E:\程序人生\Python\video\python\第4章 数据库\第1节 MySQL

简介

数据库：

功能：存储数据

特点：优化了读写，方便存储，快速查找

分布式存储：多地保存，数据更安全

数据库系统解决的问题：持久化存储、优化读写、保证数据的有效性

当前使用的数据库，主要分为两类：

文档型，如sqlite,就是一个文件，通过对文件的复制完成数据库的复制

服务型，如mysql,数据存储在一个物理文件中，但是需要使用终端以tcp/ip协议连接，进行数据库的读写操作

关系型数据库：

Oracle/Mysql/SqlServer使用了相同的设计模型（E-R设计模型），它们的区别是核心代码的实现方式不一样，即在硬盘中对数据的存储方式和读取方式不一样，但是数据结构的设计都是基于E-R结构设计

Mysql数据库是当前应用非常广泛的一款关系型数据库，免费的

官方网站：[www.mysql.com](http://www.mysql.com)

数据库排名网站：<http://db-engines.com/en/ranking>

主要操作：

数据库的操作：创建、删除

表的操作：创建、修改、删除

数据的操作：增加、修改、删除、查询，简称crud(create、read、update、delete)

E-R模型

E表示entry, 实体

R表示relationship, 关系

一个实体转换转换为数据库中的一个表

关系描述两个实体之间的对应规则，包括：

一对一

一对多

多对多

关系转换为数据库表中的一个列，在关系型数据库中一行就是一个对象

三范式

第一范式（1NF）:列不可拆分

第二范式（2NF）:唯一标识

第三范式（3NF）:引用主键

说明：后一个范式，都是在前一个范式的基础上建立的

数据完整性

一个数据库就是一个完整的业务单元，可以包含多张表，数据被存储在表中

在表中为了更加准确的存储数据，保证数据的正确有效，可以在创建表的时候，为表添加一些强制性的验证，包括数据字段的类型、约束

字段类型

在mysql中包含的数据类型很多，常用类型为：

数字：int, decimal

字符串：char, varchar, text

日期：datetime

布尔：bit

约束

主键primary key

非空not null

唯一 unique

默认 default

外键 foreign key

## 图形界面操作

Navicat for MySQL软件简介

Navicat for MySQL 是一款强大的 MySQL 数据库管理和开发工具，它为专业开发者提供了一套强大的足够尖端的工具，但对于新用户仍然易于学习。Navicat For Mysql中文网站：http://www.formysql.com/  
Navicat for MySQL 基于Windows平台，为 MySQL 量身订作，提供类似于 MySQL 的用管理界面工具。此解决方案的出现，将解放 PHP、J2EE 等程序员以及数据库设计者、管理者的大脑，降低开发成本，为用户带来更高的开发效率。  
Navicat for MySQL 使用了极好的图形用户界面（GUI），可以用一种安全和更为容易的方式快速和容易地创建、组织、存取和共享信息。 用户可完全控制 MySQL 数据库和显示不同的管理资料，包括一个多功能的图形化管理用户和访问权限的管理工具，  
方便将数据从一个数据库转移到另一个数据库中（Local to Remote、Remote to Remote、Remote to Local），进行档案备份。  
Navicat for MySQL 支援 Unicode，以及本地或远程 MySQL 服务器多连线，用户可浏览数据库、建立和删除数据库、编辑数据、建立或执行 SQL queries、管理用户权限（安全设定）、将数据库备份/复原、汇入/汇出数据（支援 CSV, TXT, DBF 和 XML 档案种类）等。

## 逻辑删除

对于重要数据，并不希望物理删除，一旦删除，数据无法找回

一般对于重要数据，会设置一个isDelete的列，类型为bit,表示逻辑删除

对于大量增长的非重要数据，可以进行物理删除

数据的重要性，要根据实际开发决定

# E:\程序人生\SQL\video\MySQL

# 数据库基础

1. 什么是数据库？

数据库：database,存储数据的仓库

数据库：高效的存储和处理数据的介质（介质主要是2种：磁盘和内存）

1. 数据库的分类？

数据库基于存储介质的不同，分为2类：关系型数据库(SQL)和非关系型数据库(NoDSQL：not only SQL)

1. 不同的数据库阵营中的产品有哪些？

关系型数据库

大型：Oracle, DB2

中型：SQL Server, mysql等

小型：access等

非关系型数据库：mongodb,redis(同步到磁盘)

1. 两种数据库阵营的区别？

关系型数据库：安全（保存磁盘基本不可能丢失），容易理解，比较浪费空间（二维表）

非关系型数据库：效率高（运行在内存），不安全（断电丢失）

## 关系型数据库

1. 什么是关系型数据库？

关系型数据库：是一种建立在关系模型（数学模型）上的数据库。

## 关键字说明

数据库：database

数据库系统：DBS(Database System),是一种虚拟系统，将多种内容关联起来的称呼

DBS=DBMS + DB

DBMS:Database Management System, 数据库管理系统，专门管理数据库

DBA:Database Administrator,数据库管理员

行/记录：row/record,本质是一个东西：都是指表中的一行（一条记录），行是从结构角度出发，记录是从数据角度出发

列/字段：column/field

## SQL

SQL:Structured Query Language,结构化查询语言（数据以查询为主：99%是在进行查询操作）

SQL分为三个部分

DDL:Data Definition Language,数据定义语言，用来维护存储数据的结构（数据库、表）

代表指令：create,drop,alter

DML：Data manipulation Language，数据操作语言，用来对数据进行操作（数据表中的内容），代表指令：insert,delete,update等：其中DML内部又单独进行了一个分类：DQL(Data Query Language:数据查询语言，如select)

