**TALLER 2**

**CRISTHIAN URREGO SALAZAR**

**CARLOS LONDOÑO**

**CENTRO DE ESTUDIOS TECNOLÓGICOS DEL NORTE DEL VALLE COTECNOVA**

**TÉCNICO PROFESIONAL EN PROGRAMACIÓN DE APLICACIONES INFORMÁTICAS**

**CARTAGO VALLE 2016**

a.¿Cómo crear una base de datos en MYSQL?

Cada conjunto de relaciones que componen un modelo completo forma una base de datos. Desde el punto de vista de SQL, una base de datos es sólo un conjunto de relaciones (o tablas), y para organizarlas o distinguirlas se accede a ellas mediante su nombre. A nivel de sistema operativo, cada base de datos se guarda en un directorio diferente.

Debido a esto, crear una base de datos es una tarea muy simple. Claro que, en el momento de crearla, la base de datos estará vacía, es decir, no contendrá ninguna tabla.

Vamos a crear y manipular nuestra propia base de datos, al tiempo que nos familiarizamos con la forma de trabajar de **MySQL**.

Para empezar, crearemos una base de datos para nosotros solos, y la llamaremos "prueba". Para crear una base de datos se usa una sentencia [CREATE DATABASE](http://mysql.conclase.net/curso/?sqlsen=CREATE_DATABASE):

mysql> CREATE DATABASE prueba;

**Query OK, 1 row affected (0.03 sec)**

**mysql>**

Podemos averiguar cuántas bases de datos existen en nuestro sistema usando la sentencia [SHOW DATABASES](http://mysql.conclase.net/curso/?sqlsen=SHOW_DATABASES):

mysql> SHOW DATABASES;

+--------------------+

| Database |

+--------------------+

| mysql |

| prueba |

| test |

+--------------------+

3 rows in set (0.00 sec)

mysql>

A partir de ahora, en los próximos capítulos, trabajaremos con esta base de datos, por lo tanto la seleccionaremos como base de datos por defecto. Esto nos permitirá obviar el nombre de la base de datos en consultas. Para seleccionar una base de datos se usa el comando [USE](http://mysql.conclase.net/curso/?sqlsen=USE), que no es exactamente una sentencia SQL, sino más bien de una opción de **MySQL**:

mysql> USE prueba;

Database changed

mysql>

B.¿Cómo conectar MYSQL con Php?

Este simple ejemplo muestra como conectar, ejecutar una consulta, imprimir las filas resultantes y desconectar a una base de datos MySQL.

Ejemplo #1 Ejemplo general de la extensión MySQL

<?php  
// Conectando, seleccionando la base de datos  
$link = mysql\_connect('mysql\_host', 'mysql\_user', 'mysql\_password')  
    or die('No se pudo conectar: ' . mysql\_error());  
echo 'Connected successfully';  
mysql\_select\_db('my\_database') or die('No se pudo seleccionar la base de datos');  
  
// Realizar una consulta MySQL  
$query = 'SELECT \* FROM my\_table';  
$result = mysql\_query($query) or die('Consulta fallida: ' . mysql\_error());  
  
// Imprimir los resultados en HTML  
echo "<table>\n";  
while ($line = mysql\_fetch\_array($result, MYSQL\_ASSOC)) {  
    echo "\t<tr>\n";  
    foreach ($line as $col\_value) {  
        echo "\t\t<td>$col\_value</td>\n";  
    }  
    echo "\t</tr>\n";  
}  
echo "</table>\n";  
  
// Liberar resultados  
mysql\_free\_result($result);  
  
// Cerrar la conexión  
mysql\_close($link);  
?>

C.¿Cómo añadir registros a una tabla desde php?

Una vez creada la tabla *clientes* en nuestra base de datos *ejemplo*, el paso siguiente sea llenarla con registros. Para ello vamos a ver este artículo, en el que se reciben datos desde un formulario y luego se insertan con PHP en la base de datos, en la tabla adecuada.

