21分钟MySQL入门教

极客学院出版

前言

MySQL（发音为"my ess cue el"，不是"my sequel"）是一种开放源代码的关系型数据库管理系统（RDBMS），MySQL 数据库系统使用最常用的数据库管理语言--结构化查询语言（SQL）进行数据库管理。本教程为 MySQL 入门教程，通过讲解对表的增删改查的操作，简明扼要的讲解了 MySQL 关系数据库的基本知识。

适应人群

本教程为初级教程，旨在帮助初学者快速掌握关系型数据库的基本操作，也可作为数据库日常知识查询手册。

学习前提

学习本教程前，你需要有电脑安装有 MySQL 数据库，动手操作是学习最有效的途径。

鸣谢：博客地址 (http://www.cnblogs.com/mr-wid/archive/2013/05/09/3068229.html#c1)

目录

前言 1

第 1 章　MySQL 的相关概念介绍 3

第 2 章　Windows 下 MySQL 的配置 5

第 3 章　MySQL 脚本的基本组成 7

第 4 章　MySQL 中的数据类型 9

第 5 章　使用 MySQL 数据库 11

– "primary key" 表示该列是表的主键, 本列的值必须唯一, MySQL将自动索引该列。下面的 char(8) 表示存储的字符长度为8, tinyint的取值范围为 -127到128, default 属性指定当该列值为空时的默认值。. 15

第 6 章　操作 MySQL 数据库 16

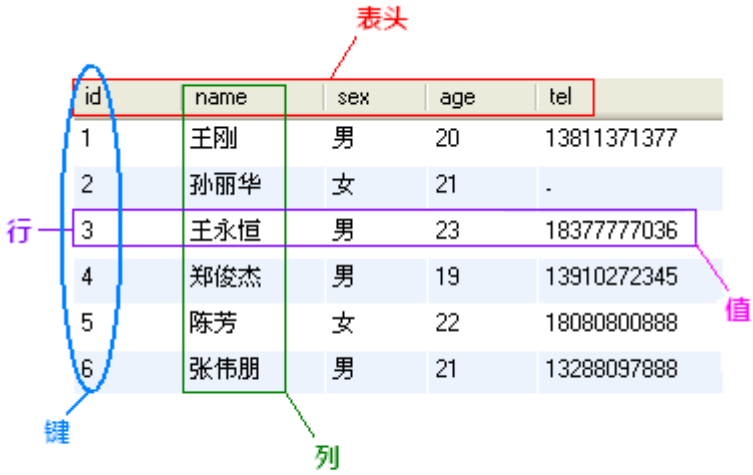
第 7 章　创建后表的修改 20

第 8 章　附录 23

1

MySQL 的相关概念介绍

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, 如图所示的一个表格:



某班级学生信息

- 表头(header): 每一列的名称;
- 列(row): 具有相同数据类型的数据的集合;
- 行(col): 每一行用来描述某个人/物的具体信息;
- 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): 表中用来识别某个特定的人\物的方法, 键的值在当前列中具有唯一性。

2

Windows 下 MySQL 的配置

以 MySQL 5.1 免安装版为例, 下载 mysql-noinstall-5.1.69-win32.zip (http://dev.mysql.com/downloads/mysql/5.1.html#downloads)

配置步骤:

1. 将下载的 mysql-noinstall-5.1.69-win32.zip 解压至需要安装的位置, 如: C:\Program Files;
2. 在安装文件夹下找到 my-small.ini 配置文件, 将其重命名为 my.ini , 打开进行编辑, 在 [client] 与 [mysqld] 下均添加一行: default-character-set = gbk
3. 打开 Windows 环境变量设置, 新建变量名 MYSQL_HOME , 变量值为 MySQL 安装目录路径, 这里为 C:\Program Files\mysql-5.1.69-win32
4. 在 环境变量 的 Path 变量中添加 ;%MYSQL_HOME%\bin;
5. 安装 MySQL 服务, 打开 Windows 命令提示符, 执行命令: mysqld --install MySQL --defaults-file="my.ini" 提示"Service successfully installed."表示成功;

MySQL 服务的启动、停止与卸载

在 Windows 命令提示符下运行:

启动: net start MySQL

停止: net stop MySQL

卸载: sc delete MySQL

3

MySQL 脚本的基本组成

与常规的脚本语言类似, MySQL 也具有一套对字符、单词以及特殊符号的使用规定, MySQL 通过执行 SQL 脚本来完成对数据库的操作, 该脚本由一条或多条 MySQL语句(SQL语句 + 扩展语句)组成, 保存时脚本文件后缀名一般为 `.sql` 。在控制台下, MySQL 客户端也可以对语句进行单句的执行而不用保存为 `.sql` 文件。

标识符

标识符用来命名一些对象, 如数据库、表、列、变量等, 以便在脚本中的其他地方引用。MySQL标识符命名规则稍微有点繁琐, 这里我们使用万能命名规则: 标识符由字母、数字或下划线(_)组成, 且第一个字符必须是字母或下划线。

对于标识符是否区分大小写取决于当前的操作系统, Windows下是不敏感的, 但对于大多数 linux\unix 系统来说, 这些标识符大小写是敏感的。

关键字: MySQL 的关键字众多, 这里不一一列出, 在学习中学习。 这些关键字有自己特定的含义, 尽量避免作为标识符。

语句: MySQL 语句是组成MySQL脚本的基本单位, 每条语句能完成特定的操作, 他是由 SQL 标准语句 + MySQL 扩展语句组成。

函数: MySQL 函数用来实现数据库操作的一些高级功能, 这些函数大致分为以下几类: 字符串函数、数学函数、日期时间函数、搜索函数、加密函数、信息函数。

4

MySQL 中的数据类型

MySQL 有三大类数据类型, 分别为 `数字` 、 `日期\时间` 、 `字符串` , 这三大类中又更细致的划分了许多子类型:

数字类型

- 整数: tinyint、smallint、mediumint、int、bigint
- 浮点数: float、double、real、decimal

日期和时间: date、time、datetime、timestamp、year

字符串类型

- 字符串: char、varchar
- 文本: tinytext、text、mediumtext、longtext
- 二进制(可用来存储图片、音乐等): tinyblob、blob、mediumblob、longblob

这里不能详细对这些类型进行介绍了, 篇幅可能会很长, 详细介绍参见:《 MySQL 数据类型 》: http://www.cnblogs.com/zbseoag/archive/2013/03/19/2970004.html

5

使用 MySQL 数据库

登录到 MySQL

当 MySQL 服务已经运行时, 我们可以通过MySQL自带的客户端工具登录到MySQL数据库中, 首先打开命令提示符, 输入以下格式的命名:

mysql -h 主机名 -u 用户名 -p

- h : 该命令用于指定客户端所要登录的MySQL主机名, 登录当前机器该参数可以省略;
- u : 所要登录的用户名;
- p : 告诉服务器将会使用一个密码来登录, 如果所要登录的用户名密码为空, 可以忽略此选项。 以登录刚刚安装在本机的MySQL数据库为例, 在命令行下输入 mysql -u root -p 按回车确认, 如果安装正确且MySQL正在运行, 会得到以下响应:

Enter password:

若密码存在, 输入密码登录, 不存在则直接按回车登录, 按照本文中的安装方法, 默认 root 账号是无密码的。登录成功后你将会看到 Welecome to the MySQL monitor... 的提示语。

然后命令提示符会一直以 mysql> 加一个闪烁的光标等待命令的输入, 输入 exit 或 quit 退出登录。

创建一个数据库

使用 create database 语句可完成对数据库的创建, 创建命令的格式如下:

```
create database 数据库名 [其他选项];
```

例如我们需要创建一个名为 samp_db 的数据库, 在命令行下执行以下命令:

```
create database samp_db character set gbk;
```

为了便于在命令提示符下显示中文, 在创建时通过 character set gbk 将数据库字符编码指定为 gbk。创建成功时会得到 Query OK, 1 row affected(0.02 sec) 的响应。

> 注意: MySQL语句以分号(;)作为语句的结束, 若在语句结尾不添加分号时, 命令提示符会以 -> 提示你继续输入(有个别特例, 但加分号是一定不会错的);

提示: 可以使用 show databases; 命令查看已经创建了哪些数据库。

选择所要操作的数据库

要对一个数据库进行操作, 必须先选择该数据库, 否则会提示错误:

ERROR 1046(3D000): No database selected

两种方式对数据库进行使用的选择:

一: 在登录数据库时指定, 命令: `mysql -D 所选择的数据库名 -h 主机名 -u 用户名 -p`

例如登录时选择刚刚创建的数据库: mysql -D samp_db -u root -p

二: 在登录后使用 use 语句指定, 命令: `use 数据库名` ;

use 语句可以不加分号, 执行 use samp_db 来选择刚刚创建的数据库, 选择成功后会提示: Database changed

创建数据库表

使用 create table 语句可完成对表的创建, create table 的常见形式:

create table 表名称(列声明);

以创建 students 表为例, 表中将存放 学号(id)、姓名(name)、性别(sex)、年龄(age)、联系电话(tel) 这些内容:

```
create table students
 (
   id int unsigned not null auto_increment primary key,
   name char(8) not null,
   sex char(4) not null,
   age tinyint unsigned not null,
   tel char(13) null default "-"
);
```

对于一些较长的语句在命令提示符下可能容易输错, 因此我们可以通过任何文本编辑器将语句输入好后保存为 createtable.sql 的文件中, 通过命令提示符下的文件重定向执行执行该脚本。

打开命令提示符, 输入: `mysql -D samp_db -u root -p < createtable.sql`

(提示: 1.如果连接远程主机请加上 -h 指令; 2. createtable.sql 文件若不在当前工作目录下需指定文件的完整路径。)

语句解说:

create table tablename(columns) 为创建数据库表的命令, 列的名称以及该列的数据类型将在括号内完成;

括号内声明了5列内容, id、name、sex、age、tel为每列的名称, 后面跟的是数据类型描述, 列与列的描述之间用逗号(,)隔开;

以 "id int unsigned not null auto_increment primary key" 行进行介绍:

- "id" 为列的名称;
- "int" 指定该列的类型为 int(取值范围为 -8388608到8388607), 在后面我们又用 "unsigned" 加以修饰, 表示该类型为无符号型, 此时该列的取值范围为 0到16777215;
- "not null" 说明该列的值不能为空, 必须要填, 如果不指定该属性, 默认可为空;
- "auto_increment" 需在整数列中使用, 其作用是在插入数据时若该列为 NULL, MySQL将自动产生一个比现存值更大的唯一标识符值。在每张表中仅能有一个这样的值且所在列必须为索引列。

– "primary key" 表示该列是表的主键, 本列的值必须唯一, MySQL将自动索引该列。下面的 char(8) 表示存储的字符长度为8, tinyint的取值范围为 -127到128, default 属性指定当该列值为空时的默认值。

更多的数据类型请参阅《 MySQL数据类型 》: http://www.cnblogs.com/zbseoag/archive/2013/03/19/2970004.html

提示: 1. 使用 show tables; 命令可查看已创建了表的名称; 2. 使用 describe 表名; 命令可查看已创建的表的详细信息。

6

操作 MySQL 数据库

向表中插入数据

insert 语句可以用来将一行或多行数据插到数据库表中, 使用的一般形式如下:

```
insert [into] 表名 [(列名1, 列名2, 列名3, ...)] values (值1, 值2, 值3, ...);
```

其中 [] 内的内容是可选的, 例如, 要给 samp_db 数据库中的 students 表插入一条记录, 执行语句:

insert into students values(NULL, "王刚", "男", 20, "13811371377");

按回车键确认后若提示 Query Ok, 1 row affected (0.05 sec) 表示数据插入成功。 若插入失败请检查是否已选择需要操作的数据库。

有时我们只需要插入部分数据, 或者不按照列的顺序进行插入, 可以使用这样的形式进行插入:

```
insert into students (name, sex, age) values("孙丽华", "女", 21);
```

查询表中的数据

select 语句常用来根据一定的查询规则到数据库中获取数据, 其基本的用法为:

```
select 列名称 from 表名称 [查询条件];
```

例如要查询 students 表中所有学生的名字和年龄, 输入语句 select name, age from students; 执行结果如下:

```
mysql> select name, age from students;
+--------+-----+
| name   | age |
+--------+-----+
| 王刚   |  20 |
| 孙丽华 |  21 |
| 王永恒 |  23 |
| 郑俊杰 |  19 |
| 陈芳   |  22 |
| 张伟朋 |  21 |
+--------+-----+
6 rows in set (0.00 sec)

mysql>
```

