

RDBMS 术语

在我们开始学习MySQL 数据库前，让我们先了解下RDBMS的一些术语：

- **数据库**: 数据库是一些关联表的集合。
- **数据表**: 表是数据的矩阵。在一个数据库中的表看起来像一个简单的电子表格。
- **列**: 一列(数据元素) 包含了相同类型的数据, 例如邮政编码的数据。
- **行**: 一行 (=元组, 或记录) 是一组相关的数据, 例如一条用户订阅的数据。
- **冗余**: 存储两倍数据, 冗余降低了性能, 但提高了数据的安全性。
- **主键**: 主键是唯一的。一个数据表中只能包含一个主键。你可以使用主键来查询数据。
- **外键**: 外键用于关联两个表。
- **复合键**: 复合键 (组合键) 将多个列作为一个索引键, 一般用于复合索引。
- **索引**: 使用索引可快速访问数据库表中的特定信息。索引是对数据库表中一列或多列的值进行排序的一种结构。类似于书籍的目录。
- **参照完整性**: 参照的完整性要求关系中不允许引用不存在的实体。与实体完整性是关系模型必须满足的完整性约束条件, 目的是保证数据的一致性。

MySQL 为关系型数据库(Relational Database Management System), 这种所谓的"关系型"可以理解
为"表格"的概念, 一个关系型数据库由一个或数个表格组成。

- 表头(header): 每一列的名称;
- 列(col): 具有相同数据类型的数据的集合;
- 行(row): 每一行用来描述某条记录的具体信息;
- 值(value): 行的具体信息, 每个值必须与该列的数据类型相同;
- 键(key): 键的值在当前列中具有唯一性。

MySQL数据库

MySQL 是一个关系型数据库管理系统, 由瑞典 MySQL AB 公司开发, 目前属于 Oracle 公司。
MySQL 是一种关联数据库管理系统, 关联数据库将数据保存在不同的表中, 而不是将所有数据放在
一个大仓库内, 这样就增加了速度并提高了灵活性。

- MySQL 是开源的, 所以你不需要支付额外的费用。
- MySQL 支持大型的数据库。可以处理拥有上千万条记录的大型数据库。
- MySQL 使用标准的 SQL 数据语言形式。
- MySQL 可以运行于多个系统上, 并且支持多种语言。这些编程语言包括 C、C++、Python、Java、Perl、PHP、Eiffel、Ruby 和 Tcl 等。
- MySQL 对PHP有很好的支持, PHP 是目前最流行的 Web 开发语言。
- MySQL 支持大型数据库, 支持 5000 万条记录的数据仓库, 32 位系统表文件最大可支持 4GB, 64 位系统支持最大的表文件为8TB。
- MySQL 是可以定制的, 采用了 GPL 协议, 你可以修改源码来开发自己的 MySQL 系统。

MySQL PHP 语法

MySQL 可应用于多种语言, 包括 PERL, C, C++, JAVA 和 PHP。在这些语言中, Mysql在PHP的
web开发中是应用最广泛。

在本教程中我们大部分实例都采用了 PHP 语言。如果你想了解 Mysql 在 PHP 中的应用，可以访问我们的 [PHP 中使用 Mysqli 介绍](#)。

PHP提供了多种方式来访问和操作Mysql数据库记录。PHP **Mysqli**函数格式如下：

```
mysqli_function(value,value,...);
```

以上格式中 function部分描述了mysql函数的功能，如

```
mysqli_connect($connect);  
mysqli_query($connect,"SQL 语句");  
mysqli_fetch_array()  
mysqli_close()
```

MySQL 连接

使用mysql二进制方式连接

您可以使用MySQL二进制方式进入到mysql命令提示符下来连接MySQL数据库。

实例

以下是从命令行中连接mysql服务器的简单实例：

```
[root@host]# mysql -u root -p  
Enter password:*****
```

在登录成功后会出现 mysql> 命令提示窗口，你可以在上面执行任何 SQL 语句。

以上命令执行后，登录成功输出结果如下：

```
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 2854760 to server version: 5.0.9  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

在以上实例中，我们使用了root用户登录到mysql服务器，当然你也可以使用其他mysql用户登录。

如果用户权限足够，任何用户都可以在mysql的命令提示窗口中进行SQL操作。

退出 mysql> 命令提示窗口可以使用 exit 命令，如下所示：

```
mysql> exit  
Bye
```

使用 PHP 脚本连接 MySQL

PHP 提供了 mysqli_connect() 函数来连接数据库。

该函数有 6 个参数，在成功链接到 MySQL 后返回连接标识，失败返回 FALSE。

语法

```
mysqli_connect(host,username,password,dbname,port,socket);
```

参数说明：

参数	描述
host	可选。规定主机名或 IP 地址。
username	可选。规定 MySQL 用户名。
password	可选。规定 MySQL 密码。
dbname	可选。规定默认使用的数据库。
port	可选。规定尝试连接到 MySQL 服务器的端口号。
socket	可选。规定 socket 或要使用的已命名 pipe。

你可以使用 PHP 的 mysqli_close() 函数来断开与 MySQL 数据库的链接。

该函数只有一个参数为 mysqli_connect() 函数创建连接成功后返回的 MySQL 连接标识符。

语法

```
bool mysqli_close ( mysqli $link )
```

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。

提示：通常不需要使用 mysqli_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。

实例

你可以尝试以下实例来连接到你的 MySQL 服务器:

连接 MySQL

```
<?php
$dbhost = 'localhost'; // mysql服务器主机地址
$dbuser = 'root';      // mysql用户名
$dbpass = '123456';    // mysql用户名密码
$conn = mysqli_connect($dbhost, $dbuser, $dbpass);
if(! $conn )
{
    die('Could not connect: ' . mysqli_error());
}
echo '数据库连接成功!';
mysqli_close($conn);
?>
```