



Laboratorio II / Laboratorio de Software II 2024

**Clase Teórica 06: Introducción a Base de datos.
Inserción. Selección (where y like). Sha.**

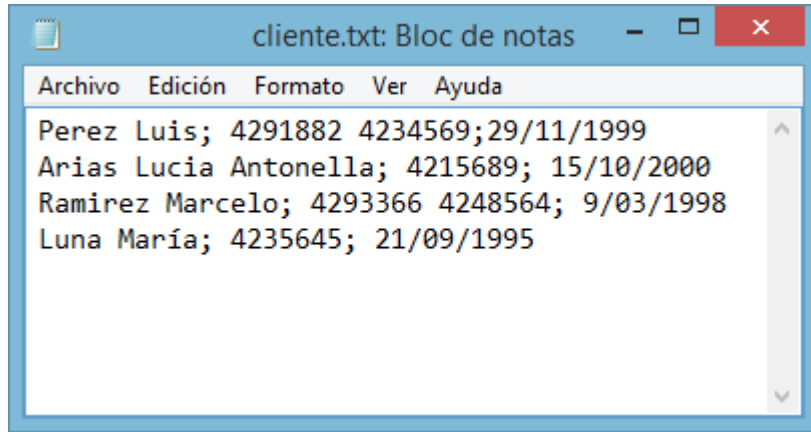


Docente: Myriam Ruiz
Licenciada en Informática
Profesora en Computación y Matemática
Programadora Universitaria



Bases de datos - Necesidad

- En la clase 05 vimos el tema de uso de archivos de texto.



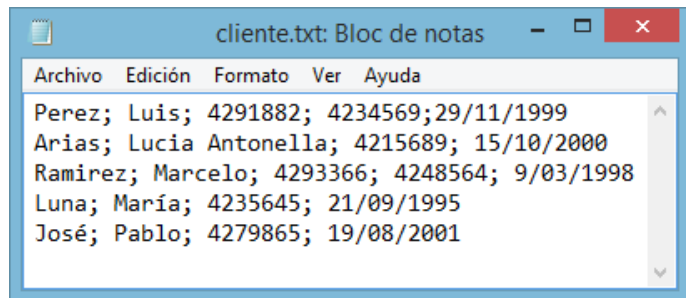
Nombre	Teléfono	Fec_Nac
Perez Luis	4291882 4234569	29/11/1999
Arias Lucia Antonella	4215689	15/10/2000
Ramirez Marcelo	4293366 4248564	9/03/1998
Luna María	4235645	21/09/1995

Cómo se vería si fuera Excel o similar

- 1) Fue sencillo buscar en un archivo de texto? Por ejemplo saber el teléfono sólo de Ramirez? Será sencillo modificar un dato?
- 2) Será claro tener más de un dato por columna (ej: teléfono)?
- 3) Será claro tener nombre y apellido juntos? Y si la persona se llama Pablo José? (apellido José)

Bases de datos - Necesidad

- Será mejor esto?

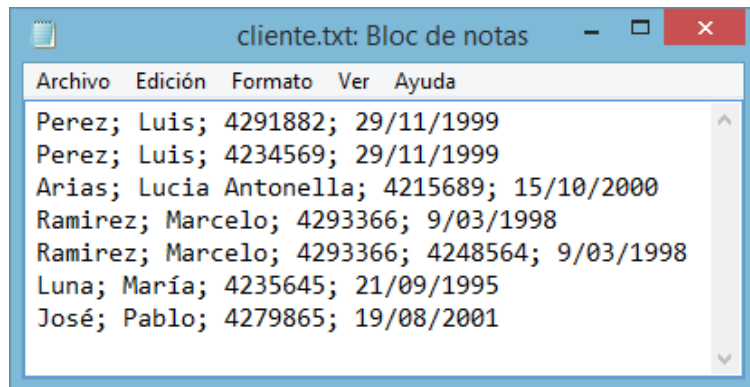


Apellido	Nombre	Teléfono1	Teléfono2	Fec_Nac
Perez	Luis	4291882	4234569	29/11/1999
Arias	Lucia Antonella	4215689		15/10/2000
Ramirez	Marcelo	4293366	4248564	9/03/1998
Luna	María	4235645		21/09/1995
José	Pablo	4279865		19/08/2001

- 1) Queda más claro al separar Apellido y Nombre, y los teléfonos?
- 2) Qué pasa con todos los casos que no existe un Teléfono2?
(los lugares que no tienen nada se llaman Nulos y se hace una reserva de disco para anotar que están vacíos, es decir, ocupan espacio)

Bases de datos - Necesidad

- Otra opción es:



Apellido	Nombre	Teléfono	Fec_Nac
Perez	Luis	4291882	29/11/1999
Perez	Luis	4234569	29/11/1999
Arias	Lucia Antonella	4215689	15/10/2000
Ramirez	Marcelo	4293366	9/03/1998
Ramirez	Marcelo	4248564	9/03/1998
Luna	María	4235645	21/09/1995
José	Pablo	4279865	19/08/2001

- 1) Se justifica crear una fila nuevo para no tener Nulos?
- 2) Otro problema: Si modifico la fecha de nacimiento en una de las filas, pero no en la repetida, será **consistente** (coherente)?

Bases de datos - Necesidad

- Estas formas de guardar datos tienen los siguientes problemas:
 - Duplicación de datos
 - Uso de mucho espacio en disco (por la duplicación) y por lo tanto aumento de costo de almacenamiento
 - Acceso lento a un dato específico
 - Mucha dificultad para Modificar un dato almacenado
 - Posibles inconsistencias en los datos

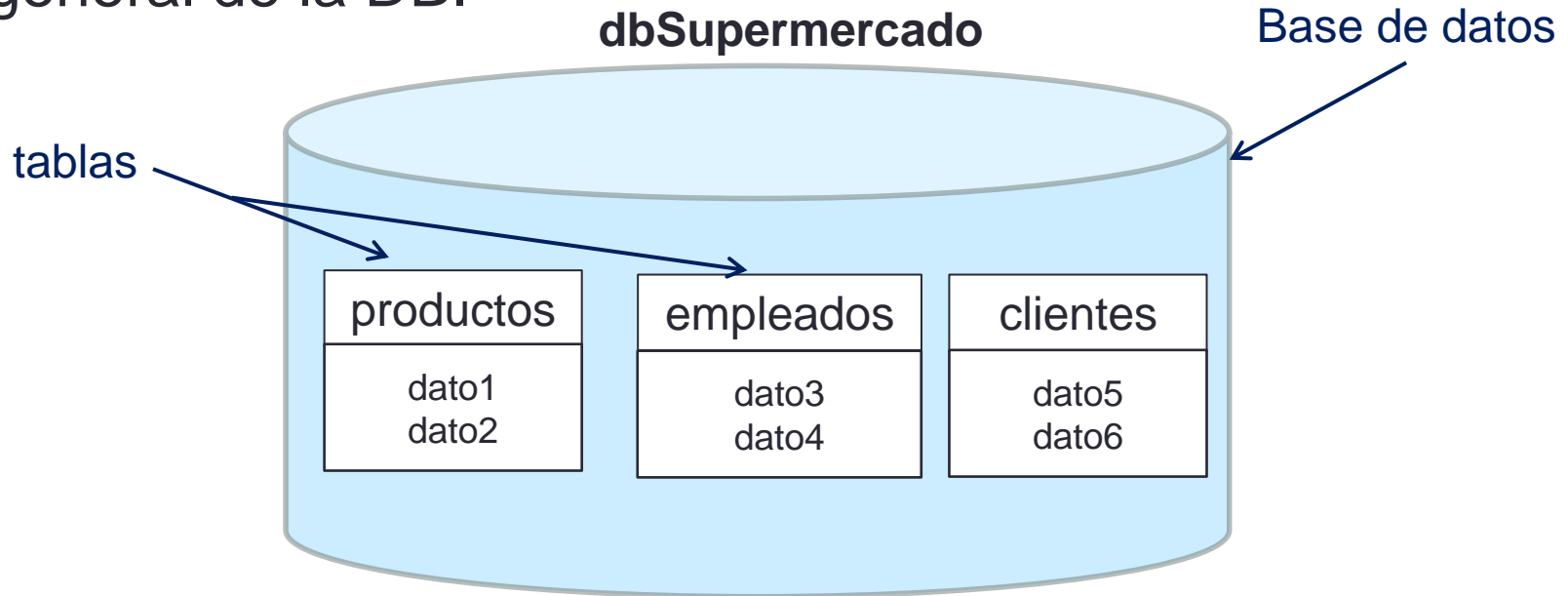
Ante estos problemas, se hace necesario una mejor forma de almacenar: Las Bases de Datos

