

### mysql数据库基础教程

数据库从最初的数据文件的简单集合发展到今天的大型数据库管理系统已经成为我们日常生活中不可缺少的组成部分。如果不借助数据库的帮助，许多简单的工作将变得冗长乏味，甚至难以实现。尤其是象银行、院校和图书馆这样的大型组织更加依靠数据库系统实现其正常的运作。再看互联网上，从搜索引擎到在线商场，从网上聊天到邮件列表，都离不开数据库。

总体来说，mysql数据库具有以下主要特点：

1. 同时访问数据库的用户数量不受限制；
2. 可以保存超过50,000,000条记录；
3. 是目前市场上现有产品中运行速度最快的数据库系统；
4. 用户权限设置简单、有效。

如今，包括siemens和silicon graphics这样的国际知名公司也开始把mysql作为其数据库管理系统，这就更加证明了mysql数据库的优越性能和广阔的市场发展前景。

本文将重点向读者介绍mysql数据库的一些基本操作，包括如何与数据库建立连接，如果设置数据库，以及如何执行基本的命令等。希望能够对读者学习和掌握mysql数据库有所助益。

### 入门

一般来说，我们访问mysql数据库时，首先需要使用telnet远程登录安装数据库系统的服务器，然后再进入mysql数据库。mysql数据库的连接命令如下：

```
mysql -h hostname -u username -p[password]
```

或者：

```
mysql -h hostname -u username --password=password
```

其中，hostname为装有mysql数据库的服务器名称，username和password分别是用户的登录名称和口令。

如果mysql数据库安装和配置正确的话，用户在输入上述命令之后会得到如下系统反馈信息：

```
welcome to the mysql monitor. commands end with ; or g.
```

```
your mysql c id is 49 to server versi 3.2 3-beta-log
```

```
type help for help.
```

```
mysql$#@62;
```

这样，用户就成功进入了mysql数据库系统，可以在mysql\$#@62;命令提示符之后输入各种命令。下面，我们列出一些mysql数据库的主要管理命令供读者参考，用户也可以通过在命令符之后输入help，h或?得到以下命令的简单说明。

```
mysql$#@62; help
```

help (h) 显示命令帮助

? (h) 作用同上

clear (c) 清除屏幕内容

c (r) 重新连接服务器，可选参数为 db（数据库）和 host（服务器）

exit () 退出mysql数据库，作用与quit命令相同

go (g) 将命令传送至mysql数据库

print (p) 打印当前命令

quit (q) 退出mysql数据库

status (s) 显示服务器当前信息

use (u) 打开数据库，以数据库名称作为命令参数

上述命令主要用于mysql数据库的系统管理，如果用户需要对某个具体的数据库进行操作，可以使用use命令进入该数据库，格式如下：

```
mysql$#@62; use dbname;
```

这里需要提醒读者注意的一点就是mysql数据库要求使用者在所有命令的结尾处使用“;”作为命令结束符。

## 数据类型和数据表

从本质上说，数据库就是一种不断增长的复杂的数据组织结构。在mysql数据库中，用于保存数据记录的结构被称为数据表。而每一条数据记录则是由更小的数据对象，即数据类型组成。因此，总体来说，一个或多个数据类型组成一条数据记录，一条或多条数据记录组成一个数据表，一个或多个数据表组成一个数据库。我们可以把上述结构理解为如下形式：

```
database $#@60; table $#@60; record $#@60; datatype
```

数据类型分为不同的格式和大小，可以方便数据库的设计人员创建最理想的数据结构。能否正确的选择恰当的数据类型对最终数据库的性能具有重要的影响，因此，我们有必要首先对数据类型的有关概念进行较为详细的介绍。

## mysql 数据类型

mysql 数据库提供了多种数据类型，其中较为常用的几种如下：

**char (m)**

char数据类型用于表示固定长度的字符串，可以包含最多达255个字符。其中m代表字符串的长度。

举例如下：

```
car_model char(10);
```

**varchar (m)**

varchar是一种比char更加灵活的数据类型，同样用于表示字符数据，但是varchar可以保存可变长度的字符串。其中m代表该数据类型所允许保存的字符串的最大长度，只要长度小于该最大值的字符串都可以被保存在该数据类型中。因此，对于那些难以估计确切长度的数据对象来说，使用varchar数据类型更加明智。varchar数据类型所支持的最大长度也是255个字符。

这里需要提醒读者注意的一点是，虽然varchar使用起来较为灵活，但是从整个系统的性能角度来说，char数据类型的处理速度更快，有时甚至可以超出varchar处理速度的50%。因此，用户在设计数据库时应当综合考虑各方面的因素，以求达到最佳的平衡。

举例如下：

```
car_model varchar(10);
```

**int (m) [unsigned]**

int数据类型用于保存从- 2147483647 到2147483648范围内的任意整数数据。如果用户使用unsigned选项，则有效数据范围调整为0-4294967295。举例如下：

```
light_years int;
```

按照上述数据类型的设置，-24567为有效数据，而3000000000则因为超出了有效数据范围成为无效数据。

再例如：

```
light_years int unsigned;
```

这时，3000000000成为有效数据，而-24567则成为无效数据。

**float [(m,d)]**

float数据类型用于表示数值较小的浮点数据，可以提供更加准确的数据精度。其中，m代表浮点数据的长度（即小数点左右数据长度的总和），d表示浮点数据位于小数点右边的数值位数。

