[**PHP MySQL 函数** 1](#_Toc278902535)

[**mysql\_affected\_rows() 函数** 6](#_Toc278902536)

[**mysql\_client\_encoding() 函数** 8](#_Toc278902537)

[**mysql\_close() 函数** 9](#_Toc278902538)

[**mysql\_connect() 函数** 10](#_Toc278902539)

[**mysql\_data\_seek() 函数** 11](#_Toc278902540)

[**mysql\_db\_name() 函数** 13](#_Toc278902541)

[**mysql\_errno() 函数** 14](#_Toc278902542)

[**mysql\_error() 函数** 15](#_Toc278902543)

[**mysql\_fetch\_array() 函数** 16](#_Toc278902544)

[**mysql\_fetch\_assoc() 函数** 18](#_Toc278902545)

[**mysql\_fetch\_field() 函数** 19](#_Toc278902546)

[**mysql\_fetch\_lengths() 函数** 22](#_Toc278902547)

[mysql\_fetch\_object() 函数 23](#_Toc278902548)

[**mysql\_fetch\_row() 函数** 25](#_Toc278902549)

[**mysql\_field\_flags() 函数** 26](#_Toc278902550)

[**mysql\_field\_len() 函数** 28](#_Toc278902551)

[mysql\_field\_name() 函数 29](#_Toc278902552)

[**mysql\_field\_seek() 函数** 30](#_Toc278902553)

[**mysql\_field\_table() 函数** 32](#_Toc278902554)

[**mysql\_field\_type() 函数** 33](#_Toc278902555)

[**mysql\_free\_result() 函数** 34](#_Toc278902556)

[mysql\_get\_client\_info() 函数 35](#_Toc278902557)

[**mysql\_get\_host\_info() 函数** 36](#_Toc278902558)

[**mysql\_get\_proto\_info() 函数** 37](#_Toc278902559)

[**mysql\_get\_server\_info() 函数** 37](#_Toc278902560)

[**mysql\_info() 函数** 38](#_Toc278902561)

[**mysql\_insert\_id() 函数** 40](#_Toc278902562)

[**mysql\_list\_dbs() 函数** 41](#_Toc278902563)

[**mysql\_list\_processes() 函数** 42](#_Toc278902564)

[**mysql\_num\_fields() 函数** 43](#_Toc278902565)

[**mysql\_num\_rows() 函数** 44](#_Toc278902566)

[**mysql\_pconnect() 函数** 46](#_Toc278902567)

[**mysql\_ping() 函数** 47](#_Toc278902568)

[**mysql\_query() 函数** 48](#_Toc278902569)

[**mysql\_real\_escape\_string() 函数** 50](#_Toc278902570)

[**mysql\_result() 函数** 53](#_Toc278902571)

[**mysql\_select\_db() 函数** 55](#_Toc278902572)

[**mysql\_stat() 函数** 56](#_Toc278902573)

[**mysql\_thread\_id() 函数** 57](#_Toc278902574)

[**mysql\_unbuffered\_query() 函数** 58](#_Toc278902575)

**PHP MySQL 函数**

* [Previous Page](http://www.w3school.com.cn/php/php_ref_math.asp)
* [Next Page](http://www.w3school.com.cn/php/php_ref_simplexml.asp)

**PHP MySQL 简介**

MySQL 函数允许您访问 MySQL 数据库服务器。

**安装**

为了能够顺利的使用本类函数，必须在编译 PHP 时添加对 MySQL 的支持。

编译时，只要使用 --with-mysql[=DIR] 配置选项即可，其中可选的 [DIR] 指向 MySQL 的安装目录。

虽然本 MySQL 扩展库兼容 MySQL 4.1.0 及其以后版本，但是它不支持这些版本提供的额外功能。要使用这些功能，请使用 MySQLi 扩展库。

如果要同时安装 mysql 扩展库和 mysqli 扩展库，必须使用同一个客户端库以避免任何冲突。

**在 Linux 系统上安装**

**PHP 4**

默认开启了 --with-mysql 选项。此默认行为可以用 --without-mysql 配置选项来禁止。如果启用 MySQL 而不指定安装目录的话，PHP 将使用绑定的 MySQL 客户端连接库。

还有其它应用程序使用 MySQL（例如 auth-mysql）的用户不要用绑定的库，而要指定 MySQL 的安装目录，如这样：--with-mysql=/path/to/mysql。这将强制 PHP 使用随 MySQL 安装的客户端连接库，就可以避免任何冲突。

**PHP 5+**

MySQL 默认未启用，也没有绑定的 MySQL 库。使用 --with-mysql[=DIR] 配置选项来加入 MySQL 的支持。可以从 [MySQL](http://www.mysql.com/) 下载头文件和库。

**在 Windows 系统上安装**

**PHP 4**

PHP MySQL 扩展已经编译入 PHP。

**PHP 5+**

MySQL 默认未启用，因此必须在 php.ini 中激活 php\_mysql.dll 动态连接库。此外，PHP 还需要访问 MySQL 客户端连接库。PHP 的 Windows 发行版包括了一个 libmysql.dll，为了让 PHP 能和 MySQL 对话，此文件必须放在 Windows 的系统路径 PATH 中。

要激活任何 PHP 扩展库（例如 php\_mysql.dll），PHP 指令 extension\_dir 要被设为 PHP 扩展库所在的目录。PHP 5 下 extension\_dir 取值的一个例子是 c:\php\ext。

注释：如果启动 web 服务器时出现类似如下的错误："Unable to load dynamic library './php\_mysql.dll'"，这是因为系统找不到 php\_mysql.dll 和 / 或 libmysql.dll。

**Runtime 配置**

MySQL 函数的行为受到 php.ini 中设置的影响。

MySQL 配置选项：

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **默认** | **描述** | **可更改** |
| mysql.allow\_persistent | "1" | 是否允许 MySQL 的持久连接。 | PHP\_INI\_SYSTEM |
| mysql.max\_persistent | "-1" | 每个进程中最大的持久连接数目。 | PHP\_INI\_SYSTEM |
| mysql.max\_links | "-1" | 每个进程中最大的连接数，包括持久连接。 | PHP\_INI\_SYSTEM |
| mysql.trace\_mode | "0" | 跟踪模式。从 PHP 4.3.0 起可用。 | PHP\_INI\_ALL |
| mysql.default\_port | NULL | 指定默认连接数据库的 TCP 端口号。 | PHP\_INI\_ALL |
| mysql.default\_socket | NULL | 默认的 socket 名称。PHP 4.0.1起可用。 | PHP\_INI\_ALL |
| mysql.default\_host | NULL | 默认的服务器地址。不适用于SQL安全模式。 | PHP\_INI\_ALL |
| mysql.default\_user | NULL | 默认使用的用户名。不适用于SQL安全模式。 | PHP\_INI\_ALL |
| mysql.default\_password | NULL | 默认使用的密码。不适用于SQL安全模式。 | PHP\_INI\_ALL |
| mysql.connect\_timeout | "60" | 连接超时秒数。 | PHP\_INI\_ALL |