Bases de datos - Concepto

- Base de Datos (DB, Data Base): es un conjunto de datos perteneciente a **un mismo contexto (o temática)**, ordenada de modo sistemático para su posterior **recuperación, análisis y/o transmisión**.
- Por lo tanto, de cada situación que necesitemos almacenar, sólo se guardarán los datos (o atributos) relevantes

Bases de datos - Tablas

- Una DB contiene una o más tablas.
- Cada tabla trata de un tema particular, parte del tema general de la DB.



Bases de datos - Tablas

- Partes de una tabla

Registro o Fila o Tupla →

Columna o Campo

nombre	marca	modelo	Particularidad	cantidad	precio
Procesador	AMD	Ryzen 3	3,7Ghz	5	6899
RAM	Kingston	Hyperx Fury Ddr4 2400	x 4GB	20	2529
Fuente Pc	Thermaltake	Smart Sp-650m	650w 80 Plus Bronze 2	8	7409

Bases de datos – Formas Normales

- En la asignatura Conceptos de Bases de datos I verán lo que se llaman las *Formas Normales*. Tomando ese marco, pero sin mencionarlas, seguiremos las siguientes recomendaciones al momento de crear una Tabla:
 - ✓ Almacenar un único dato por campo (Ej: Apellido en un campo distinto a Nombre)
 - ✓ Almacenar un único dato del mismo tipo por campo (ej. Si hay más de un teléfono se deben guardar por separado de alguna forma)
 - ✓ Cada tabla debe tener un campo que sea único para ser la clave identificadora del registro.

Bases de datos – Gestión de Base 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.

Bases de datos – Gestión de Base de Datos

- Para crear una DB se pueden usar diversos DBMS



ORACLE®

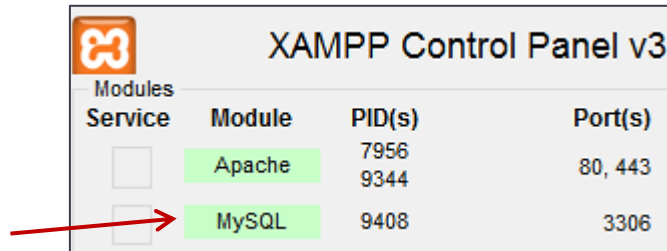


- Nosotros usaremos **MariaDB**
 - Derivado de MySQL, con sus características y algunas mejoras.
 - Surge cuando MySQL es adquirido por Oracle, a fin de seguir la filosofía Open Source.
 - Es totalmente compatible con MySQL.

Creación de Bases de datos

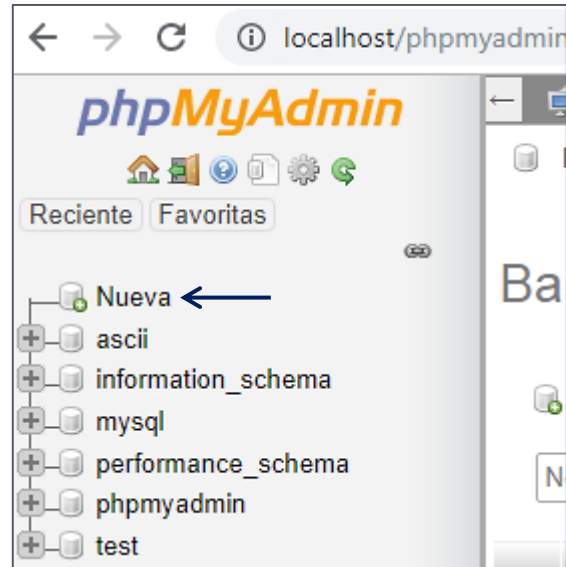
Creación de Base de Datos

- Para interactuar con MariaDB vamos a utilizar phpmyadmin
 - Es una herramienta de software gratuita escrita en PHP
 - Permite manejar la administración vía Web
- Para utilizarlo, tener encendido en XAMPP: Apache y MYSQL e ingresar a: <http://localhost/phpmyadmin/>



Creación de DB - Pasos

- P1. Hacer clic en Nueva



Creación de DB - Pasos

- P2. Colocarle un nombre a la Base de Datos y elegir Cotejamiento **utf-8_general_ci**. Luego presionar Crear



Tipos de Datos MariaDB - Cadena

Tipo	Valores	Tamaño
CHAR (M)	255 Letras. Por defecto 1	Fijo: M Bytes
VARCHAR (M)	65.532 Letras	Variable: M + 1 Bytes
TINYTEXT	255 Letras.	Longitud + 1 Bytes
TEXT (M)	65.535 Letras	Longitud + 2 Bytes
MEDIUMTEXT	16.777.215 Letras	Longitud + 3 Bytes
LONGTEXT	4.294.967.295 o 4GB Letras	Longitud + 4 Bytes

Tipos de Datos MariaDB - Enteros

Tipo	Valores	Tamaño
TINYINT (M)	-128 a 127 o 0 a 255 (con atributo unsigned)	1 byte
SMALLINT (M)	-32768 a 32767 o 0 a 65535	2 bytes
MEDIUMINT (M)	-8388608 a 8388607, o 0 a 16777215	3 bytes
INT (M)	-2147483648 a 2147483647, o 0 a 4294967295	4 bytes
BIGINT (M)	-9223372036854775808 a 9223372036854775807 o 0 a 18446744073709551615	8 bytes

Tipos de Datos MariaDB - Decimales

Tipo	Valores	Tamaño
DECIMAL(M,D)	65 dígitos para entero y 30 para decimal	M + 2 bytes si D > 0, sino M +1 bytes
FLOAT(M,D)	3.402823466E+38 a -1.175494351E-38 0 1.175494351E-38 a 3.402823466E+38	4 bytes si M <= 24, sino 8 bytes
DOUBLE (M,D)	-1.7976931348623157E+308 a - 2.2250738585072014E-308 0 2.2250738585072014E-308 a 1.7976931348623157E+308	8 bytes
REAL	Es sinónimo de DOUBLE	

Tipos de Datos MariaDB - Fecha

Tipo	Valores	Tamaño
DATE	'1000-01-01' a '9999-12-31'	3 bytes
DATETIME	'1000-01-01 00:00:00.000000' a '9999-12-31 23:59:59.999999'	8 bytes
TIMESTAMP	'1970-01-01 00:00:01' a '2038-01-19 03:14:07'	4 bytes
TIME	'-838:59:59.999999' a '838:59:59.999999'	3 bytes
YEAR	1901 a 2155, no se usa en 2 dígitos. El nulo se toma como 0000	1 byte

