

PHP

MySql

Tipos de campo en MySql

char(longitud):

Cualquier carácter.

El campo tendrá siempre la misma longitud.

Máximo de 255 caracteres.

varchar(longitud):

Cualquier carácter.

El campo tendrá el tamaño del contenido.

Máximo de 255 caracteres.

int(longitud):

Número entero -214783648 a +2147483647

El campo longitud limitará el número de caracteres que se visualizan, no el contenido.

PHP

MySql

Tipos de campo en MySql

int(longitud) unsigned:

Número entero: 0 a 4294967295

text:

Cualquier carácter.

El campo el tamaño máximo será de 65536 caracteres.

date:

Fechas.

Formato: aaaa-mm-dd

PHP

MySql

Tipos de campo en MySql

datetime:

Fecha y hora: aaaa-mm-dd hh:mm:ss

smallint (longitud):

Entero.

-32768 a + 32767

smallint(longitud) unsigned:

Entero

0 a + 65535

PHP

MySql

Null y Not Null

Null permite que el campo esté vacío.

Not Null no permite que el campo esté vacío.

Auto Increment

Campo que se quiere incrementar automáticamente.

Se puede indicar que el primer valor no sea el 1.

Índices

Índices primarios.

Índices secundarios.

Unique

No se permiten duplicados en ese campo.

PHP

MySql

Tipos de tablas en MySql

MyISAM:

Tabla predeterminada.
No permite integridad referencial.
Requiere indice primario.
No permite transacciones.
Muy rápidas.

HEAP

Modalidad empleada para tablas temporales.
Se guardan en memoria.
Muy rápidas.
No permiten autoincrement ni campos blob, entre otras cosas.

PHP

MySql

Tipos de tablas en MySql

BDB:

Transacciones

Muy potentes en situaciones de muchos accesos concurrentes.

InnoDB

Transacciones.

Ideales para aplicaciones de gran tamaño.

Bloqueo de filas.

La inmensa mayoría de aplicaciones WEB utilizan tablas MyISAM.

PHP

MySql

¿Dónde guarda MySql los datos?

En el caso, más común, de tablas MyISAM crea un subdirectorío del directorio "datadir" indicado en my.ini

El subdirectorío tendrá el nombre de la base de datos.

Por cada tabla creará los siguientes ficheros:

<nombre_tabla>.**FRM**:
DEFINICIÓN DE LA TABLA.

<nombre_tabla>.**MYD**:
DATOS DE LA TABLA.

<nombre_tabla>.**MYI**:
ÍNDICES DE LA TABLA.

InnoDB y DBD pueden trabajar de igual manera que MyISAM o creando tablespaces.

PHP

MySql

Sintaxis y comandos MySQL

USE <base de datos>: selecciona la base de datos donde vamos a trabajar.

DESCRIBE <tabla>: muestra las propiedades de la tabla.

CREATE TABLE <tabla>: crea una tabla:

```
CREATE TABLE gente (  
    identificador      int Not Null auto_increment,  
    nombre             varchar(50) Not Null,  
    apellidos          varchar(75) Not Null,  
  
    primary key (identificador),  
    type=MyISAM AUTO_INCREMENT=7)
```

PHP

MySql

Sintaxis y comandos MySQL

Añadir campos:

ALTER TABLE <tabla> **ADD COLUMN** (direccion varchar(150))

Eliminar campos:

ALTER TABLE <tabla> **DROP COLUMN** (direccion)

Añadir índices:

ALTER TABLE <tabla> **ADD** (<campo> [,<campo2>,...])

Eliminar una tabla:

DROP TABLE <tabla>

Eliminar una base de datos:

DROP DATABASE <BASE DE DATOSa>

PHP

MySql

Sintaxis y comandos MySQL

Consulta de selección:

```
SELECT <CAMPOS> AS <alias>  
FROM <tablas>  
WHERE criterios de selección  
ORDER BY <campo1>, <campo2> ASC [ó DESC]  
LIMIT <inicio>, <máximo de filas>
```

Modificación:

```
UPDATE tabla  
SET  
    <campo1> = <valor>,  
    <campo2> = <valor>  
WHERE  
    criterios
```

PHP

MySql

Sintaxis y comandos MySQL

Inserción:

```
INSERT INTO tabla  
(<campo1>, <campo2>.....,<campo3>)  
VALUES  
(<valor campo1>, <valor campo2>, .....,<valor campo3>)
```

Borrado

```
DELETE FROM tabla  
WHERE <condición>
```

PHP

MySql y PHP

Una conexión a un servidor o a una base de datos está sujeta a problemas:

La base de datos no existe o no está "arriba" el servidor
El usuario no existe o no tiene permisos de conexión

Para controlar estas excepciones vamos a utilizar dos funciones:

die(cadena):

Sale del programa mostrando el mensaje indicado en cadena

mysql_error():

Devuelve el mensaje de error producido por MySQL.

Utilizaremos entonces:

```
<proceso de conexión> or die(mysql_error());
```

PHP

MySql y PHP

El protocolo que seguiremos para trabajar con la base de datos será el siguiente:

- 1.-** Nos conectaremos al servidor de base de datos. Capturaremos el error.
- 2.-** Seleccionaremos la base de datos en la que vamos a trabajar. Capturaremos el error.
- 3.-** Crearemos la consulta SQL asignándosela a una cadena.
- 4.-** Realizaremos la consulta a la base de datos.
- 5.-** En el caso de una consulta de devolución de datos, recorreremos el array devuelto mostrando el resultado en el navegador.

PHP

MySQL y PHP

El paso uno en PHP (conexión al servidor):

```
$conexion=mysql_connect($host,$usuario,$password) or die(mysql_error());
```

El paso dos en PHP (selección de base de datos):

```
mysql_select_db("base de datos") or die(mysql_error());
```

El paso tres en PHP (creación de la consulta):

```
$consulta="SELECT campo1, campo2 FROM tabla where campo1='1'";
```

PHP

MySql y PHP

El paso cuatro en PHP (lanzamos la consulta):

```
$resultado=mysql_query($consulta);
```

El paso cinco en PHP (recorrido del resultado, en el caso de consulta de selección)

```
while ($filas=mysql_fetch_assoc($resultado))  
    {  
        echo $filas["campo1"] . "--" . $filas["campo2"]."<br>";  
    }
```

Ya que `mysql_fetch_assoc`(resultado de query) devuelve un array asociativo cada vez que se la invoca.

Cuando termina de realizar el recorrido devuelve false por lo que un while es ideal para recorrer los datos de consulta.

PHP

MySQL y PHP

Quedando:

```
$conexion=mysql_connect($host,$usuario,$password) or die(mysql_error());  
mysql_select_db("base de datos") or die(mysql_error());  
$consulta="SELECT campo1, campo2 FROM tabla where campo1='1';  
  
$resultado=mysql_query($consulta);  
  
while ($filas=mysql_fetch_assoc($resultado))  
    {  
        echo $filas["campo1"] . "--" . $filas["campo2"]."<br>";  
    }
```