



# Bases de Datos

---

# Bases de Datos

Definiciones

# Definiciones

- Una **Base de Datos** es una colección de datos que se encuentran almacenados juntos, organizados y relacionados entre sí.
- Un **Sistema Gestor de Bases de Datos** es el software que nos ayuda a comunicarnos con una Base de Datos

# Definiciones

- ◉ Existe muchos tipos de SGBD, el más utilizado es el SGBD relacionales.
- ◉ Las Bases de Datos relacionales están compuestas por:
  - ◉ Tablas
    - ◉ Filas
    - ◉ Columnas
  - ◉ Relaciones entre las tablas

# Bases de Datos Relacionales

Elementos

# Tablas

- Una Base de Datos **relacional** tiene como elemento principal las tablas.
- Las tablas estarán compuestas por una serie de **columnas** (atributos) y una serie de **filas** (datos).
- Cada tabla representa a un conjunto de datos de la realidad.

# Tablas

- Cada **columna** de una tabla debe tener:
  - Un nombre
  - Un tipo de dato
  - Una longitud
- Se debe de poder **diferenciar** de forma univoca a cada fila de la tabla.
- Cada tabla debe tener una **clave principal**
  - Una columna de la tabla que permite diferenciar a todos los elementos del conjunto sin lugar a error.

# Tablas

- Para que un atributo pueda ser clave principal debe cumplir que:
  - **No** se pueda **repetir** el mismo valor en elementos distintos.
  - **No** pueda **cambiar** a lo largo del tiempo
  - **No** puede quedar **vacío** nunca
- La clave principal de una tabla no tiene por qué ser un único atributo, puede estar compuesta por **varios atributos juntos**.



# Ejemplos

## PERSONA

| Nombre | DNI | Teléfono |
|--------|-----|----------|
|        |     |          |

## LIBRO

| ISBN | Título | Autor | Año publicación |
|------|--------|-------|-----------------|
|      |        |       |                 |

## FECHA

| Día semana | Día | Mes | Año |
|------------|-----|-----|-----|
|            |     |     |     |

## MASCOTA

| Nombre | Edad | Peso | Raza | Color |
|--------|------|------|------|-------|
|        |      |      |      |       |

## VEHICULO

| Matrícula | Color | Marca | Modelo | Precio |
|-----------|-------|-------|--------|--------|
|           |       |       |        |        |

# Relaciones

- Una Base de Datos es una colección de datos que se encuentran almacenados juntos, organizados y relacionados entre sí.
- Algunas de las tablas que componen una Base de Datos pueden tener **relación** entre sí

PERSONA

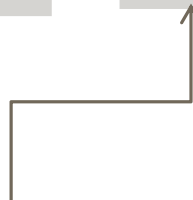
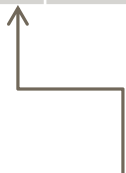
| Nombre | DNI | Teléfono |
|--------|-----|----------|
|        |     |          |

LIBRO

| ISBN | Título | Autor | Año publicación |
|------|--------|-------|-----------------|
|      |        |       |                 |

VENTAS

| Persona | Libro | Unidades | Fecha |
|---------|-------|----------|-------|
|         |       |          |       |



# Relaciones

- Estas relaciones se traducen en:
  - Para que un elemento pueda existir en una tabla que hace referencia a otra, antes deberá existir en la tabla referenciada.
- **En nuestro ejemplo**
  - Para que una persona pueda comprar un libro, antes debemos tener registrada a la persona y al libro

# Creando Bases de Datos

El lenguaje SQL

# El lenguaje SQL

- ◉ **SQL** (Structured Query Language) es un lenguaje de consulta de Bases de Datos que nos permite realizar cualquier operación que queramos.
- ◉ Vamos a ver como:
  - ◉ Crear tablas
  - ◉ Insertar datos en las tablas
  - ◉ Modificar datos de las tablas
  - ◉ Eliminar datos de las tablas
  - ◉ Consultar información de las tablas

# EL lenguaje SQL

- Las operaciones de SQL reciben el nombre de **sentencias**.
- Están formadas por diferentes partes que denominamos **cláusulas**.

Sentencia

```
[ SELECT codigo_producto, nombre_producto, tipo — Cláusula  
  FROM productos — Cláusula  
  WHERE precio > 1000; — Cláusula
```

# El lenguaje SQL y MySQL

Creando una Base de Datos

# MySQL

- ◉ **MySQL** es un SGBD relacional, libre y de código abierto.
- ◉ **XAMPP** viene con él instalado.
- ◉ Para acceder a panel de control de MySQL, escribiremos en el navegador:

`localhost/phpmyadmin`



# MySQL

► Accederemos a una página parecida esta:

The screenshot shows the phpMyAdmin web interface in a browser window. The address bar displays the URL: `localhost/phpmyadmin/#PMAURL=0:index.php?db=&table=&server=1&target=&lang=es&collation_connection=utf8_general_ci&token=62671e2ac1dfbcee346be730ac252125`. The interface includes a top navigation bar with tabs for 'Bases de datos', 'SQL', 'Estado actual', 'Usuarios', 'Exportar', 'Importar', 'Configuración', 'Replicación', 'Variables', 'Juegos de caracteres', and 'Motores'. On the left, a sidebar shows a tree view of databases: 'cdcol', 'data', 'information\_schema', 'mysql', 'performance\_schema', 'phpmyadmin', 'test', and 'webauth'. The main content area is divided into two sections: 'Configuraciones generales' and 'Configuraciones de apariencia'. The 'Configuraciones generales' section shows the 'Cotejamiento de la conexión al servidor' set to 'utf8\_general\_ci'. The 'Configuraciones de apariencia' section shows the 'Idioma - Language' set to 'Español - Spanish' and the 'Tema' set to 'pmahomme'. A right sidebar contains information about the 'Servidor de base de datos' (MySQL 5.6.11), 'Servidor web' (Apache/2.4.4), and 'phpMyAdmin' (4.0.4.1). A footer message states: 'Una versión más reciente de phpMyAdmin está disponible y le recomendamos que la obtenga. La versión más reciente es 4.1.6, y existe desde el 2014-01-26.'

