

Sintaxis MySQL de selección de registros

Las *sentencias* de selección de registros requieren utilizar -entre otras- *palabras clave* como las que enumeramos a continuación.

Observa que hay dos tipos: *obligatorias* y *opcionales*, y que algunas de las palabras clave son *alternativas* y por lo tanto, incompatibles en una misma sentencia.

El uso de estas palabras clave *requiere* que sean insertadas en un *determinado orden* tal y como se enumera aquí debajo.

Si alteráramos ese orden (p. ejemplo: colocando GROUP BY antes de WHERE) nos daría un **error** y no se ejecutaría la sentencia.

SELECT

Es la **primera palabra** de la sentencia de búsqueda y tiene carácter **obligatorio**.

[STRAIGHT_JOIN]

Es una *palabra clave* de uso **opcional** (la marcamos con corchetes para indicar su condición de opcional) que fuerza al *optimizador MySQL* a organizar las tablas en el mismo orden en el que han sido especificados los campos en la cláusula FROM.

Sirve para mejorar -en casos muy concretos- la velocidad de gestión de tablas de gran tamaño.

[SQL_BIG_RESULT]

Es una *cláusula* **opcional** que se usa para indicar al *optimizador* que el resultado va a tener **una gran cantidad de registros**.

En ese caso, MySQL utilizará *tablas temporales* cuando sea necesario para optimizar la velocidad de gestión de la información.

Esta *cláusula* también puede ser utilizada dentro de GROUP BY.

[SQL_BUFFER_RESULT]

Es **opcional** y su finalidad es la de **forzar** a MySQL a tratar el resultado en un *fichero temporal*.

Ese tratamiento ayuda a MySQL a *liberar recursos más rápidamente* y es de

gran utilidad (siempre desde el punto de vista de la rapidez) cuando es necesario un largo proceso de cálculo antes de enviar los resultados al *cliente*.

[HIGH_PRIORITY]

Esta cláusula, **opcional** da prioridad al comando SELECT sobre otros comandos que simultáneamente pudieran estar intentando acceder a la tabla para *escribir* en ella (añadir o modificar registros).

Si esta opción está activa, los intentos de escritura que pudieran producirse de forma simultánea deberían esperar al final de este proceso para ejecutarse.

campo1, campo2, ...

Tienen carácter **obligatorio** y **señalan los campos de la tabla** que deben incluirse en la consulta.

La función SELECT sólo devolverá información de aquellos campos que estén enumerados aquí.

Si se desea que la consulta **incluya a todos campos** bastará con incluir en esta posición un *****, que es el carácter *comodín* que indica a MySQL que se desea incluir todos los campos en la consulta.

Los **campos numéricos** tienen la opción de llevar asociadas **funciones** MySQL que devuelven información *estadística*.

Algunas de esas funciones son las siguientes:

- **MAX(campo..)**
Devuelve el valor **máximo** de ese campo en todos los registros de la tabla, salvo que tenga la opción GROUP BY, en cuyo caso devolverá el máximo de cada grupo, o cuando tenga activada la opción WHERE, en cuyo caso la función sólo será aplicada a los registros que resulten de tal *filtrado*.
- **MIN(campo..)**
Idéntica a la anterior en cuanto a criterios de selección, esta función devuelve el *mínimo*.
- **AVG(campo..)**
Devuelve el valor *promedio* de todos los registros numéricos seleccionados con los mismos criterios del caso anterior.
- **SUM(campo..)**
Devuelve la **suma** de los valores del **campo** y sigue idénticos criterios de selección de campos que en los casos anteriores.
- **STDDEV(campo..)**
Devuelve la estimación de la *desviación típica* de la población.
- **COUNT(campo..)**
Cuenta los valores *no nulos* del campo indicado.

En el caso de aplicar estas funciones, el **resultado de la consulta** contiene una sola línea, salvo que active la opción GROUP BY, en cuyo caso devolverá **tantas líneas** como **grupos resulten**.

FROM *tabla*

Esta *expresión* -que aunque no tiene carácter obligatorio podría tomarse como tal- indica a MySQL el **nombre de la tabla** en el que debe efectuarse la consulta.

WHERE *definición*

Esta *instrucción* tiene carácter opcional y su utilidad es la de *filtrarla* consulta estableciendo los criterios de selección de los **registros** que debe devolver.

Si se omite WHERE, la consulta devolverá **todos** los registros de la tabla.

En la parte derecha tienes información sobre la manera de **definir** los criterios de selección de esta opción.

GROUP BY *definición*

Tiene carácter **opcional** y su finalidad es la de presentar los resultados de la consulta **agrupados** según el criterio establecido en su **definición**.