**资源类型**

在 MySQL 模块中使用了两种资源类型。第一种是数据库的连接句柄，第二种是 SQL 查询返回的结果集。

**PHP MySQL 函数**

PHP：指示支持该函数的最早的 PHP 版本。

|  |  |  |
| --- | --- | --- |
| **函数** | **描述** | **PHP** |
| [mysql\_affected\_rows()](http://www.w3school.com.cn/php/func_mysql_affected_rows.asp) | 取得前一次 MySQL 操作所影响的记录行数。 | 3 |
| mysql\_change\_user() | 不赞成。改变活动连接中登录的用户 | 3 |
| [mysql\_client\_encoding()](http://www.w3school.com.cn/php/func_mysql_client_encoding.asp) | 返回当前连接的字符集的名称 | 4 |
| [mysql\_close()](http://www.w3school.com.cn/php/func_mysql_close.asp) | 关闭非持久的 MySQL 连接。 | 3 |
| [mysql\_connect()](http://www.w3school.com.cn/php/func_mysql_connect.asp) | 打开非持久的 MySQL 连接。 | 3 |
| mysql\_create\_db() | 不赞成。新建 MySQL 数据库。使用 mysql\_query() 代替。 | 3 |
| [mysql\_data\_seek()](http://www.w3school.com.cn/php/func_mysql_data_seek.asp) | 移动记录指针。 | 3 |
| [mysql\_db\_name()](http://www.w3school.com.cn/php/func_mysql_db_name.asp) | 从对 mysql\_list\_dbs() 的调用返回数据库名称。 | 3 |
| mysql\_db\_query() | 不赞成。发送一条 MySQL 查询。  使用 mysql\_select\_db() 和 mysql\_query() 代替。 | 3 |
| mysql\_drop\_db() | 不赞成。丢弃（删除）一个 MySQL 数据库。  使用 mysql\_query() 代替。 | 3 |
| [mysql\_errno()](http://www.w3school.com.cn/php/func_mysql_errno.asp) | 返回上一个 MySQL 操作中的错误信息的数字编码。 | 3 |
| [mysql\_error()](http://www.w3school.com.cn/php/func_mysql_error.asp) | 返回上一个 MySQL 操作产生的文本错误信息。 | 3 |
| mysql\_escape\_string() | 不赞成。转义一个字符串用于 mysql\_query。  使用 mysql\_real\_escape\_string() 代替。 | 4 |
| [mysql\_fetch\_array()](http://www.w3school.com.cn/php/func_mysql_fetch_array.asp) | 从结果集中取得一行作为关联数组，或数字数组，或二者兼有。 | 3 |
| [mysql\_fetch\_assoc()](http://www.w3school.com.cn/php/func_mysql_fetch_assoc.asp) | 从结果集中取得一行作为关联数组。 | 4 |
| [mysql\_fetch\_field()](http://www.w3school.com.cn/php/func_mysql_fetch_field.asp) | 从结果集中取得列信息并作为对象返回。 | 3 |
| [mysql\_fetch\_lengths()](http://www.w3school.com.cn/php/func_mysql_fetch_lengths.asp) | 取得结果集中每个字段的内容的长度。 | 3 |
| [mysql\_fetch\_object()](http://www.w3school.com.cn/php/func_mysql_fetch_object.asp) | 从结果集中取得一行作为对象。 | 3 |
| [mysql\_fetch\_row()](http://www.w3school.com.cn/php/func_mysql_fetch_row.asp) | 从结果集中取得一行作为数字数组。 | 3 |
| [mysql\_field\_flags()](http://www.w3school.com.cn/php/func_mysql_field_flags.asp) | 从结果中取得和指定字段关联的标志。 | 3 |
| [mysql\_field\_len()](http://www.w3school.com.cn/php/func_mysql_field_len.asp) | 返回指定字段的长度。 | 3 |
| [mysql\_field\_name()](http://www.w3school.com.cn/php/func_mysql_field_name.asp) | 取得结果中指定字段的字段名。 | 3 |
| [mysql\_field\_seek()](http://www.w3school.com.cn/php/func_mysql_field_seek.asp) | 将结果集中的指针设定为指定的字段偏移量。 | 3 |
| [mysql\_field\_table()](http://www.w3school.com.cn/php/func_mysql_field_table.asp) | 取得指定字段所在的表名。 | 3 |
| [mysql\_field\_type()](http://www.w3school.com.cn/php/func_mysql_field_type.asp) | 取得结果集中指定字段的类型。 | 3 |
| [mysql\_free\_result()](http://www.w3school.com.cn/php/func_mysql_free_result.asp) | 释放结果内存。 | 3 |
| [mysql\_get\_client\_info()](http://www.w3school.com.cn/php/func_mysql_get_client_info.asp) | 取得 MySQL 客户端信息。 | 4 |
| [mysql\_get\_host\_info()](http://www.w3school.com.cn/php/func_mysql_get_host_info.asp) | 取得 MySQL 主机信息。 | 4 |
| [mysql\_get\_proto\_info()](http://www.w3school.com.cn/php/func_mysql_get_proto_info.asp) | 取得 MySQL 协议信息。 | 4 |
| [mysql\_get\_server\_info()](http://www.w3school.com.cn/php/func_mysql_get_server_info.asp) | 取得 MySQL 服务器信息。 | 4 |
| [mysql\_info()](http://www.w3school.com.cn/php/func_mysql_info.asp) | 取得最近一条查询的信息。 | 4 |
| [mysql\_insert\_id()](http://www.w3school.com.cn/php/func_mysql_insert_id.asp) | 取得上一步 INSERT 操作产生的 ID。 | 3 |
| [mysql\_list\_dbs()](http://www.w3school.com.cn/php/func_mysql_list_dbs.asp) | 列出 MySQL 服务器中所有的数据库。 | 3 |
| mysql\_list\_fields() | 不赞成。列出 MySQL 结果中的字段。  使用 mysql\_query() 代替。 | 3 |
| [mysql\_list\_processes()](http://www.w3school.com.cn/php/func_mysql_list_processes.asp) | 列出 MySQL 进程。 | 4 |
| mysql\_list\_tables() | 不赞成。列出 MySQL 数据库中的表。  使用Use mysql\_query() 代替。 | 3 |
| [mysql\_num\_fields()](http://www.w3school.com.cn/php/func_mysql_num_fields.asp) | 取得结果集中字段的数目。 | 3 |
| [mysql\_num\_rows()](http://www.w3school.com.cn/php/func_mysql_num_rows.asp) | 取得结果集中行的数目。 | 3 |
| [mysql\_pconnect()](http://www.w3school.com.cn/php/func_mysql_pconnect.asp) | 打开一个到 MySQL 服务器的持久连接。 | 3 |
| [mysql\_ping()](http://www.w3school.com.cn/php/func_mysql_ping.asp) | Ping 一个服务器连接，如果没有连接则重新连接。 | 4 |
| [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) | 发送一条 MySQL 查询。 | 3 |
| [mysql\_real\_escape\_string()](http://www.w3school.com.cn/php/func_mysql_real_escape_string.asp) | 转义 SQL 语句中使用的字符串中的特殊字符。 | 4 |
| [mysql\_result()](http://www.w3school.com.cn/php/func_mysql_result.asp) | 取得结果数据。 | 3 |
| [mysql\_select\_db()](http://www.w3school.com.cn/php/func_mysql_select_db.asp) | 选择 MySQL 数据库。 | 3 |
| [mysql\_stat()](http://www.w3school.com.cn/php/func_mysql_stat.asp) | 取得当前系统状态。 | 4 |
| mysql\_tablename() | 不赞成。取得表名。使用 mysql\_query() 代替。 | 3 |
| [mysql\_thread\_id()](http://www.w3school.com.cn/php/func_mysql_thread_id.asp) | 返回当前线程的 ID。 | 4 |
| [mysql\_unbuffered\_query()](http://www.w3school.com.cn/php/func_mysql_unbuffered_query.asp) | 向 MySQL 发送一条 SQL 查询（不获取 / 缓存结果）。 | 4 |

