

# MySQL y Sesiones en PHP



**Área de Ingeniería Telemática**

# Contenido

- Repaso de SQL
- PHP y MySQL

## Repaso SQL: tipo de datos

- Numéricos
  - Standard:
    - INTEGER o INT, SMALLINT, DECIMAL o DEC, NUMERIC
    - FLOAT, REAL, DOUBLE PRECISION
    - BIT
  - Extras:
    - TINYINT, MEDIUMINT, BIGINT
- Fecha y hora
  - DATETIME, DATE, TIMESTAMP, TIME, YEAR
  - Las fechas siempre deben darse en el orden **año-mes-día**

## Repaso SQL: tipo de datos

Tipo de Columna	“Cero” Valor
DATETIME	'0000-00-00 00:00:00'
DATE	'0000-00-00'
TIMESTAMP	0000000000000000
TIME	'00:00:00'
YEAR	0000

- Cadena de caracteres
  - CHAR y VARCHAR
    - Char: longitud fija (0-255). Asigna espacios si la cadena es más corta
    - Varchar: longitud variable (0-255 v5.0.3-, 0-65535 MySQLv5.0.3+)

## Repaso SQL: tipo de datos

- BINARY y VARBINARY
  - Similares a CHAR y VARCHAR
  - Contienen cadenas de caracteres binarias (cadenas de bytes) en lugar de cadenas de caracteres no binarias (cadenas de caracteres)
  - Principalmente se usan para almacenar ficheros
- BLOB y TEXT
  - No pueden tener valores DEFAULT
  - Su tamaño está prefijado
  - TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB
  - TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT
  - BLOB: cadenas de caracteres binarias
  - TEXT: cadenas de caracteres no binarias

## Repaso SQL: tipo de datos

- ENUM
  - Objeto de cadenas de caracteres con un valor elegido de una lista de valores permitidos que se enumeran explícitamente en la especificación de columna en tiempo de creación de la tabla
  - Cada valor de la enumeración tiene un índice
  - `ENUM('uno', 'dos', 'tres')` => 'uno' 'dos' 'tres'
- SET
  - objeto de cadenas de caracteres que tiene cero o más valores, cada uno de ellos debe elegirse de una lista de valores posibles especificada cuando se crea la tabla.
  - Los miembros de SET no pueden contener comas
  - Puede tener un máximo de 64 miembros distintos
  - `SET('one', 'two') NOT NULL` => " 'one' 'two' 'one,two'

## Repaso SQL: SELECT

- `SELECT * FROM tabla`
- `SELECT nombre_campo as nuevo_nombre FROM tabla`
- `SELECT * FROM tabla WHERE nombre_campo='valor_campo'`
- `SELECT * FROM tabla WHERE nombre_campo='valor_campo' AND/OR nombre_campo='valor_campo'`
- `SELECT * FROM tabla GROUP BY nombre_campo`
- `SELECT DISTINCT nombre_campo FROM tabla`
- `SELECT * FROM tabla nombre_campo LIKE '%valor_campo?'`
- `SELECT * FROM tabla ORDER BY nombre_campo ASC/DESC`

## **Repaso SQL: INSERT INTO**

- `INSERT INTO tabla (campo_1, campo_2, ...) VALUES ('valor_1', 'valor_2', ...)`
- Recomendación: siempre poner los nombres de los campos



## Repaso SQL: UPDATE

- UPDATE tabla SET campo\_1='valor\_1', campo\_2='valor\_2',  
...
- UPDATE tabla SET campo\_1='valor\_1', campo\_2='valor\_2',  
... WHERE campo\_x='valor\_x'

## Repaso SQL: DELETE

- DELETE FROM tabla WHERE
- DELETE FROM tabla WHERE nombre\_campo = 'valor\_campo'

## Repaso SQL: Otros

- TRUNCATE TABLE tabla => reinicia tabla
- SELECT \* FROM tabla\_1 WHERE nombre\_campo=(SELECT valor FROM tabla\_2 WHERE nombre\_campo\_2='valor\_campo\_2')
- Referencias:
  - <http://www.w3schools.com/sql/default.asp>
  - <http://dev.mysql.com/doc/refman/5.0/es/index.html>

# PHP y MySQL

- Veamos algunas funciones para conectarse y realizar queries a un servidor de MySQL:
  - `mysql_connect()`
  - `mysql_list_dbs()`, `mysql_select_db()`
  - `mysql_select_db()`
  - `mysql_query()`
  - `mysql_fetch_array()`
  - `mysql_num_rows()`, `mysql_insert_id()`
  - `mysql_error()`, `mysql_errno()`

