摘自:

http://www.runoob.com/mysql/mysql-administration.html

# MySQL 教程



Mysql是最流行的关系型数据库管理系统，在WEB应用方面MySQL是最好的RDBMS(Relational Database Management System：关系数据库管理系统)应用软件之一。

在本教程中，会让大家快速掌握Mysql的基本知识，并轻松使用Mysql数据库。

## 什么是数据库？

数据库（Database）是按照数据结构来组织、存储和管理数据的仓库，

每个数据库都有一个或多个不同的API用于创建，访问，管理，搜索和复制所保存的数据。

我们也可以将数据存储在文件中，但是在文件中读写数据速度相对较慢。

所以，现在我们使用关系型数据库管理系统（RDBMS）来存储和管理的大数据量。所谓的关系型数据库，是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。

RDBMS即关系数据库管理系统(Relational Database Management System)的特点：

* 1.数据以表格的形式出现
* 2.每行为各种记录名称
* 3.每列为记录名称所对应的数据域
* 4.许多的行和列组成一张表单
* 5.若干的表单组成database

## 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，冗余可以使系统速度更快。
* **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可以允许于多个系统上，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 安装

所有平台的Mysql下载地址为： [MySQL 下载](http://www.mysql.com/downloads). 挑选你需要的 *MySQL Community Server* 版本及对应的平台。

## Linux/UNIX上安装Mysql

Linux平台上推荐使用RPM包来安装Mysql,MySQL AB提供了以下RPM包的下载地址：

* **MySQL** - MySQL服务器。你需要该选项，除非你只想连接运行在另一台机器上的MySQL服务器。
* **MySQL-client** - MySQL 客户端程序，用于连接并操作Mysql服务器。
* **MySQL-devel** - 库和包含文件，如果你想要编译其它MySQL客户端，例如Perl模块，则需要安装该RPM包。
* **MySQL-shared** - 该软件包包含某些语言和应用程序需要动态装载的共享库(libmysqlclient.so\*)，使用MySQL。
* **MySQL-bench** - MySQL数据库服务器的基准和性能测试工具。

以下安装Mysql RMP的实例是在SuSE Linux系统上进行，当然该安装步骤也适合应用于其他支持RPM的Linux系统，如:Centos。

安装步骤如下：

使用root用户登陆你的Linux系统。

下载Mysql RPM包，下载地址为：[MySQL 下载](http://www.mysql.com/downloads)。

通过以下命令执行Mysql安装，rpm包为你下载的rpm包：

[root@host]# rpm -i MySQL-5.0.9-0.i386.rpm

以上安装mysql服务器的过程会创建mysql用户，并创建一个mysql配置文件my.cnf。

你可以在/usr/bin和/usr/sbin中找到所有与MySQL相关的二进制文件。所有数据表和数据库将在/var/lib/mysql目录中创建。

以下是一些mysql可选包的安装过程，你可以根据自己的需要来安装：

[root@host]# rpm -i MySQL-client-5.0.9-0.i386.rpm

[root@host]# rpm -i MySQL-devel-5.0.9-0.i386.rpm

[root@host]# rpm -i MySQL-shared-5.0.9-0.i386.rpm

[root@host]# rpm -i MySQL-bench-5.0.9-0.i386.rpm

## Window上安装Mysql

Window上安装Mysql相对来说会较为简单，你只需要载 [MySQL 下载](http://www.mysql.com/downloads)中下载window版本的mysql安装包，并解压安装包。

双击 setup.exe 文件，接下来你只需要安装默认的配置点击"next"即可，默认情况下安装信息会在C:\mysql目录中。

接下来你可以通过"开始" =》在搜索框中输入 " cmd" 命令 =》 在命令提示符上切换到 C:\mysql\bin 目录，并输入一下命令：

mysqld.exe --console

如果安装成功以上命令将输出一些mysql启动及InnoDB信息。

## 验证Mysql安装

在成功安装Mysql后，一些基础表会表初始化，在服务器启动后，你可以通过简单的测试来验证Mysql是否工作正常。

使用 mysqladmin 工具来获取服务器状态：

使用 mysqladmin 命令俩检查服务器的版本,在linux上该二进制文件位于 /usr/bin on linux ，在window上该二进制文件位于C:\mysql\bin 。