也可以使用通配符 * 查询表中所有的内容, 语句: select * from students;

按特定条件查询

where 关键词用于指定查询条件, 用法形式为: `select 列名称 from 表名称 where 条件` ;

以查询所有性别为女的信息为例, 输入查询语句: `select * from students where sex="女"` ;

where 子句不仅仅支持 "where 列名 = 值" 这种名等于值的查询形式, 对一般的比较运算的运算符都是支持的, 例如 =、>、<、>=、<、!= 以及一些扩展运算符 is [not] null、in、like 等等。 还可以对查询条件使用 or 和 and 进行组合查询, 以后还会学到更加高级的条件查询方式, 这里不再多做介绍。

示例:

查询年龄在 21 岁以上的所有人信息: `select * from students where age > 21` ;

查询名字中带有 "王" 字的所有人信息: `select * from students where name like "%王%"` ;

查询 id 小于 5 且年龄大于 20 的所有人信息: `select * from students where id<5 and age>20` ;

更新表中的数据

update 语句可用来修改表中的数据, 基本的使用形式为:

update 表名称 set 列名称=新值 where 更新条件;

使用示例:

将 id 为 5 的手机号改为默认的"-": `update students set tel=default where id=5` ;

将所有人的年龄增加 1: update students set age=age+1;

将手机号为 13288097888 的姓名改为 "张伟鹏", 年龄改为 19: `update students set name="张伟鹏", age=19 where tel="13288097888"` ;

删除表中的数据

delete 语句用于删除表中的数据, 基本用法为:

delete from 表名称 where 删除条件;

使用示例:

删除 id 为 2 的行: `delete from students where id=2` ;

删除所有年龄小于 21 岁的数据: `delete from students where age<20` ;

删除表中的所有数据: `delete from students` ;

7

创建后表的修改

alter table 语句用于创建后对表的修改, 基础用法如下:

添加列

基本形式: `alter table 表名 add 列名 列数据类型 [after 插入位置]` ;

示例:

在表的最后追加列 `address: alter table students add address char(60)` ;

在名为 age 的列后插入列 `birthday: alter table students add birthday date after age` ;

修改列

基本形式: `alter table 表名 change 列名称 列新名称 新数据类型` ;

示例:

将表 tel 列改名为 `telphone: alter table students change tel telphone char(13) default "-"` ;

将 name 列的数据类型改为 char(16): `alter table students change name name char(16) not null` ;

删除列

基本形式: `alter table 表名 drop 列名称` ;

示例:

删除 birthday 列: `alter table students drop birthday` ;

重命名表

基本形式: `alter table 表名 rename 新表名` ;

示例:

重命名 students 表为 workmates: `alter table students rename workmates` ;

删除整张表

基本形式: `drop table 表名;`

示例: 删除 workmates 表: `drop table workmates;`

删除整个数据库

基本形式: `drop database 数据库名;`

示例: 删除 samp_db 数据库: `drop database samp_db;`

8 附录

修改 root 用户密码

按照本文的安装方式, root 用户默认是没有密码的, 重设 root 密码的方式也较多, 这里仅介绍一种较常用的方式。

使用 mysqladmin 方式:

打开命令提示符界面, 执行命令: `mysqladmin -u root -p password 新密码`

执行后提示输入旧密码完成密码修改, 当旧密码为空时直接按回车键确认即可。

可视化管理工具 MySQL Workbench

尽管我们可以在命令提示符下通过一行行的输入或者通过重定向文件来执行 mysql 语句, 但该方式效率较低, 由于没有执行前的语法自动检查, 输入失误造成的一些错误的可能性会大大增加, 这时不妨试试一些可视化的 MySQL 数据库管理工具, MySQL Workbench 就是 MySQL 官方 为 MySQL 提供的一款可视化管理工具, 你可以在里面通过可视化的方式直接管理数据库中的内容, 并且 MySQL Workbench 的 SQL 脚本编辑器支持语法高亮以及输入时的语法检查, 当然, 它的功能强大, 绝不仅限于这两点。

MySQL Workbench官方介绍: http://www.mysql.com/products/workbench/

MySQL Workbench 下载页: http://dev.mysql.com/downloads/tools/workbench/

极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台

更多信息请访问

http://wiki.jikexueyuan.com/project/mysql-21-minutes/