**PHP MySQL 常量**

在 PHP 4.3.0 以后的版本中，允许在 mysql\_connect() 函数和 mysql\_pconnect() 函数中指定更多的客户端标记：

PHP：指示支持该常量的最早的 PHP 版本。

|  |  |  |
| --- | --- | --- |
| **常量** | **描述** | **PHP** |
| MYSQL\_CLIENT\_COMPRESS | 使用压缩的通讯协议。 | 4.3 |
| MYSQL\_CLIENT\_IGNORE\_SPACE | 允许在函数名后留空格位。 | 4.3 |
| MYSQL\_CLIENT\_INTERACTIVE | 在关闭连接前所允许的交互超时非活动时间。 | 4.3 |
| MYSQL\_CLIENT\_SSL | 使用 SSL 加密（仅在 MySQL 客户端库版本为 4+ 时可用）。 | 4.3 |

mysql\_fetch\_array() 函数使用一个常量来表示所返回数组的类型：

|  |  |  |
| --- | --- | --- |
| **常量** | **描述** | **PHP** |
| MYSQL\_ASSOC | 返回的数据列使用字段名作为数组的索引名。 |  |
| MYSQL\_BOTH | 返回的数据列使用字段名及数字索引作为数组的索引名。 |  |
| MYSQL\_NUM | 返回的数据列使用数字索引作为数组的索引名。  索引从 0 开始，表示返回结果的第一个字段。 |  |

**mysql\_affected\_rows() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_affected\_rows() 函数返回前一次 MySQL 操作所影响的记录行数。

**语法**

mysql\_affected\_rows(link\_identifier)

|  |  |
| --- | --- |
| **参数** | **描述** |
| link\_identifier | 必需。MySQL 的连接标识符。如果没有指定，默认使用最后被 [mysql\_connect()](http://www.w3school.com.cn/php/func_mysql_connect.asp) 打开的连接。如果没有找到该连接，函数会尝试调用 [mysql\_connect()](http://www.w3school.com.cn/php/func_mysql_connect.asp) 建立连接并使用它。如果发生意外，没有找到连接或无法建立连接，系统发出 E\_WARNING 级别的警告信息。 |

**说明**

取得最近一次与 link\_identifier 关联的 INSERT，UPDATE 或 DELETE 查询所影响的记录行数。

**返回值**

执行成功，则返回受影响的行的数目，如果最近一次查询失败的话，函数返回 -1。

如果最近一次操作是没有任何条件（WHERE）的 DELETE 查询，在表中所有的记录都会被删除，但本函数返回值在 4.1.2 版之前都为 0。

当使用 UPDATE 查询，MySQL 不会将原值与新值一样的列更新。这样使得 mysql\_affected\_rows() 函数返回值不一定就是查询条件所符合的记录数，只有真正被修改的记录数才会被返回。

REPLACE 语句首先删除具有相同主键的记录，然后插入一个新记录。该函数返回的是被删除的记录数加上被插入的记录数。

**例子**

<?php

$con = mysql\_connect("localhost","mysql\_user","mysql\_pwd");

if (!$con)

{

die("Could not connect: " . mysql\_error());

}

mysql\_select\_db("mydb");

mysql\_query("DELETE FROM mytable WHERE id < 5");

$rc = mysql\_affected\_rows();

echo "Records deleted: " . $rc;

mysql\_close($con);

?>

输出：

Records deleted: 4

**mysql\_client\_encoding() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_client\_encoding() 函数返回当前连接的字符集的名称。

**语法**

mysql\_client\_encoding(link\_identifier)

|  |  |
| --- | --- |
| **参数** | **描述** |
| link\_identifier | 必需。MySQL 的连接标识符。如果没有指定，默认使用最后被 [mysql\_connect()](http://www.w3school.com.cn/php/func_mysql_connect.asp) 打开的连接。如果没有找到该连接，函数会尝试调用 [mysql\_connect()](http://www.w3school.com.cn/php/func_mysql_connect.asp) 建立连接并使用它。如果发生意外，没有找到连接或无法建立连接，系统发出 E\_WARNING 级别的警告信息。 |

**说明**

从 MySQL 中取得 character\_set 变量的值。

**返回值**

返回当前连接的默认字符集名称。

**例子**

<?php

$con = mysql\_connect("localhost","mysql\_user","mysql\_pwd");

if (!$con)

{

die("Could not connect: " . mysql\_error());

}

$charset = mysql\_client\_encoding($con);

echo "The current character set is: $charset";

mysql\_close($con);

?>

输出：

The current character set is: latin1

**mysql\_close() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_close() 函数关闭非持久的 MySQL 连接。

**语法**

mysql\_close(link\_identifier)

|  |  |
| --- | --- |
| **参数** | **描述** |
| link\_identifier | 必需。MySQL 的连接标识符。如果没有指定，默认使用最后被 [mysql\_connect()](http://www.w3school.com.cn/php/func_mysql_connect.asp) 打开的连接。如果没有找到该连接，函数会尝试调用 [mysql\_connect()](http://www.w3school.com.cn/php/func_mysql_connect.asp) 建立连接并使用它。如果发生意外，没有找到连接或无法建立连接，系统发出 E\_WARNING 级别的警告信息。 |

**说明**

本函数关闭指定的连接标识所关联的到 MySQL 服务器的非持久连接。如果没有指定 *link\_identifier*，则关闭上一个打开的连接。

**返回值**

如果成功则返回 true，失败则返回 false。

**提示和注释**

提示：通常不需要使用 mysql\_close()，因为已打开的非持久连接会在脚本执行完毕后自动关闭。

注释：mysql\_close() 不会关闭由 mysql\_pconnect() 建立的持久连接。

**例子**

<?php

$con = mysql\_connect("localhost","mysql\_user","mysql\_pwd");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

// 一些代码...

mysql\_close($con);

?>

**mysql\_connect() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_connect() 函数打开非持久的 MySQL 连接。

**语法**

mysql\_connect(server,user,pwd,newlink,clientflag)

|  |  |
| --- | --- |
| **参数** | **描述** |
| server | 可选。规定要连接的服务器。  可以包括端口号，例如 "hostname:port"，或者到本地套接字的路径，例如对于 localhost 的 ":/path/to/socket"。  如果 PHP 指令 mysql.default\_host 未定义（默认情况），则默认值是 'localhost:3306'。 |
| user | 可选。用户名。默认值是服务器进程所有者的用户名。 |
| pwd | 可选。密码。默认值是空密码。 |
| newlink | 可选。如果用同样的参数第二次调用 mysql\_connect()，将不会建立新连接，而将返回已经打开的连接标识。参数 new\_link 改变此行为并使 mysql\_connect() 总是打开新的连接，甚至当 mysql\_connect() 曾在前面被用同样的参数调用过。 |
| clientflag | 可选。client\_flags 参数可以是以下常量的组合：   * MYSQL\_CLIENT\_SSL - 使用 SSL 加密 * MYSQL\_CLIENT\_COMPRESS - 使用压缩协议 * MYSQL\_CLIENT\_IGNORE\_SPACE - 允许函数名后的间隔 * MYSQL\_CLIENT\_INTERACTIVE - 允许关闭连接之前的交互超时非活动时间 |

