



APRENDERAPROGRAMAR.COM

EJEMPLO DE CONSULTA  
PHP MYSQL. SELECT. USO  
DEL BUCLE WHILE CON  
MYSQLI\_FETCH\_ARRAY  
PARA EXTRAER LOS  
RESULTADOS DE UNA  
CONSULTA. (CU00842B)

Sección: Cursos

Categoría: Tutorial básico del programador web: PHP desde cero

Fecha revisión: 2029

**Resumen:** Entrega nº42 del Tutorial básico "PHP desde cero".

Autor: Enrique González Gutiérrez y César Krall

## CONSULTAS MEJORADAS A BASE DE DATOS MySQL CON PHP.

Ahora que conocemos de una forma básica como interactuar con una base de datos MySQL haciendo uso de distintas funciones o instrucciones PHP, vamos a profundizar un poco más en dichas consultas para, por ejemplo, recorrer todos los registros de la tabla agenda y visualizarlos de manera correcta (recordar que en apartados anteriores del curso sólo se mostraban los valores para el primer registro que se encontraba en la base de datos).



Para conseguir nuestro objetivo nos basaremos en el mismo código que ya hemos utilizado anteriormente y realizaremos una serie de cambios.

Para empezar, añada algunas filas más a la tabla agenda de la base de datos para que quede así:

nombre	apellidos	direccion	telefono	edad	altura
Manuel Jesús	López de la Rosa	C/Juan Bautista N° 3	658954875	32	1.80
María	Manzano Cabezas	C/Arco del triunfo N° 7	695001002	19	1.99
Pedro	Somoza Castro	C/Solera n°88, B	88834321	44	1.66
Juana	Pérez Rozas	Avda. Luarcato n°22	888111222	22	1.77

Lo primero que vamos a tratar de ver es cómo una invocación de `mysqli_fetch_array` devuelve un registro (una fila) de los resultados obtenidos en una consulta a la tabla de la base de datos. ¿Qué fila de los resultados es la que nos devuelve? Si la fila ha sido establecida usando `mysqli_data_seek`, nos devolverá la fila que hayamos especificado. Si no usamos `mysqli_data_seek` nos devuelve la primera fila de los resultados.

Si `mysqli_fetch_array` vuelve a ser invocada sin especificar la fila a extraer, devuelve el siguiente registro o fila, siguiendo el orden que tienen los resultados de la consulta. Decimos que `mysqli_fetch_array` tiene vinculado un puntero o referencia a la fila que debe devolver. Este puntero o referencia puede definirse usando `data seek`, o en caso de no definirse, es inicialmente 0 (lo que significa que nos devolverá inicialmente la fila 0). Una vez invocada `mysqli_fetch_array` y devueltos los resultados de la fila 0, el puntero queda apuntando al siguiente registro, es decir, a la fila 1. Este proceso puede repetirse tantas veces como se desee y por cada llamada `mysqli_fetch_array` va avanzando una fila. Cuando ya no existen más filas en los resultados de la consulta la función devuelve NULL (no hay resultados).