Los datos del registro pueden ser recogidos, por ejemplo, a partir de un formulario. Aquí os proponemos un simple documento HTML que recoge los datos y los envía a una página PHP que se encarga de procesarlos:

<HTML>  
<HEAD>  
<TITLE>Insertar.html</TITLE>  
</HEAD>  
<BODY>  
<div align="center">  
<h1>Insertar un registro</h1>  
<br>  
<FORM METHOD="POST" ACTION="insertar.php">  
Nombre<br>  
<INPUT TYPE="TEXT" NAME="nombre"><br>  
Teléfono<br>  
<INPUT TYPE="TEXT" NAME="telefono"><br>  
<INPUT TYPE="SUBMIT" value="Insertar">  
</FORM>  
</div>  
</BODY>  
</HTML>

Llegados a la página destino del formulario (*insertar.php*), lo primero que habrá que hacer es establecer un vínculo entre el programa y la base de datos. Esta conexión se lleva a cabo con la función *mysql\_connect*. A continuación, deberemos generar una orden de inserción del registro en lenguaje SQL. Esta orden será ejecutada por medio de la función *mysql\_db\_query*. En esta función especificaremos primeramente la base de datos sobre la que queremos actuar y a continuación introduciremos la sentencia SQL:

<HTML>  
<HEAD>  
<TITLE>Insertar.php</TITLE>  
</HEAD>  
<BODY>  
<?  
//Conexion con la base  
mysql\_connect("localhost","tu\_user","tu\_password");   
  
//selección de la base de datos con la que vamos a trabajar   
mysql\_select\_db("mi\_base\_datos");   
  
//Ejecucion de la sentencia SQL  
mysql\_query("insert into clientes (nombre,telefono) values ('$nombre','$telefono')");  
?>  
<h1><div align="center">Registro Insertado</div></h1>  
<div align="center"><a href="lectura.php">Visualizar el contenido de la base</a></div>  
</BODY>  
</HTML>

D.¿cómo actualizar registros a una tabla desde php?

Para mostrar cómo se actualiza un registro presente en nuestra base de datos, vamos a hacerlo a partir de un caso un poco más complejo para que empecemos a familiarizarnos con estas operaciones. Realizaremos un par de scripts que permitan cambiar el numero de teléfono de las distintas personas presentes en nuestra base. El nombre de estas personas, así como el nuevo numero de teléfono, serán recogidos por medio de un formulario.

El archivo del formulario va a ser esta vez un script PHP en el que efectuaremos una llamada a nuestra base de datos para construir un menú desplegable donde aparezcan todos los nombres. La cosa quedaría así:

<HTML>  
<HEAD>  
<TITLE>Actualizar1.php</TITLE>  
</HEAD>  
<BODY>  
<div align="center">  
<h1>Actualizar un registro</h1>  
<br>  
<?  
//Conexion con la base  
mysql\_connect("localhost","tu\_user","tu\_password");  
  
//selección de la base de datos con la que vamos a trabajar   
mysql\_select\_db("mi\_base\_datos");   
  
echo '<FORM METHOD="POST" ACTION="actualizar2.php">Nombre<br>';  
  
//Creamos la sentencia SQL y la ejecutamos  
$sSQL="Select nombre From clientes Order By nombre";  
$result=mysql\_query($sSQL);  
  
echo '<select name="nombre">';  
  
//Generamos el menu desplegable  
while ($row=mysql\_fetch\_array($result))  
{echo '<option>'.$row["nombre"];}  
?>  
</select>  
<br>  
Teléfono<br>  
<INPUT TYPE="TEXT" NAME="telefono"><br>  
<INPUT TYPE="SUBMIT" value="Actualizar">  
</FORM>  
</div>  
  
</BODY>  
</HTML>

La manera de operar para construir el menú desplegable es la misma que para visualizar la tabla. De nuevo empleamos un bucle *while* en combinación con la función *mysql\_fetch\_array* lo que nos permite mostrar cada una de las opciones.

E.¿cómo eliminar registros a una tabla desde php?