[root@host]# mysqladmin --version

linux上该命令将输出以下结果，该结果基于你的系统信息：

mysqladmin Ver 8.23 Distrib 5.0.9-0, for redhat-linux-gnu on i386

如果以上命令执行后未输入任何信息，说明你的Mysql未安装成功。

## 使用 MySQL Client(Mysql客户端) 执行简单的SQL命令

你可以在 MySQL Client(Mysql客户端) 使用 mysql 命令连接到Mysql服务器上，默认情况下Mysql服务器的密码为空，所以本实例不需要输入密码。

命令如下：

[root@host]# mysql

以上命令执行后会输出 mysql>提示符，这说明你已经成功连接到Mysql服务器上，你可以在 mysql> 提示符执行SQL命令：

mysql> SHOW DATABASES;

+----------+

| Database |

+----------+

| mysql |

| test |

+----------+

2 rows in set (0.13 sec)

## Mysql安装后需要做的

Mysql安装成功后，默认的root用户密码为空，你可以使用以下命令来创建root用户的密码：

[root@host]# mysqladmin -u root password "new\_password";

现在你可以通过以下命令来连接到Mysql服务器：

[root@host]# mysql -u root -p

Enter password:\*\*\*\*\*\*\*

**注意：**在输入密码时，密码是不会显示了，你正确输入即可。

## Linux系统启动时启动 MySQL

如果你需要在Linux系统启动时启动 MySQL 服务器，你需要在 /etc/rc.local 文件中添加以下命令：

/etc/init.d/mysqld start

同样，你需要将 mysqld 二进制文件添加到 /etc/init.d/ 目录中。

# MySQL 管理

## 启动及关闭 MySQL 服务器

首先，我们需要通过以下命令来检查MySQL服务器是否启动：

ps -ef | grep mysqld

如果MySql已经启动，以上命令将输出mysql进程列表， 如果mysql未启动，你可以使用以下命令来启动mysql服务器:

root@host# cd /usr/bin

./mysqld\_safe &

如果你想关闭目前运行的 MySQL 服务器, 你可以执行以下命令:

root@host# cd /usr/bin

./mysqladmin -u root -p shutdown

Enter password: \*\*\*\*\*\*

## MySQL 用户设置

如果你需要添加 MySQL 用户，你只需要在 mysql 数据库中的 user 表添加新用户即可。

以下为添加用户的的实例，用户名为guest，密码为guest123，并授权用户可进行 SELECT, INSERT 和 UPDATE操作权限：

root@host# mysql -u root -p

Enter password:\*\*\*\*\*\*\*

mysql> use mysql;

Database changed

mysql> INSERT INTO user

(host, user, password,

select\_priv, insert\_priv, update\_priv)

VALUES ('localhost', 'guest',

PASSWORD('guest123'), 'Y', 'Y', 'Y');

Query OK, 1 row affected (0.20 sec)

mysql> FLUSH PRIVILEGES;

Query OK, 1 row affected (0.01 sec)

mysql> SELECT host, user, password FROM user WHERE user = 'guest';

+-----------+---------+------------------+

| host | user | password |

+-----------+---------+------------------+

| localhost | guest | 6f8c114b58f2ce9e |

+-----------+---------+------------------+

1 row in set (0.00 sec)

在添加用户时，请注意使用MySQL提供的 PASSWORD() 函数来对密码进行加密。 你可以在以上实例看到用户密码加密后为： 6f8c114b58f2ce9e.