## mysql\_connect()

- Establecer una conexión con un servidor MySQL

```
resource mysql_connect ( [string server [, string username  
[, string password [, bool new_link [, int  
client_flags]]]])
```

- Devuelve un recurso que identifica a esa conexión si se lleva a cabo con éxito, si no devuelve FALSE

- Ejemplo:

```
$link = mysql_connect("localhost", "mysql_user",  
"mysql_password");
```

- Para cerrar esa conexión se usa `mysql_close()`, aunque no es necesario puesto que se cerrará al terminar el script de PHP. Es recomendable cerrarla.

## mysql\_list\_dbs()

- Lista las bases de datos disponibles en el servidor (si se tiene permiso)

```
resource mysql_list_dbs ( [resource link_identifier])
```

- El argumento es el recurso que identifica a la conexión con la base de datos (`$link` en el ejemplo anterior)
- El resultado se puede recorrer como cualquier resultado de un query

## mysql\_select\_db()

- Selecciona una base de datos para todas las queries siguientes

```
bool mysql_select_db ( string database_name [, resource  
link_identifier])
```

- Si se le pasa el segundo argumento (la conexión a la BD), pasa a estar activa la base de datos seleccionada en el resource dado
- Sin segundo argumento, activa esa base de datos en la conexión más reciente al servidor

## mysql\_query()

- Envía una query al servidor

```
resource mysql_query ( string query [, resource  
link_identifier])
```

- Envía una query (primer argumento) al servidor al que hace referencia el segundo argumento, a la base de datos que tenga activa
- Si no hay segundo argumento se emplea la última conexión con base de datos que se haya creado
- Para `SELECT`, `SHOW`, `EXPLAIN` y `DESCRIBE` devuelve un identificador de recurso o `FALSE`
- Para otras sentencias SQL devuelve `TRUE` o `FALSE`



## mysql\_fetch\_array()

- Extrae una fila del resultado de una query

```
array mysql_fetch_array ( resource result [, int result_type])
```

- El primer argumento es el resultado de la query
- El segundo argumento nos permite especificar cómo queremos que nos devuelva la fila:
  - `MYSQL_ASSOC` para que sea un array asociativo cuyas claves son los nombres de las columnas
  - `MYSQL_NUM` para que sea un array indexado por el número de columna
  - `MYSQL_BOTH` para que tenga ambas claves (nombres e índice)
- Cada vez que se llama a la función devuelve la siguiente fila del resultado
- Devuelve `FALSE` cuando no quedan más filas

## mysql\_num\_rows()

- Devuelve el número de filas que tiene el resultado de una query de tipo SELECT

```
int mysql_num_rows ( resource result)
```

- El argumento es el resultado de la query

## mysql\_insert\_id()

- Si la última operación de tipo INSERT fue en una tabla con una columna con el atributo AUTO\_INCREMENT devuelve el ID nuevo que se le asignó

```
int mysql_insert_id ( [resource link_identifier])
```

- El argumento es la conexión con la base de datos y si no se especifica se empleará el último que se haya creado.

## mysql\_error()

- Devuelve el texto asociado al error que produjo la última función de MySQL

```
string mysql_error ( [resource link_identifier])
```

- Si no se produjo un error devuelve la cadena vacía.

## mysql\_errno()

- Devuelve el código numérico del error que produjo la última función de MySQL

```
int mysql_errno ( [resource link_identifier])
```

## PHP+MySQL: recomendaciones

- Crear un fichero "config.php" donde guardar los parámetros de conexión MySQL
- Mediante "include" usar ese fichero en estos los scripts => actualizar parámetros MySQL tan sencillo como modificar "config.php"
- Opcional:
  - Crear en "config.php" una función que devuelva una conexión MySQL a la BBDD => ahorrar tener que escribir las mismas líneas en todos los scripts

# Lectura sencilla de MySQL

```
<?php
session_start();
include("inicio.php");
?>
<html><head><title>Uso PHP, MySQL y HTML</title></head>
<body>
<table border="0">
  <tr><td>ID</td><td>Usuario</td></tr>
<?php
$query = "SELECT * FROM Usuarios";
$result = mysql_query($query, $link);
while($row = mysql_fetch_array($result))
{
  print "<tr><td>".$row['id']. "</td><td>".$row['usuario']. "</td></tr>\n";
}
?>
</table>
</body>
<?php
include("fin.php");
?>
```