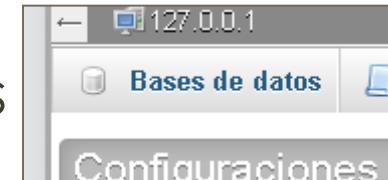
Una versión más reciente de phpMyAdmin está disponible y le recomendamos que la obtenga. La versión más reciente es 4.1.6, y existe desde el 2014-01-26.

! Su archivo de configuración contiene parámetros (root sin contraseña) que corresponden a la cuenta privilegiada predeterminada de MySQL. Su servidor de MySQL está usando estos valores, lo que constituye una vulnerabilidad. Se le recomienda corregir esta brecha de seguridad. Por ejemplo, desde la página de inicio de phpMyAdmin seleccione Privilegios y agregue la contraseña a root@localhost. Deberá escribir la misma contraseña en config.inc.php de phpMyAdmin.

# MySQL

- Lo primero que debemos hacer es **crear una base de datos** donde estarán nuestras tablas.
- Para ello:

- Pulsamos en el botón Bases de Datos



- Elegimos un nombre para la Base de Datos y le damos a Crear

A screenshot of the 'Crear base de datos' (Create database) form in the MySQL interface. It features a title 'Crear base de datos' with a question mark icon. Below the title is a text input field for the database name, a dropdown menu currently showing 'Cotejamiento', and a 'Crear' button.

- En ese momento aparecerá la nueva Base de Datos en el árbol de la izquierda

# El lenguaje SQL y MySQL

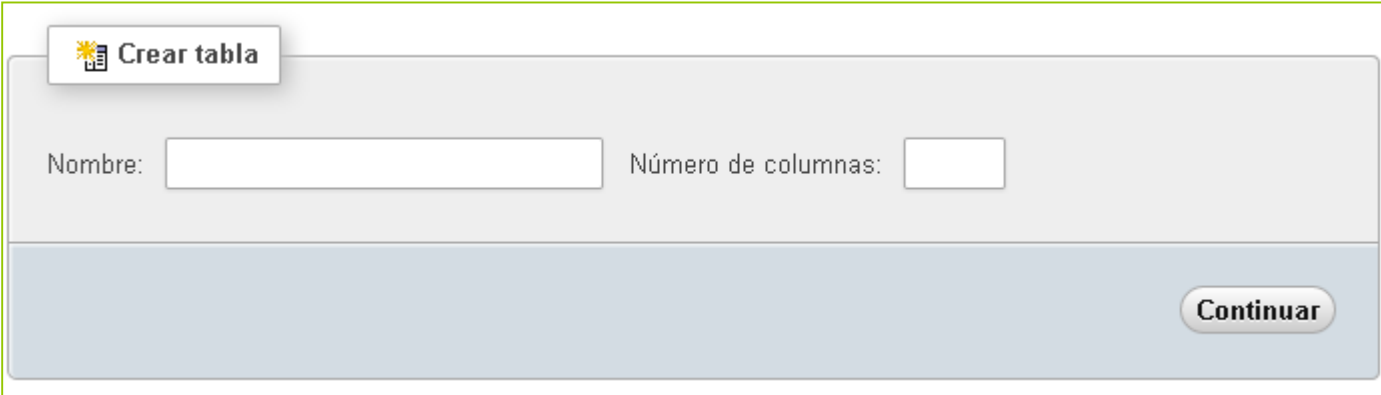
Creando tablas

# Creación de Tablas. Pasos a seguir

1. Decidir el **nombre** de la tabla
2. Nombre de cada uno de las **columnas**
3. A cada una de las columnas asignar un **tipo de datos**
  - ✓ También podremos dar definiciones por defecto y restricciones de columna.
4. Sólo nos quedará dar las **relaciones** de la tabla.

# Creación de tablas

- Crear tablas en desde phpMyAdmin es también muy sencillo.
- Una vez seleccionada la Base de Datos con la que vamos a trabajar nos aparece un formulario donde podemos elegir el nombre de la tabla y el número de filas que va a tener.



The screenshot shows the 'Crear tabla' (Create table) form in phpMyAdmin. The form has a title bar with a sun icon and the text 'Crear tabla'. Below the title bar, there are two input fields: 'Nombre:' followed by a text box, and 'Número de columnas:' followed by a text box. At the bottom right of the form, there is a button labeled 'Continuar'.

# Columnas. Tipos de datos

- Una vez creada la tabla, nos aparece un formulario con una fila por cada una de las columnas de la tabla

| Nombre               | Tipo ? | Longitud/Valores ?   | Predeterminado ? |
|----------------------|--------|----------------------|------------------|
| <input type="text"/> | INT ▼  | <input type="text"/> | Ninguno ▼        |

- Los **tipos de datos** más utilizados son:
  - varchar o text → cadenas de texto
  - int → números sin decimales
  - Decimal → números con decimales
  - date → fechas
  - Set → Conjuntos de valores
  - Serial → Valores automáticos

# Columnas. Tipos de datos

- En el campo Longitud/Valores se debe introducir una de estas 3 cosas
  - **Cadenas:** longitud máxima de la cadena
  - **Enteros:** longitud máxima del entero
  - **Decimales:** dos números separados por comas. El primero indicará el número total de dígitos permitidos. El segundo indicará de todos ellos cuántos se reservan para decimales
  - **Set:** Los valores que podrá contener la columna

# Restricciones de columna

- Una vez elegidos las columnas y asignados los tipos de datos, hay que decidir si las columnas tienen alguna **restricción**.
- Las restricciones pueden ser del tipo:
  - Si es o no **clave** principal
  - Debe ser **único** en toda la tabla o puede repetirse.
  - Puede quedar **vacío** o debe estar siempre relleno.