**注意：**在 MySQL5.7 中 user 表的 password 已换成了**authentication\_string**。

**注意：**在注意需要执行 **FLUSH PRIVILEGES** 语句。 这个命令执行后会重新载入授权表。

如果你不使用该命令，你就无法使用新创建的用户来连接mysql服务器，除非你重启mysql服务器。

你可以在创建用户时，为用户指定权限，在对应的权限列中，在插入语句中设置为 'Y' 即可，用户权限列表如下：

* Select\_priv
* Insert\_priv
* Update\_priv
* Delete\_priv
* Create\_priv
* Drop\_priv
* Reload\_priv
* Shutdown\_priv
* Process\_priv
* File\_priv
* Grant\_priv
* References\_priv
* Index\_priv
* Alter\_priv

另外一种添加用户的方法为通过SQL的 GRANT 命令，你下命令会给指定数据库TUTORIALS添加用户 zara ，密码为 zara123 。

root@host# mysql -u root -p password;

Enter password:\*\*\*\*\*\*\*

mysql> use mysql;

Database changed

mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP

-> ON TUTORIALS.\*

-> TO 'zara'@'localhost'

-> IDENTIFIED BY 'zara123';

以上命令会在mysql数据库中的user表创建一条用户信息记录。

**注意:** MySQL 的SQL语句以分号 (;) 作为结束标识。

## /etc/my.cnf 文件配置

一般情况下，你不需要修改该配置文件，该文件默认配置如下：

[mysqld]

datadir=/var/lib/mysql

socket=/var/lib/mysql/mysql.sock

[mysql.server]

user=mysql

basedir=/var/lib

[safe\_mysqld]

err-log=/var/log/mysqld.log

pid-file=/var/run/mysqld/mysqld.pid

在配置文件中，你可以指定不同的错误日志文件存放的目录，一般你不需要改动这些配置。

## 管理MySQL的命令

以下列出了使用Mysql数据库过程中常用的命令：

* **USE 数据库名** :  
  选择要操作的Mysql数据库，使用该命令后所有Mysql命令都只针对该数据库。
* mysql> use RUNOOB;

Database changed

* **SHOW DATABASES:**  
  列出 MySQL 数据库管理系统的数据库列表。
* mysql> SHOW DATABASES;
* +--------------------+
* | Database |
* +--------------------+
* | information\_schema |
* | RUNOOB |
* | cdcol |
* | mysql |
* | onethink |
* | performance\_schema |
* | phpmyadmin |
* | test |
* | wecenter |
* | wordpress |
* +--------------------+

10 rows in set (0.02 sec)

* **SHOW TABLES:**  
  显示指定数据库的所有表，使用该命令前需要使用 use 命令来选择要操作的数据库。
* mysql> use RUNOOB;
* Database changed
* mysql> SHOW TABLES;
* +------------------+
* | Tables\_in\_runoob |
* +------------------+
* | employee\_tbl |
* | runoob\_tbl |
* | tcount\_tbl |
* +------------------+

3 rows in set (0.00 sec)

* **SHOW COLUMNS FROM 数据表:**  
  显示数据表的属性，属性类型，主键信息 ，是否为 NULL，默认值等其他信息。
* mysql> SHOW COLUMNS FROM runoob\_tbl;
* +-----------------+--------------+------+-----+---------+-------+
* | Field | Type | Null | Key | Default | Extra |
* +-----------------+--------------+------+-----+---------+-------+
* | runoob\_id | int(11) | NO | PRI | NULL | |
* | runoob\_title | varchar(255) | YES | | NULL | |
* | runoob\_author | varchar(255) | YES | | NULL | |
* | submission\_date | date | YES | | NULL | |
* +-----------------+--------------+------+-----+---------+-------+

4 rows in set (0.01 sec)