Hay más tipos:

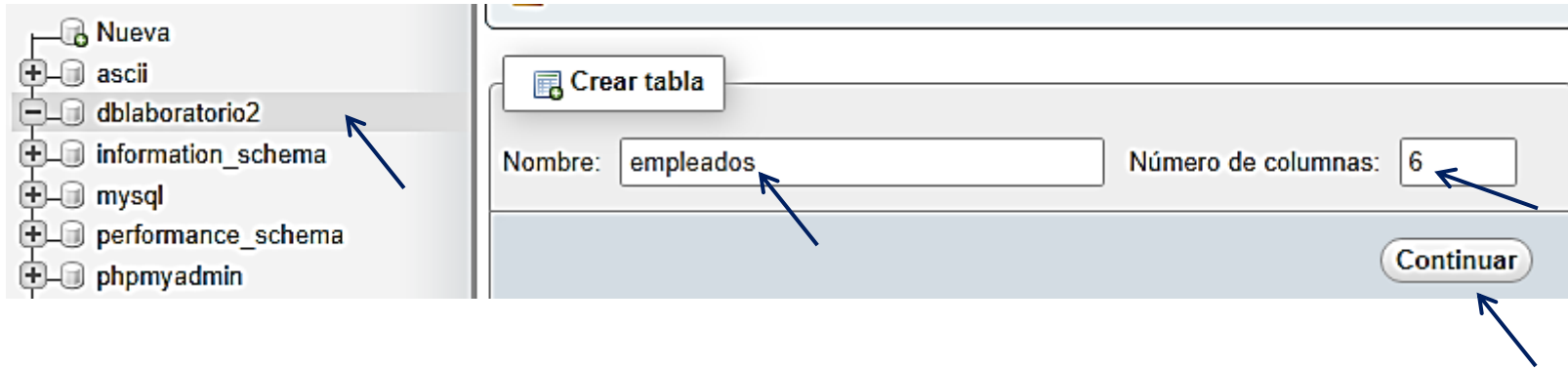
<https://mariadb.com/kb/en/library/data-types/>

Tipos de Datos MariaDB – Atributos

- Se puede modificar los tipos de datos, mediante la elección de Atributos en phpmyadmin:
 - unsigned (sin signo)
 - unsigned zerofill (sin signo con ceros antepuestos)
- También se podría aceptar Nulos

Creación de Tablas - Pasos

- P1. Seleccionar la DB creada, al lado derecho ingresar el nombre de la tabla y la cantidad de columnas que tendrá. Luego presionar continuar



Creación de Tablas - Pasos

- P2. Darle nombre a las columnas y establecer el tipo de dato que tendrán.
- Proponer los campos y tipos de dato para el siguiente caso:

Tabla Empleados

legajo	apellido	nombre	sueldo	fecha_nac	foto

Creación de Tablas - Pasos

- P2.

Nueva
+ ascii
- dblaboratorio2
+ information_schema
+ mysql
+ performance_schema
+ phpmyadmin
+ test

Nombre	Tipo	Longitud/Valores
--------	------	------------------

legajo	INT	4
--------	-----	---

Seleccionar desde las columnas centrales

apellido	VARCHAR	30
----------	---------	----

Seleccionar desde las columnas centrales

nombre	VARCHAR	50
--------	---------	----

Seleccionar desde las columnas centrales

suelo	FLOAT	6,2
-------	-------	-----

Seleccionar desde las columnas centrales

fecha_nac	DATE	
-----------	------	--

Seleccionar desde las columnas centrales

foto	VARCHAR	50
------	---------	----

Seleccionar desde las

Creación de Tablas - Pasos

- P2. De ser necesario, se debe establecer cuál es el campo que identifica de manera única a un registro (fila), indicando que es índice PRIMARY:

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice
<input type="text" value="legajo"/> <small>Seleccionar desde las columnas centrales</small>	INT	4	Ninguno		UNSIGNED	<input type="checkbox"/>	---
<input type="text" value="apellido"/> <small>Seleccionar desde las columnas centrales</small>	VARCHAR	30	Ninguno			<input type="checkbox"/>	PRIMARY UNIQUE INDEX FULLTEXT SPATIAL
<input type="text"/>							

Creación de Tablas - Pasos

- P2. Aparece lo siguiente. Presionar Continuar.

Agregar índice

Nombre del índice:

PRIMARY

Opción de índice:

PRIMARY

+ Advanced Options

	Columna	Tamaño
	legajo [int]	

Continuar

Cancelar

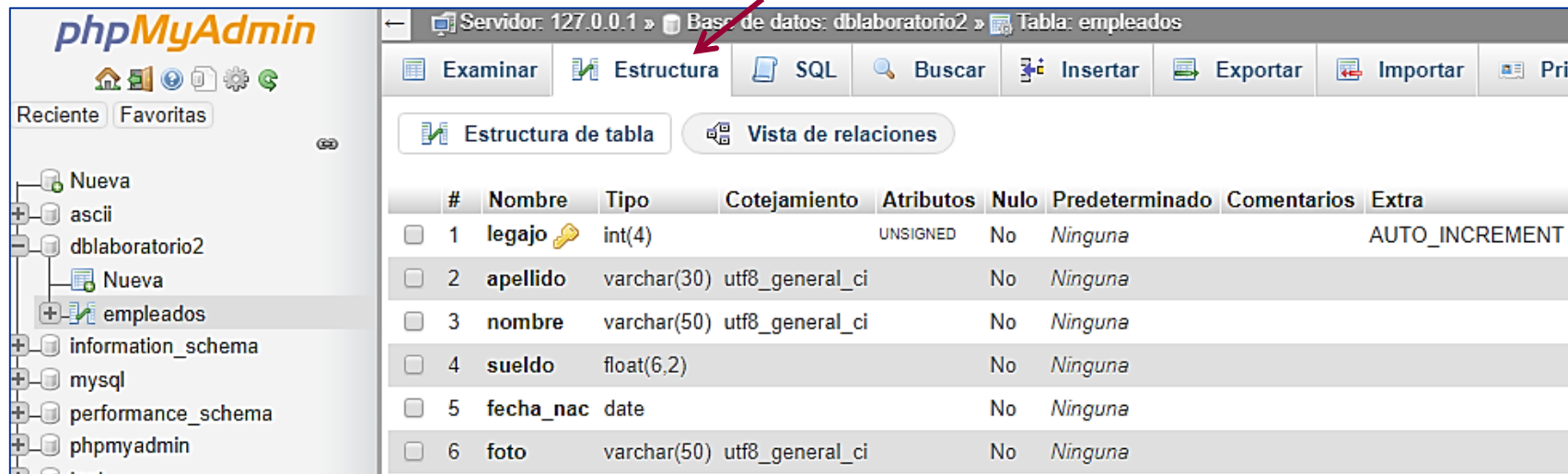
Creación de Tablas - Pasos

- P2. También es útil que el campo que sea clave PRIMARY sea autoincremental (A_I), lo que quiere decir que se numera de manera automática

Atributos	Nulo	Indice	A_I	Comentarios
UNSIGNED ▼	<input type="checkbox"/>	PRIMARY ▼	<input checked="" type="checkbox"/>	
		PRIMARY		