Para comprender el funcionamiento que hemos descrito ejecuta este código (que podemos denominar "bruto", pero no te preocupes ahora por la eficiencia del código, céntrate en entender el funcionamiento de `mysqli_fetch_array`). Recuerda que debes cambiar los datos relacionados con la base de datos (nombre de servidor, nombre de base de datos, usuario, password) y escribir los que estés usando tú.

```
<html>
<head><meta charset="utf-8"> </head>
<body>
<?php
//Ejemplo aprenderaprogramar.com
function mostrarDatos ($resultados) {
if ($resultados !=NULL) {
echo "- Nombre: ".$resultados['nombre']."<br/> ";
echo "- Apellidos: ".$resultados['apellidos']."<br/>";
echo "- Dirección: ".$resultados['direccion']."<br/>";
echo "- Teléfono: ".$resultados['telefono']."<br/>";
echo "- Edad: ".$resultados['edad']."<br/>";}
else {echo "<br/>No hay más datos: <br/>".$resultados;}
}
$link = mysqli_connect("sql203.byethost7.com", "b7_10356956", "*****");
mysqli_select_db($link, " b7_10356956_mibasededatos");
$tildes = $link->query("SET NAMES 'utf8'"); //Para que se muestren las tildes correctamente
$result = mysqli_query($link, "SELECT * FROM agenda");
$extraido1= mysqli_fetch_array($result);
mostrarDatos($extraido1);
$extraido2= mysqli_fetch_array($result);
mostrarDatos($extraido2);
$extraido3= mysqli_fetch_array($result);
mostrarDatos($extraido3);
$extraido4= mysqli_fetch_array($result);
mostrarDatos($extraido4);
$extraido5= mysqli_fetch_array($result);
mostrarDatos($extraido5);
$extraido6= mysqli_fetch_array($result);
mostrarDatos($extraido6);
mysqli_free_result($result);
mysqli_close($link);
?>
</body>
</html>
```

En este ejemplo sólo tenemos cuatro registros. Si tuviéramos más registros el listado se alargaría para mostrarlos todos ellos. ¿Qué ocurre si la información existente en base de datos comprende muchos registros? En este caso la información se habrá almacenado en memoria y esto consume recursos del sistema. Para liberar esta memoria se utiliza la instrucción `mysqli_free_result($result)`;

Escribe ahora este código y fíjate cómo hemos sintetizado el proceso eliminando variables auxiliares. Recuerda que debes cambiar los datos relacionados con la base de datos (nombre de servidor, nombre de base de datos, usuario, password) y escribir los que estés usando tú.

```
<html><head><meta charset="utf-8" > </head>
<body>
<?php
//Ejemplo aprenderaprogramar.com
function mostrarDatos ($resultados) {
if ($resultados !=NULL) {
echo "- Nombre: ".$resultados['nombre']."<br/> ";
echo "- Apellidos: ".$resultados['apellidos']."<br/>";
echo "- Dirección: ".$resultados['direccion']."<br/>";
echo "- Teléfono: ".$resultados['telefono']."<br/>";
echo "- Edad: ".$resultados['edad']."<br/>";
else {echo "<br/>No hay más datos!!! <br/>";}
}
$link = mysqli_connect("sql203.byethost7.com", "b7_10356956", "*****");
mysqli_select_db($link, " b7_10356956_mibasededatos");
$tildes = $link->query("SET NAMES 'utf8'"); //Para que se muestren las tildes correctamente
$result = mysqli_query($link, "SELECT * FROM agenda");
mostrarDatos(mysqli_fetch_array($result));
mostrarDatos(mysqli_fetch_array($result));
mostrarDatos(mysqli_fetch_array($result));
mostrarDatos(mysqli_fetch_array($result));
mostrarDatos(mysqli_fetch_array($result));
mostrarDatos(mysqli_fetch_array($result));
mysqli_free_result($result);
mysqli_close($link);
?>
</body></html>
```

Fijate como `mysqli_fetch_array($result)` nos devuelve un array con los datos de una fila de las obtenidas como resultado y cada vez que lo invocamos nos devuelve la siguiente fila hasta que ya no existen más filas de resultados, momento en que nos devuelve NULL o vacío.

Con una expresión como `mostrarDatos(mysqli_fetch_array($result))`; estamos pasándole a la función `mostrarDatos` el array que nos devuelve `mysqli_fetch_array`. Si la consulta nos ha devuelto 4 filas, podemos hacer 4 invocaciones obteniendo en cada invocación una fila. En la quinta invocación, ya obtenemos NULL (vacío) como resultado.

