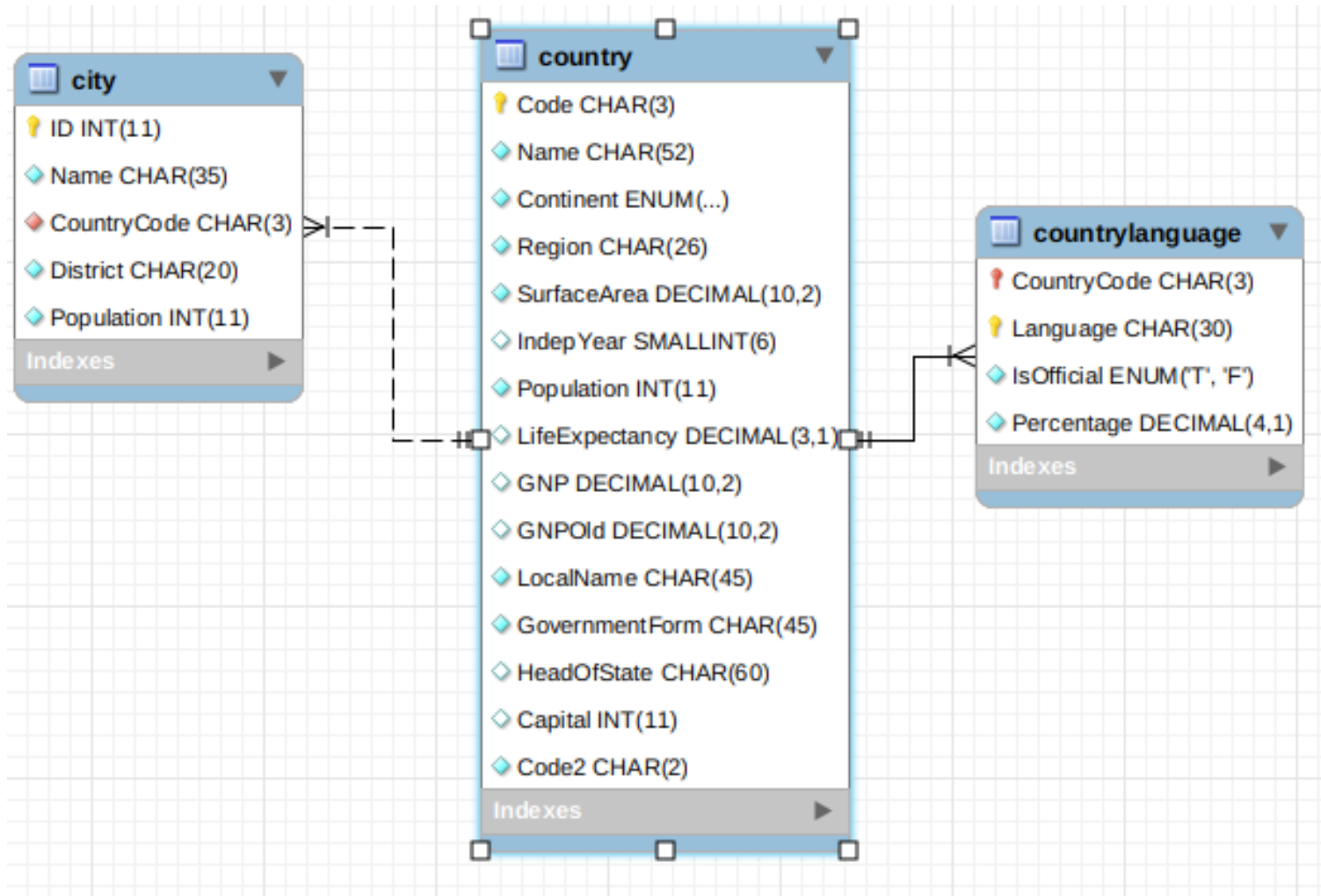
MySQL Workbench



- Cliente visual
- Permite modelar
 - Genera DDL a partir del modelo



Modelo Relacional Workbench



Instalación MySQL Workbench



- Descargar paquete .deb

```
> sudo apt install ./mysqlworkbech.deb
```

- 2ª opción, con Ubuntu store y luego

```
> snap connect mysql-workbench-community:password-manager-service
```

```
> snap connect mysql-workbench-community:ssh-keys
```


PhpMyAdmin



- Herramientas de administración web
 - > sudo apt install phpmyadmin
 - O instalar XAMPP

The screenshot shows the phpMyAdmin interface for a server at 127.0.0.1, connected to the 'world' database. The left sidebar shows a tree view of databases and tables. The main area displays a table list for the 'world' database. The table 'country' is selected. Below the table list, there is a 'Crear tabla' (Create table) section with a form to create a new table.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> city	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	4,079	InnoDB	utf8mb4_general_ci	480.0 KB	-
<input checked="" type="checkbox"/> country	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	239	InnoDB	utf8mb4_general_ci	16.0 KB	-
<input type="checkbox"/> countrylanguage	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	984	InnoDB	utf8mb4_general_ci	160.0 KB	-
3 tablas	Número de filas	5,302	InnoDB	utf8mb4_general_ci	656.0 KB	0 B

Nombre: Número de columnas:



¿Alguna pregunta?