**PHP MySQL 简介**

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

**MySQL 是最流行的开源数据库服务器。**

**什么是 MySQL？**

MySQL 是一种数据库。数据库定义了存储信息的结构。

在数据库中，存在着一些表。类似 HTML 表格，数据库表含有行、列以及单元。

在分类存储信息时，数据库非常有用。一个公司的数据库可能拥有这些表："Employees", "Products", "Customers" 以及 "Orders"。

**数据库表**

数据库通常包含一个或多个表。每个表都一个名称（比如 "Customers" 或 "Orders"）。每个表包含带有数据的记录（行）。

下面是一个名为 "Persons" 的表的例子：

|  |  |  |  |
| --- | --- | --- | --- |
| **LastName** | **FirstName** | **Address** | **City** |
| Hansen | Ola | Timoteivn 10 | Sandnes |
| Svendson | Tove | Borgvn 23 | Sandnes |
| Pettersen | Kari | Storgt 20 | Stavanger |

上面的表含有三个记录（每个记录是一个人）和四个列（LastName, FirstName, Address 以及 City）。

**查询**

查询是一种询问或请求。

通过 MySQL，我们可以向数据库查询具体的信息，并得到返回的记录集。

请看下面的查询：

SELECT LastName FROM Persons

上面的查询选取了 Persons 表中 LastName 列的所有数据，并返回类似这样的记录集：

|  |
| --- |
| **LastName** |
| Hansen |
| Svendson |
| Pettersen |

**下载 MySQL 数据库**

如果您的 PHP 服务器没有 MySQL 数据库，可以在此下载 MySQL：<http://www.mysql.com/downloads/index.html>

**Facts About MySQL Database**

关于 MySQL 的一点很棒的特性是，可以对它进行缩减，来支持嵌入的数据库应用程序。也许正因如此，许多人认为 MySQL 仅仅能处理中小型的系统。

事实上，对于那些支持巨大数据和访问量的网站，MySQL 是事实上的标准数据库（比如 Friendster, Yahoo, Google）。这个地址提供了使用 MySQL 的公司的概览：<http://www.mysql.com/customers/>。

# MySQL 连接数据库

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

**免费的 MySQL 数据库通常是通过 PHP 来使用的。**

## 连接到一个 MySQL 数据库

在您能够访问并处理数据库中的数据之前，您必须创建到达数据库的连接。

在 PHP 中，这个任务通过 mysql\_connect() 函数完成。

### 语法

mysql\_connect(servername,username,password);

|  |  |
| --- | --- |
| **参数** | **描述** |
| servername | 可选。规定要连接的服务器。默认是 "localhost:3306"。 |
| username | 可选。规定登录所使用的用户名。默认值是拥有服务器进程的用户的名称。 |
| password | 可选。规定登录所用的密码。默认是 ""。 |

注释：虽然还存在其他的参数，但上面列出了最重要的参数。请访问 W3School 提供的 [PHP MySQL 参考手册](http://www.w3school.com.cn/php/php_ref_mysql.asp)，获得更多的细节信息。

### 例子

在下面的例子中，我们在一个变量中 ($con) 存放了在脚本中供稍后使用的连接。如果连接失败，将执行 "die" 部分：

<?php

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

if (!$con)

{

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

}

// some code

?>

## 关闭连接

脚本一结束，就会关闭连接。如需提前关闭连接，请使用 mysql\_close() 函数。

<?php

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

if (!$con)

{

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

}

// some code

mysql\_close($con);

?>

# MySQL 创建数据库和表

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

**数据库存有一个或多个表。**

## 创建数据库

CREATE DATABASE 语句用于在 MySQL 中创建数据库。

### 语法

CREATE DATABASE database\_name

为了让 PHP 执行上面的语句，我们必须使用 mysql\_query() 函数。此函数用于向 MySQL 连接发送查询或命令。

### 例子

在下面的例子中，我们创建了一个名为 "my\_db" 的数据库：

<?php

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

if (!$con)

{

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

}

if (mysql\_query("CREATE DATABASE my\_db",$con))

{

echo "Database created";

}

else

{

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

}

mysql\_close($con);

?>

## 创建表

CREATE TABLE 用于在 MySQL 中创建数据库表。

### 语法

CREATE TABLE table\_name

(

column\_name1 data\_type,

column\_name2 data\_type,

column\_name3 data\_type,

.......