Resulta de gran utilidad cuando se pretende obtener **valores estadísticos** de los registros que **cumplen determinadas condiciones** (las condiciones del *agrupamiento*).

ORDER BY *definición*

También tiene carácter **opcional** y su utilidad es la de presentar la información de la consulta **ordenada** por los contenidos de **uno** o **varios** campos.

Siempre tiene como opción complementaria de que **en cada campo utilizado para la ordenación** puede establecerse uno de estos criterios **ASC** (ascendente, es el valor por defecto) o **DESC**.

Si no se establece ningún orden, los resultados de la consulta aparecerán en el mismo orden en el que fueron añadidos los registros.

LIMIT *m, n*

Esta cláusula es **opcional** y permite establecer **cuántos** y **cuáles** registros han de presentarse en la salida de la consulta.

Por ejemplo: *LIMIT 4, 8* indicaría a MySQL que la consulta debería *mostrar* **OCHO** registros contados a partir del **quinto** (sí, el quinto porque LIMIT considera el primer registro como CERO).

El criterio límite se aplica sobre los resultados de la salida, es decir, sobre los resultados **seleccionados, ordenados y filtrados** siguiendo los criterios establecidos por las cláusulas anteriores.

Si se escribe como un solo parámetro (LIMIT k), MySQL lo interpretará como que **k** es el segundo de ellos y que el primero es CERO, es decir:
LIMIT 0, k

Recuento de resultados

PHP dispone de dos funciones que permiten conocer el número de registros de la tabla afectados por una sentencia MySQL.

mysql_num_rows (\$c)

Esta función devuelve un valor numérico que recoge el número de registros que cumplen las condiciones establecidas en una **consulta**. Sólo es válido para sentencia tipo SELECT

mysql_affected_rows(\$c)

En este caso la función devuelve también el número de registros afectados, pero sólo en el caso de que la sentencia MySQL haya producido modificaciones en los contenidos de la tabla. Es decir, sólo recoge resultados de sentencias que: **añaden, modifican o borran** registros.

Manejo de fechas en las consultas

MySQL dispone de algunas cláusulas de gestión de fechas que pueden tener una gran utilidad a la hora de gestionar consultas. Son las siguientes:

DATE_FORMAT(campo, formato)

Las diferentes opciones de formato las tienes en la siguiente tabla. Es importante tener en cuenta que la sintaxis correcta es **%Y** (sin espacio) ya que si hubiera un espacio **% Y** interpretaría la letra Y como un texto a incluir.

Los formatos soportados por la función DATE_FORMAT format son los siguientes:

Formato	Descripción	Sintaxis
%d	Día del mes en formato de dos dígitos	DATE_FORMAT(Nacimiento,'%d')
%e	Día del mes en formato de uno ó dos dígitos	DATE_FORMAT(Nacimiento,'%e')
%D	Número de día seguido del sufijo en inglés	DATE_FORMAT(Nacimiento,'%D')
%m	Número del mes en formato de dos dígitos	DATE_FORMAT(Nacimiento,'%m')
%c	Número del mes en formato de uno o dos dígitos	DATE_FORMAT(Nacimiento,'%c')
%M	Nombre del mes (en inglés)	DATE_FORMAT(Nacimiento,'%M')
%b	Nombre del mes abreviado (en inglés)	DATE_FORMAT(Nacimiento,'%b')

%y	Número del año en formato de dos dígitos	DATE_FORMAT(Nacimiento,'%y')
%Y	Número del año en formato de cuatro dígitos	DATE_FORMAT(Nacimiento,'%Y')
%w	Número de día de la semana 0=Domingo ... 6=Sábado	DATE_FORMAT(Nacimiento,'%w')
%W	Nombre del día de la semana (en inglés)	DATE_FORMAT(Nacimiento,'%W')
%W	Nombre abreviado del día de la semana (en inglés)	DATE_FORMAT(Nacimiento,'%W')
%j	Número de día del año en formato de 3 dígitos	DATE_FORMAT(Nacimiento,'%j')
%U	Número de semana del año considerando el DOMINGO como primer día de la semana (en formato de dos dígitos)	DATE_FORMAT(Nacimiento,'%U')
%u	Número de semana del año considerando el LUNES como primer día de la semana (en formato de dos dígitos)	DATE_FORMAT(Nacimiento,'%u')
%H	Hora con dos dígitos (formato 0 a 24 horas)	DATE_FORMAT(\$fecha,'%H')
%k	Hora con uno ó dos dígitos (formato 0 a 24 horas)	DATE_FORMAT(\$fecha,'%k')
%h	Hora con dos dígitos (formato 0 a 12 horas)	DATE_FORMAT(\$fecha,'%h')
%l	Hora con uno ó dos dígitos (formato 0 a 12 horas)	DATE_FORMAT(\$fecha,'%l')
%i	Minutos con dos dígitos	DATE_FORMAT(\$fecha,'%i')
%s	Segundos con dos dígitos	DATE_FORMAT(\$fecha,'%s')
%r	Hora completa (HH:mm:ss) en formato de 12 horas indicando AM ó PM	DATE_FORMAT(\$fecha,'%r')
%T	Hora completa (HH:mm:ss) en formato de 24 horas	DATE_FORMAT(\$fecha,'%T')
% texto	Incluye el texto que se indica detrás del %	DATE_FORMAT(\$fecha,'% texto')
%p	Añade AM ó PM dependiendo de la Hora	DATE_FORMAT(\$fecha,'%p')
<p>Se pueden combinar a voluntad varias opciones utilizando una sintaxis de este tipo: '% Hoy es: %d - %m - %Y % es %W % estamos en el mes de %M %
y van transcurridos %j % días de este año.
Son las %r'</p>		

