

Bases de Datos

Unidad 3: Diseño Físico de Bases de Datos

UNIDAD 3: DISEÑO FÍSICO DE BASES DE DATOS

1. Características físicas del almacenamiento de la base de datos
2. Herramientas gráficas para la implementación de la base de datos
3. El lenguaje de definición de datos
4. Creación modificación y eliminación de bases de datos.
5. Tipos de datos. Valores. Operadores
6. Administración de tablas
 - 6.1 Sintaxis de la instrucción CREATE TABLE
 - 6.2 Propiedades de tablas
 - 6.3 Modificación de tablas
 - 6.4 Eliminación de tablas
7. El lenguaje de control de datos
8. Vistas
9. Usuarios y privilegios

1. Características del diseño físico

El diseño físico se realiza a partir del diseño lógico (grafo relacional). Consta de todas las instrucciones SQL necesarias para implementar la base de datos en el DBMS.

- ☐ Hay que crear las tablas eligiendo un nombre adecuado.
- ☐ Para cada tabla se definen las columnas, sus nombres y los tipos de datos que contienen.
- ☐ Se establecen las restricciones necesarias sobre las columnas de las tablas (PRIMARY KEY, NOT NULL, UNIQUE, FOREIGN KEY, etc.)
- ☐ Se crean vistas. (Tablas virtuales que permiten simplificar búsquedas complejas)
- ☐ Se crean procedimientos (conjuntos de queries), funciones y triggers (objetos que se activan cuando sucede una acción).
- ☐ Se establecen propiedades sobre las tablas (motor de almacenamiento, carpeta de almacenamiento, valor autoincrement, particiones, etc.)

2 Herramientas gráficas para la implementación de la base de datos

Herramientas gráficas gratuitas:

☐ **MySQL Workbench**

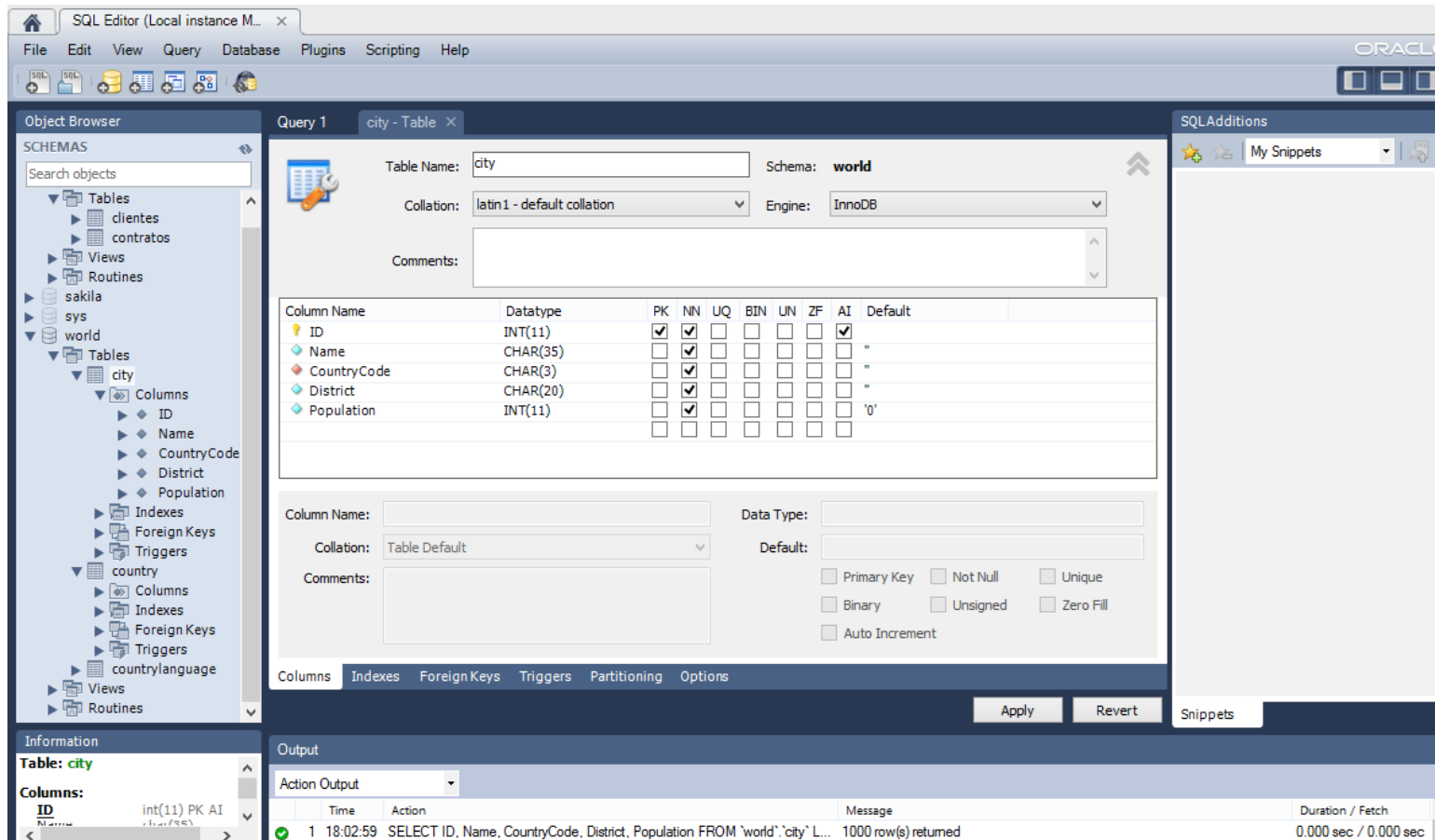
☐ **phpMyAdmin** (Requiere servicio Apache con motor PHP)