Creación de Tablas - Pasos

- Se puede ver como quedó la tabla yendo a Estructura



The screenshot shows the phpMyAdmin interface. On the left is the database navigation tree with 'empleados' selected under 'dblaboratorio2'. The main panel shows the 'Estructura' (Structure) tab for the 'empleados' table. A red arrow points to the 'Estructura' tab in the top navigation bar. Below the tabs, there are two buttons: 'Estructura de tabla' (active) and 'Vista de relaciones'. A table lists the columns of the 'empleados' table.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1 legajo	int(4)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2 apellido	varchar(30)	utf8_general_ci		No	Ninguna		
<input type="checkbox"/>	3 nombre	varchar(50)	utf8_general_ci		No	Ninguna		
<input type="checkbox"/>	4 sueldo	float(6,2)			No	Ninguna		
<input type="checkbox"/>	5 fecha_nac	date			No	Ninguna		
<input type="checkbox"/>	6 foto	varchar(50)	utf8_general_ci		No	Ninguna		

Creación de Tablas - Pasos

- Si quisiéramos cambiar las características de algún campo, estando en Estructura elegimos cambiar



#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	legajo 	int(4)		UNSIGNED	No	Ninguna		AUTO_INCREMENT	 Cambiar  Eliminar  Más
<input type="checkbox"/> 2	apellido	varchar(30)	utf8_general_ci		No	Ninguna			 Cambiar  Eliminar  Más
<input type="checkbox"/> 3	nombre	varchar(50)	utf8_general_ci		No	Ninguna			 Cambiar  Eliminar  Más
<input type="checkbox"/> 4	sueldo	float(7,2)			Sí	NULL			 <u>Cambiar</u>  Eliminar  Más
<input type="checkbox"/> 5	fecha_nac	date			No	Ninguna			 Cambiar  Eliminar  Más
<input type="checkbox"/> 6	foto	varchar(50)	utf8_general_ci		Sí	NULL			 Cambiar  Eliminar  Más

Creación de Tablas - Pasos

- Se pueden agregar más campos yendo debajo de los campos y llenando cuantas columnas queremos agregar:

The screenshot shows the phpMyAdmin interface. On the left, a tree view shows the database structure with 'empleados' selected under 'Nueva'. The main area displays the 'Add new column' dialog for the 'empleados' table. The dialog shows a list of existing columns: 'sueldo' (float(7,2)), 'fecha_nac' (date), and 'foto' (varchar(50)). Below this, there are buttons for 'Seleccionar todo', 'Agregar a columnas centrales', and 'Eliminar de las columnas centrales'. The 'Agregar' button is highlighted, and a dropdown menu is open, showing options for where to add the new column: 'después de foto', 'Al comienzo de la tabla', 'después de legajo', 'después de apellido', 'después de nombre', 'después de sueldo', 'después de fecha_nac', and 'después de foto' (highlighted). The 'Continuar' button is also visible.

	4	sueldo	float(7,2)	Sí	NULL
	5	fecha_nac	date	No	Ninguna
	6	foto	varchar(50) utf8_general_ci	Sí	NULL

↑ ☐ Seleccionar todo Para los elementos que están marcados: [Examinar](#)

[Agregar a columnas centrales](#) [Eliminar de las columnas centrales](#)

[Imprimir](#) [Planteamiento de la estructura de tabla](#) [Hacer seguimiento de la estructura de la tabla](#)

[Agregar](#) 1 columna(s) después de foto [Continuar](#)

Índices [?](#)

Acción	Nombre de la	paquetado	Columna
Editar Eliminar	PRIMARY		legajo

Crear un índice en 1 columna(s) [Continuar](#)

Importar Bases de datos

Importar Base de Datos

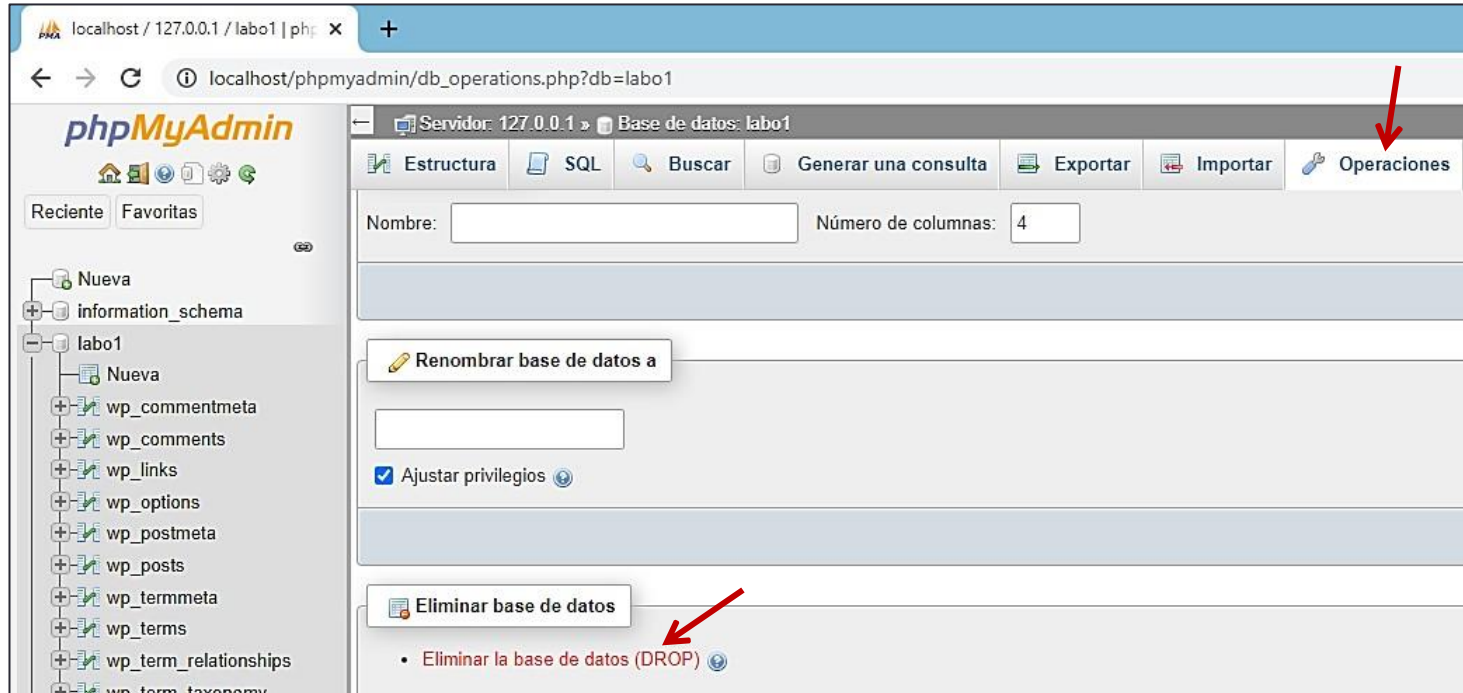
1. Cada vez que se quiera importar una base de datos, se deberá crear una base de datos con el mismo nombre que la original, seleccionarla e ir a importar
2. Si ya existe una base de datos se deberá primero eliminar la que ya existe y recién crear la base de datos en blanco

Importar Base de Datos

- Eliminar la que ya existe y recién crear la base de datos en blanco
 - Ir a Operaciones y hacer clic en Eliminar la base de datos (DROP)

Importar Base de Datos

- Ir a Operaciones y hacer clic en Eliminar la base de datos (DROP)



Importar Base de Datos

- Seleccione la base de datos vacía, vaya a **Importar** y busque la base de datos desde **Seleccionar archivo**. Luego vaya hacia abajo y seleccione **Continuar**. Espere unos segundos hasta que se importe