CURDATE()

Dentro de **DATE_FORMAT** se puede incluir -en vez del nombre del campo- una cadena en la que se indique una fecha en formato **YYYY-MM-DD hh:mm:ss**. Puedes verlo en los ejemplos. De igual modo es posible sustituir el nombre del campo -o la cadena- por la función **CURDATE()** que recoge la **fecha actual del sistema** (únicamente día, mes y año). A efectos de horas, minutos y segundos **CURDATE()** va a tomar el *mediodía* de la fecha actual.

Ejemplo de CURDATE()

```
$resultado= mysql_query("SELECT DATE_FORMAT(CURDATE(),'% Hoy es: %d
- %m -
%Y % es %W % estamos en el mes de %M % <br>y van transcurridos %j %
días de este año de gracia<br> cuando son las %r')", $conexion);
echo mysql_result($resultado,0);
```

Hoy es: 09 - 01 - 2012 es Monday estamos en el mes de January
y van transcurridos 009 días de este año de gracia
cuando son las 12:00:00 AM

CURTIME()

Se comporta de forma similar a CURDATE().

Devuelve la hora actual del sistema que alberga el servidor MySQL en formato **hh:mm:ss**

CURRENT_TIMESTAMP()

Se comporta de forma similar a CURDATE().

Devuelve la fecha y hora actual del sistema en formato **YYYY-MM-DD hh:mm:ss**

NOW()

Es un *alias* de **CURRENT_TIMESTAMP()**.

mysql_result(\$resultado,num, campo)

Esta función PHP permite obtener un solo campo de uno solo de los registros obtenidos como resultado de una consulta MySQL.

El parámetro *\$resultado* es la variable que recoge el resultado obtenido de la ejecución de *mysql_query* de forma idéntica a como lo hacíamos en otras consultas.

El valor *num* es un número entero que indica el número de fila de la que queremos extraer el valor contenido en uno de sus campos.

El valor *campo* indica el *número del campo* que tratamos de extraer. Este número (la primera posición siempre es **cero**) indica el número de orden del campo tal como está especificado en la sentencia SELECT. Si en esta sentencia se incluyera * (extraer todos los campos) consideraría el orden en el que está creada la estructura de la tabla que los contiene.

EJEMPLO DE USO DE `mysql_result($resultado,num, campo)`

<?

```
# recogemos en una variable el nombre de BASE DE DATOS
```

```
$base="ejemplos";
```

```
# recogemos en una variable el nombre de la TABLA
```

```
$tabla="demo4";
```

```
# establecemos la conexión con el servidor
```

```
$conexion=mysql_connect("localhost","alumno1","alumno1");
```

```
#asiganamos la conexión a una base de datos determinada
```

```
mysql_select_db($base,$conexion);
```

```
# establecemos el criterio de SELECCION
```

```
# en este caso los campos Nombre, Apellido1, Apellido2 unicamente
```

```
$resultado= mysql_query("SELECT Nombre, Apellido1, Apellido2 FROM $tabla ORDER BY Nombre DESC" , $conexion);
```

```
print mysql_result($resultado,3,1);
```

```
print "<br>o también<br>";
```

```
print mysql_result($resultado,3);
```

```
$resultado= mysql_query("SELECT * FROM $tabla ORDER BY Nombre DESC" , $conexion);
```

```
print mysql_result($resultado,5,0);
```

```
print "<br>o también<br>";
```

```
print mysql_result($resultado,5);
```

```
mysql_close();
```

?>