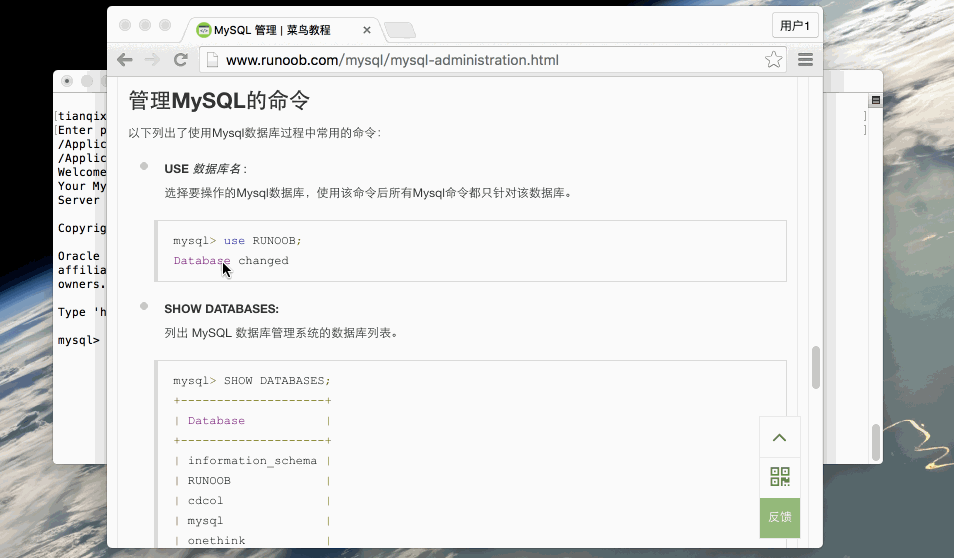
* **SHOW INDEX FROM 数据表:**  
  显示数据表的详细索引信息，包括PRIMARY KEY（主键）。
* mysql> SHOW INDEX FROM runoob\_tbl;
* +------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
* | Table | Non\_unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment | Index\_comment |
* +------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+
* | runoob\_tbl | 0 | PRIMARY | 1 | runoob\_id | A | 2 | NULL | NULL | | BTREE | | |
* +------------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+

1 row in set (0.00 sec)

* **SHOW TABLE STATUS LIKE [FROM db\_name] [LIKE 'pattern'] \G:**   
  该命令将输出Mysql数据库管理系统的性能及统计信息。
* mysql> SHOW TABLE STATUS FROM RUNOOB; # 显示数据库 RUNOOB 中所有表的信息
* mysql> SHOW TABLE STATUS from RUNOOB LIKE 'runoob%'; # 表名以runoob开头的表的信息

mysql> SHOW TABLE STATUS from RUNOOB LIKE 'runoob%'\G; # 加上 \G，查询结果按列打印

Gif 图演示：



# 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

# MySQL 创建数据库

## 使用 mysqladmin 创建数据库

使用普通用户，你可能需要特定的权限来创建或者删除 MySQL 数据库。

所以我们这边使用root用户登录，root用户拥有最高权限，可以使用 mysql mysqladmin 命令来创建数据库。

### 实例

以下命令简单的演示了创建数据库的过程，数据名为 RUNOOB:

[root@host]# mysqladmin -u root -p create RUNOOB

Enter password:\*\*\*\*\*\*

以上命令执行成功后会创建 MySQL 数据库 RUNOOB。

# MySQL 删除数据库

## 使用 mysqladmin 删除数据库

使用普通用户登陆mysql服务器，你可能需要特定的权限来创建或者删除 MySQL 数据库。

所以我们这边使用root用户登录，root用户拥有最高权限，可以使用 mysql mysqladmin 命令来创建数据库。

在删除数据库过程中，务必要十分谨慎，因为在执行删除命令后，所有数据将会消失。

以下实例删除数据库RUNOOB(该数据库在前一章节已创建)：

[root@host]# mysqladmin -u root -p drop RUNOOB

Enter password:\*\*\*\*\*\*

执行以上删除数据库命令后，会出现一个提示框，来确认是否真的删除数据库：

Dropping the database is potentially a very bad thing to do.

Any data stored in the database will be destroyed.

