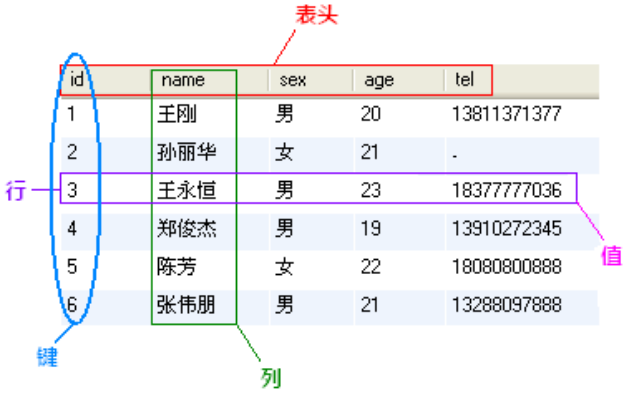
Chapter 1. MySQL 的相关概念介绍

1、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

Chapter 2. Windows 下 MySQL 的配置

Chapter 3. MySQL 脚本的基本组成

1、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

2、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

3、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

4、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

5、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

Chapter 4. MySQL 中的数据类型

1、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

2、数字类型

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

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

4、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

Chapter 5. 使用 MySQL 数据库

5.1. 登陆到 MySQL

1、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，输入以下格式的命令

- mysql [-h 主机名] [-u 用户名] -p[密码]
- -h：指定客户端所要登陆的 MySQL 主机名，登陆当前机器该参数可以省略
- -u：所要登陆的用户名
- -p：告诉服务器将会使用一个密码来登陆，如果所要登陆的用户名密码为空，可以忽略此选项
  - 可以直接在 p 后面接密码，注意，密码与 p 之间没有任何字符
  - 在 p 后不接密码，按回车后再输入密码

5.2. 创建一个数据库

1、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]

2、例子

- create database samp_db character set gbk;
- character set gbk 将数据库字符编码指定为 gbk
- MySQL 语句以分号";"结尾

3、<show>

- show databases; <==查看已经创建了哪些数据库
- show tables; <==查看已经创建了哪些表

5. 3. 选择所要操作的数据库

1、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

2、有以下两种方式对数据库进行使用的选择

- 在登陆数据库时指定：
  - mysql [-D 数据库名] [-h 主机] [-u 用户名] -p
  - 例如：mysql -D samp_db -u root -p
- 在登陆后使用 use 语句指定：
  - use [数据库名]
  - use 语句可以不加分号

5. 4. 创建数据库表

1、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(列名称);

2、例子

```
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 "-"
);
```

- 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。

3、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.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

- mysql -D samp_db -u root -p < [.sql 文件路径]

5.5. 使用脚本来执行 sql 语句

1、方法 1：在 bash 命令行下(未连接数据库)，输入：

mysql -h localhost -u root -p123456 < [.sql 文件路径]

- 注意，p 与密码之间不要有任何间隔

2、方法 2：在 mysql 命令行下(已连接数据库)，输入：

source [.sql 文件路径]

\. [.sql 文件路径] <==注意，这里与 bash 不同，需要加反斜杠

Chapter 6. 操作 MySQL 数据库

6.1. 向表中插入数据

1、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：insert into students values(NULL,"王刚","男",20,"13811371377")
- 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);

6.2. 查询表中的数据

1、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];

- 例如查找表中所有学生的名字和年龄

select name, age from students;

- 也可以使用通配符"*"查询表中所有的内容

select * from students;

2、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

    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

  - 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;

6.3. 更新表中的数据

1、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];

- 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";

6.4. 删除表中的数据

1、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];

- 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;

Chapter 7. 创建后的修改

1、alter table 语句用于创建后对表的修改

7.1. 添加列

1、基本形式

alter table [表名] add [列名] [列数据类型] [after 插入位置];

- after 可以省略：省略表示在末尾追加
- 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;

7.2. 修改列

1、基本形式

alter table [表名] change [列名称] [列新名称] [新数据类型];

- 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;

7.3. 删除列

1、基本形式

alter table [表名] drop [列名称];

- 删除 birthday 列:

alter table students drop birthday;

7.4. 重命名表

1、基本形式

alter table [表名] rename [新表名];

- 重命名 students 表为 workmates:

alter table students rename workmates;

7.5. 删除整张表

1、基本形式

drop table [表名];

- 删除 workmates 表:

drop table workmates;

7.6. 删除整个数据库

1、基本形式

drop database [数据库名];

- 删除 samp_db 数据库:

drop database samp_db;

附录

1、修改 root 用户密码

- mysqladmin -u root -p password [新密码]
  - 会让你输入旧密码