

Cómo hacer backup de una base de datos MySQL utilizando PHP



Dentro del mundo de la informática se sabe de la importancia de realizar copias de seguridad periódicas de la información para que, en caso de pérdida de datos, poder acudir a ellas para su restauración. En el campo de las **páginas web** también es habitual hablar de backups, tanto de los archivos que forman parte del portal como de la base de datos que se utilice. La mayoría de los proveedores realizan copias de seguridad de forma periódica de sus máquinas, pero puede ser que haya ocasiones en las que nos interese realizar nosotros mismos esas copias de seguridad cada cierto tiempo.

En nuestro White Paper de este mes lo que os queremos enseñar es cómo programar un backup de una base de datos **MySQL** utilizando PHP para ello. Una vez que tengamos generado el script, podremos activar una tarea programada para que se lleve a cabo ese backup sin que tengamos que intervenir.

Configurar acceso a base de datos

Lo primero que deberemos hacer en nuestro código será crear la conexión a la base de datos de la cual queremos hacer la copia de seguridad. Para ello necesitaremos conocer los siguientes datos:

- Host del servidor
- Usuario de acceso a la base de datos
- Nombre de la base de datos
- Contraseña del usuario de la base de datos

Una vez que tengamos esta información, deberemos establecer la conexión a la base de datos.

```
$host="localhost";  
$username="root";  
$password="dueudh%53hWW";  
$database_name="myBBDD";  
  
$conn=mysqli_connect($host,$username,$password,$database_name);
```

En el código anterior hemos creado variables con nuestros datos de conexión a la **base de datos**. Luego, hemos hecho uso de la función “**mysqli_connect**” de PHP para establecer la conexión que hemos almacenado en la variable “\$conn”.

Recuperar nombres de las tablas de la base de datos

Una vez que hemos creado la conexión con el código anterior, el siguiente paso que haremos será recuperar el nombre de todas las tablas que forman parte de nuestra base de datos. Estos nombres los almacenaremos en un array para utilizarlos en el siguiente paso.

Para llevar a cabo este paso, lo primero que haremos será declararnos el array donde almacenaremos los nombres de nuestras tablas.

```
$tables=array();
```

A continuación escribiremos la sentencia SQL que nos devolverá el listado de las tablas.

```
$sql="SHOW TABLES";
```

Una vez declarado lo anterior, será el momento de ejecutar esa instrucción SQL dentro de nuestro script programado en PHP. Esto lo hacemos mediante el uso de la función “**mysqli_query**” a la que hay que pasarle como parámetros la conexión creada y la instrucción SQL que queremos ejecutar.

```
$result=mysqli_query($conn,$sql);
```

El siguiente pasoserá recorrer los resultados devueltos en el paso anterior para ir sacando el nombre de cada una de las tablas e ir almacenándolas en nuestro array. Para ello utilizaremos un bucle “**while**” que se encargue de recorrer el resultado devuelto.

```
while($row=mysqli_fetch_row($result)){  
    $tables[]=$row[0];  
}
```

Una vez que hayamos hecho todo esto, en nuestro array “**\$tables**” tendremos el nombre de todas las tablas, paso necesario para poder llevar a cabo nuestro backup.

Código PHP para la exportación de datos

Tras el paso anterior lo siguiente será recorrer todas y cada una de esas tablas que forman parte de nuestra base de datos para ir generando el código SQL de nuestro backup. Este código SQL estará formado por las estructuras de las tablas y sus correspondientes instrucciones INSERT, que contendrán toda la información almacenada en cada una de ellas.

Declararemos una variable donde iremos almacenando todas las instrucciones SQL de nuestro backup. En nuestro caso lo hemos llamado “**\$backupSQL**”.

Para recorrer todas las tablas haremos uso de un bucle “**foreach**” que se irá repitiendo mientras haya elementos en el array de tablas.

```
foreach($tables as $table){  
    .....  
}
```

Para la obtención de la estructura de la tabla tendremos que ejecutar el comando SQL “**SHOW CREATE TABLE \$table**”, donde “**\$table**” será el nombre de la tabla, valor que se irá recuperando con cada iteración.

```
$query="SHOW CREATE TABLE $table";  
$result=mysqli_query($conn,$query);  
$row=mysqli_fetch_row($result);
```

Estas instrucciones ya las hemos visto anteriormente, por lo que no las vamos a volver a explicar. Lo que sí diremos es que en “**\$row**” se guardará la estructura de la tabla en formato de array. Para recuperar la información deberemos utilizar el siguiente código.

```
$backupSQL.="\\n\\n".$row[1].";\\n\\n";
```