Otra de las operaciones elementales que se pueden realizar sobre una base de datos es borrar un registro. Para hacerlo, SQL nos propone sentencias del tipo *Delete*. Veámoslo con un ejemplo aplicado a nuestra agenda.

Cabe señalar que primero debemos seleccionar el registro que se desea borrar y luego realizar el borrado propiamente dicho. Para ello crearemos un menú desplegable dinámico, donde se podrá seleccionar el elemento que se desea borrar. Luego se pasará a una página PHP una referencia al elemento seleccionado, para borrarlo de la base de datos.

<HTML>  
<HEAD>  
<TITLE>Borrar1.php</TITLE>  
</HEAD>  
<BODY>  
<div align="center">  
<h1>Borrar un registro</h1>  
<br>  
  
<?  
//Conexion con la base  
mysql\_connect("localhost","tu\_user","tu\_password");  
  
//selección de la base de datos con la que vamos a trabajar   
mysql\_select\_db("mi\_base\_datos");   
  
echo '<FORM METHOD="POST" ACTION="borrar2.php">Nombre<br>';  
  
//Creamos la sentencia SQL y la ejecutamos  
$sSQL="Select nombre From clientes Order By nombre";  
$result=mysql\_query($sSQL);  
  
echo '<select name="nombre">';  
  
//Mostramos los registros en forma de menú desplegable  
while ($row=mysql\_fetch\_array($result))  
{echo '<option>'.$row["nombre"];}  
mysql\_free\_result($result)  
?>

</select>  
<br>  
<INPUT TYPE="SUBMIT" value="Borrar">  
</FORM>  
</div>  
  
</BODY>  
</HTML>

El siguiente paso es hacer efectiva la operación a partir de la ejecución de la sentencia SQL que construimos a partir de los datos del formulario:

<HTML>  
<HEAD>  
<TITLE>Borrar2.php</TITLE>  
</HEAD>  
<BODY>  
<?  
//Conexion con la base  
mysql\_connect("localhost","tu\_user","tu\_password");  
  
//selección de la base de datos con la que vamos a trabajar   
mysql\_select\_db("mi\_base\_datos");

//Creamos la sentencia SQL y la ejecutamos  
$sSQL="Delete From Clientes Where nombre='$nombre'";  
mysql\_query($sSQL);  
?>

<h1><div align="center">Registro Borrado</div></h1>  
<div align="center"><a href="lectura.php">Visualizar el contenido de la base</a></div>  
  
</BODY>  
</HTML>

f.¿como buscar registros a una tabla desde php?

Dentro de una base de datos, organizada por tablas, la selección de una tabla entera o de un cierto numero de registros resulta una operación rutinaria.

Aquí os mostramos una forma bastante clásica de mostrar en pantalla a partir de un bucle los registros seleccionados por una sentencia SQL:

<HTML>  
<HEAD>  
<TITLE>lectura.php</TITLE>  
</HEAD>  
<BODY>  
<h1><div align="center">Lectura de la tabla</div></h1>  
<br>  
<br>  
<?  
//Conexion con la base  
mysql\_connect("localhost","tu\_user","tu\_password");   
  
//selección de la base de datos con la que vamos a trabajar   
mysql\_select\_db("mi\_base\_datos");   
  
//Ejecutamos la sentencia SQL  
$result=mysql\_query("select \* from clientes");  
?>  
<table align="center">  
<tr>  
<th>Nombre</th>  
<th>Teléfono</th>  
</tr>  
<?  
//Mostramos los registros  
while ($row=mysql\_fetch\_array($result))  
{  
echo '<tr><td>'.$row["nombre"].'</td>';  
echo '<td>'.$row["telefono"].'</td></tr>';  
}  
mysql\_free\_result($result)  
?>  
</table>  
  
<div align="center">  
<a href="insertar.html">Añadir un nuevo registro</a><br>  
<a href="actualizar1.php">Actualizar un registro existente</a><br>  
<a href="borrar1.php">Borrar un registro</a><br>  
</div>  
  
</BODY>  
</HTML>

[**Ejecutar script**](http://www.desarrolloweb.com/articulos/ejemplos/php/lectura.php)

Los pasos a realizar son, en un principio, los vistos para la inserción de un registro: Conexión a la base y ejecución de la sentencia. Esta vez, la información de dicha ejecución será almacenada en una variable (*$result*).