The screenshot shows a web-based database management interface. At the top, a horizontal menu bar contains several icons and labels: 'Estructura', 'SQL', 'Buscar', 'Generar una consulta', 'Exportar', and 'Importar'. A red arrow points to the 'Importar' button. Below this menu, the main content area is titled 'Importando en la base de datos "labo1"'. Under the heading 'Archivo a importar:', there is a text box with instructions: 'El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido. A compressed file's name must end in .[format].[compression]. Example: .sql.zip'. Below this, a label 'Buscar en su ordenador:' is followed by a button labeled 'Seleccionar archivo' and the text 'Sin archivos seleccionados (Máximo: 2,048KB)'. A red arrow points to the 'Seleccionar archivo' button. Below this, there is a note 'También puede arrastrar un archivo en cualquier página.' and a label 'Conjunto de caracteres del archivo:' followed by a dropdown menu showing 'utf-8'. To the right of the main content area, there is a vertical dashed line with a downward arrow. Further right, there is a dropdown menu showing 'SQL'. Below this, the section 'Opciones específicas al formato:' contains a label 'Modalidad SQL compatible:' followed by a dropdown menu showing 'NONE'. Below this, there is a checked checkbox followed by the text 'No utilizar AUTO_INCREMENT con el valor 0'. At the bottom, there is a button labeled 'Continuar' with a red arrow pointing to it.

Importando en la base de datos "labo1"

Archivo a importar:

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.
A compressed file's name must end in .[format].[compression]. Example: .sql.zip

Buscar en su ordenador: **Seleccionar archivo** Sin archivos seleccionados (Máximo: 2,048KB)

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo: utf-8

SQL

Opciones específicas al formato:

Modalidad SQL compatible: NONE

☒ No utilizar AUTO_INCREMENT con el valor 0

Continuar

PHP y SQL para Bases de datos

SQL: Lenguaje para comunicarnos con la BD

- Si bien usaremos instrucciones PHP para interactuar con la Base de Datos, es necesario "hablarle" en un lenguaje que pueda "comprender".
- Este lenguaje es SQL (Structured Query Language, Lenguaje de Consulta Estructura).
- Es un lenguaje diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.
- Iremos viendo a lo largo de esta clase, cuales son las instrucciones o **sentencias** que deberemos escribir en este lenguaje, para poder comunicar lo que queremos hacer

SQL:

Tipos de Consultas:

- **Select** (de selección, con o sin where)
- **Insert** (de inserción)
- Update (de actualización)
- Delete (de eliminación)

SQL: Consulta SELECT

- SELECT se utiliza para Seleccionar (recuperar, "traer") datos de la tabla.

- Forma general:

SELECT campos de la tabla separados por comas
FROM tabla

- Ejemplo:

SELECT legajo, apellido, nombre
FROM empleados

SQL: Consulta SELECT

- También se pueden reemplazar todos los nombres de campos de una tabla con asterisco * (aunque no se recomienda traer siempre todos los campos, porque se usa recursos de más)
- Ejemplo:

SELECT *

FROM empleados

SQL: **SELECT ... WHERE ... =**

- Se puede establecer condiciones utilizando la cláusula **where**

SELECT campos de la tabla separados por comas

FROM tabla

WHERE campo1 = ? **AND** campo2 = ? **AND** ...

Donde los signos ? luego serán reemplazados por datos en variables de php. En este caso establecemos que los campos deberán ser iguales a los valores que le enviemos

PHP: Interactuar con una DB

- Posibles condiciones en WHERE:

Operadores	Lo que hace
campo > \$valor	campo mayor que \$valor
campo >= \$valor	campo Mayor o igual a \$valor
campo < \$valor	campo Menor que \$valor
campo <= \$valor	campo Menor o igual a \$valor
campo = \$valor	campo Igual a valor
campo <> \$valor	campo Distinto de \$valor

SQL: Consulta INSERT

- INSERT permite ingresar (o agregar), filas (o tuplas) a una base de datos, es decir, guardar los datos.
- Forma general:

INSERT INTO tabla (campo1, campo2, campo3, ...)
VALUES (?, ?, ?, ...);

- Ejemplo:

INSERT INTO alumnos (legajo, apellido, nombres, foto, correo)
VALUES (?, ?, ?, ?, ?);

SQL: Consulta INSERT

- Se podría omitir la lista de los nombres de los campos, pero sólo si se van a ingresar todos. **En el caso de tener un campo autoincrementado no se podría.**
- Forma general:

INSERT INTO tabla
VALUES (?, ?, ?, ...);

PHP: Instrucciones para Bases de Datos

- `mysqli_connect`: conectar con base de datos
- `mysqli_close`: desconectar
- `mysqli_prepare`: preparar consulta
- `mysqli_stmt_bind_param`: vincular variables de entrada
- `mysqli_stmt_execute`: ejecutar consulta
- `mysqli_stmt_bind_result`: vincular variables de salida
- `mysqli_stmt_fetch`: avanzar en los resultados

PHP: Instrucciones para Bases de Datos

- `mysqli_connect('servidor', 'usuario', contraseña, 'nombre_bd')`:
`mysqli`
 - Conectar con la base de datos
 - `$conexion = mysqli_connect('localhost', 'root', '', 'nombre');`
- `mysqli_close(mysqli $conexion)`: `bool`
 - Cerrar conexión (devuelve true o false si pudo o no cerrar la conexión.
 - `$estado = mysqli_close($conexion);` // puede no asignarse
 - Sólo se puede cerrar una conexión que esté abierta

PHP: Crear en archivo aparte, función para conectar:

```
<?php
function conectar ()
{
    $servidor = 'localhost';
    $usuario = 'root';
    $clave = '';
    $nombreBaseDato = 'labo2';

    set_error_handler (function() { // establece una función controladora de errores
        | throw new Exception("Error");
    });

    try { // intentará conectar con la base de datos
        | $conexion = mysqli_connect($servidor, $usuario, $clave, $nombreBaseDato);
        | }
    catch (Exception $e) { // atrapa el error
        | $conexion = false;
        | }

    return ($conexion);
}
```

PHP: Crear función para desconectar:

```
function desconectar ($miConexion) {  
    if ($miConexion) {  
        mysqli_close ($miConexion);  
        echo '<p>Desconexion exitosa</p>';  
    }  
}
```

PHP: Cómo usar funciones conectar y desconectar:

```
<?php
    require_once 'php/funciones.php';

    $conexion = conectar();
    if (!$conexion) {
        echo '<p>Error al conectar con la BD</p>'; // este mensaje siempre
    } else {
        echo '<p>Conexión exitosa</p>'; // sólo poner este mensaje para pruebas
        // el resto del código debe ir aquí

        desconectar($conexion);
    }
}
```


PHP: Comandos para Consultas a Base de datos

- Para minimizar daños por ataques a nuestra base de datos, se trabajará instrucciones para **consultas preparadas**
- **mysqli_prepare**(mysqli \$link, string \$query): mysqli_stmt
 - Prepara una sentencia SQL para su ejecución ("relaciona" la conexión con la consulta que se quiere hacer a la base de datos)

\$consulta = 'SELECT apellido, nombres FROM alumnos'; // armo la consulta SQL (es sólo texto hasta aquí)

\$sentencia = **mysqli_prepare**(\$conexion, \$consulta); // preparo consulta (queda fijada la estructura de la consulta)

PHP: Comandos para Consultas a Base de datos

En los casos que debemos involucrar variables php en la consulta, usaremos este comando:

- **mysqli_stmt_bind_param** (**mysqli_stmt** \$statement, **string** \$types, **mixed** \$var1, **mixed** \$var2, ...): **bool**
 - Agrega variables a una sentencia preparada

Los **\$types** del 2do parámetro pueden ser:

- 'i': la variable correspondiente es de tipo entero
- 'd': la variable correspondiente es de tipo double
- 's': la variable correspondiente es de tipo string
- Si es más de un parámetro se ponen de corrido 'ss', 'sid', ...