)

为了执行此命令，我必须向 mysql\_query() 函数添加 CREATE TABLE 语句。

### 例子

下面的例子展示了如何创建一个名为 "person" 的表，此表有三列。列名是 "FirstName", "LastName" 以及 "Age"：

<?php

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

if (!$con)

{

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

}

// Create database

if (mysql\_query("CREATE DATABASE my\_db",$con))

{

echo "Database created";

}

else

{

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

}

// Create table in my\_db database

mysql\_select\_db("my\_db", $con);

$sql = "CREATE TABLE person

(

FirstName varchar(15),

LastName varchar(15),

Age int

)";

mysql\_query($sql,$con);

mysql\_close($con);

?>

重要事项：在创建表之前，必须首先选择数据库。通过 mysql\_select\_db() 函数选取数据库。

注释：当您创建 varchar 类型的数据库字段时，必须规定该字段的最大长度，例如：varchar(15)。

## MySQL 数据类型

下面的可使用的各种 MySQL 数据类型：

|  |  |
| --- | --- |
| **数值类型** | **描述** |
| * int(size) * smallint(size) * tinyint(size) * mediumint(size) * bigint(size) | 仅支持整数。在 size 参数中规定数字的最大值。 |
| * decimal(size,d) * double(size,d) * float(size,d) | 支持带有小数的数字。  在 size 参数中规定数字的最大值。在 d 参数中规定小数点右侧的数字的最大值。 |

|  |  |
| --- | --- |
| **文本数据类型** | **描述** |
| char(size) | 支持固定长度的字符串。（可包含字母、数字以及特殊符号）。  在 size 参数中规定固定长度。 |
| varchar(size) | 支持可变长度的字符串。（可包含字母、数字以及特殊符号）。  在 size 参数中规定最大长度。 |
| tinytext | 支持可变长度的字符串，最大长度是 255 个字符。 |
| * text * blob | 支持可变长度的字符串，最大长度是 65535 个字符。 |
| * mediumtext * mediumblob | 支持可变长度的字符串，最大长度是 16777215 个字符。 |
| * longtext * longblob | 支持可变长度的字符串，最大长度是 4294967295 个字符。 |

|  |  |
| --- | --- |
| **日期数据类型** | **描述** |
| * date(yyyy-mm-dd) * datetime(yyyy-mm-dd hh:mm:ss) * timestamp(yyyymmddhhmmss) * time(hh:mm:ss) | 支持日期或时间 |

|  |  |
| --- | --- |
| **杂项数据类型** | **描述** |
| enum(value1,value2,ect) | ENUM 是 ENUMERATED 列表的缩写。可以在括号中存放最多 65535 个值。 |
| set | SET 与 ENUM 相似。但是，SET 可拥有最多 64 个列表项目，并可存放不止一个 choice |

## 主键和自动递增字段

每个表都应有一个主键字段。

主键用于对表中的行进行唯一标识。每个主键值在表中必须是唯一的。此外，主键字段不能为空，这是由于数据库引擎需要一个值来对记录进行定位。

主键字段永远要被编入索引。这条规则没有例外。你必须对主键字段进行索引，这样数据库引擎才能快速定位给予该键值的行。

下面的例子把 personID 字段设置为主键字段。主键字段通常是 ID 号，且通常使用 AUTO\_INCREMENT 设置。AUTO\_INCREMENT 会在新纪录被添加时逐一增加该字段的值。要确保主键字段不为空，我们必须向该字段添加 NOT NULL 设置。

### 例子

$sql = "CREATE TABLE person

(

personID int NOT NULL AUTO\_INCREMENT,

PRIMARY KEY(personID),

FirstName varchar(15),

LastName varchar(15),

Age int

)";

mysql\_query($sql,$con);

# MySQL Insert Into

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

**INSERT INTO 语句用于向数据库表中插入新纪录。**

## 向数据库表插入数据

INSERT INTO 语句用于向数据库表添加新纪录。

### 语法

INSERT INTO table\_name

VALUES (value1, value2,....)

您还可以规定希望在其中插入数据的列：

INSERT INTO table\_name (column1, column2,...)

VALUES (value1, value2,....)