El siguiente paso será plasmar en pantalla la información recogida en $result. Esto lo haremos mediante la función *mysql\_fetch\_array* que devuelve una variable array con los contenidos de un registro a la vez que se posiciona sobre el siguiente. El bucle *while* nos permite leer e imprimir secuencialmente cada uno de los registros.

La función *mysql\_free\_result* se encarga de liberar la memoria utilizada para llevar a cabo la consulta. Aunque no es necesaria su utilización, resulta altamente aconsejable.

G.¿ que hace la función require en php?

require es idéntico a include excepto que en caso de fallo producirá un error fatal de nivel E\_COMPILE\_ERROR. En otras palabras, éste detiene el script mientras que include sólo emitirá una advertencia (E\_WARNING) lo cual permite continuar el script.

Remember, when using require that it is a statement, not a function. It's not necessary to write:  
<?php  
require('somefile.php');  
?>  
  
The following:  
<?php  
require 'somefile.php';  
?>  
  
Is preferred, it will prevent your peers from giving you a hard time and a trivial conversation about what require really is.

h.¿ que hace la función require\_once en php?

La sentencia require\_once es idéntica a require excepto que PHP verificará si el archivo ya ha sido incluido y si es así, no se incluye (require) de nuevo.

Si el código se ejecuta en varios servidores con diferentes ambientes (donde las poblaciones de las secuencias de comandos de ejecución) la siguiente idea pueden ser útiles para usted:

No le dé ruta absoluta para incluir archivos en el servidor.

Dinámicamente el cálculo de la ruta completa (ruta absoluta)

consejos:

Use una combinación de dirname (\_\_ FILE\_\_) y las llamadas posteriores a sí mismo hasta llegar a la casa de su 'index.php'. A continuación, fije esta variable (que contiene la ruta de acceso) a los archivos incluidos.

Uno de mi ejemplo típico es:

<?php   
define('\_\_ROOT\_\_', dirname(dirname(\_\_FILE\_\_)));   
require\_once(\_\_ROOT\_\_.'/config.php');   
?>   
  
en lugar de:

<?php require\_once('/var/www/public\_html/config.php'); ?>

Después de esto, si copia pegar sus códigos a otros servidores, que seguirá funcionando, sin necesidad de re-configuraciones adicionales.

i.¿ que hace la función include en php?

Los archivos son incluidos con base en la ruta de acceso dada o, si ninguna es dada, el [include\_path](http://php.net/manual/es/ini.core.php#ini.include-path) especificado. Si el archivo no se encuentra en el [include\_path](http://php.net/manual/es/ini.core.php#ini.include-path), include finalmente verificará en el propio directorio del script que hace el llamado y en el directorio de trabajo actual, antes de fallar. El constructor include emitirá una advertencia si no puede encontrar un archivo, éste es un comportamiento diferente al de [require](http://php.net/manual/es/function.require.php), el cual emitirá un error fatal..

Si una ruta es definida — ya sea absoluta (comenzando con una letra de unidad o \ en Windows o / en sistemas Unix/Linux) o relativa al directorio actual (comenzando con . o ..) — el [include\_path](http://php.net/manual/es/ini.core.php" \l "ini.include-path) será ignorado por completo. Por ejemplo, si un nombre de archivo comienza con../, el intérprete buscará en el directorio padre para encontrar el archivo solicitado.

Para más información sobre como PHP maneja la inclusión de archivos y la ruta de accesos para incluir, ver la documentación de [include\_path](http://php.net/manual/es/ini.core.php#ini.include-path).

Cuando se incluye un archivo, el código que contiene hereda el ámbito de las variables de la línea en la cual ocurre la inclusión. Cualquier variable disponible en esa línea del archivo que hace el llamado, estará disponible en el archivo llamado, desde ese punto en adelante. Sin embargo, todas las funciones y clases definidas en el archivo incluido tienen el ámbito global.