**返回值**

如果成功，则返回一个 MySQL 连接标识，失败则返回 FALSE。

**提示和注释**

注释：脚本一结束，到服务器的连接就被关闭，除非之前已经明确调用 [mysql\_close()](http://www.w3school.com.cn/php/func_mysql_close.asp) 关闭了。

提示：要创建一个持久连接，请使用 [mysql\_pconnect()](http://www.w3school.com.cn/php/func_mysql_pconnect.asp) 函数。

**例子**

<?php

$con = mysql\_connect("localhost","mysql\_user","mysql\_pwd");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

// 一些代码...

mysql\_close($con);

?>

**mysql\_data\_seek() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_data\_seek() 函数移动内部结果的指针。

**语法**

mysql\_data\_seek(data,row)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。返回类型为 resource 的结果集。该结果集从 mysql\_query() 的调用中得到。 |
| row | 必需。想要设定的新的结果集指针的行数。0 指示第一个记录。 |

**说明**

mysql\_data\_seek() 将 *data* 参数指定的 MySQL 结果内部的行指针移动到指定的行号。

接着调用 [mysql\_fetch\_row()](http://www.w3school.com.cn/php/func_mysql_fetch_row.asp) 将返回那一行。

*row* 从 0 开始。*row* 的取值范围应该从 0 到 mysql\_num\_rows - 1。

但是如果结果集为空（mysql\_num\_rows() == 0），要将指针移动到 0 会失败并发出 E\_WARNING 级的错误，mysql\_data\_seek() 将返回 false。

**返回值**

如果成功则返回 true，失败则返回 false。

**提示和注释**

注释：mysql\_data\_seek() 只能和 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 一起使用，而不能用于 [mysql\_unbuffered\_query()](http://www.w3school.com.cn/php/func_mysql_unbuffered_query.asp)。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

print\_r(mysql\_fetch\_row($result));

mysql\_data\_seek($result,3);

print\_r(mysql\_fetch\_row($result));

mysql\_close($con);

?>

输出：

Array

(

[0] => Adams

[1] => John

[2] => London

)

Array

(

[0] => Carter

[1] => Thomas

[2] => Beijing

)

**mysql\_db\_name() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_db\_name() 函数取得 mysql\_list\_dbs() 调用所返回的数据库名。

**语法**

mysql\_db\_name(list,row,field)

|  |  |
| --- | --- |
| **参数** | **描述** |
| list | 必需。[mysql\_list\_dbs()](http://www.w3school.com.cn/php/func_mysql_list_dbs.asp) 调用所返回的结果指针。 |
| row | 必需。规定结果集中的行号。以 0 开始。 |
| field | 可选。规定字段名。 |

**说明**

如果成功，则返回数据库名，失败返回 false。如果返回了 false，用 [mysql\_error()](http://www.w3school.com.cn/php/func_mysql_error.asp) 来判断错误的种类。

**例子**

<?php

$con = mysql\_connect("localhost", "peter", "abc123");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_list = mysql\_list\_dbs($con);

$i = 0;

$db\_count = mysql\_num\_rows($db\_list);

while ($i < $db\_count)

{

echo mysql\_db\_name($db\_list, $i) . "<br />";

$i++;

}

mysql\_close($con);

?>

输出类似：

mysql

customers

movies

**mysql\_errno() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_errno() 函数返回上一个 MySQL 操作中的错误信息的数字编码。

返回上一个 MySQL 函数的错误号码，如果没有出错则返回 0（零）。

**语法**

mysql\_errno(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 SQL 连接标识符。如果未规定，则使用上一个打开的连接。 |

**例子**

在本例中，我们将尝试使用错误的用户名和密码来登录一个 MySQL 服务器：

<?php

$con = mysql\_connect("localhost","wrong\_user","wrong\_pwd");

if (!$con)

{

die('Could not connect: ' . mysql\_errno());

}

mysql\_close($con);

?>

输出类似：

Could not connect: 1045

**mysql\_error() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_error() 函数返回上一个 MySQL 操作产生的文本错误信息。

本函数返回上一个 MySQL 函数的错误文本，如果没有出错则返回 ''（空字符串）。

**语法**

mysql\_error(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 SQL 连接标识符。如果未规定，则使用上一个打开的连接。 |

**例子**

在本例中，我们将尝试使用错误的用户名和密码来登录一个 MySQL 服务器：

<?php

$con = mysql\_connect("localhost","wrong\_user","wrong\_pwd");

if (!$con)

{

die(mysql\_error());

}

mysql\_close($con);

?>

输出类似：

Access denied for user 'wrong\_user'@'localhost'

(using password: YES)

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_fetch\_array() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_fetch\_array() 函数从结果集中取得一行作为关联数组，或数字数组，或二者兼有

返回根据从结果集取得的行生成的数组，如果没有更多行则返回 false。

**语法**

mysql\_fetch\_array(data,array\_type)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 可选。规定规定要使用的数据指针。该数据指针是 mysql\_query() 函数产生的结果。 |
| array\_type | 可选。规定返回哪种结果。可能的值：   * MYSQL\_ASSOC - 关联数组 * MYSQL\_NUM - 数字数组 * MYSQL\_BOTH - 默认。同时产生关联和数字数组 |

**提示和注释**

注释：mysql\_fetch\_array() 是 [mysql\_fetch\_row()](http://www.w3school.com.cn/php/func_mysql_fetch_row.asp) 的扩展版本。除了将数据以数字索引方式储存在数组中之外，还可以将数据作为关联索引储存，用字段名作为键名。

提示：有很重要的一点必须指出，用 mysql\_fetch\_array() 并不明显比用 [mysql\_fetch\_row()](http://www.w3school.com.cn/php/func_mysql_fetch_row.asp) 慢，而且还明显提供了更多的值。

注释：本函数返回的字段名是区分大小写的。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person WHERE Lastname='Adams'";

$result = mysql\_query($sql,$con);

print\_r(mysql\_fetch\_array($result));

mysql\_close($con);

?>

输出类似：

Array

(

[0] => Adams

[LastName] => Adams

[1] => John

[FirstName] => John

[2] => London

[City] => London

)

**mysql\_fetch\_assoc() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_fetch\_assoc() 函数从结果集中取得一行作为关联数组。

返回根据从结果集取得的行生成的关联数组，如果没有更多行，则返回 false。

**语法**

mysql\_fetch\_assoc(data)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 mysql\_query() 返回的结果。 |

**提示和注释**

注释：mysql\_fetch\_assoc() 和用 mysql\_fetch\_array() 加上第二个可选参数 MYSQL\_ASSOC 完全相同。它仅仅返回关联数组。这也是 mysql\_fetch\_array() 初始的工作方式。

提示：如果在关联索引之外还需要数字索引，用 mysql\_fetch\_array()。

注释：本函数返回的字段名是区分大小写的。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person WHERE Lastname='Adams'";