注释：SQL 语句对大小写不敏感。INSERT INTO 与 insert into 相同。

为了让 PHP 执行该语句，我们必须使用 mysql\_query() 函数。该函数用于向 MySQL 连接发送查询或命令。

### 例子

在前面的章节，我们创建了一个名为 "Person" 的表，有三个列："Firstname", "Lastname" 以及 "Age"。我们将在本例中使用同样的表。下面的例子向 "Person" 表添加了两个新纪录：

<?php

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

if (!$con)

{

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

}

mysql\_select\_db("my\_db", $con);

mysql\_query("INSERT INTO person (FirstName, LastName, Age)

VALUES ('Peter', 'Griffin', '35')");

mysql\_query("INSERT INTO person (FirstName, LastName, Age)

VALUES ('Glenn', 'Quagmire', '33')");

mysql\_close($con);

?>

## 把来自表单的数据插入数据库

现在，我们创建一个 HTML 表单，这个表单可把新纪录插入 "Person" 表。

这是这个 HTML 表单：

<html>

<body>

<form action="insert.php" method="post">

Firstname: <input type="text" name="firstname" />

Lastname: <input type="text" name="lastname" />

Age: <input type="text" name="age" />

<input type="submit" />

</form>

</body>

</html>

当用户点击上例中 HTML 表单中的提交按钮时，表单数据被发送到 "insert.php"。"insert.php" 文件连接数据库，并通过 $\_POST 变量从表单取回值。然后，mysql\_query() 函数执行 INSERT INTO 语句，一条新的记录会添加到数据库表中。

下面是 "insert.php" 页面的代码：

<?php

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

if (!$con)

{

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

}

mysql\_select\_db("my\_db", $con);

$sql="INSERT INTO person (FirstName, LastName, Age)

VALUES

('$\_POST[firstname]','$\_POST[lastname]','$\_POST[age]')";

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

{

die('Error: ' . mysql\_error());

}

echo "1 record added";

mysql\_close($con)

?>

# MySQL Select

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

**SELECT 语句用于从数据库中选取数据。**

## 从数据库表中选取数据

SELECT 语句用于从数据库中选取数据。

### 语法

SELECT column\_name(s) FROM table\_name

注释：SQL 语句对大小写不敏感。SELECT 与 select 等效。

为了让 PHP 执行上面的语句，我们必须使用 mysql\_query() 函数。该函数用于向 MySQL 发送查询或命令。

### 例子

下面的例子选取存储在 "Person" 表中的所有数据（\* 字符选取表中所有数据）：

<?php

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

if (!$con)

{

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

}

mysql\_select\_db("my\_db", $con);

$result = mysql\_query("SELECT \* FROM person");

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

{

echo $row['FirstName'] . " " . $row['LastName'];

echo "<br />";

}

mysql\_close($con);

?>

上面这个例子在 $result 变量中存放由 mysql\_query() 函数返回的数据。接下来，我们使用 mysql\_fetch\_array() 函数以数组的形式从记录集返回第一行。每个随后对 mysql\_fetch\_array() 函数的调用都会返回记录集中的下一行。 while loop 语句会循环记录集中的所有记录。为了输出每行的值，我们使用了 PHP 的 $row 变量 ($row['FirstName'] 和 $row['LastName'])。

以上代码的输出：

Peter Griffin

Glenn Quagmire

## 在 HTML 表格中显示结果

下面的例子选取的数据与上面的例子相同，但是将把数据显示在一个 HTML 表格中：

<?php

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

if (!$con)

{

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

}

mysql\_select\_db("my\_db", $con);

$result = mysql\_query("SELECT \* FROM person");

echo "<table border='1'>

<tr>

<th>Firstname</th>

<th>Lastname</th>

</tr>";

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

{

echo "<tr>";

echo "<td>" . $row['FirstName'] . "</td>";

echo "<td>" . $row['LastName'] . "</td>";

echo "</tr>";

}

echo "</table>";

mysql\_close($con);

?>

以上代码的输出：

|  |  |
| --- | --- |
| **Firstname** | **Lastname** |
| Glenn | Quagmire |
| Peter | Griffin |