**Ejemplo #1 Ejemplo básico de** include

vars.php  
<?php  
  
$color = 'verde';  
$fruta = 'manzana';  
  
?>  
  
test.php  
<?php  
  
echo "Una $fruta $color"; // Una  
  
include 'vars.php';  
  
echo "Una $fruta $color"; // Una manzana verde  
  
?>

Si la inclusión ocurre al interior de una función dentro del archivo que hace el llamado, entonces todo el código contenido en el archivo llamado se comportará como si hubiera sido definida dentro de esa función. Por lo tanto, seguirá el ámbito de las variables de esa función. Una excepción a esta regla son las constantes mágicas las cuales son evaluadas por el intérprete antes que ocurra la inclusión.

**Ejemplo #2 Incluyendo dentro de funciones**

<?php  
  
function foo()  
{  
    global $color;  
  
    include 'vars.php';  
  
    echo "Una $fruta $color";  
}  
  
/\* vars.php está en el ámbito de foo() así que \*  
\* $fruta NO está disponible por fuera de éste  \*  
\* ámbito. $color sí está porque fue declarado \*  
\* como global.                                 \*/  
  
foo();                      // Una manzana verde  
echo "Una $fruta $color";   // Una verde  
  
?>

j.¿ que hace la función include\_once en php?

La sentencia include\_once incluye y evalúa el fichero especificado durante la ejecución del script. Tiene un comportamiento similar al de la sentencia [include](https://php.net/manual/es/function.include.php), siendo la única diferencia de que si el código del fichero ya ha sido incluido, no se volverá a incluir, e include\_once devolverá **TRUE**. Como su nombre indica, el fichero será incluido solamente una vez.

include\_once se puede utilizar en casos donde el mismo fichero podría ser incluido y evaluado más de una vez durante una ejecución particular de un script, así que en este caso, puede ser de ayuda para evitar problemas como la redefinición de funciones, reasignación de valores de variables, etc.

Véase la documentación de [include](https://php.net/manual/es/function.include.php) para más información sobre cómo funciona esta función.

**Nota**:

Con PHP 4, la funcionalidad de \_once difiere entre sistemas operativos que ignoran las mayúsculas/minúsculas (como Windows). Por ejemplo:

**Ejemplo #1** include\_once **con un SO que no distinque mayúsculas de minúsculas en PHP 4**

<?php  
include\_once "a.php"; // esto incluirá a.php  
include\_once "A.php"; // esto incluirá a.php ¡otra vez! (sólo PHP 4)  
?>

Este comportamiento cambió en PHP 5, así que, por ejemplo con Windows primero se normaliza la ruta para que C:\PROGRA~1\A.php sea identificado con C:\Program Files\a.php y el fichero sea incluido sólo una vez.

BIBLIOGRAFIA

desarrolloweb. (s.f.). *desarrolloweb*. Obtenido de desarrolloweb: http://www.desarrolloweb.com/articulos/324.php

desarrolloweb.com. (s.f.). *desarrolloweb.com*. Obtenido de desarrolloweb.com: http://www.desarrolloweb.com/articulos/323.php

desarrolloweb.com. (s.f.). *desarrolloweb.com*. Obtenido de desarrolloweb.com: http://www.desarrolloweb.com/articulos/326.php

mysql.conclase.net. (s.f.). *mysql.conclase.net*. Obtenido de mysql.conclase.net: http://mysql.conclase.net/curso/?cap=007

php.net. (s.f.). *php.net*. Obtenido de php.net: http://php.net/manual/es/mysql.examples-basic.php

php.net. (s.f.). *php.net*. Obtenido de php.net: http://php.net/manual/es/function.require.php

php.net/. (s.f.). *php.net/*. Obtenido de php.net/: http://php.net/manual/es/function.require-once.php

php.net/. (s.f.). *php.net/*. Obtenido de php.net/: http://php.net/manual/es/function.include.php

php.net/. (s.f.). *php.net/*. Obtenido de php.net/: https://secure.php.net/manual/es/function.include-once.php