**7 consulta.**

**a-**

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):

<http://mysql.conclase.net/curso/?cap=007>

**b-**

-parta conectar una BD mysql con php se utiliza una serie de pasos que seria de la siguiente manera se debe crear una primer archivo llamado por ejemplo conexión.php

Después incluimos en conexión.php el código visto en clase entonces después de incluir el código vamos al localhost que nos indica a donde se conecta con la BD

Luego después de todo ese proceso para incluir la conexión en cualquier archivo desde el que queramos conectarnos, solo se debe incluir el código conexión.php

**c-**

Creamos la consulta de inserción.

$sql = "INSERT INTO tabla (campo1, campo2, campo3) VALUES ('value1','value2',value3)";

Para ejecutar la consulta necesitamos escribir el siguiente código.

mysql\_query($sql);

**d-**

Lo primero, es lo primero, para modificar hay que tener permiso para ello en el servidor de BD, el resto nos viene de corrido. Primero seleccionamos el registro que deseamos modificar, y luego, mandamos una consulta con las modificaciones, o ambas cosas a la vez. Suponemos que las modificaciones las recogemos de un formulario como el de la lección anterior .

**Modificar registros opcion A**

<html>

<body>

<?php

if (isset($id)){

// process form

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

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

$sql = "SELECT \* FROM agenda WHERE id = $id"

$result = mysql\_query($sql);

$sql = "UPDATE agenda SET nombre='$nombre', direccion='$direccion',".

"telefono='$telefono', email='$email'";

$result = mysql\_query($sql);

}else{

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

}

</body>

</html>

<http://www.pablin.com.ar/computer/cursos/phpmysql/modificacion.htm>

**E-**

El proceso de borrar un registro es idéntico al de modificar, solo que en vez de utilizar UPDATE utilizamos DELETE en la sentencia **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>

**http://www.pablin.com.ar/computer/cursos/phpmysql/borrado.htm**

**F-**

$consulta\_mysql="select \*

                    from Empresas

                        where Actividad = 'Registro de dominios y Web Hosting'";

$resultado\_consulta\_mysql=mysql\_query($consulta\_mysql,$conexion);

while($registro=mysql\_fetch\_array($resultado\_consulta\_mysql)){

 echo $registro['Nombre']."

";

}

Para poder tomar cada registro que retornó nuestra consulta Mysql, tal como hemos visto, utilizamos la sentencia mysql\_fetch\_array

<http://www.blogdephp.com/codigo-fuente-php-para-buscar-un-dato-dentro-de-tabla-mysql/>

**G-**

Establece que el código del archivo invocado es requerido, es decir, obligatorio para el funcionamiento del programa. Por ello, si el archivo especificado en la función require() no se encuentra saltará un error “PHP Fatal error” y el programa PHP se detendrá.

**H-**

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.

**I-**

include(), por el contrario, si no se encuentra dicho código, saltará un error tipo “Warning” y el programa seguirá ejecutándose (aunque como consecuencia de no incluirse el código puede que no funcione correctamente, o sí, depende de la situación).Por tanto, es más común utilizar require(), pues lo más normal es que si llamamos al código de otro archivo es porque lo necesitamos. No obstante, se suele utilizar require para invocar código que, si no es incluido, el programa puede llegar a errores muy graves y por lo tanto, en dichas circunstancias, es mejor parar la ejecución del programa. Y utilizar include() para la llamada a archivos cuyo código no afecta a otras partes de la aplicación y que, por tanto, si no están, no afectará al resto del programa.

j-

es la misma que la h