# MySQL Where 子句

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

**如需选取匹配指定条件的数据，请向 SELECT 语句添加 WHERE 子句。**

## WHERE 子句

如需选取匹配指定条件的数据，请向 SELECT 语句添加 WHERE 子句。

### 语法

SELECT column FROM table

WHERE column operator value

下面的运算符可与 WHERE 子句一起使用：

|  |  |
| --- | --- |
| **运算符** | **说明** |
| = | 等于 |
| != | 不等于 |
| > | 大于 |
| < | 小于 |
| >= | 大于或等于 |
| <= | 小于或等于 |
| BETWEEN | 介于一个包含范围内 |
| LIKE | 搜索匹配的模式 |

注释：SQL 语句对大小写不敏感。WHERE 与 where 等效。

为了让 PHP 执行上面的语句，我们必须使用 mysql\_query() 函数。该函数用于向 SQL 连接发送查询和命令。

### 例子

下面的例子将从 "Person" 表中选取所有 FirstName='Peter' 的行：

<?php

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

if (!$con)

{

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

}

mysql\_select\_db("my\_db", $con);

$result = mysql\_query("SELECT \* FROM person

WHERE FirstName='Peter'");

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

{

echo $row['FirstName'] . " " . $row['LastName'];

echo "<br />";

}

?>

以上代码的输出：

Peter Griffin

# MySQL Order By 关键词

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

**ORDER BY 关键词用于对记录集中的数据进行排序。**

## ORDER BY 关键词

ORDER BY 关键词用于对记录集中的数据进行排序。

### 语法

SELECT column\_name(s)

FROM table\_name

ORDER BY column\_name

注释：SQL 对大小写不敏感。ORDER BY 与 order by 等效。

### 例子

下面的例子选取 "Person" 表中的存储的所有数据，并根据 "Age" 列对结果进行排序：

<?php

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

if (!$con)

{

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

}

mysql\_select\_db("my\_db", $con);

$result = mysql\_query("SELECT \* FROM person ORDER BY age");

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

{

echo $row['FirstName'];

echo " " . $row['LastName'];

echo " " . $row['Age'];

echo "<br />";

}

mysql\_close($con);

?>

以上代码的输出：

Glenn Quagmire 33

Peter Griffin 35

## 升序或降序的排序

如果您使用 ORDER BY 关键词，记录集的排序顺序默认是升序（1 在 9 之前，"a" 在 "p" 之前）。

请使用 DESC 关键词来设定降序排序（9 在 1 之前，"p" 在 "a" 之前）：

SELECT column\_name(s)

FROM table\_name

ORDER BY column\_name DESC

## 根据两列进行排序

可以根据多个列进行排序。当按照多个列进行排序时，只有第一列相同时才使用第二列：

SELECT column\_name(s)

FROM table\_name

ORDER BY column\_name1, column\_name2

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

# MySQL Update

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

**UPDATE 语句用于中修改数据库表中的数据。**

## 更新数据库中的数据

UPDATE 语句用于在数据库表中修改数据。

### 语法

UPDATE table\_name

SET column\_name = new\_value

WHERE column\_name = some\_value

注释：SQL 对大小写不敏感。UPDATE 与 update 等效。

为了让 PHP 执行上面的语句，我们必须使用 mysql\_query( 函数。该函数用于向 SQL 连接发送查询和命令。

### 例子

稍早时，我们在本教程中创建了一个名为 "Person" 的表。它看起来类似这样：

|  |  |  |
| --- | --- | --- |
| **FirstName** | **LastName** | **Age** |
| Peter | Griffin | 35 |
| Glenn | Quagmire | 33 |

下面的例子更新 "Person" 表的一些数据：

<?php

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

if (!$con)

{

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

}

mysql\_select\_db("my\_db", $con);

mysql\_query("UPDATE Person SET Age = '36'

WHERE FirstName = 'Peter' AND LastName = 'Griffin'");

mysql\_close($con);

?>

在这次更新后，"Person" 表格是这样的：

|  |  |  |
| --- | --- | --- |
| **FirstName** | **LastName** | **Age** |
| Peter | Griffin | 36 |
| Glenn | Quagmire | 33 |