$result = mysql\_query($sql,$con);

print\_r(mysql\_fetch\_assoc($result));

mysql\_close($con);

?>

输出：

Array

(

[LastName] => Adams

[FirstName] => John

[City] => London

)

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_fetch\_field() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_fetch\_field() 函数从结果集中取得列信息并作为对象返回。

mysql\_fetch\_field() 可以用来从查询结果中取得字段的信息。如果没有指定字段偏移量，则提取下一个尚未被 mysql\_fetch\_field() 取得的字段。

该函数返回一个包含字段信息的对象。

被返回的对象的属性为：

* name - 列名
* table - 该列所在的表名
* max\_length - 该列最大长度
* not\_null - 1，如果该列不能为 NULL
* primary\_key - 1，如果该列是 primary key
* unique\_key - 1，如果该列是 unique key
* multiple\_key - 1，如果该列是 non-unique key
* numeric - 1，如果该列是 numeric
* blob - 1，如果该列是 BLOB
* type - 该列的类型
* unsigned - 1，如果该列是无符号数
* zerofill - 1，如果该列是 zero-filled

**语法**

mysql\_fetch\_field(data,field\_offset)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 mysql\_query() 返回的结果。 |
| field\_offset | 必需。规定从哪个字段开始。0 指示第一个字段。如果未设置，则取回下一个字段。 |

**提示和注释**

注释：本函数返回的字段名是区分大小写的。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

while ($property = mysql\_fetch\_field($result))

{

echo "Field name: " . $property->name . "<br />";

echo "Table name: " . $property->table . "<br />";

echo "Default value: " . $property->def . "<br />";

echo "Max length: " . $property->max\_length . "<br />";

echo "Not NULL: " . $property->not\_null . "<br />";

echo "Primary Key: " . $property->primary\_key . "<br />";

echo "Unique Key: " . $property->unique\_key . "<br />";

echo "Mutliple Key: " . $property->multiple\_key . "<br />";

echo "Numeric Field: " . $property->numeric . "<br />";

echo "BLOB: " . $property->blob . "<br />";

echo "Field Type: " . $property->type . "<br />";

echo "Unsigned: " . $property->unsigned . "<br />";

echo "Zero-filled: " . $property->zerofill . "<br /><br />";

}

mysql\_close($con);

?>

输出：

Field name: LastName

Table name: Person

Default value:

Max length: 8

Not NULL: 0

Primary Key: 0

Unique Key: 0

Mutliple Key: 0

Numeric Field: 0

BLOB: 0

Field Type: string

Unsigned: 0

Zero-filled: 0

Field name: FirstName

Table name: Person

Default value:

Max length: 7

Not NULL: 0

Primary Key: 0

Unique Key: 0

Mutliple Key: 0

Numeric Field: 0

BLOB: 0

Field Type: string

Unsigned: 0

Zero-filled: 0

Field name: City

Table name: Person

Default value:

Max length: 9

Not NULL: 0

Primary Key: 0

Unique Key: 0

Mutliple Key: 0

Numeric Field: 0

BLOB: 0

Field Type: string

Unsigned: 0

Zero-filled: 0

Field name: Age

Table name: Person

Default value:

Max length: 2

Not NULL: 0

Primary Key: 0

Unique Key: 0

Mutliple Key: 0

Numeric Field: 1

BLOB: 0

Field Type: int

Unsigned: 0

Zero-filled: 0

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_fetch\_lengths() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_fetch\_lengths() 函数取得一行中每个字段的内容的长度。

行是由这些函数返回的：[mysql\_fetch\_array()](http://www.w3school.com.cn/php/func_mysql_fetch_array.asp)、[mysql\_fetch\_assoc()](http://www.w3school.com.cn/php/func_mysql_fetch_assoc.asp)、[mysql\_fetch\_object()](http://www.w3school.com.cn/php/func_mysql_fetch_object.asp) 或 [mysql\_fetch\_row()](http://www.w3school.com.cn/php/func_mysql_fetch_row.asp)。

若成功，则该函数返回一个数字数组，若出错或没有其他的行，则返回 false。

**语法**

mysql\_fetch\_lengths(data)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 返回的结果。 |

**例子**

<?php

$con = mysql\_connect("localhost", "peter", "abc123");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person WHERE Lastname='Refsnes'";

$result = mysql\_query($sql,$con);

print\_r(mysql\_fetch\_row($result));

print\_r(mysql\_fetch\_lengths($result));

mysql\_close($con);

?>

输出：

Array

(

[0] => Adams

[1] => John

[2] => London

)

Array

(

[0] => 5

[1] => 4

[2] => 6

)

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

# mysql\_fetch\_object() 函数

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

## 定义和用法

mysql\_fetch\_object() 函数从结果集（记录集）中取得一行作为对象。

若成功的话，本函数从 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 获得一行，并返回一个对象。如果失败或没有更多的行，则返回 false。

### 语法

mysql\_fetch\_object(data)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 返回的结果。 |

## 提示和注释

注释：每个随后对 mysql\_fetch\_object() 的调用都会返回记录集中的下一行。

注释：mysql\_fetch\_object() 与 [mysql\_fetch\_array()](http://www.w3school.com.cn/php/func_mysql_fetch_array.asp) 类似，只有一点区别 - 返回的是对象而不是数组。间接地，也意味着只能通过字段名来访问数组，而不是偏移量。

## 例子

<?php

$con = mysql\_connect("localhost", "peter", "abc123");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

while ($row = mysql\_fetch\_object($result))

{

echo $row->FirstName . "<br />";

}

mysql\_close($con);

?>

输出：

John

George

Thomas

窗体顶端

**mysql\_fetch\_row() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_fetch\_row() 函数从结果集中取得一行作为数字数组。

**语法**

mysql\_fetch\_row(data)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 mysql\_query() 返回的结果。 |

**说明**

mysql\_fetch\_row() 从和结果标识 *data* 关联的结果集中取得一行数据并作为数组返回。每个结果的列储存在一个数组的单元中，偏移量从 0 开始。

依次调用 mysql\_fetch\_row() 将返回结果集中的下一行，如果没有更多行则返回 FALSE。

**返回值**

返回根据所取得的行生成的数组，如果没有更多行则返回 false。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person WHERE Lastname='Adams'";

$result = mysql\_query($sql,$con);

print\_r(mysql\_fetch\_row($result));

mysql\_close($con);

?>

输出：

Array

(

[0] => Adams

[1] => John

[2] => London

)

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_field\_flags() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_field\_flags() 函数从结果中取得和指定字段关联的标志。

本函数返回指定字段的字段标志。

如果 MySQL 版本足够新，则会支持以下的标志：

* auto\_intcrement
* binary
* blob
* enum
* multiple\_key
* not\_null
* primary\_key
* timestamp
* unique\_key
* unsigned
* zerofill

**语法**

mysql\_field\_flags(data,field\_offset)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 返回的结果。 |
| field\_offset | 必需。指示从哪个字段开始返回。0 指示第一个字段。 |

**提示和注释**

提示：每个标志都用一个单词表示，之间用一个空格分开，因此，您可以通过 explode() 函数把返回的字符串打散到一个数组中。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

$flags = mysql\_field\_flags($result, 0);

echo $flags;

mysql\_close($con);