Do you really want to drop the 'RUNOOB' database [y/N] y

Database "RUNOOB" dropped

# MySQL 选择数据库

在你连接到 MySQL 数据库后，可能有多个可以操作的数据库，所以你需要选择你要操作的数据库。

## 从命令提示窗口中选择MySQL数据库

在 mysql> 提示窗口中可以很简单的选择特定的数据库。你可以使用SQL命令来选择指定的数据库。

### 实例

以下实例选取了数据库 RUNOOB:

[root@host]# mysql -u root -p

Enter password:\*\*\*\*\*\*

mysql> use RUNOOB;

Database changed

mysql>

执行以上命令后，你就已经成功选择了 RUNOOB 数据库，在后续的操作中都会在 RUNOOB 数据库中执行。

**注意:**所有的数据库名，表名，表字段都是区分大小写的。所以你在使用SQL命令时需要输入正确的名称。

# MySQL 数据类型

MySQL中定义数据字段的类型对你数据库的优化是非常重要的。

MySQL支持多种类型，大致可以分为三类：数值、日期/时间和字符串(字符)类型。

## 数值类型

MySQL支持所有标准SQL数值数据类型。

这些类型包括严格数值数据类型(INTEGER、SMALLINT、DECIMAL和NUMERIC)，以及近似数值数据类型(FLOAT、REAL和DOUBLE PRECISION)。

关键字INT是INTEGER的同义词，关键字DEC是DECIMAL的同义词。

BIT数据类型保存位字段值，并且支持MyISAM、MEMORY、InnoDB和BDB表。

作为SQL标准的扩展，MySQL也支持整数类型TINYINT、MEDIUMINT和BIGINT。下面的表显示了需要的每个整数类型的存储和范围。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类型** | **大小** | **范围（有符号）** | **范围（无符号）** | **用途** |
| TINYINT | 1 字节 | (-128，127) | (0，255) | 小整数值 |
| SMALLINT | 2 字节 | (-32 768，32 767) | (0，65 535) | 大整数值 |
| MEDIUMINT | 3 字节 | (-8 388 608，8 388 607) | (0，16 777 215) | 大整数值 |
| INT或INTEGER | 4 字节 | (-2 147 483 648，2 147 483 647) | (0，4 294 967 295) | 大整数值 |
| BIGINT | 8 字节 | (-9 233 372 036 854 775 808，9 223 372 036 854 775 807) | (0，18 446 744 073 709 551 615) | 极大整数值 |
| FLOAT | 4 字节 | (-3.402 823 466 E+38，1.175 494 351 E-38)，0，(1.175 494 351 E-38，3.402 823 466 351 E+38) | 0，(1.175 494 351 E-38，3.402 823 466 E+38) | 单精度 浮点数值 |
| DOUBLE | 8 字节 | (1.797 693 134 862 315 7 E+308，2.225 073 858 507 201 4 E-308)，0，(2.225 073 858 507 201 4 E-308，1.797 693 134 862 315 7 E+308) | 0，(2.225 073 858 507 201 4 E-308，1.797 693 134 862 315 7 E+308) | 双精度 浮点数值 |
| DECIMAL | 对DECIMAL(M,D) ，如果M>D，为M+2否则为D+2 | 依赖于M和D的值 | 依赖于M和D的值 | 小数值 |

## 日期和时间类型

表示时间值的日期和时间类型为DATETIME、DATE、TIMESTAMP、TIME和YEAR。

每个时间类型有一个有效值范围和一个"零"值，当指定不合法的MySQL不能表示的值时使用"零"值。

TIMESTAMP类型有专有的自动更新特性，将在后面描述。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类型** | **大小 (字节)** | **范围** | **格式** | **用途** |
| DATE | 3 | 1000-01-01/9999-12-31 | YYYY-MM-DD | 日期值 |
| TIME | 3 | '-838:59:59'/'838:59:59' | HH:MM:SS | 时间值或持续时间 |
| YEAR | 1 | 1901/2155 | YYYY | 年份值 |
| DATETIME | 8 | 1000-01-01 00:00:00/9999-12-31 23:59:59 | YYYY-MM-DD HH:MM:SS | 混合日期和时间值 |
| TIMESTAMP | 8 | 1970-01-01 00:00:00/2037 年某时 | YYYYMMDD HHMMSS | 混合日期和时间值，时间戳 |