# MySQL Delete From

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

**DELETE FROM 语句用于从数据库表中删除行。**

## 删除数据库中的数据

DELETE FROM 语句用于从数据库表中删除记录。

### 语法

DELETE FROM table\_name

WHERE column\_name = some\_value

注释：SQL 对大小写不敏感。DELETE FROM 与 delete from 等效。

为了让 PHP 执行上面的语句，我们必须使用 mysql\_query( 函数。该函数用于向 SQL 连接发送查询和命令。

### 例子

稍早时，我们在本教程中创建了一个名为 "Person" 的表。它看起来类似这样：

|  |  |  |
| --- | --- | --- |
| **FirstName** | **LastName** | **Age** |
| Peter | Griffin | 35 |
| Glenn | Quagmire | 33 |

下面的例子删除 "Person" 表中所有 LastName='Griffin' 的记录：

<?php

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

if (!$con)

{

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

}

mysql\_select\_db("my\_db", $con);

mysql\_query("DELETE FROM Person WHERE LastName='Griffin'");

mysql\_close($con);

?>

在这次删除之后，表是这样的：

|  |  |  |
| --- | --- | --- |
| **FirstName** | **LastName** | **Age** |
| Glenn | Quagmire | 33 |

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

# Database ODBC

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

**ODBC 指的是（Application Programming Interface，API），使我们有能力连接到某个数据源（比如一个 MS Access 数据库）。**

## 创建 ODBC 连接

通过一个 ODBC 连接，您可以连接到您的网络中的任何计算机上的任何数据库，只要 ODBC 连接是可用的。

这是创建到达 MS Access 数据的 ODBC 连接的方法：

1. 在控制面板中打开管理工具
2. 双击其中的数据源 (ODBC) 图标
3. 选择系统 DSN 选项卡
4. 点击系统 DSN 选项卡中的“添加”按钮
5. 选择 Microsoft Access Driver。点击完成。
6. 在下一个界面，点击“选择”来定位数据库。
7. 为这个数据库取一个数据源名 (DSN)。
8. 点击确定。

请注意，必须在您的网站所在的计算机上完成这个配置。如果您的计算机上正在运行 Internet 信息服务器 (IIS)，上面的指令会生效，但是假如您的网站位于远程服务器，您必须拥有对该服务器的物理访问权限，或者请您的主机提供商为您建立 DSN。

## 连接到 ODBC

odbc\_connect() 函数用于连接到 ODBC 数据源。该函数有四个参数：数据源名、用户名、密码以及可选的指针类型参数。

odbc\_exec() 函数用于执行 SQL 语句。

### 例子

下面的例子创建了到达名为 northwind 的 DSN 的连接，不没有用户名和密码。然后创建并执行一条 SQL 语句：

$conn=odbc\_connect('northwind','','');

$sql="SELECT \* FROM customers";

$rs=odbc\_exec($conn,$sql);

## 取回记录

odbc\_fetch\_row() 函数用于从结果集中返回记录。如果能够返回行，则返回 true，否则返回 false。

该函数有两个参数：ODBC 结果标识符和可选的行号：

odbc\_fetch\_row($rs)

## 从记录中取回字段

odbc\_result() 函数用于从记录中读取字段。该函数有两个参数：ODBC 结果标识符和字段编号或名称。

下面的代码行从记录中返回第一个字段的值：

$compname=odbc\_result($rs,1);

The code line below returns the value of a field called "CompanyName":

$compname=odbc\_result($rs,"CompanyName");

## 关闭 ODBC 连接

odbc\_close()函数用于关闭 ODBC 连接。

odbc\_close($conn);

## ODBC 实例

下面的例子展示了如何首先创建一个数据库连接，然后是结果集，然后在 HTML 表格中显示数据。

<html>

<body>

<?php

$conn=odbc\_connect('northwind','','');

if (!$conn)

{exit("Connection Failed: " . $conn);}

$sql="SELECT \* FROM customers";

$rs=odbc\_exec($conn,$sql);

if (!$rs)

{exit("Error in SQL");}

echo "<table><tr>";

echo "<th>Companyname</th>";

echo "<th>Contactname</th></tr>";

while (odbc\_fetch\_row($rs))

{

$compname=odbc\_result($rs,"CompanyName");

$conname=odbc\_result($rs,"ContactName");

echo "<tr><td>$compname</td>";

echo "<td>$conname</td></tr>";

}

odbc\_close($conn);

echo "</table>";

?>

</body>

</html>