# Restricciones de columna

| Restricción | descripción  |
|-------------|--|
| NULO        | En la columna no puede haber valores nulos                               |
| UNIQUE      | En toda la columna no se pueden repetir valores                          |
| PRIMARY     | La columna es clave primaria: no puede tener valores nulos ni repetidos. |

# Creación de tablas: Ejercicios

- La Base de Datos “Productora” contiene la siguientes tablas:

**Estrella** (nombre, dirección, sexo, f\_nacimiento)

**Estudio** (nombre, dirección)

**Película** (título, año, duración, nombre\_estudio)

**Protagonistas** (título película, año pel, nomb estr)

# PRODUCTORA: Estrella y estudio

## ESTRELLA

Estructura

| Nombre       | Tipo    | Longitud/Valores | Predeterminado | Cotejamiento | Atributos | Nulo                     | Índice  |
|--------------|---------|------------------|----------------|--------------|-----------|--------------------------|---------|
| nombre       | VARCHAR | 20               | Ninguno        |              |           | <input type="checkbox"/> | PRIMARY |
| direccion    | VARCHAR | 50               | Ninguno        |              |           | <input type="checkbox"/> | ---     |
| sexo         | SERIAL  | h, m             | Ninguno        |              |           | <input type="checkbox"/> | ---     |
| f_nacimiento | DATE    |                  | Ninguno        |              |           | <input type="checkbox"/> | ---     |

## ESTUDIO

| Nombre    | Tipo    | Longitud/Valores | Predeterminado | Cotejamiento | Atributos | Nulo                     | Índice  |
|-----------|---------|------------------|----------------|--------------|-----------|--------------------------|---------|
| nombre    | VARCHAR | 20               | Ninguno        |              |           | <input type="checkbox"/> | PRIMARY |
| direccion | VARCHAR | 50               | Ninguno        |              |           | <input type="checkbox"/> | ---     |

# PRODUCTORA: películas y protagonistas

## PELÍCULAS










| Nombre         | Tipo    | Longitud/Valores | Predeterminado | Cotejamiento | Atributos | Nulo                     | Índice  |
|----------------|---------|------------------|----------------|--------------|-----------|--------------------------|---------|
| titulo         | VARCHAR | 50               | Ninguno        |              |           | <input type="checkbox"/> | PRIMARY |
| anio           | INT     | 4                | Ninguno        |              |           | <input type="checkbox"/> | PRIMARY |
| duracion       | INT     | 3                | Ninguno        |              |           | <input type="checkbox"/> | ---     |
| nombre_estudio | VARCHAR | 20               | Ninguno        |              |           | <input type="checkbox"/> | ---     |

## PROTAGONISTAS

| Nombre          | Tipo    | Longitud/Valores | Predeterminado | Cotejamiento | Atributos | Nulo                     | Índice  |
|-----------------|---------|------------------|----------------|--------------|-----------|--------------------------|---------|
| titulo_pelicula | VARCHAR | 50               | Ninguno        |              |           | <input type="checkbox"/> | PRIMARY |
| anio_peli       | INT     | 4                | Ninguno        |              |           | <input type="checkbox"/> | PRIMARY |
| nomb_estr       | VARCHAR | 20               | Ninguno        |              |           | <input type="checkbox"/> | PRIMARY |





# Relaciones de las tablas

- Una vez creadas todas las tablas necesarias en nuestra Base de Datos, habrá que relacionar unas con otras.
- Para ello seleccionamos la tabla que va a hacer referencia a otras y nos vamos a **SU** sección Estructura


|  |  |
|--|--|
| <input type="checkbox"/> estudio             |  Examinar  Estructura  BU       |
| <input checked="" type="checkbox"/> pelicula |  Examinar  Estructura  BU |
| <input type="checkbox"/> protagonistas       |  Examinar  Estructura  BU |





# Relaciones de las tablas




- Dentro de la tabla, seleccionamos Vista de relaciones



 **Examinar**  **Estructura**  **SQL** 

| #                          | Nombre                | Tipo        | Cotejamiento   |
|----------------------------|-----------------------|-------------|----------------|
| <input type="checkbox"/> 1 | <b><u>titulo</u></b>  | varchar(50) | latin1_swedish |
| <input type="checkbox"/> 2 | <b><u>anio</u></b>    | int(4)      |                |
| <input type="checkbox"/> 3 | <b>duracion</b>       | int(3)      |                |
| <input type="checkbox"/> 4 | <b>nombre_estudio</b> | varchar(20) | latin1_swedish |

 ☐ Marcar todos *Para los elementos que e*

 Eliminar  Primaria  Único  Índice

 Vista de impresión  Vista de relaciones  Plan

 Hacer seguimiento a la tabla  Mover columnas



Vista de relaciones

# Relaciones de las tablas

- Una vez aquí sólo hay que elegir dentro de la lista desplegable, a qué campo va a hacer referencia cada una de las columnas

The screenshot shows a software interface for defining table relationships. A dropdown menu is open, displaying a list of available columns from a table named 'productora'. The columns listed are 'estrella.nombre', 'estrella.sexo', and 'estudio.nombre'. The option 'productora.estudio.nombre' is currently selected and highlighted in blue. The background shows a table with columns like 'nombre\_estudio' and a label 'Elegir la columna a'.

| nombre_estudio |
|----------------|
|                |
|                |
|                |
|                |

Elegir la columna a

- productora.estrella.nombre
- productora.estrella.sexo
- productora.estudio.nombre

# Ejercicios



# Creación de tablas: Ejercicios

Persona (nombre, calle, ciudad)

Trabaja (nombre persona, nombre\_empresa, salario)

Empresa (nombre, ciudad)

Director\_empresa (nombre\_empresa, nombre\_director)

NOTA: El salario por defecto será de 1000€

# Creación de tablas: Ejercicios

**Pieza** (codigo pieza, fabricante, tipo)

**PC** (#modelo, pieza, velocidad, ram, hd, cd, precio)

Sabiendo que:

- El precio del PC por defecto será de 500.
- El tipo de producto sólo puede ser 'p' para 'PC' y 'o' para 'otros'
- Los PC pueden no tener CD, pero siempre deben tener HD



Conjunto de tablas para  
trabajar

# Conjunto de tablas

- ◉ **Alumnos** (DNI, nom\_alum, fecha\_nac, provincia, beca)
- ◉ **Asignaturas** (cod\_asig, nom\_asig, credits, curso)
- ◉ **Matriculas** (cod\_asig, DNI, convocatoria, calificacion)
- ◉ **Profesores** (NRP, nom\_prof, categoria, area, cod\_dpto)
- ◉ **Aula** (cod\_aula, capacidad)
- ◉ **Grupo** (cod\_gru, cod\_asig, NRP, cod\_aula, tipo, max\_al)

# Conjunto de tablas

- **Alumnos** (DNI, nom\_alum, fecha\_nac, provincia, beca)
  - **DNI**: cadena de 9 caracteres
  - **Nom\_alum**: cadena de 50 caracteres.
  - **Fecha\_nac**: fecha
  - **Provincia**: Sólo podrán ser “Granada”, “Jaén” o “Málaga”
  - **Beca**: Sólo podrá ser “SI” o “NO”

# Conjunto de tablas

- **Asignaturas** (cod\_asig, nom\_asig, credits, curso)
  - **Cod\_asig**: valor autonumérico
  - **Nom\_asig**: cadena de 30 caracteres
  - **Credits**: Número entero de 2 dígitos. Debe poder ser NULL
  - **Curso**: Sólo podrá ser 1, 2 o 3

# Conjunto de tablas

- **Matriculas** (cod\_asig, DNI, convocatoria, calificacion)
- **Cod\_asig:** Hará referencia al código de la asignatura de la que se matricula el alumno
- **DNI:** Hará referencia al DNI del alumnos que se matricula
- **Convocatoria:** número entero de 1 dígito
- **Calificación:** número que debe permitir 2 enteros y 2 decimales

# Conjunto de tablas

- **Profesores** (NRP, nom\_prof, categoria, area, cod\_dpto)
- **NRP**: Número entero de dos dígitos
- **Nom\_prof**: Cadena de 15 caracteres
- **Categoria**: Sólo puede ser "Titular" o "Suplente"
- **Area**: Cadena de 20 caracteres
- **Cod\_dpto**: Número entero de un dígito



# Conjunto de tablas

- **Aula** (cod\_aula, capacidad)
  - **Cod\_aula**: Cadena de caracteres de dos posiciones
  - **Capacidad**: Número entero de 3 dígitos

# Conjunto de tablas

- **Grupo** (cod\_gru, cod\_asig, NRP, cod\_aula, tipo, max\_al)
- **Cod\_gru**: Cadena de caracteres de 3 posiciones
- **Cod\_asig**: Hará referencia al código de la asignatura que se imparte en el grupo
- **NRP**: Hará referencia al NRP del profesor que imparte clase en el grupo
- **Cod\_aula**: Hará referencia al código del aula que tiene asignada el grupo
- **Tipo**: Cadena de caracteres. Sólo podrá ser 'M' para grupos de mañana y 'T' para grupos de tarde
- **Max\_al**: Número entero de 3 dígitos



Inserción de filas

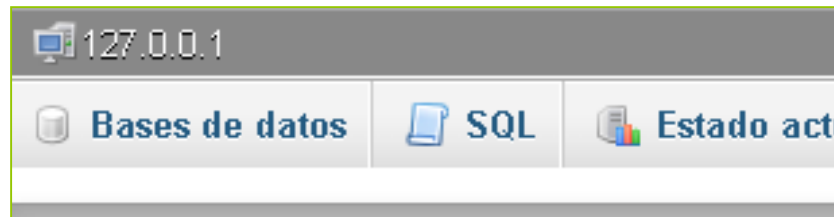
# Inserción de filas

- La inserción de filas en MySQL se podría hacer igualmente con varios clic
- Nosotros necesitamos aprender la sentencia SQL puesto que será lo que utilizaremos en PHP para insertar nuevos datos
- Para insertar datos en una tabla, la sintaxis es sencilla:

```
INSERT INTO <nombre_tabla> VALUES (<valor1>,  
.....,<valorN>);
```

# Inserción de filas en MySQL

- Para ejecutar la sentencia que inserta una fila en MySQL tenemos que ir a la sección SQL



- Se abrirá un cuadro donde podremos escribir la sentencia:

# Inserción de filas en MySQL

Ejecute la o las consultas SQL en el servidor "127.0.0.1":

1

Limpiar

Guardar esta consulta en favoritos:

[ Delimitador  ]

☒ Mostrar esta consulta otra vez

☐ Mantener la caja de texto con la consulta

Continuar



# ALGUNAS ACLARACIONES

- Datos de tipo **cadena** y **fecha** van entrecomillados.
- Los valores de tipo fecha suelen expresarse como cadenas, pero el formato específico depende de SGBD concreto.
  - En MySQL 'aaaa-mm-dd'
- Los valores han de suministrarse en el mismo orden en el que están creados en la tabla.
- Si nos interesa dar un valor nulo a un campo concreto, podemos utilizar la palabra reservada **null** o dejar un espacio vacío en la posición que le corresponda.