?>

输出：

not\_null primary\_key auto\_increment

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_field\_len() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_field\_len() 函数返回指定字段的长度。

如果成功，则返回字段的长度，如果失败，则返回 false。

**语法**

mysql\_field\_len(data,field\_offset)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 返回的结果。 |
| field\_offset | 必需。指示从哪个字段开始返回。0 指示第一个字段。 |

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

$length = mysql\_field\_len($result, 0);

echo $length;

mysql\_close($con);

?>

输出：

20

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

# mysql\_field\_name() 函数

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

## 定义和用法

mysql\_field\_name() 函数取得结果中指定字段的字段名。

如果成功，则返回字段的名称，如果失败，则返回 false。

### 语法

mysql\_field\_name(data,field\_offset)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 返回的结果。 |
| field\_offset | 必需。指示从哪个字段开始返回。0 指示第一个字段。 |

## 提示和注释

注释：本函数返回的字段名是区分大小写的。

## 例子

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

$name = mysql\_field\_name($result, 0);

echo $name;

mysql\_close($con);

?>

输出：

LastName

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

窗体顶端

Search:





窗体底端

##### [PHP 参考手册](http://www.w3school.com.cn/php/php_ref.asp)

##### [PHP 测验](http://www.w3school.com.cn/php/php_quiz.asp)

## 赞助商链接

W3School提供的内容仅用于培训。我们不保

**mysql\_field\_seek() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_field\_seek() 函数将结果集中的指针设定为指定的字段偏移量。

如果成功，则返回 true，如果失败，则返回 false。

**语法**

mysql\_field\_seek(data,field\_offset)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 返回的结果。 |
| field\_offset | 必需。指示从哪个字段开始返回。0 指示第一个字段。 |

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

// 跳到第四个字段

mysql\_field\_seek($result,3);

print\_r(mysql\_fetch\_field($result));

mysql\_close($con);

?>

输出：

stdClass Object

(

[name] => Age

[table] => Person

[def] =>

[max\_length] => 2

[not\_null] => 0

[primary\_key] => 0

[multiple\_key] => 0

[unique\_key] => 0

[numeric] => 1

[blob] => 0

[type] => int

[unsigned] => 0

[zerofill] => 0

)

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_field\_table() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_field\_table() 函数返回指定字段所在的表名。

如果成功，则返回指定字段的表名，如果失败，则返回 false。

**语法**

mysql\_field\_table(data,field\_offset)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 返回的结果。 |
| field\_offset | 必需。指示从哪个字段开始返回。0 指示第一个字段。 |

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

$table = mysql\_field\_table($result, 0);

echo $table;

mysql\_close($con);

?>

输出：

Person

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_field\_type() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_field\_type() 函数返回结果集中指定字段的类型。

如果成功，则返回指定字段的类型，如果失败，则返回 false。

**语法**

mysql\_field\_type(data,field\_offset)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要使用的数据指针。该数据指针是从 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 返回的结果。 |
| field\_offset | 必需。指示从哪个字段开始返回。0 指示第一个字段。 |

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

$type = mysql\_field\_type($result, 0);

echo $type;

mysql\_close($con);

?>

输出：

string

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_free\_result() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_free\_result() 函数释放结果内存。

如果成功，则返回 true，如果失败，则返回 false。

**语法**

mysql\_free\_result(data)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。要释放的结果标识符。该结果标识符是从 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 返回的结果。 |

**提示和注释**

注释：mysql\_free\_result() 仅需要在考虑到返回很大的结果集时会占用多少内存时调用。在脚本结束后所有关联的内存都会被自动释放。

**例子**

<?php

$con = mysql\_connect("localhost", "peter", "abc123");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

print\_r(mysql\_fetch\_row($result));

// 释放内存

mysql\_free\_result($result);

$sql = "SELECT \* from Customers";

$result = mysql\_query($sql,$con);

print\_r(mysql\_fetch\_row($result));

mysql\_close($con);

?>

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

# mysql\_get\_client\_info() 函数

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

## 定义和用法

mysql\_get\_client\_info() 函数返回 MySQL 客户端信息。

如果成功，则返回 MySQL 客户端的版本号，如果失败，则返回 false。

### 语法

mysql\_get\_client\_info()

## 例子

<?php

echo "MySQL client info: " . mysql\_get\_client\_info();

?>

输出：

MySQL client info: 5.0.18

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_get\_host\_info() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_get\_host\_info() 函数返回 MySQL 主机的信息。

如果成功，则返回当前正在使用的 MySQL 连接的类型，如果失败，则返回 false。

**语法**

mysql\_get\_host\_info(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 MySQL 连接。如果未规定，则使用上一个连接。 |

**说明**

mysql\_get\_host\_info() 返回一个字符串，说明了连接 *connection* 所使用的连接方式，包括服务器的主机名。如果省略 *connection*，则使用上一个打开的连接。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

echo "MySQL host info: " . mysql\_get\_host\_info($con);

?>

输出：

MySQL host info: localhost via TCP/IP

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_get\_proto\_info() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_get\_proto\_info() 函数返回 MySQL 协议的信息。

如果成功，则返回 MySQL 协议的版本，如果失败，则返回 false。

**语法**

mysql\_get\_proto\_info(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 MySQL 连接。如果未规定，则使用上一个连接。 |

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

echo "MySQL protocol info: " . mysql\_get\_proto\_info($con);

?>

输出：

MySQL protocol info: 10

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_get\_server\_info() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_get\_server\_info() 函数返回 MySQL 服务器的信息。

如果成功，则返回 MySQL 服务器的版本号，如果失败，则返回 false。

**语法**

mysql\_get\_server\_info(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 MySQL 连接。如果未规定，则使用上一个连接。 |

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

echo "MySQL server info: " . mysql\_get\_server\_info($con);

?>

输出：

MySQL server info: 5.0.18-nt

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_info() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_info() 函数返回最近一条查询的信息。

如果成功，则返回有关该语句的信息，如果失败，则返回 false。

**语法**

mysql\_info(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 MySQL 连接。如果未规定，则使用上一个连接。 |

**说明**

mysql\_info() 返回通过给定的 *connection* 所进行的最新一条查询的详细信息。如果没有指定 *connection*，则假定为上一个打开的连接。

mysql\_info() 对以下列出的所有语句返回一个字符串：

INSERT INTO ... SELECT ...

INSERT INTO ... VALUES (...),(...),(...)...

LOAD DATA INFILE ...