PHP: Comandos para Consultas a Base de datos

Por ejemplo, si haremos una consulta en la que se utilice un entero y un string:

```
$edad = $_POST['edad'];
```

```
$apel = $_POST['apellido'];
```

```
$consulta = 'SELECT edad, apellido, nombre  
            FROM alumnos  
            WHERE edad = ? AND apellido = ?';
```

```
$sentencia = mysqli_prepare($conexion, $consulta);
```

```
mysqli_stmt_bind_param ($sentencia, 'is', $edad, $apel);
```

PHP: Comandos para Consultas a Base de datos

- **mysqli_stmt_execute** (mysqli_stmt \$statement): bool
 - Ejecuta una sentencia preparada

\$consulta = 'SELECT apellido, nombres FROM alumnos'; // armo la consulta SQL (es sólo texto hasta aquí)

\$sentencia = mysqli_prepare(\$conexion, \$consulta); // preparo consulta

mysqli_stmt_execute(\$sentencia); // podemos o no usar lo que devuelve

PHP: Comandos para Consultas a Base de datos

- **mysqli_stmt_bind_result** (mysqli_stmt \$statement, tipo1 \$var1, tipo2 \$var2, tipo3 \$var3, ...): bool
 - Vincula campos de la tabla, en el conjunto de resultados, a variables de PHP, para poder mostrar los resultados.

```
$sentencia = mysqli_prepare($conexion, $consulta); // preparo  
consulta
```

```
mysqli_stmt_execute($sentencia); // podemos o no usar lo que devuelve  
mysqli_stmt_bind_result($sentencia, $apellido, $nombres); // vincula los  
resultados con variables
```

PHP: Comandos para Consultas a Base de datos

- **mysqli_stmt_fetch** (mysqli_stmt \$statement): bool
 - Avanza sobre los resultados obtenidos de la consulta y los asigna a las variables vinculadas.
 - Cuando se obtienen muchos resultados, se usará un **while**, para usar varias veces la instrucción, hasta finalizar los resultados

```
mysqli_stmt_execute($sentencia); // podemos o no usar lo que devuelve  
mysqli_stmt_bind_result($sentencia, $apellido, $nombres); // vincula los  
resultados con variables  
while (mysqli_stmt_fetch($sentencia)) { // avanza por los resultados  
    echo '<p>' . $apellido . ', ' . $nombres . '</p>'; // variables vinculadas  
}
```

PHP: Interactuar con una DB

- Qué hace el **mysqli_stmt_fetch()**?

Supongamos que la variables de resultado tienen lo siguiente:



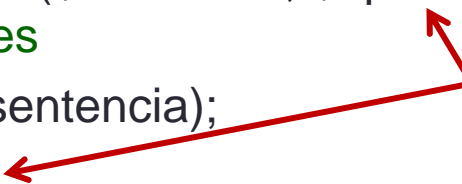
id	apellido	nombre	dni	correo	fecha
1	Acosta	Antonio	34555666	aacosta@gmail.com	1995-12-21
2	Bulacio	Fernando	30444555	fbulacio@gmail.com	2000-02-20
3	Correa	Mariana	38111222	mcorrea@gmail.com	2001-10-28
4	Ruiz	Myriam	27123456	mruiz@herrera.unt.edu.ar	2021-08-24
5	Ruiz	Alejandro	39365978	aruiz@gmail.com	1996-10-14

- Al principio, no podemos tomar los datos, ya que apunta a la parte superior de resultados (flecha llena).
- Lo que hace `mysqli_stmt_fetch` cada vez que lo llamamos, es ubicarse en la siguiente fila dentro del conjunto que recibió (flecha de trazo roja), permitiendo leer los datos de la misma.

PHP: Cómo saber si hubo resultados en un SELECT?

Forma 1: Luego del `bind_result`, se hace el `fetch` y se controla que haya algo en alguna de las variables vinculadas. Trabaja sin búfer.

```
mysqli_stmt_execute($sentencia); // podemos o no usar lo que devuelve
mysqli_stmt_bind_result($sentencia, $apellido, $nombres); // vincula los
resultados con variables
mysqli_stmt_fetch($sentencia);
if (!empty($apellido)) {
    do {
        // usar las variables
    } while (mysqli_stmt_fetch($sentencia));
}
```



PHP: Cómo saber si hubo resultados en un SELECT?

Forma 1:

```
$sql = 'SELECT titulo, temporadas, fecha_estreno, productora
      FROM series
      WHERE productora = ?'; // texto en lenguaje SQL
$sentencia = mysqli_prepare($conexion, $sql); // dejo fijada
mysqli_stmt_bind_param($sentencia, 's', $produc); // vinculo variable
mysqli_stmt_execute ($sentencia); // tengo las respuestas
mysqli_stmt_bind_result($sentencia, $titu, $temporas, $fecha, $productora);
if (!empty($titu)) {
    do {
        echo '<p> Título: ' . $titu . '</p>';
        echo '<p> Temporadas: ' . $temporas . '</p>';
        echo '<p> Fecha Estreno: ' . $fecha . '</p>';
        echo '<p> Productora: ' . $productora . '</p>';
        echo '<hr>';
    } while (mysqli_stmt_fetch($sentencia));
}
```

PHP: Cómo saber si hubo resultados en un SELECT?

Forma 2: Luego del `bind_result`, se utilizan dos funciones:

`mysqli_stmt_store_result(mysqli_stmt $statement): bool`

- Almacena un conjunto de resultados en un búfer interno

`mysqli_num_rows(mysqli_result $result): int`

- Obtiene la cantidad de filas de, resultado de una consulta

PHP: Cómo saber si hubo resultados en un SELECT?

Forma 2: Ejemplo

```
mysqli_stmt_execute($sentencia);
mysqli_stmt_bind_result($sentencia, $apellido, $nombres);
mysqli_stmt_store_result($sentencia);
$cantidadFilas = mysqli_num_rows($sentencia);
if ($cantidadFilas > 0) {
    while(mysqli_stmt_fetch($sentencia)) {
        // usar las variables
    }
}
```

PHP: Cómo saber si hubo resultados en un SELECT?

Diferencia entre forma 1 y 2:

Forma 1: Los datos se transfieren sin búfer, lo que puede disminuir el rendimiento (pero reduce el costo de memoria).

Forma 2: Almacena un conjunto de resultados en un búfer interno, mejor rendimiento (mayor costo en memoria)

PHP: Interactuar con una DB

- Pasando en limpio, el Orden de Pasos es:
 - 1) Conectarse al servidor de DB.
 - 2) Preparar una Consulta (hoy vemos sólo 2 tipos)
 - 3) Vincular variables (si las tuviera).
 - 4) Ejecutar Consulta.
 - 5) Mostrar resultados (si los hubiera).
 - 6) Desconectarse de la DB.