# Ejercicios

| DNI       | Nom_alum | Fecha_nac  | Provincia | beca |
|-----------|----------|------------|-----------|------|
| 11111111Z | Lucía    | 1992-05-12 | Granada   | No   |
| 22222222B | Mónica   | 1998-12-18 | Jaén      | Si   |
| 12345678C | Luis     | 1995-01-03 | Granada   | Si   |
| 33333333R | César    | 1993-09-08 | Granada   | No   |
| 55555555T | Roberto  | 1998-11-24 | Málaga    | Si   |

| Cod_asig | Nom_asig    | Creditos | curso |
|----------|-------------|----------|-------|
| 1        | Lengua      | 14       | 1     |
| 2        | Matemáticas | 8        | 1     |
| 3        | Ciencias    | null     | 2     |
| 4        | Literatura  | 7        | 2     |
| 5        | Historia    | null     | 1     |
| 6        | Dibujo      | 12       | 2     |

# Ejercicios

| Cod_asig | DNI        | Convocatoria | Calificación |
|----------|------------|--------------|--------------|
| 1        | 111111111Z | 1            | 3            |
| 1        | 12345678C  | 1            | 8            |
| 2        | 111111111Z | 1            | 5            |
| 2        | 55555555T  | 1            | 7            |
| 3        | 111111111Z | 1            | 7            |
| 4        | 12345678C  | 1            | 3            |
| 2        | 33333333R  | 1            | 4            |
| 1        | 111111111Z | 2            | 6            |
| 1        | 12345678C  | 2            | 2            |

| NRP | Nom_prof  | Categoria | Area        | Cod_depto |
|-----|-----------|-----------|-------------|-----------|
| 21  | Alfonso   | Titular   | Científica  | 1         |
| 34  | Helena    | Titular   | Científica  | 2         |
| 15  | Francisco | Suplente  | Lingüística | 3         |

| Cod_aula | Capacidad |
|----------|-----------|
| A1       | 25        |
| A2       | 20        |
| A3       | 22        |

| Cod_gru | Cod_asig | NRP | cod_aula | tipo | Max_al |
|---------|----------|-----|----------|------|--------|
| G1      | 2        | 21  | A1       | M    | 23     |
| G2      | 1        | 15  | A2       | M    | 18     |
| G3      | 3        | 21  | A3       | M    | 20     |
| G4      | 3        | 15  | A3       | T    | 20     |

# Eliminación de filas

# Eliminación de filas

- El borrado de filas es muy sencillo en SQL. La sentencia es:

```
DELETE FROM <nombre_tabla>;
```

- Ejemplo:** *Borrar todas las filas de la tabla alumnos.*

```
DELETE FROM alumnos;
```

# Eliminación de filas

- No siempre nos interesa borrar todas las filas de una tabla.
- SQL incluye una segunda cláusula dentro de la sentencia DELETE
- **WHERE** permite incluir la condición que deben cumplir las filas que se borrarán:

```
DELETE FROM <nombre_tabla>  
WHERE condición;
```

# Eliminación de filas

- **Ejemplo:** *Borrar todas las filas de los alumnos que tienen beca.*

`DELETE FROM alumnos WHERE beca='SI';`

- **Ejemplo:** *Borrar todas las asignaturas de 2º curso*

`DELETE FROM asignaturas WHERE curso=2;`

- **Ejemplo:** *Borrar todos los alumnos que vienen de Granada*

`DELETE FROM alumnos WHERE provincia='Granada'`

# Eliminación de filas: Ejercicios

- Realizar los siguientes cambios en la base de datos:
  - Borrar todos los alumnos de Jaén
  - Borrar todas las asignaturas con menos de 10 créditos
  - Borrar a todos los profesores del departamento 3
  - Borrar los grupos con una capacidad inferior a 25 alumnos



# Actualización de filas

# Actualización de filas

- Se realiza mediante el uso de la sentencia.

- UPDATE**

- La sintaxis es la siguiente:

```
UPDATE <nombre_tabla>  
SET <nombre_columna>=<nuevo_valor>  
[{,<nombre_columna>=<nuevo_valor>}]  
[WHERE condicion];
```

# Ejemplos

- ◉ **Ejemplo:** Cambiar el nombre del alumno con DNI=44444444 por el de JUAN LOPEZ

```
UPDATE alumnos  
SET nombre='Juan López'  
WHERE DNI = '44444444';
```

- ◉ **Ejemplo:** Subir un punto la nota del alumno de DNI '44444444' en la asignatura de código 3 y aumentarle una convocatoria

```
UPDATE matriculas  
SET calificación = calificación + 1,  
    convocatoria = convocatoria + 1  
WHERE DNI = '44444444' and cod_asig = 3;
```

# Actualización de filas: Ejercicios

- Cambiar de la base de datos los siguientes datos:
  - La alumna Lucía se ha mudado a Málaga
  - A todos los alumnos se les baja la nota 1 punto
  - El profesor con código 34 pasa a ser Suplente
  - Aquellas aulas con capacidad entre 20 y 23, subirles la capacidad en 10 alumnos más
  - El profesor con código 21 pasa a ser Suplente y además del departamento 3

# Manipulando datos con SQL

# Manipulación de datos

- Para recuperar los datos almacenados en las tablas de nuestra base de datos, SQL dispone de la sentencia **SELECT**.
- La sintaxis básica de esta sentencia tiene la siguiente forma:

```
SELECT <nombre_coluna> [{,nombre_columna}]  
FROM <nombre_tabla>  
[WHERE <condicion>];
```

- Hay tres cláusulas principales:
  - **SELECT** contiene las columnas que queremos mostrar.
  - **FROM** indica sobre que tabla queremos consultar.
  - **WHERE** impone una condición booleana que deben cumplir las tuplas para ser recuperadas.

# Consultas Sencillas

**Mostrar sólo los nombres de los alumnos.**

- El proceso a seguir para conseguirlo:
  - Decidir qué tabla o tablas necesitamos:
    - Alumnos
  - Decidir qué columna o columnas queremos mostrar:
    - Nom\_alum
  - Construir la sentencia SQL:

```
SELECT nom_alum
```

```
FROM alumnos;
```

# Consultas Sencillas

- El resultado de la ejecución de esta sentencia es el siguiente:

| Nom_alum |
|----------|
| Lucía    |
| Mónica   |
| Luis     |
| César    |
| Roberto  |



# Consultas Sencillas

## Mostrar la provincia de los alumnos

- Decidir qué tabla o tablas necesitamos:
  - alumnos
- Decidir qué columna o columnas queremos mostrar:
  - provincia
- Construir la sentencia SQL:

```
SELECT provincia  
FROM alumnos;
```