☐ **HeidiSQL**

☐ **MyDB Studio**

2 Herramientas gráficas para la implementación de la base de datos

MySQL Workbench



2 Herramientas gráficas para la implementación de la base de datos

phpMyAdmin

The screenshot displays the phpMyAdmin web interface. On the left sidebar, the 'Current Server' is 'phpMyAdmin demo - MySQL', and the 'Recent tables' list includes 'City'. The main panel shows the 'Structure' tab for the 'City' table. The table structure is as follows:

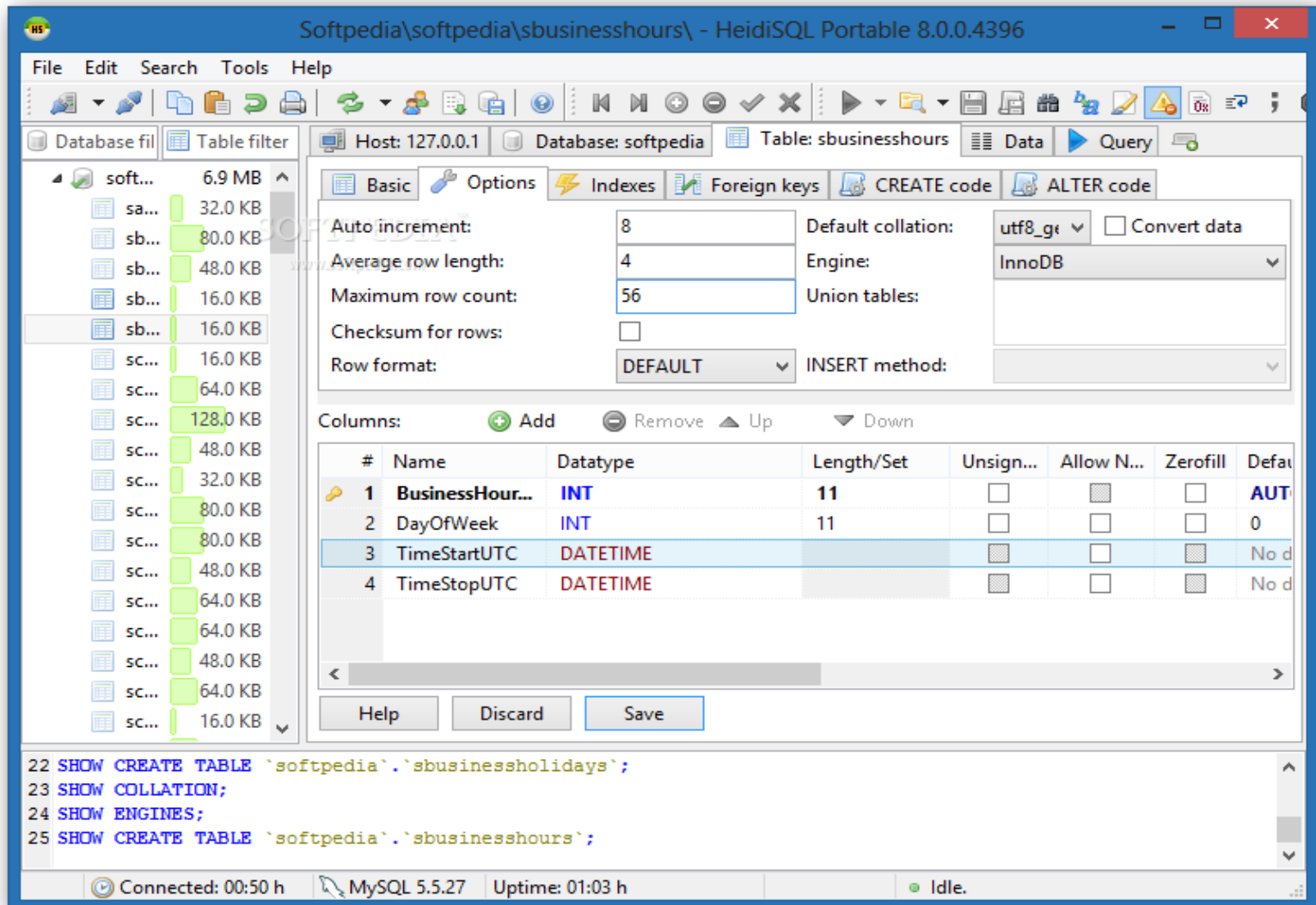
#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	ID	int(11)			No	None	AUTO_INCREMENT	Change Drop More
2	Name	char(35)	latin1_swedish_ci		No			Change Drop More
3	CountryCode	char(3)	latin1_swedish_ci		No			Change Drop More
4	District	char(20)	latin1_swedish_ci		No			Change Drop More
5	Population	int(11)			No	0		Change Drop More

Below the table structure, there are options to 'Check All', 'With selected', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', and 'Index'. The 'Indexes' section shows the 'Information' tab with the following data:

Space usage		Row Statistics	
Data	266.9 KiB	Format	static
Index	42 KiB	Collation	latin1_swedish_ci
Total	308.9 KiB	Rows	4,079
		Row length	67
		Row size	78 B
		Next autoindex	4,080
		Creation	Apr 03, 2013 at 01:30 PM
		Last update	Apr 03, 2013 at 01:30 PM

2 Herramientas gráficas para la implementación de la base de datos

HeidiSQL



Desde este momento comenzamos a usar el lenguaje SQL (*Structured Query Language*).

- ☐ Se trata de un lenguaje estandarizado para interactuar mediante consultas sobre sistemas de bases de datos relacionales. Se entiende por consulta cualquier petición que se hace al SGBD.
- ☐ Los SGBD relacionales incluyen siempre alguna herramienta para ejecutar instrucciones SQL.
- ☐ Desde el primer estándar ANSI-SQL de 1986, se han ido desarrollando varios estándares o versiones del SQL. El último es SQL-2016.
- ☐ Un estándar establece reglas de sintaxis y funcionamiento del repertorio de instrucciones SQL.