INSERT para Fecha

- En qué formato insertar en una tabla las Fechas?

fecha_publici	foto_destacada
2019-08-22	curso-de-posgrado-misiones-espaciales.pn
2019-10-09	curso-de-posgrado-misiones-espaciales.pn
2019-10-09	homero.png



- Debe guardarse con formato 'año-mes-día'. Ej:

```
$fecha = date('Y-m-d'); // fecha actual
```

INSERT de clave

- Cómo encriptar una clave?
 - Usar funciones que limpien lo que venga de un formulario (veremos en clases posteriores)
 - Las claves deben utilizar alguna **función** que las encripte (que no las deje tan visibles):
- `sha1(string $str [, bool $raw_output = false]): string`
- Ej: `$encriptada = sha1($clave)`

INSERT de Clave encriptada

```
<?php
if (!empty($_POST['usuario']) && !empty($_POST['clave'])) {
    include 'funciones.php';
    $conexion = conectar();
    if ($conexion) {
        $usu = $_POST['usuario'];
        $clave = sha1($_POST['clave']);
        $consulta = 'INSERT INTO usuarios(usuario, clave)
        |         |         | VALUES (?, ?)';
        $sentencia = mysqli_prepare($conexion, $consulta); // preparo consulta
        mysqli_stmt_bind_param($sentencia, 'ss', $usu, $clave);
        $q = mysqli_stmt_execute($sentencia); // ejecuto consulta

        if ($q) {
            echo '<p>guardado exitoso</p>';
        } else {
            echo '<p>error al guardar</p>';
        }
        desconectar($conexion);
    }
}
```



INSERT de Clave

Ingreso de Usuarios

Usuario:

Clave:

+ Opciones



id_usuario

usuario

clave



Editar



Copiar



Borrar

1


mruiz










7ce0359f12857f2a90c7de465f40a95f01cb5da9



Consultas INSERT

- Cómo se guarda con otro tipo de Encriptación la palabra admin?:



<input type="checkbox"/>	 Editar	 Copiar	 Borrar	13	jsilva	admin	Joaquín	Silva	jsilva@hotmail.com
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	14	josilva	21232f297a57a5a743894a0e4a801fc3	Joaquín	Silva	josilva@gmail.com
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	15	joasilva	d033e22ae348aeb5660fc2140aec35850c4da997	Joaquín	Silva	joasilva@yahoo.com.ar

- El 1er caso la clave no está encriptada
- En 2do caso se encriptó con md5(): longitud de 16 caracteres
- En 3er caso se encriptó con sha1(): longitud de 40 caracteres

INSERT

- Aunque ninguna función hará imposible que descifren nuestra clave, tampoco la dejemos sin encriptar, así por lo menos les tome tiempo



Ejemplo de Insert:

- **formulario.html** envía datos que serán guardados por el archivo **guardar.php**.
- En **guardar.php** controlar que recibe los datos que son obligatorios
- Si los tiene, incluye el archivo **funciones.php** y se conecta a la base de datos. Entonces inserta los datos enviados

Ejemplo de Insert:

← → ↻ ⓘ localhost/clase06/05-insert/formulario.html

Ingreso de Clientes

Apellido:

Nombre:

Fecha Nacimiento:
 

Foto:
 Programador.jpg

Ejemplo de Insert:

- guardar.php

```
<?php
if (!empty($_FILES['foto']['size']) && !empty($_POST['apellido']) &&
    !empty($_POST['nombre']) && !empty($_POST['fechaNac'])) {
    include 'funciones.php';
    include '../html/encabezado.html';
    $conexion = conectar();
    if ($conexion) { // controla que se haya creado la conexión
        $apellido = $_POST['apellido'];
        $nombre = $_POST['nombre'];
        $fecha = $_POST['fechaNac'];
        $foto_nombre = $_FILES['foto']['name'];
        $consulta = 'INSERT INTO empleados(apellido, nombre, sueldo, fecha_nac, foto)
            VALUES (?, ?, NULL, ?, ?)';

        $sentencia = mysqli_prepare($conexion, $consulta); // preparo consulta
        mysqli_stmt_bind_param($sentencia, 'ssss', $apellido, $nombre, $fecha, $foto_nombre);
        $q = mysqli_stmt_execute($sentencia); // ejecuto consulta

        // falta mover foto
    }
}
```

Si el dato no existe en ese momento, guardamos NULL (la tabla debe permitirlo)

Ejemplo de Insert:

- En caso de no poseer los datos del formulario, se puede redireccionar con header()

```
// falta mover foto
if ($q) {
    echo '<p>guardado exitoso</p>';
} else {
    echo '<p>error al guardar</p>';
}
desconectar($conexion);
}
} else {
    echo '<p>Debe Ingresar datos en el formulario</p>';
    header( 'refresh:3;url=../formulario.html' ); // abre otra página
}

include '../html/pie.html';

?>
```

Ejemplo de Insert:

✓ Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0013 segundos.)

```
SELECT * FROM `empleados`
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 ▼ Filtrar filas: Sort by key:

+ Opciones

				legajo	apellido	nombre	sueldo	fecha_nac	foto
<input type="checkbox"/>	Editar	Copiar	Borrar	1	Gola	Mayra	57864.25	1997-07-07	mgola.jpg
<input type="checkbox"/>	Editar	Copiar	Borrar	3	Diaz	Jorge	NULL	2010-10-25	Programador.jpg

PHP: Ejemplo de Select

- Cuando se ejecute la consulta de SELECT, se almacenará en una variable, muchas filas, una sola o ninguna, dependiendo de si se establecen condiciones. Por ejemplo si establecemos que el campo apellido es igual a 'Ruiz', nos devolvería las filas de la 2da imagen:

←T→				id	apellido	nombre	dni	correo	fecha
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	Acosta	Antonio	34555666	aacosta@gmail.com	1995-12-21
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	Bulacio	Fernando	30444555	fbulacio@gmail.com	2000-02-20
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	Correa	Mariana	38111222	mcorrea@gmail.com	2001-10-28
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	4	Ruiz	Myriam	27123456	mruiz@herrera.unt.edu.ar	2021-08-24
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5	Ruiz	Alejandro	39365978	aruiz@gmail.com	1996-10-14

+ Opciones									
←T→			id	apellido	nombre	dni	correo	fecha	
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	4	Ruiz	Myriam	27123456	mruiz@herrera.unt.edu.ar	2021-08-24
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5	Ruiz	Alejandro	39365978	aruiz@gmail.com	1996-10-14

SELECT de Clave

- Cómo usar SELECT una clave encriptada?

```
if (!empty($_POST['usuario']) && !empty($_POST['clave'])) {  
    include 'funciones.php';  
    $conexion = conectar();  
    if ($conexion) {  
        $usuForm = $_POST['usuario'];  
        $claveForm = sha1($_POST['clave']);  
        $consulta = 'SELECT usuario, clave  
                    FROM usuarios  
                    WHERE usuario = ? AND clave = ?';  
        $sentencia = mysqli_prepare($conexion, $consulta); // preparo consulta  
        mysqli_stmt_bind_param($sentencia, 'ss', $usuForm, $claveForm);  
        $q = mysqli_stmt_execute($sentencia); // ejecuto consulta  
        mysqli_stmt_bind_result($sentencia, $usuarioBD, $claveBD);  
  
        if ($q) {  
            mysqli_stmt_fetch($sentencia);  
            echo '<p>Usuario y Clave Correctos -> usuario: ' . $usuarioBD . '</p>'; // control visual
```

PHP: Mostrar resultados

- Ejemplo sin WHERE:

```
<article>
    <?php
        include 'php/funciones.php';
        $conexion = conectar();
        $consulta = 'SELECT apellido, nombre, fecha_nac FROM empleados';
        $sentencia = mysqli_prepare($conexion, $consulta); // preparo consulta
        $q = mysqli_stmt_execute($sentencia); // ejecuto consulta
        mysqli_stmt_bind_result($sentencia, $apellido, $nombre, $fecha);

        if ($q) {
            while(mysqli_stmt_fetch($sentencia)) {
                $fecha_nac = date_create($fecha);
                $fecha_nac = date_format($fecha_nac, 'd-m-Y');
                echo '<p>Nombre: ' . $apellido . ', ' . $nombre . '</p>';
                echo '<p>Fecha de Nac.: ' . $fecha_nac . '</p><hr>';
            }
        } else {
            echo '<p>No hubo resultados</p>';
        }
    }
```

PHP: Interactuar con una DB

- Ejemplo sin WHERE:

← → ↻ ⓘ localhost/clase06/02-insert-select/php/mostrar.php	
Empleados	
SELECT apellido, nombre, fecha_nac FROM empleados	
Nombre: Ruiz, Myriam	
Fecha de Nac.: 29-12-1979	
<hr/>	
Nombre: Acosta, Juan	
Fecha de Nac.: 14-06-1981	
<hr/>	
Nombre: Ledesma, Manuel	
Fecha de Nac.: 27-05-1985	
<hr/>	
Nombre: Romero, Mariana	
Fecha de Nac.: 17-10-1986	
<hr/>	

PHP: Mostrar resultados

- Ejemplo con WHERE:

```
<article>
  <?php
    include 'php/funciones.php';
    $conexion = conectar();
    $fechaControl = '1982-03-20';
    $consulta = 'SELECT apellido, nombre, fecha_nac FROM empleados
                WHERE fecha_nac > ?';
    $sentencia = mysqli_prepare($conexion, $consulta); // preparo consulta
    mysqli_stmt_bind_param($sentencia, 's', $fechaControl);
    $q = mysqli_stmt_execute($sentencia); // ejecuto consulta
    mysqli_stmt_bind_result($sentencia, $apellido, $nombre, $fecha);

    if ($q) {
      while(mysqli_stmt_fetch($sentencia)) {
        $fecha_nac = date_create($fecha);
        $fecha_nac = date_format($fecha_nac, 'd-m-Y');
        echo '<p>Nombre: ' . $apellido . ', ' . $nombre . '</p>';
        echo '<p>Fecha de Nac.: ' . $fecha_nac . '</p><hr>';
      }
    }
  </?php>
</article>
```

PHP: Interactuar con una DB

- Ejemplo con WHERE:

Empleados

Nombre: Ledesma, Manuel

Fecha de Nac.: 27-05-1985

Nombre: Romero, Mariana

Fecha de Nac.: 17-10-1986

Nombre: Diaz, Jorge

Fecha de Nac.: 25-10-2002

Nombre: Franco, Myriam

Fecha de Nac.: 09-11-2023

SELECT ... WHERE ... LIKE

PHP: Consultas SELECT con WHERE LIKE

- **LIKE** permite hacer una búsqueda por coincidencia parcial.
- Si bien la instrucción en SQL tiene la forma:

LIKE '%cadena%'

la consulta preparada sólo tendrá un signo **?** y se armará el resto por separado.

```
$consulta = 'SELECT campos  
            FROM tabla  
            WHERE campo LIKE ?';
```


PHP: Consultas SELECT con WHERE LIKE

- Ejemplo que busca parte del apellido en el medio:

```
$ape = $_POST['apellido'];
```

```
$buscado = '%' . $ape . '%';
```

```
$consulta = 'SELECT apellido, nombre
```

```
FROM clientes
```

```
WHERE apellido LIKE ?';
```

```
mysqli_stmt_bind_param($sentencia, "s", $buscado);
```

PHP: Interactuar con una DB

- Posibles condiciones en WHERE:

Operadores	Lo que hace
campo > \$valor	campo mayor que \$valor
campo >= \$valor	campo Mayor o igual a \$valor
campo < \$valor	campo Menor que \$valor
campo <= \$valor	campo Menor o igual a \$valor
campo = \$valor	campo Igual a valor
campo <> \$valor	campo Distinto de \$valor
campo BETWEEN \$valor1 AND \$valor2	campo está entre \$valor1 y \$valor2
campo LIKE '%\$valor%'	Campo similar a '%\$valor%'

PHP: Consultas SELECT con WHERE like

Ej: SELECCIONAR los productos cuyos nombres **empiezan** con Ar

Tabla Productos:

id	nombre	stock	fecha_vencimiento	categoria	foto
1	Arroz Marolio 1kg	32	2022-03-10	Granos	Arroz Marolio 1kg.jpg
2	Leche Entera Milkaut 1lt	50	2021-01-26	Lacteos	Leche Entera Milkaut 1lt.jpg
3	Queso Cremoso La Lacteo	10	2020-12-20	Lácteos	Queso Cremoso La Lacteo.jpg
4	Arroz Largo Fino 00000 Molinos Ala 1 Kg	254	2024-12-31	Granos	NULL

Buscar Producto sabiendo como inicia su nombre

Nombre:



mostrar-where-like-inicio.php

PHP: Consultas SELECT con WHERE LIKE

- mostrar-where-like-inicio.php

```
if (!empty($_POST['nombre'])) {  
    $conexion = conectar();  
    $nombre = $_POST['nombre'];  
    $consulta = 'SELECT nombre, stock, categoria  
                FROM productos  
                WHERE nombre LIKE ?';  
  
    $sentencia = mysqli_prepare($conexion, $consulta); //  
    $nomb = $nombre . '%'; ←  
    mysqli_stmt_bind_param($sentencia, "s", $nomb);  
    mysqli_stmt_execute($sentencia); // ejecuto consulta
```

PHP: Consultas SELECT con WHERE LIKE

- mostrar-where-like-inicio.php

```
mysqli_stmt_execute($sentencia); // ejecuto consulta
mysqli_stmt_bind_result($sentencia, $nombre, $stock, $categoria); // asocio variables
mysqli_stmt_store_result($sentencia);
$cantFilas = mysqli_stmt_num_rows($sentencia);
if ($cantFilas > 0) {
    while (mysqli_stmt_fetch($sentencia)) { // avanzo por los resultados
        echo '<h2>Nombre: ' . $nombre . '</h2>';
        echo '<p> Categoría: ' . $categoria . '</p>';
        echo '<p> Stock: ' . $stock . '</p>';
        echo '<hr>';
    }
} else {
    echo '<h2>No hubo resultados</h2>';
}
}
include_once '../html/pie.html';
```

PHP: Consultas SELECT con WHERE LIKE

- `mostrar-where-like-inicio.php`

Resultado:

Productos Seleccionados

Nombre: Arroz Marolio 1kg

Categoría: Granos

Stock:32

Nombre: Arroz Largo Fino 00000 Molinos Ala 1 Kg

Categoría: Granos

Stock:254

PHP: Consultas SELECT con WHERE like

Ej: SELECCIONAR los productos cuyos nombres que *terminan* con 'lt'

Buscar Producto por terminación de nombre

Nombre:



mostrar-where-like-fin.php

```
$nomb = '%' . $nombre;
```

PHP: Consultas SELECT con WHERE like

Ej: SELECCIONAR los productos cuyos nombres que *terminan* con 'lt'

Productos Seleccionados

Nombre: Leche Entera Milkaut 1lt

Categoría: Lacteos

Stock:50

Exportar la Base de Datos

1. Ingresar a <http://localhost/phpmyadmin/>
2. Seleccionar la base de datos haciendo clic sobre la misma
3. Hacer clic en Exportar
4. Luego no tocar nada y hacer clic en Continuar
5. Se descargará un archivo con el nombre de la base de datos y extensión **sql**. Eso deberemos enviar para quien quiera continuar trabajando en el sitio y para presentarlo

Exportar base de datos

3. Hacer clic en Exportar



Exportar Base de Datos

4. Luego no tocar nada y hacer clic en Continuar