# Consultas Sencillas

- El resultado de la ejecución de esta sentencia es el siguiente:

| Provincia |
|-----------|
| Granada   |
| Jaén      |
| Granada   |
| Granada   |
| Málaga    |

# Consultas Sencillas

## **Mostrar el DNI y el nombre de los alumnos**

- Decidir qué tabla o tablas necesitamos:
  - alumnos
- Decidir qué columna o columnas queremos mostrar:
  - DNI, nom\_alum
- Construir la sentencia SQL:

```
SELECT DNI, nom_alum  
  
FROM alumnos;
```

# Consultas Sencillas

- El resultado de la ejecución de esta sentencia es el siguiente:

| DNI       | Nom_alum |
|-----------|----------|
| 11111111Z | Lucía    |
| 22222222B | Mónica   |
| 12345678C | Luis     |
| 33333333R | César    |
| 55555555T | Roberto  |

# Consultas Sencillas

## Mostrar todos los datos de los alumnos

- Decidir qué tabla o tablas necesitamos:
  - alumnos
- Decidir qué columna o columnas queremos mostrar:
  - Todas
- Construir la sentencia SQL:

```
SELECT DNI, nom_alum, fecha_nac, provincia,  
beca
```

```
FROM alumnos;
```

```
SELECT *
```

```
FROM alumnos;
```

# Consultas Sencillas

- El resultado de la ejecución de esta sentencia es el siguiente:

| DNI       | Nom_alum | Fecha_nac  | Provincia | beca |
|-----------|----------|------------|-----------|------|
| 11111111Z | Lucía    | 12/05/1992 | Granada   | No   |
| 22222222B | Mónica   | 18/12/1998 | Jaén      | Si   |
| 12345678C | Luis     | 03/01/1995 | Granada   | Si   |
| 33333333R | César    | 08/09/1993 | Granada   | No   |
| 55555555T | Roberto  | 24/11/1998 | Málaga    | Si   |

# Consultas Sencillas

## **Mostrar el nombre de los alumnos que son de Granada**

- Decidir qué tabla o tablas necesitamos:
  - alumnos
- Decidir qué columna o columnas queremos mostrar:
  - Nom\_alum
- Decidir qué condición o condiciones deben cumplir las filas mostradas:
  - Provincia = 'Granada'
- Construir la sentencia SQL:  

```
SELECT nom_alum  
FROM alumnos  
WHERE provincia = 'Granada';
```

# Consultas Sencillas

- El resultado de la ejecución de esta sentencia es el siguiente:

| Nom_alum |
|----------|
| Lucía    |
| Luis     |
| César    |



# Aclaración

Las condiciones de la cláusula WHERE pueden ser tan complicadas como el usuario desee.

- Identificadores de columnas
- Literales
- Operadores de comparación (<, >, >=, <=, <>, =)
- Operadores lógicos (AND, OR, NOT)

# Consultas Sencillas

## **Mostrar el nombre de los alumnos que tienen beca**

- Decidir qué tabla o tablas necesitamos:
  - alumnos
- Decidir qué columna o columnas queremos mostrar:
  - Nom\_alum
- Decidir qué condición o condiciones deben cumplir las filas mostradas:
  - beca = 'Si'
- Construir la sentencia SQL:

```
SELECT nom_alum  
FROM alumnos  
WHERE beca= 'Si';
```

# Consultas Sencillas

- El resultado de la ejecución de esta sentencia es el siguiente:

| Nom_alum |
|----------|
| Mónica   |
| Luis     |
| Roberto  |

# Consultas Sencillas

## Mostrar el nombre y DNI de los alumnos que NO son de Granada

- Decidir qué tabla o tablas necesitamos:
  - alumnos
- Decidir qué columna o columnas queremos mostrar:
  - Nom\_alum, DNI
- Decidir qué condición o condiciones deben cumplir las filas mostradas:
  - Provincia <> 'Granada'

- Construir la sentencia SQL:

```
SELECT nom_alum, DNI  
FROM alumnos  
WHERE provincia <> 'Granada';
```

# Consultas Sencillas

- El resultado de la ejecución de esta sentencia es el siguiente:

| Nom_alum | DNI       |
|----------|-----------|
| Mónica   | 22222222B |
| Roberto  | 55555555T |

# Consultas Sencillas

## **Mostrar todos los datos del alumno llamado Lucía**

- Decidir qué tabla o tablas necesitamos:
  - alumnos
- Decidir qué columna o columnas queremos mostrar:
  - Todos
- Decidir qué condición o condiciones deben cumplir las filas mostradas:
  - Nom\_alum= 'Lucía'
- Construir la sentencia SQL:

```
SELECT *  
FROM alumnos  
WHERE nom_alum= 'Lucía';
```

# Consultas Sencillas

- El resultado de la ejecución de esta sentencia es el siguiente:

| DNI       | Nom_alum | Fecha_nac  | Provincia | beca |
|-----------|----------|------------|-----------|------|
| 11111111Z | Lucía    | 12/05/1992 | Granada   | No   |

# Consultas Sencillas

- Supongamos que el contenido de la tabla asignaturas es el siguiente:

| Cod_asig | Nom_asig    | Creditos | curso |
|----------|-------------|----------|-------|
| 1        | Lengua      | 14       | 1     |
| 2        | Matemáticas | 8        | 1     |
| 3        | Ciencias    | 19       | 2     |
| 4        | Literatura  | 7        | 2     |
| 5        | Historia    | 25       | 1     |
| 6        | Dibujo      | 12       | 2     |



# Consultas sencillas

- Realicemos las siguientes consultas:
  - Mostrar el nombre de todas las asignaturas.
  - Mostrar los créditos de todas las asignaturas.
  - Mostrar código de asignatura y curso de todas las asignaturas.
  - Mostrar los créditos de las asignaturas de 2º curso.
  - Mostrar todos los datos de la asignatura con código 4
  - Mostrar todos los datos de aquellas asignaturas con más de 15 créditos.
  - Mostrar los nombres de las asignaturas que tienen entre 12 y 25 créditos.