Recuperada la estructura de la tabla, será la hora de recuperar los valores almacenados en ella. Para ello tendremos que utilizar la siguiente instrucción SQL.

```
$query="SELECT * FROM $table";
```

Una vez ejecutada esa sentencia mediante la función “`mysqli_query`” de PHP, será cuestión de ir recorriendo cada uno de los registros devueltos para ir creando los correspondientes INSERT que formarán parte de nuestro backup. Todo esto se hace mediante la siguiente estructura de bucles.

```
for ($i=0;$i<$columnCount;$i++) {
while ($row=mysqli_fetch_row($result)) {
$backupSQL.="INSERT INTO $table VALUES (";
for ($j=0;$j<$columnCount;$j++) {
$row[$j]=$row[$j];

if (isset($row[$j])) {
$backupSQL.="'".$row[$j]."'";
} else {
$backupSQL.="''";
}
if ($j<($columnCount-1)) {
$backupSQL.=',';
}
}
$backupSQL.="");\n";
}
}
```

Descargar el backup

Tras recorrer todas las tablas en el paso anterior e ir almacenando la información en la variable “`$backupSQL`”, lo último que nos faltaría hacer sería la creación de un fichero de extensión SQL donde grabaremos los datos de la variable “`$backupSQL`” y su posterior descarga. Para ello, lo primero será comprobar que nuestra variable no está vacía.

```
if (!empty($backupSQL)) {
```

A continuación asignamos un nombre al archivo SQL que contendrá las instrucciones de nuestro backup.

```
$backup_file_name=$database_name.'_backup_'.time().'.sql';
```

Creamos un manejador de archivo donde escribiremos nuestro contenido.

```
$fileHandler=fopen($backup_file_name,'w');
```

Para finalizar, escribimos la información y cerramos el manejador de archivo.

```
$number_of_lines=fwrite($fileHandler,$backupSQL);
fclose($fileHandler);
```

Por último, lanzamos la descarga del archivo SQL en el navegador.

```
header('Content-Description: File Transfer');
header('Content-Type: application/octet-stream');
header('Content-Disposition: attachment;
filename='.basename($backup_file_name));
header('Content-Transfer-Encoding: binary');
header('Expires: 0');
header('Cache-Control: must-revalidate');
header('Pragma: public');
header('Content-Length: '.filesize($backup_file_name));
ob_clean();
flush();
```

Esto hará que se lleve a cabo la descarga del archivo SQL que hemos creado para que lo podamos guardar donde queramos.

Código completo del script

Para finalizar, os dejamos el código completo del script en PHP que nos permitirá realizar copias de seguridad de nuestras bases de datos.

```
<?php
$host="localhost";
$username="root";
$password="dueudh%53hWW";
$database_name="myBBDD";

$conn=mysqli_connect($host,$username,$password,$database_name);

$tables=array();
$sql="SHOW TABLES";
$result=mysqli_query($conn,$sql);

while($row=mysqli_fetch_row($result)){
    $tables[]=$row[0];
}

$backupSQL="";
foreach($tables as $table){
    $query="SHOW CREATE TABLE $table";
    $result=mysqli_query($conn,$query);
    $row=mysqli_fetch_row($result);
    $backupSQL.="\\n\\n".$row[1].".\\n\\n";
}

$query="SELECT * FROM $table";
$result=mysqli_query($conn,$query);

$columnCount=mysqli_num_fields($result);

for($i=0;$i<$columnCount;$i++){
```

```
while($row=mysqli_fetch_row($result)){
$backupSQL.="INSERT INTO $table VALUES (";
for($j=0;$j<$columnCount;$j++){
$row[$j]=$row[$j];

if(isset($row[$j])){
$backupSQL.=' '.$row[$j].' ';
}else{
$backupSQL.=' ' ';
}
if($j<($columnCount-1)){
$backupSQL.=' , ';
}
}
$backupSQL.=")";
}
$backupSQL.=" \n";

if(!empty($backupSQL)){
$backup_file_name=$database_name.'_backup_'.time().'.sql';
$fileHandler=fopen($backup_file_name,'w+');
$number_of_lines=fwrite($fileHandler,$backupSQL);
fclose($fileHandler);

header('Content-Description: File Transfer');
header('Content-Type: application/octet-stream');
header('Content-Disposition: attachment;
filename='.basename($backup_file_name));
header('Content-Transfer-Encoding: binary');
header('Expires: 0');
header('Cache-Control: must-revalidate');
header('Pragma: public');
header('Content-Length: '.filesize($backup_file_name));
ob_clean();
flush();
}
```

Para utilizar este código lo único que deberíais hacer es cambiar los datos de conexión a la base de datos de vuestro **servidor** y poner los vuestros.