ALTER TABLE

UPDATE

对于其它任何语句返回 false。字符串的格式取决于给出的语句。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "INSERT INTO person VALUES ('John','Doe','Utah','17')";

$result = mysql\_query($sql,$con);

$info = mysql\_info($con);

echo $info;

mysql\_close($con);

?>

输出类似：

String format: Records: 15 Duplicates: 0 Warnings: 0

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_insert\_id() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_insert\_id() 函数返回上一步 INSERT 操作产生的 ID。

如果上一查询没有产生 AUTO\_INCREMENT 的 ID，则 mysql\_insert\_id() 返回 0。

**语法**

mysql\_insert\_id(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 MySQL 连接。如果未规定，则使用上一个连接。 |

**说明**

mysql\_insert\_id() 返回给定的 *connection* 中上一步 INSERT 查询中产生的 AUTO\_INCREMENT 的 ID 号。如果没有指定 *connection* ，则使用上一个打开的连接。

**提示和注释**

注释：如果需要保存该值以后使用，要确保在产生了值的查询之后立即调用 mysql\_insert\_id()。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "INSERT INTO person VALUES ('Carter','Thomas','Beijing')";

$result = mysql\_query($sql,$con);

echo "ID of last inserted record is: " . mysql\_insert\_id();

mysql\_close($con);

?>

输出类似：

ID of last inserted record is: 5

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_list\_dbs() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_list\_dbs() 函数列出 MySQL 服务器中所有的数据库。

**语法**

mysql\_list\_dbs(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 SQL 连接标识符。如果未规定，则使用上一个打开的连接。 |

**说明**

mysql\_list\_dbs() 将返回一个结果指针，包含了当前 MySQL 进程中所有可用的数据库。

用 mysql\_tablename() 函数来遍历此结果指针，或者任何使用结果表的函数，例如 [mysql\_fetch\_array()](http://www.w3school.com.cn/php/func_mysql_fetch_array.asp)。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_list = mysql\_list\_dbs($con);

while ($db = mysql\_fetch\_object($db\_list))

{

echo $db->Database . "<br />";

}

mysql\_close($con);

?>

输出类似：

mysql

test

test\_db

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_list\_processes() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_list\_processes() 函数列出 MySQL 进程。

若成功，则 mysql\_list\_processes() 返回一个结果指针，说明了当前服务器的线程。如果失败，则返回 false。

**语法**

mysql\_list\_processes(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 MySQL 连接。如果未规定，则使用上一个连接。 |

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$my\_list = mysql\_list\_processes($con);

while ($row = mysql\_fetch\_assoc($my\_list))

{

print\_r($row);

}

mysql\_close($con);

?>

输出类似：

Array

(

[Id] => 2

[User] => hello

[Host] => localhost:1038

[db] =>

[Command] => Processlist

[Time] => 0

[State] =>

[Info] =>

)

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_num\_fields() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_num\_fields() 函数返回结果集中字段的数。

如果失败，则返回 false。

**语法**

mysql\_num\_fields(data)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。规定规定要使用的数据指针。该数据指针是 mysql\_query() 函数产生的结果。 |

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* FROM person";

$result = mysql\_query($sql,$con);

echo mysql\_num\_fields($result);

mysql\_close($con);

?>

输出类似：

3

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_num\_rows() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_num\_rows() 函数返回结果集中行的数目。

**语法**

mysql\_num\_rows(data)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。结果集。该结果集从 mysql\_query() 的调用中得到。 |

**说明**

mysql\_num\_rows() 返回结果集中行的数目。此命令仅对 SELECT 语句有效。要取得被 INSERT，UPDATE 或者 DELETE 查询所影响到的行的数目，用 [mysql\_affected\_rows()](http://www.w3school.com.cn/php/func_mysql_affected_rows.asp)。

**提示和注释**

注释：如果使用 [mysql\_unbuffered\_query()](http://www.w3school.com.cn/php/func_mysql_unbuffered_query.asp)，则直到结果集中的所有行都被提取后 mysql\_num\_rows() 才能返回正确的值。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db",$con);

$sql = "SELECT \* FROM person";

$result = mysql\_query($sql,$con);

echo mysql\_num\_rows($result);

mysql\_close($con);

?>

输出类似：

3

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_pconnect() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_pconnect() 函数打开一个到 MySQL 服务器的持久连接。

mysql\_pconnect() 和 mysql\_connect() 非常相似，但有两个主要区别：

1. 当连接的时候本函数将先尝试寻找一个在同一个主机上用同样的用户名和密码已经打开的（持久）连接，如果找到，则返回此连接标识而不打开新连接。
2. 其次，当脚本执行完毕后到 SQL 服务器的连接不会被关闭，此连接将保持打开以备以后使用（mysql\_close() 不会关闭由 mysql\_pconnect() 建立的连接）。

**语法**

mysql\_pconnect(server,user,pwd,clientflag)

|  |  |
| --- | --- |
| **参数** | **描述** |
| server | 可选。规定要连接的服务器。  可以包括端口号，例如 "hostname:port"，或者到本地套接字的路径，例如对于 localhost 的 ":/path/to/socket"。  如果 PHP 指令 mysql.default\_host 未定义（默认情况），则默认值是 'localhost:3306'。 |
| user | 可选。用户名。默认值是服务器进程所有者的用户名。 |
| pwd | 可选。密码。默认值是空密码。 |
| clientflag | 可选。client\_flags 参数可以是以下常量的组合：   * MYSQL\_CLIENT\_SSL - 使用 SSL 加密 * MYSQL\_CLIENT\_COMPRESS - 使用压缩协议 * MYSQL\_CLIENT\_IGNORE\_SPACE - 允许函数名后的间隔 * MYSQL\_CLIENT\_INTERACTIVE - 允许关闭连接之前的交互超时非活动时间 |

**返回值**

如果成功，则返回一个 MySQL 持久连接标识符，出错则返回 FALSE。

**提示和注释**

注释：可选参数 *clientflag* 自 PHP 4.3.0 版起可用。

提示：要创建一个非持久连接，请使用 [mysql\_connect()](http://www.w3school.com.cn/php/func_mysql_connect.asp) 函数。

**例子**

<?php

$con = mysql\_pconnect("localhost","mysql\_user","mysql\_pwd");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

?>

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_ping() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_ping() 函数 Ping 一个服务器连接，如果没有连接则重新连接。

如果存在连接，则返回 true。如果失败，则返回 false。

**语法**

mysql\_ping(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 MySQL 连接。如果未规定，则使用上一个连接。 |

**提示和注释**

注释：本函数可用于空闲很久的脚本来检查服务器是否关闭了连接。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db", $con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$link);

mysql\_ping();

// 一些代码...

mysql\_close($con);

?>

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_query() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_query() 函数执行一条 MySQL 查询。

**语法**

mysql\_query(query,connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| query | 必需。规定要发送的 SQL 查询。注释：查询字符串不应以分号结束。 |
| connection | 可选。规定 SQL 连接标识符。如果未规定，则使用上一个打开的连接。 |

**说明**

如果没有打开的连接，本函数会尝试无参数调用 mysql\_connect() 函数来建立一个连接并使用之。

**返回值**

mysql\_query() 仅对 SELECT，SHOW，EXPLAIN 或 DESCRIBE 语句返回一个资源标识符，如果查询执行不正确则返回 FALSE。

对于其它类型的 SQL 语句，mysql\_query() 在执行成功时返回 TRUE，出错时返回 FALSE。

非 FALSE 的返回值意味着查询是合法的并能够被服务器执行。这并不说明任何有关影响到的或返回的行数。很有可能一条查询执行成功了但并未影响到或并未返回任何行。

**提示和注释**

注释：该函数自动对记录集进行读取和缓存。如需运行非缓存查询，请使用 [mysql\_unbuffered\_query()](http://www.w3school.com.cn/php/func_mysql_unbuffered_query.asp)。

**例子**

**例子 1**

<?php

$con = mysql\_connect("localhost","mysql\_user","mysql\_pwd");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$sql = "SELECT \* FROM Person";

mysql\_query($sql,$con);

// 一些代码

mysql\_close($con);