# Consultas sencillas

- Realicemos las siguientes consultas:
  - Mostrar el DNI de los alumnos matriculados.
  - Mostrar el DNI de los alumnos matriculados en la asignatura 3.
  - Mostrar el DNI de los alumnos con algo suspenso.
  - Mostrar el DNI de los alumnos que han aprobado la asignatura con código 1.
  - Mostrar el DNI y el código de la asignatura de los alumnos que hayan sacado en algo más de un 7.
  - Mostrar el DNI de los alumnos con asignaturas aprobadas.



# Anotaciones

# Eliminación de valores duplicados

| DNI       | Nom_alum | Fecha_nac  | Provincia | beca |
|-----------|----------|------------|-----------|------|
| 11111111Z | Lucía    | 12/05/1992 | Granada   | No   |
| 22222222B | Mónica   | 18/12/1998 | Jaén      | Si   |
| 12345678C | Luis     | 03/01/1995 | Granada   | Si   |
| 33333333R | César    | 08/09/1993 | Granada   | No   |
| 55555555T | Roberto  | 24/11/1998 | Málaga    | Si   |

# Eliminación de valores duplicados

**Ejemplo:** Recuperar las ciudades de las que provienen los Alumnos.

```
SELECT provincia  
FROM alumnos;
```

Esta consulta daría como resultado lo siguiente:

| Provincia |
|-----------|
| Granada   |
| Jaén      |
| Granada   |
| Granada   |
| Málaga    |

# Eliminación de valores duplicados

- Con SQL, si queremos eliminar las filas duplicadas del resultado debemos solicitarlo mediante **DISTINCT**

```
SELECT DISTINCT provincia  
FROM alumnos;
```

- El resultado en este caso sería:

| Provincia |
|-----------|
| Granada   |
| Jaén      |
| Málaga    |

# Ordenación de los resultados

- SQL, al ejecutar una consulta, muestra los datos en el mismo orden en el que se encuentran en la tabla original.
- Sin embargo, nosotros podemos indicar que queremos que los resultados se muestren en otro orden.
- Utilizamos para ello la cláusula **ORDER BY**.  
**ORDER BY <nombre\_atributo>**
- La cláusula ORDER BY admite además indicar el tipo de ordenación.
  - ASC
  - DESC

# Ordenación de los resultados

**Ejemplo:** *Mostrar la lista de los becarios ordenados en orden alfabético.*

```
SELECT *  
FROM alumnos  
WHERE beca='SI'  
ORDER BY nomb_alumn ASC;
```

| DNI       | Nom_alum | Fecha_nac  | Provincia | beca |
|-----------|----------|------------|-----------|------|
| 12345678C | Luis     | 03/01/1995 | Granada   | Si   |
| 22222222B | Mónica   | 18/12/1998 | Jaén      | Si   |
| 55555555T | Roberto  | 24/11/1998 | Málaga    | Si   |



# Ordenación de los resultados

**Ejemplo:** *Mostrar la lista de los alumnos ordenados por su provincia de procedencia de forma descendente y dentro de cada provincia, ordenados alfabéticamente.*

```
SELECT provincia, nomb_alum  
FROM alumnos  
ORDER BY provincia DESC, nomb_alumn ASC;
```

| Provincia | Nom_alum |
|-----------|----------|
| Málaga    | Roberto  |
| Jaén      | Mónica   |
| Granada   | César    |
| Granada   | Lucía    |
| Granada   | Luis     |

# Control de los resultados

# Cantidad de resultados

- Podemos pedirle que sólo nos muestre una cantidad limitada de filas.
- Con la cláusula **LIMIT** podremos indicar qué filas queremos ver.
- LIMIT** espera dos números:
  - Fila en la que empezar (no se mostrará)
  - Cantidad de filas a mostrar

# Cantidad de resultados

- Mostrar sólo el nombre de los dos primeros alumnos de Granada

```
SELECT nom_alum  
FROM alumnos  
WHERE provincia='Granada'  
ORDER BY nom_alum  
LIMIT 0, 2;
```

- Mostrar el nombre del alumno más viejo

```
SELECT nom_alum  
FROM alumnos  
ORDER BY fecha_nac asc  
LIMIT 0, 1;
```

# Consultas sobre varias tablas

# Consultas sobre varias tablas

- Hay veces que es necesario utilizar datos que están contenidos en distintas tablas.
- En SQL, se pueden incluir varias tablas en la cláusula **FROM**.
- En este caso el sistema hace el **producto cartesiano** de todas las tablas incluidas y luego realiza la consulta sobre la tabla resultado de ese producto cartesiano.
- Esto quiere decir que tenemos que eliminar las **filas no reales**.

# Consultas sobre varias tablas

- Para explicar esto, vamos a utilizar dos nuevas tablas:
- Dueños:
- Perros:

| DNI     | Nom_dueño |
|---------|-----------|
| 111111S | Rocío     |
| 333333E | Paloma    |
| 666666B | Víctor    |

| Num_perro | Nom_perro | Dni_dueño |
|-----------|-----------|-----------|
| 1         | Ali       | 111111S   |
| 2         | Buda      | 333333E   |
| 3         | Pico      | 111111S   |
| 4         | Rufo      | 666666B   |

# Consultas sobre varias tablas

- Supongamos que nos piden mostrar los perros de cada uno de los dueños.