## 字符串类型

字符串类型指CHAR、VARCHAR、BINARY、VARBINARY、BLOB、TEXT、ENUM和SET。该节描述了这些类型如何工作以及如何在查询中使用这些类型。

|  |  |  |
| --- | --- | --- |
| **类型** | **大小** | **用途** |
| CHAR | 0-255字节 | 定长字符串 |
| VARCHAR | 0-65535 字节 | 变长字符串 |
| TINYBLOB | 0-255字节 | 不超过 255 个字符的二进制字符串 |
| TINYTEXT | 0-255字节 | 短文本字符串 |
| BLOB | 0-65 535字节 | 二进制形式的长文本数据 |
| TEXT | 0-65 535字节 | 长文本数据 |
| MEDIUMBLOB | 0-16 777 215字节 | 二进制形式的中等长度文本数据 |
| MEDIUMTEXT | 0-16 777 215字节 | 中等长度文本数据 |
| LONGBLOB | 0-4 294 967 295字节 | 二进制形式的极大文本数据 |
| LONGTEXT | 0-4 294 967 295字节 | 极大文本数据 |

CHAR和VARCHAR类型类似，但它们保存和检索的方式不同。它们的最大长度和是否尾部空格被保留等方面也不同。在存储或检索过程中不进行大小写转换。

BINARY和VARBINARY类类似于CHAR和VARCHAR，不同的是它们包含二进制字符串而不要非二进制字符串。也就是说，它们包含字节字符串而不是字符字符串。这说明它们没有字符集，并且排序和比较基于列值字节的数值值。

BLOB是一个二进制大对象，可以容纳可变数量的数据。有4种BLOB类型：TINYBLOB、BLOB、MEDIUMBLOB和LONGBLOB。它们只是可容纳值的最大长度不同。

有4种TEXT类型：TINYTEXT、TEXT、MEDIUMTEXT和LONGTEXT。这些对应4种BLOB类型，有相同的最大长度和存储需求。

# MySQL 创建数据表

创建MySQL数据表需要以下信息：

* 表名
* 表字段名
* 定义每个表字段

### 语法

以下为创建MySQL数据表的SQL通用语法：

CREATE TABLE table\_name (column\_name column\_type);

以下例子中我们将在 RUNOOB 数据库中创建数据表runoob\_tbl：

runoob\_tbl(

runoob\_id INT NOT NULL AUTO\_INCREMENT,

runoob\_title VARCHAR(100) NOT NULL,

runoob\_author VARCHAR(40) NOT NULL,

submission\_date DATE,

PRIMARY KEY ( runoob\_id )

);

实例解析：

* 如果你不想字段为 **NULL** 可以设置字段的属性为 **NOT NULL**， 在操作数据库时如果输入该字段的数据为**NULL** ，就会报错。
* AUTO\_INCREMENT定义列为自增的属性，一般用于主键，数值会自动加1。
* PRIMARY KEY关键字用于定义列为主键。 您可以使用多列来定义主键，列间以逗号分隔。

## 通过命令提示符创建表

通过 mysql> 命令窗口可以很简单的创建MySQL数据表。你可以使用 SQL 语句 **CREATE TABLE** 来创建数据表。

### 实例

以下为创建数据表 runoob\_tbl 实例:

root@host# mysql -u root -p

Enter password:\*\*\*\*\*\*\*

mysql> use RUNOOB;

Database changed

mysql> CREATE TABLE runoob\_tbl(

-> runoob\_id INT NOT NULL AUTO\_INCREMENT,

-> runoob\_title VARCHAR(100) NOT NULL,

-> runoob\_author VARCHAR(40) NOT NULL,

-> submission\_date DATE,

-> PRIMARY KEY ( runoob\_id )

-> );

Query OK, 0 rows affected (0.16 sec)

mysql>

**注意：**MySQL命令终止符为分号 (;) 。