SQL consta de un repertorio de instrucciones. En general, los SGBD incluyen prácticamente todo el repertorio de instrucciones del estándar SQL y con la misma sintaxis. Pero puede ocurrir que:

- ☐ **No incluyan alguna instrucción.**
- ☐ **Incluyan alguna instrucción propia no perteneciente al estándar.**
- ☐ **En algunas instrucciones la sintaxis pueda variar ligeramente por no incluir alguna funcionalidad o por incluir alguna funcionalidad propia.**

Interpretación de la sintaxis de una instrucción SQL:

Cuando nos dan la sintaxis completa de una instrucción SQL, por ejemplo, en la documentación oficial de MySQL, tenemos algo como esto:

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name  
[create_specification]
```

Esto tenemos que saber interpretarlo para construir correctamente las instrucciones.

<https://dev.mysql.com/doc/refman/8.0/en/>

Interpretación de la sintaxis de una instrucción SQL:

- ☐ Las palabras en mayúsculas son palabras reservadas SQL.
- ☐ Las palabras en minúsculas y cursiva son parámetros sustituibles y cuyo valor decide el usuario. Así, por ejemplo `db_name` indica que tenemos que escribir el nombre de la base de datos.
- ☐ Algo entre corchetes indica que es opcional, es decir, que podemos escribir la instrucción sin esa parte. Si se escribe lo opcional, no hay que escribir los corchetes.
- ☐ Algo entre llaves indica que tenemos que elegir entre uno de los elementos (separados por `|`) que hay dentro de las llaves.
- ☐ Puntos suspensivos indica que podemos introducir una lista de valores de lo anterior que hay.

Subconjuntos del lenguaje SQL

En función del tipo de operaciones realizadas, el conjunto de instrucciones SQL se puede considerar dividido en tres subconjuntos de instrucciones.

DDL (Data Definition Language)

Son todas las instrucciones que permiten establecer la estructura de los datos en las bases de datos. En definitiva, son las instrucciones para realizar el diseño físico de las bases de datos.

DML (Data Manipulation Language)

Son las instrucciones que sirven para manipular los datos que se almacenan en las bases de datos (consultar, insertar, modificar, eliminar)

DCL (Data Control Language)

Son las instrucciones de control de acceso a los datos (gestionar usuarios y privilegios, realizar transacciones, bloquear, etc.)

Principales instrucciones del lenguaje DDL:

- ☐ CREATE DATABASE
- ☐ CREATE TABLE
- ☐ CREATE INDEX
- ☐ CREATE VIEW
- ☐ CREATE PROCEDURE
- ☐ CREATE FUNCTION
- ☐ CREATE TRIGGER
- ☐ ALTER (DATABASE, TABLE, VIEW, ...)
- ☐ DROP (DATABASE, TABLE, VIEW,)

Crear una base de datos. Sintaxis.

CREATE {**DATABASE** | **SCHEMA**} [**IF NOT EXISTS**] *db_name* [*create_specification*]

Create_specification: [**DEFAULT**] **CHARACTER SET** [=] *charset_name* | [**DEFAULT**] **COLLATE** [=] *collation_name*

- ☐ En MySql es lo mismo usar **DATABASE** o **SCHEMA**.
- ☐ La cláusula **IF NOT EXISTS** hace que no se intente crear la base de datos si es que existe. Así no se produce un error de ejecución.
- ☐ **CHARACTER SET** permite especificar el conjunto de caracteres o, lo que es lo mismo, como se codifican internamente los caracteres (utf8, latin1, etc.)
- ☐ **COLLATION** establece los criterios para ordenar y comparar datos alfabéticamente (por ejemplo, **spanish_ci**).

Ejemplo. Crear una base de datos EmpTransportes.

```
CREATE DATABASE EmpTransportes;
```

Si queremos que la instrucción no de error en caso de existir **EmpTransportes**:

```
CREATE DATABASE IF NOT EXISTS EmpTransportes;
```

Si queremos que la base de datos se cree para usar el conjunto de caracteres **latin1** (en lugar de **utf8** usado por defecto) y con ordenación alfabética para el español (por defecto, se usa **general_ci**):

```
CREATE DATABASE IF NOT EXISTS EmpTransportes CHARSET latin1 COLLATE latin1_spanish_ci;
```

Mostrar las bases de datos montadas en el servidor.

SHOW DATABASES;

	Database
▶	information_schema
	clinica
	fp
	fp2
	jardineria
	musica

Modificar una base de datos