## USO DE WHILE PARA EXTRAER TODOS LOS RESULTADOS DE UNA CONSULTA

Hasta ahora hemos tratado de comprender el funcionamiento de `mysqli_fetch_array`. Pero obviamente ir recuperando resultados uno a uno no resulta eficiente. Piensa que el resultado de una consulta pueden ser cientos o miles de filas y no sería eficiente escribir línea a línea una extracción de datos.

Vamos a recurrir por tanto a crear un bucle de forma que se extraigan de forma automatizada todos los registros que nos devuelve una consulta.

La sintaxis que emplearemos es de este tipo:

```
while ($fila = mysqli_fetch_array($result)){  
    // Operaciones con los resultados que tenemos en $fila  
}
```

Esta expresión es una forma compacta donde se hacen varias cosas al mismo tiempo. Por un lado, se ejecuta `$fila = mysqli_fetch_array($result)` con lo cual `$fila` toma un valor que puede ser un array resultado ó vacío.

Luego se evalúa el `while`. `while ($variable)` lo que hace es entrar al bucle si `$variable` existe (no está vacío). `while ($fila = mysqli_fetch_array($result))` da lugar a la entrada al bucle mientras no se hayan extraído todas las filas del resultado.

Esta expresión sintética es útil porque nos permite recorrer muchas filas escribiendo un mínimo de código. También podríamos escribir esto:

```
$hayResultados = true; //Forzamos la entrada al bucle  
while ($hayResultados==true){  
    $fila = mysqli_fetch_array($result);  
    if ($fila) { //operaciones a realizar  
    } else {$hayResultados = false;}  
}
```

O también esto:

```
while (($fila = mysqli_fetch_array($result))!=NULL){  
    //operaciones a realizar  
}
```

De las diversas alternativas planteadas la que requiere menos código y una vez se entiende resulta más clara es `while ($fila = mysqli_fetch_array($result))`, por eso es la que vamos a usar. Escribe el siguiente código y comprueba los resultados. Recuerda que debes cambiar los datos relacionados con la base de datos (nombre de servidor, nombre de base de datos, usuario, password) y escribir los que estés usando tú.

De una forma u otra cada vez que invocamos la función `mysqli_fetch_array($result)` obtenemos un array asociativo con índice del array los nombres de las columnas, por ejemplo `$fila['nombre']`,

\$fila['apellidos'], \$fila['direccion'], \$fila['telefono'], \$fila['edad'], \$fila['altura']. Si ya se han extraido todas las filas se produce la salida del bucle while. Los programadores suelen llamar a la variable donde se van extrayendo los resultados \$row (en inglés, row es fila). Con mysql\_fetch\_array(\$result) también podemos acceder a través de índices numéricos, por ejemplo \$fila[0], \$fila[1], \$fila[2], \$fila[3], \$fila[4], \$fila[5].

```

<html><head><meta charset="utf-8"> </head>
<body>
<?php
//Ejemplo aprenderaprogramar.com
function mostrarDatos ($resultados) {
if ($resultados !=NULL) {
echo "- Nombre: ".$resultados['nombre']."<br/> ";
echo "- Apellidos: ".$resultados['apellidos']."<br/>";
echo "- Dirección: ".$resultados['direccion']."<br/>";
echo "- Teléfono: ".$resultados['telefono']."<br/>";
echo "- Edad: ".$resultados['edad']."<br/>";
echo "*****<br/>";
}
else {echo "<br/>No hay más datos!!! <br/>";}
}
$link = mysqli_connect("sql203.byethost7.com", "b7_10356956", "*****");
mysqli_select_db($link, " b7_10356956_mibasededatos");
$tildes = $link->query("SET NAMES 'utf8'"); //Para que se muestren las tildes correctamente
$result = mysqli_query($link, "SELECT * FROM agenda");
while ($fila = mysqli_fetch_array($result)){
mostrarDatos($fila);
}
mysqli_free_result($result);
mysqli_close($link);
?>
</body></html>
    
```