举例如下：

```
rainfall float (4,2);
```

按照上述数据类型的设置，42.35为有效数据，而324.45和3.542则因为超过数据长度限制或者小数点右边位数大于规定值2成为无效数据。

**date**

date数据类型用于保存日期数据，默认格式为yyyy-mm-dd。mysql提供了许多功能强大的日期格式化和操作命令，本文无法在此一一进行介绍，感兴趣的读者可以参看mysql的技术文档。

date数据类型举例如下：

the\_date date;

text / blob

text和blob数据类型可以用来保存255 到65535个字符，如果用户需要把大段文本保存到数据库内的话，可以选用text或blob数据类型。text和blob这两种数据类型基本相同，唯一的区别在于text不区分大小写，而blob对字符的大小写敏感。

set

set数据类型是多个数据值的组合，任何部分或全部数据值都是该数据类型的有效数据。set数据类型最大可以包含64个指定数据值。

举例如下：

transport set ("truck", "wag not null;

根据上述数据类型的设置，truck、wag

enum

enum数据类型和set基本相同，唯一的区别在于enum只允许选择一个有效数据值。例如：

transport enum ("truck", "wag not null;

根据上述设置，truck或wag

以上，我们只是对用户使用mysql数据库的过程中经常用到的数据类型进行了简单介绍，有兴趣的读者，可以参看mysql技术文档的详细说明。

## 数据记录

一组经过声明的数据类型就可以组成一条记录。记录小到可以只包含一个数据变量，大到可以满足用户的各种复杂需求。多条记录组合在一起就构成了数据表的基本结构。

## 数据表

在我们执行各种数据库命令之前，首先需要创建用来保存信息的数据表。我们可以通过以下方式在mysql数据库中创建新的数据表：

```
mysql$#@62; create table test (  
$#@62; name varchar (15),  
$#@62; email varchar (25),  
$#@62; ph int,  
$#@62; id int not null auto_increment,  
$#@62; primary key (id));
```

系统反馈信息为：

```
query ok, 0 rows affected (0.10 sec)
```

```
mysql$#@62;
```

这样，我们就在数据库中创建了一个新的数据表。注意，同一个数据库中不能存在两个名称相同的数据表。

这里，我们使用create table命令创建的test数据表中包含name，email，ph

下面，我们来看一看创建数据表时所用到的几个主要的参数选项。

primary key

具有primary key限制条件的字段用于区分同一个数据表中的不同记录。因为同一个数据表中不会存在两个具有相同值的primary key字段，所以对于那些需要严格区分不同记录的数据表来说，primary key具有相当重要的作用。

auto\_increment

具有auto\_increment限制条件的字段值从1开始，每增加一条新记录，值就会相应地增加1。一般来说，我们可以把auto\_increment字段作为数据表中每一条记录的标识字段。

not null

not null限制条件规定用户不得在该字段中插入空值。

#### 其它数据表命令

除了创建新的数据表之外，mysql数据库还提供了其它许多非常实用的以数据表作为操作对象的命令。

#### 显示数据表命令

```
mysql$#@62; show tables;
```

该命令将会列出当前数据库下的所有数据表。

#### 显示字段命令

```
mysql$#@62; show columns from tablename;
```

该命令将会返回指定数据表的所有字段和字段相关信息。

#### 数据操作

对mysql数据库中数据的操作可以划分为四种不同的类型，分别是添加、删除、修改和查询，我们将会在本节中对此进行介绍。但是，首先我们需要强调的一点就是mysql数据库所采用的sql语言同其它绝大多数计算机编程语言一样，对命令的语法格式有严格的规定。任何语法格式上的错误，例如不正确的使用括号、逗号或分号等都可能致命命令执行过程中的错误。因此，建议用户在学习时一定要多留心语法格式的使用。

#### 添加记录

用户可以使用insert命令向数据库中添加新的记录。

例如：

```
mysql$#@62; insert into test values  
mysql$#@62; (john, carrots@mail.com,  
mysql$#@62; 5554321, null);
```

上述命令正确执行后会返回以下信息：

```
query ok, 1 row affected (0.02 sec)  
mysql$#@62;
```

对上述命令有几点我们需要说明。首先，所有的字符类型数据都必须使用单引号括起来。其次，null关键字与auto\_increment限制条件相结合可以为字段自动赋值。最后，也是最重要的一点就是新记录的字段值必须与数据表中的原字段相对应，如果原数据表中有4个字段，而用户所添加的记录包含3个或5个字段的话都会导致错误出现。

mysql数据库的一个非常显著的优势就是可以对整数、字符串和日期数据进行自动转换。因此，用户在添加新记录时就不必担心因为数据类型不相符而出现错误。

#### 查询数据

如果我们无法从数据库中查找和读取数据的话，数据库就丧失了其存在和使用的价值。

在mysql数据库中，用户可以使用select命令进行数据的查询。

例如：

```
mysql$#@62; select * from test  
mysql$#@62; where (name = "john");
```

上述命令会返回如下结果：

name

email

ph

id

john

carrots@mail.com

5554321

1

### 删除数据

用户除了可以向数据表中添加新的记录之外，还可以删除数据表中的已有记录。删除记录可以使用delete命令。

例如：

```
mysql$#@62; delete from test
```

```
mysql$#@62; where (name = "");
```

该命令将会删除test数据表中name字段的值为john的记录。同样，

```
mysql$#@62; delete from test
```

```
mysql$#@62; where (ph = 5554321);
```

将会从数据表中删除ph

### 修改数据

mysql数据库还支持用户对已经输入到数据表中的数据进行修改。修改记录可以使用update命令。

例如：

```
mysql$#@62; update test set name = mary
```

```
mysql$#@62; where name = "john";
```

上述命令的执行结果如下：

name

email

ph

id

mary

carrots@mail.com

5554321

1