```
ALTER {DATABASE | SCHEMA} [db_name] alter_specification;
```

alter_specification: [DEFAULT] CHARACTER SET [=] *charset_name* | [DEFAULT] COLLATE [=] *collation_name*

Eliminar una base de datos

```
DROP {DATABASE | SCHEMA} [IF EXISTS] db_name;
```

Usar una base de datos

Para que podamos ejecutar instrucciones sobre una base de datos existente, es necesario tenerla en uso o abrirla:

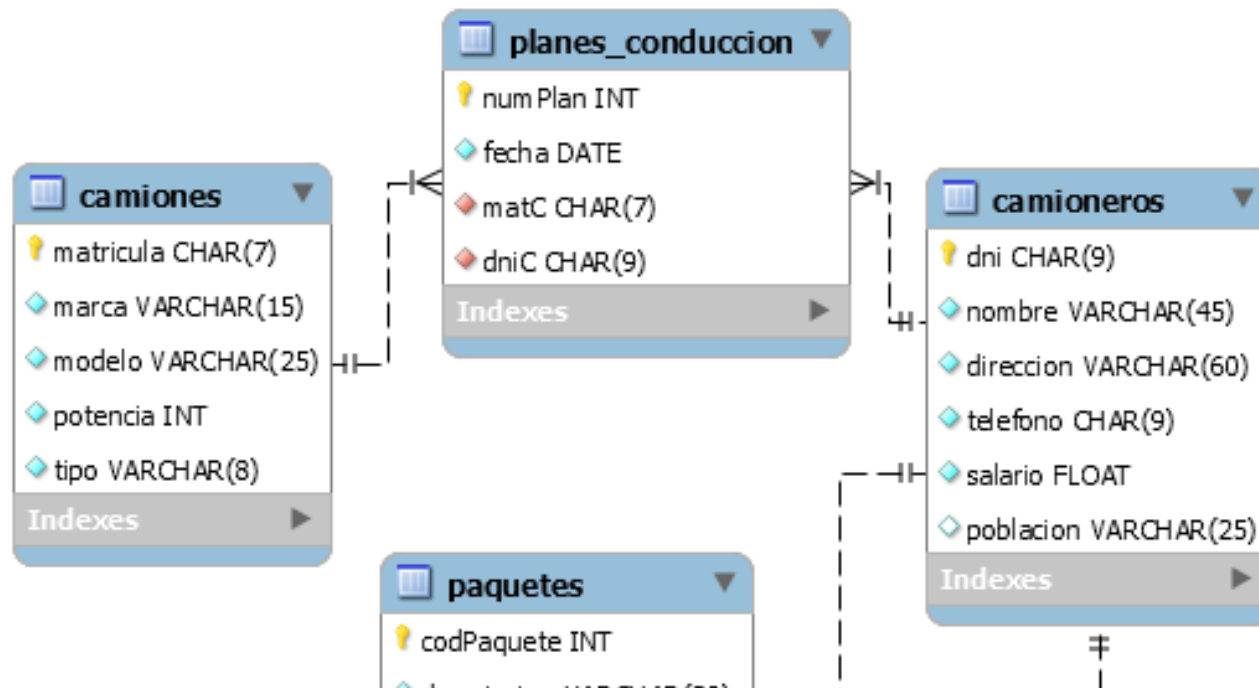
```
USE db_name;
```

MySQL Workbench: Ejercicio de aprendizaje I

- 1.- Crea una base de datos EmpTransportes**
- 2.- ¿Cuál es la instrucción SQL para crear la base de datos?**
- 3.- Selecciona la nueva base de datos e identifica los botones de la barra de herramientas para añadir tablas, vistas y rutinas**

MySQL Workbench: Ejercicio de aprendizaje I

4.- Crea gráficamente en la base de datos EmpTransportes, la tabla camiones y copia la instrucción SQL de creación de la tabla.



MySQL Workbench: Ejercicio de aprendizaje

5.- Abre una ventana para editar y ejecutar instrucciones SQL. Edita la instrucción para crear la tabla camioneros.