El resultado esperado es que se muestren por pantalla todos los resultados de la consulta:

```

- Nombre: Manuel Jesús - Apellidos: López de la Rosa - Dirección: C/Juan Bautista Nº 3
- Teléfono: 658954875 - Edad: 32
*****
- Nombre: María - Apellidos: Manzano Cabezas - Dirección: C/Arco del triunfo Nº 7
- Teléfono: 695001002 - Edad: 19
*****
- Nombre: Pedro - Apellidos: Somoza Castro - Dirección: C/Solera nº88, B
- Teléfono: 88834321 - Edad: 44
*****
- Nombre: Juana - Apellidos: Pérez Rozas - Dirección: Avda. Luarcato nº22
- Teléfono: 888111222 - Edad: 22
*****
    
```

## SINTAXIS NO MYSQLI (NO RECOMENDADA)

Recordar que si estamos revisando código antiguo o en servidores no actualizados nos podemos encontrar que no se usan las funciones de tipo mysqli sino las mysql. A modo de ejemplo de cómo sería esta sintaxis, mostramos este código:

```
<html> // Ejemplo php con bases de datos aprenderaprogramar.com
<body>
<?php
$link = mysql_connect("sql203.byethost7.com", "b7_10356956", "*****");
mysql_select_db("b7_10356956_mibasededatos", $link);
$result = mysql_query("SELECT * FROM agenda", $link);
// comienza un bucle que leerá todos los registros existentes
while($row = mysql_fetch_array($result)) {
    // $row es un array con todos los campos existentes en la tabla
    echo "<hr>";
    echo "Nombre: ".$row['nombre']."<br>";
    echo "Apellidos: ".$row['apellidos']."<br>";
    echo "Dirección: ".$row['direccion']."<br>";
    echo "Teléfono: ".$row['telefono']."<br>";
    echo "Edad: ".$row['edad']."<br>";
    echo "Altura: ".$row['altura']."<br>";
} // fin del bucle de instrucciones
mysql_free_result($result); // Liberamos los registros
mysql_close($link); // Cerramos la conexión con la base de datos
echo "<hr>";
?>
</body>
</html>
```

### EJERCICIO 1

Utilizando la tabla agenda que hemos empleado en los ejemplos y funciones para manejo de cadenas PHP, realiza una consulta a la base de datos para extraer todos los nombres y determina cuántos nombres en la base de datos empiezan por M.

Ejemplo de resultados a obtener: [En la base de datos los nombres que empiezan por M son: 2](#)

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros [aprenderaprogramar.com](#).

### EJERCICIO 2

Utilizando la tabla <<ciudades>> que hemos utilizado para ejercicios de entregas anteriores del curso, cuyo contenido era:

id	ciudad	país	habitantes	superficie	tieneMetro
1	México D.F.	México	555666	23434.34	1
2	Barcelona	España	444333	1111.11	0
3	Buenos Aires	Argentina	888111	333.33	1
4	Medellín	Colombia	999222	888.88	0
5	Lima	Perú	999111	222.22	0
6	Caracas	Venezuela	111222	111.11	1
7	Santiago	Chile	777666	222.22	1
8	Antigua	Guatemala	444222	877.33	0
9	Quito	Ecuador	333111	999.11	1
10	La Habana	Cuba	111222	333.11	0

Crea un archivo php donde se realice una consulta a la base de datos para extraer todos los datos de la tabla y a continuación calcula la suma de habitantes de todos los países y cuántas de estas ciudades tienen metro. Muestra el resultado por pantalla.

Ejemplo de resultados a obtener:

La suma de habitantes de México D.F., Barcelona, Buenos Aires, Medellín, Lima, Caracas, Santiago, Antigua, Quito y La Habana es: 5663886 habitantes

De estas ciudades tienen metro: 5 ciudades.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros [aprenderaprogramar.com](http://www.aprenderaprogramar.com).

**Próxima entrega:** CU00843B

**Acceso al curso completo** en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:  
[http://www.aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=70&Itemid=193](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=70&Itemid=193)