- La consulta:

```
Select *
```

```
From dueños, perros;
```

- Dará como resultado:



# Consultas sobre varias tablas

| DNI     | Nom_dueño | Num_perro | Nom_perro | DNI_dueño |
|---------|-----------|-----------|-----------|-----------|
| 111111S | Rocío     | 1         | Ali       | 111111S   |
| 111111S | Rocío     | 2         | Buda      | 333333E   |
| 111111S | Rocío     | 3         | Pico      | 111111S   |
| 111111S | Rocío     | 4         | Rufo      | 666666B   |
| 333333E | Paloma    | 1         | Ali       | 111111S   |
| 333333E | Paloma    | 2         | Buda      | 333333E   |
| 333333E | Paloma    | 3         | Pico      | 111111S   |
| 333333E | Paloma    | 4         | Rufo      | 666666B   |
| 666666B | Víctor    | 1         | Ali       | 111111S   |
| 666666B | Víctor    | 2         | Buda      | 333333E   |
| 666666B | Víctor    | 3         | Pico      | 111111S   |
| 666666B | Víctor    | 4         | Rufo      | 666666B   |

# Consultas sobre varias tablas

- Dentro de esta tabla, hay muchas filas que **no son “REALES”** ya que no unen de verdad a cada perro con su dueño:

# Consultas sobre varias tablas

| DNI     | Nom_dueño | Num_perro | Nom_perro | DNI_dueño |
|---------|-----------|-----------|-----------|-----------|
| 111111S | Rocío     | 1         | Ali       | 111111S   |
| 111111S | Rocío     | 2         | Buda      | 333333E   |
| 111111S | Rocío     | 3         | Pico      | 111111S   |
| 111111S | Rocío     | 4         | Rufo      | 666666B   |
| 333333E | Paloma    | 1         | Ali       | 111111S   |
| 333333E | Paloma    | 2         | Buda      | 333333E   |
| 333333E | Paloma    | 3         | Pico      | 111111S   |
| 333333E | Paloma    | 4         | Rufo      | 666666B   |
| 666666B | Víctor    | 1         | Ali       | 111111S   |
| 666666B | Víctor    | 2         | Buda      | 333333E   |
| 666666B | Víctor    | 3         | Pico      | 111111S   |
| 666666B | Víctor    | 4         | Rufo      | 666666B   |

# Consultas sobre varias tablas

- ¿Qué pasa con las filas que SI son reales? ¿Qué tienen en común todas ellas?
  - DNI = DNI\_dueño
- Por tanto, una vez unidas las tablas, deberemos quedarnos solo con las filas que nos interesan:

```
Select *  
From dueños, perros  
Where dni = dni_dueño;
```

| DNI     | Nom_dueño | Num_perro | Nom_perro | DNI_dueño |
|---------|-----------|-----------|-----------|-----------|
| 111111S | Rocío     | 1         | Ali       | 111111S   |
| 111111S | Rocío     | 3         | Pico      | 111111S   |
| 333333E | Paloma    | 2         | Buda      | 333333E   |
| 666666B | Víctor    | 4         | Rufo      | 666666B   |

## Consultas sobre varias tablas

Volvamos ahora a las  
tablas que teníamos...

# Consultas sobre varias tablas

**Mostrar para cada matricula, el nombre de la asignatura a la que corresponde.**

- Tablas que necesitamos:
  - Matriculas, asignaturas.
- Columnas que queremos visualizar:
  - Todas las de matriculas, nombre de asignatura.
- Condición que deben cumplir las filas visualizadas:
  - `Matriculas.cod_asig = asignaturas.cod_asig`

# Consultas sobre varias tablas

- Consulta:

```
Select matriculas.*, asignaturas.nom_asig  
From matriculas, asignaturas  
Where matriculas.cod_asig = asignaturas.cod_asig;
```

| Cod_asig | DNI       | Convocatoria | Calificación | Nom_asig    |
|----------|-----------|--------------|--------------|-------------|
| 1        | 11111111Z | 1            | 3            | Lengua      |
| 1        | 12345678C | 1            | 8            | Lengua      |
| 2        | 11111111Z | 1            | 5            | Matemáticas |
| 2        | 55555555T | 1            | 7            | Matemáticas |
| 3        | 11111111Z | 1            | 7            | Ciencias    |
| 4        | 12345678C | 1            | 3            | Literatura  |
| 2        | 33333333R | 1            | 4            | Matemáticas |
| 1        | 11111111Z | 2            | 6            | Lengua      |
| 1        | 12345678C | 2            | 2            | Lengua      |

# Consultas sobre varias tablas

- Mostrar la tabla matriculas, incluyendo el nombre de cada alumno.
- Mostrar el nombre de las asignaturas en las que está matriculado cada alumno.
- Mostrar el nombre de aquellos alumnos con más de un 4 en la asignatura con código 2.
- Visualizar el nombre de las asignaturas con alumnos suspensos.
- Mostrar todos los datos de las asignaturas en las que está matriculado el alumno llamado "Mónica"



# Consultas sobre varias tablas

- Mostrar el nombre de los profesores que dan clase en el aula A3.
- Mostrar los códigos de los grupos en los que se imparte Ciencias.
- Mostrar los códigos de los grupos en los que imparte clase el profesor Francisco.
- Mostrar la capacidad de las aulas en las que se imparte Lengua.
- Listar el nombre y la categoría de los profesores que dan clase en el grupo G2.
- Visualizar todos los datos de los grupos en los que imparte clase Helena

# Consultas sobre varias tablas

- Mostrar los nombres y las categorías de los profesores que imparten la asignatura de “Ciencias”.
- Visualizar los nombres de las asignaturas que se imparten por la mañana.
- Mostrar los nombres de las asignaturas que imparten los profesores del departamento 2.
- Mostrar los nombres de las asignaturas que tienen aprobadas los alumnos de Granada.
- Listar las provincias de las que vienen los alumnos con más de un 7 en matemáticas.

# Consultas sobre varias tablas

- Mostrar los nombres y las notas de todos los alumnos de Málaga, ordenados alfabéticamente.
- Mostrar las provincias de los alumnos que han sacado un 7 en Matemáticas.
- Mostrar las aulas en las que se da clase de asignaturas que aun no tienen asignados créditos.
- Mostrar los nombres y las áreas de los profesores, que imparten clase por la mañana.
- Visualizar una lista de los grupos en los que imparten clase los profesores del área de Científica.
- Listar los datos de los alumnos que cursan Matemáticas, ordenados por su fecha de nacimiento.



# Bases de Datos

---