**TALLER NO.2 PROGRAMACIÓN WEB**

**PRESENTADO POR**

**JOULINNE ANDREA RAMIREZ TABORDA**

**PRESENTADO A**

**CARLOS ALBERTO LONDOÑO LOAIZA**

**INGENIERO EN SISTEMAS**

**CORPOTACION DE ESTUDIOS TECNOLOGICOS DEL NORTE DEL VALLE**

**CARTAGO-VALLE**

**2016**

Crear una base de datos en mysql:

1. Crea una base. En la línea de comandos de MySQL, introduce el comando CREATE DATABASE <DATABASENAME>; Remplaza <DATABASENAMEs> con el nombre de la base de datos. No puedes incluir espacios
2. Obtén la lista de las bases de datos disponibles. Introduce el comando SHOW DATABASES; para listar todas las bases de datos que tengas almacenadas
3. Selecciona la base de datos. Una vez que hayas creado la base de datos, debes seleccionarla para comenzar a editarla. Introduce el comando USE estados; Verás el mensaje Database changed, que indica que la base de datos activa es ahora estados.
4. Crea una tabla. Una tabla alberga la información en la base de datos. Para crear una tabla, tendrás que introducir todo el formato de la tabla en el comando inicial. Para crear una tabla, introduce el comando siguiente: CREATE TABLE.
5. Crea una entrada en la tabla. Ahora que has creado la tabla, es hora de comenzar a introducir la información. Usa el comando siguiente para introducir la primera entrada: INSERT INTO.
6. Crea más entradas. Puedes crear múltiples entradas con un solo comando. Para introducir los siguientes tres estados.
7. Realiza una consulta en la nueva base de datos. Ahora que has creado la base de datos básica, puedes introducir consultas para obtener resultados específicos. Primero, introduce el comando SELECT.

b.

1- Creamos un primer archivo llamado por ejemplo “conexion.php”.

2- Finalmente para incluir la conexión en cualquier archivo desde el que querramos conectarnos, sólo se debe incluir el siguiente código.

|  |  |
| --- | --- |
| 1 | <? |
| 2 | include("conexion.php"); | |

|  |  |
| --- | --- |
| 3 | ?> |
|  |  |

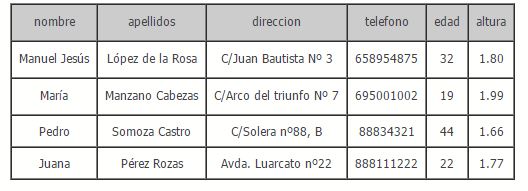
c.

Para realizar la inserción de datos vamos a utilizar una sintaxis de este tipo:

mysqli\_query($link, "INSERT INTO nombreTabla VALUES (datoCol1, datoCol2, … datoColN)");

Ejemplo:

Partiremos de nuestra base de datos donde tenemos una tabla denominada agenda con los siguientes datos:



Donde mysqli\_query indica que vamos a realizar una operación (en general se denominan consultas, aunque en este caso la operación sea una inserción de datos), $link es el identificador de conexión que hayamos obtenido con mysqli\_connect, nombreTabla es el nombre de la tabla en la cual vamos a realizar la inserción de datos y datoCol1, datoCol2, datoColN representa los datos que vamos a insertar en la tabla. Tener en cuenta que debemos indicar los datos en el mismo orden en que se encuentran las columnas en la tabla. Por ejemplo para nuestra tabla agenda tendremos que seguir el orden: nombre, apellidos, dirección, telefono, edad, altura (orden de las columnas).

d.

Para actualizar datos, lo mismo que para mostrar todos los datos correspondientes a un registro desde un listado, necesitamos identificar cada registro. En este caso vamos a identificar los registros por su “id”.

Por tanto, para iniciar la actualización vamos a crear un formulario en el que introduciremos el id que queremos actualizar, y seguidamente tras enviar este dato, nos aparecerá un formulario que contiene los datos que queremos actualizar. En este formulario los modificaremos y los actualizaremos con la correspondiente consulta.

Por tanto, ahora tendremos que crear tres páginas, de ellas quizás pueda pasar desapercibido que en el form2.php la variable “id” se envía a través de un campo oculto (hidden) para ejecutar la actualización en la página de actualización.

e.

Borrar registros de nuestra base de datos

El proceso de borrar un registro es identico al de modificar, solo que en vez de utilizar UPDATE utilizamos DELETE en la sentenica SQL. Por tanto el script quedaría como sigue.

Borrado registros de BD

<html>

<body>

<?php

if (isset($id)){

// process form

$link = mysql\_connect("localhost", "root");

mysql\_select\_db("mydb",$db);

$sql = "DELETE agenda WHERE id=$id")

$result = mysql\_query($sql);

}else{

echo "Debe especificar un 'id'.\n";

}

</body>

</html>

f..

**Buscar registros**

<title>Buscar y modificar registros en PHP</title>

</head>

<body>

<H1>Consulta de registros</H1>

<FORM ACTION="**busca.php**">

Nombre:

<INPUT TYPE="text" NAME="nombre" SIZE="20" MAXLENGTH="30">

<INPUT TYPE="submit" NAME="accion" VALUE="Buscar">

</FORM>

</body>

</html>

g.

El require es un codigo bien sencillo utilizado para cuando al procesar la página de php, se necesita otro código.

h.

La sentencia "require\_once ()" es similar a "require ()" salvo que se compruebe si el archivo ya ha sido incluido y si es así, no lo incluirá de nuevo.

Ejemplo

<?php

include 'test1.php';

require\_once ("test1.php");

?>

i.

La sentencia include incluye y evalúa el archivo especificado.

La siguiente documentación también se aplica a require.

Los archivos son incluidos con base en la ruta de acceso dada o, si ninguna es dada, el include\_path especificado. Si el archivo no se encuentra en el 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, el cual emitirá un error fatal.