?>

**例子 2**

通过 mysql\_query() 函数创建一个新数据库：

<?php

$con = mysql\_connect("localhost","mysql\_user","mysql\_pwd");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$sql = "CREATE DATABASE my\_db";

if (mysql\_query($sql,$con))

{

echo "Database my\_db created";

}

else

{

echo "Error creating database: " . mysql\_error();

}

?>

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_real\_escape\_string() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_real\_escape\_string() 函数转义 SQL 语句中使用的字符串中的特殊字符。

下列字符受影响：

* \x00
* \n
* \r
* \
* '
* "
* \x1a

如果成功，则该函数返回被转义的字符串。如果失败，则返回 false。

**语法**

mysql\_real\_escape\_string(string,connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| string | 必需。规定要转义的字符串。 |
| connection | 可选。规定 MySQL 连接。如果未规定，则使用上一个连接。 |

**说明**

本函数将 *string* 中的特殊字符转义，并考虑到连接的当前字符集，因此可以安全用于 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp)。

**提示和注释**

提示：可使用本函数来预防数据库攻击。

**例子**

**例子 1**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

// 获得用户名和密码的代码

// 转义用户名和密码，以便在 SQL 中使用

$user = mysql\_real\_escape\_string($user);

$pwd = mysql\_real\_escape\_string($pwd);

$sql = "SELECT \* FROM users WHERE

user='" . $user . "' AND password='" . $pwd . "'"

// 更多代码

mysql\_close($con);

?>

**例子 2**

数据库攻击。本例演示如果我们不对用户名和密码应用 mysql\_real\_escape\_string() 函数会发生什么：

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$sql = "SELECT \* FROM users

WHERE user='{$\_POST['user']}'

AND password='{$\_POST['pwd']}'";

mysql\_query($sql);

// 不检查用户名和密码

// 可以是用户输入的任何内容，比如：

$\_POST['user'] = 'john';

$\_POST['pwd'] = "' OR ''='";

// 一些代码...

mysql\_close($con);

?>

那么 SQL 查询会成为这样：

SELECT \* FROM users

WHERE user='john' AND password='' OR ''=''

这意味着任何用户无需输入合法的密码即可登陆。

**例子 3**

预防数据库攻击的正确做法：

<?php

function check\_input($value)

{

// 去除斜杠

if (get\_magic\_quotes\_gpc())

{

$value = stripslashes($value);

}

// 如果不是数字则加引号

if (!is\_numeric($value))

{

$value = "'" . mysql\_real\_escape\_string($value) . "'";

}

return $value;

}

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

// 进行安全的 SQL

$user = check\_input($\_POST['user']);

$pwd = check\_input($\_POST['pwd']);

$sql = "SELECT \* FROM users WHERE

user=$user AND password=$pwd";

mysql\_query($sql);

mysql\_close($con);

?>

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_result() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_result() 函数返回结果集中一个字段的值。

如果成功，则该函数返回字段值。如果失败，则返回 false。

**语法**

mysql\_result(data,row,field)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。规定要使用的结果标识符。该标识符是 mysql\_query() 函数返回的。 |
| row | 必需。规定行号。行号从 0 开始。 |
| field | 可选。规定获取哪个字段。可以是字段偏移值，字段名或 table.fieldname。  如果该参数未规定，则该函数从指定的行获取第一个字段。 |

**说明**

当作用于很大的结果集时，应该考虑使用能够取得整行的函数。这些函数在一次函数调用中返回了多个单元的内容，比 mysql\_result() 快得多。

此外请注意，在字段参数中指定数字偏移量比指定字段名或者 tablename.fieldname 要快得多。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db", $con);

$sql = "SELECT \* from Person";

$result = mysql\_query($sql,$con);

echo mysql\_result($result,0);

mysql\_close($con);

?>

输出类似：

Adams

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_select\_db() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_select\_db() 函数设置活动的 MySQL 数据库。

如果成功，则该函数返回 true。如果失败，则返回 false。

**语法**

mysql\_select\_db(database,connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| database | 必需。规定要选择的数据库。 |
| connection | 可选。规定 MySQL 连接。如果未指定，则使用上一个连接。 |

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db", $con);

if (!$db\_selected)

{

die ("Can\'t use test\_db : " . mysql\_error());

}

mysql\_close($con);

?>

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_stat() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_stat() 函数返回 MySQL 服务器的当前系统状态。

如果成功，则该函数返回状态。如果失败，则返回 false。

**语法**

mysql\_stat(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 MySQL 连接。如果未指定，则使用上一个连接。 |

**提示和注释**

注释：mysql\_stat() 目前只返回 uptime、threads、queries、open tables、flush tables 和 queries per second。要得到其它状态变量的完整列表，只能使用 SQL 命令 SHOW STATUS。

**例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db", $con);

echo mysql\_stat();

?>

输出类似：

Uptime: 99410 Threads: 1 Questions: 162

Slow queries: 0 Opens: 0 Flush tables: 1

Open tables: 0 Queries per second avg: 0.002

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_thread\_id() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_thread\_id() 函数返回当前线程的 ID。

如果成功，则该函数线程的 ID。如果失败，则返回 false。

**语法**

mysql\_thread\_id(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 MySQL 连接。如果未指定，则使用上一个连接。 |

**提示和注释**

注释：每个 MySQL 连接的 线程 ID 是不同的。

**例子**

<?php

$con = mysql\_connect("localhost", "peter", "abc123");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

echo mysql\_thread\_id();

mysql\_close($con);

?>

输出类似：

38

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**mysql\_unbuffered\_query() 函数**

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)

**定义和用法**

mysql\_unbuffered\_query() 函数向 MySQL 发送一条 SQL 查询（不获取 / 缓存结果）。

**语法**

mysql\_unbuffered\_query(query,connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| query | 必需。规定要发送的 SQL 查询。注释：查询字符串不应以分号结束。 |
| connection | 可选。规定 SQL 连接标识符。如果未规定，则使用上一个打开的连接。 |

**说明**

mysql\_unbuffered\_query() 向 MySQL 发送一条 SQL 查询 query ，但不像 [mysql\_query()](http://www.w3school.com.cn/php/func_mysql_query.asp) 那样自动获取并缓存结果集。一方面，这在处理很大的结果集时会节省可观的内存。另一方面，可以在获取第一行后立即对结果集进行操作，而不用等到整个 SQL 语句都执行完毕。

当使用多个数据库连接时，必须指定可选参数 *connection*。

**提示和注释**

注释：mysql\_unbuffered\_query() 的好处是有代价的：在 mysql\_unbuffered\_query() 返回的结果集之上不能使用 [mysql\_num\_rows()](http://www.w3school.com.cn/php/func_mysql_num_rows.asp) 和 [mysql\_data\_seek()](http://www.w3school.com.cn/php/func_mysql_data_seek.asp)。此外在向 MySQL 发送一条新的 SQL 查询之前，必须提取所有未缓存的 SQL 查询所产生的结果行。

**例子**

<?php

$con = mysql\_connect("localhost","mysql\_user","mysql\_pwd");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

// 大型查询

$sql = "SELECT \* FROM Person";

mysql\_unbuffered\_query($sql,$con);

// 开始处理数据...

mysql\_close($con);

?>

[PHP MySQL 函数](http://www.w3school.com.cn/php/php_ref_mysql.asp)