DCL：Data control Language，数据控制语言，主要是负责权限管理（用户）

代表指令：grant，revoke等

SQL是关系型数据库的操作指令，SQL是一种约束，但不强制（类似W3C）:不同的数据库产品（如Oracle,mysql）可能内部会有一些细微的差别

## Mysql数据库

Mysql数据库是一种c/s结构的软件：客户端/服务端，若想访问服务器必须通过客户端（服务器一直运行，客户端在需要使用的时候运行）

交互方式：

1. 客户端连接认证：连接服务器，认证身份：mysql.exe -hpup（-hlocalhost -p3306 -uroot -p）
2. 发送SQL指令
3. 服务器接受SQL指令：处理SQL指令：返回操作结果
4. 客户端接受结果：显示结果
5. 断开连接（释放资源：服务器并发限制）(exit/quit/\q)

## SQL基本操作

基本操作：CRUD

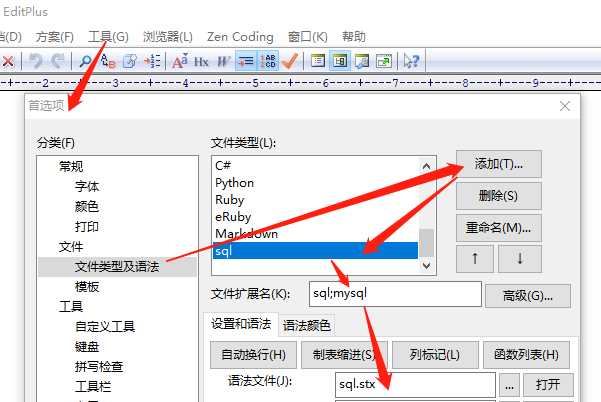
将SQL的基本操作根据操作对象进行分类，分为3类：库操作，表操作（字段），数据操作

### SQL编辑器

可使用EditPlus

配置sql语法模板（有颜色区分，高亮显示等）

1. 下载sql.stx
2. 将sql.stx放到EditPlus.exe所在目录
3. 配置



### 库操作

对数据库的增删改查

新增数据库

语法：create database 数据库名字 [库选项];

库选项：用来约束数据库，分为两个选项

字符集设定：charset/character set 具体字符集（数据存储的编码格式）：常用字符集：GBK和UTF8

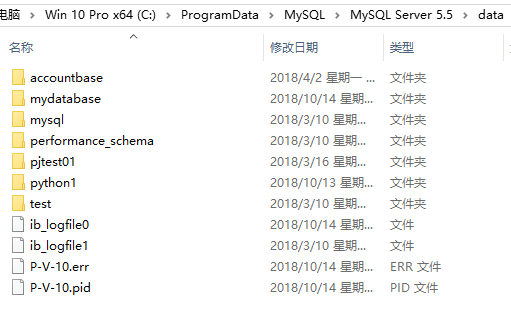
校对集设定：collate 具体校对集（数据比较的规则）

-- 创建数据库

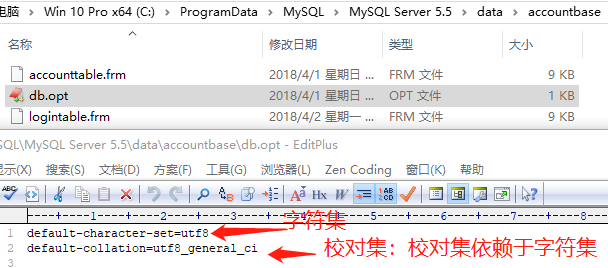
create database mydatabase charset utf8;

当创建数据库的SQL语句执行之后，发生了什么？

1. 在数据库系统中，增加了对应的数据库信息
2. 会在保存数据的文件夹下：Data目录，创建一个对应数据库名字的文件夹



1. 每个数据库下都有一个opt文件：保存了库选项



## 查看数据库

1. 查看所有数据库：show databases;
2. 查看指定部分的数据库：模糊查询

Show databases like ‘pattern’; -- pattern是匹配模式

%：表示匹配多个字符

\_：表示匹配单个字符

-- 查看以abc\_开始的数据库：\_需要被转义

Show databases like ‘information\\_%’;

1. 查看数据库的创建语句：show create database 数据库名字；

查看结果会与创建时的sql语句不太一样，因为数据库在执行SQL语句之前会优化SQL:系统保存的结果是优化的结果

## 更新数据库

数据库名字不可修改。

数据库的修改仅限库选项：字符集和校对集（校对集依赖字符集）

Alter database 数据库名字 [库选项];

Charset/character set [=] 字符集

Collate [=] 校对集

## 删除数据库

Drop database 数据库名字;

注意：数据库删除不可逆，删除要慎重，应该先进行备份后操作。

## 表操作

表与字段是密不可分的。

### 新增数据表

Create table [if not exists] 表名(

字段名字 数据类型，

字段名字 数据类型 --最后一行不需要逗号

)[表选项];

表选项:控制表的表现

字符集:charset/character set 具体字符集;

校对集:collate 具体校对集;

存储引擎:engine 具体的存储引擎(innodb和myisam)

方案1：显式的指定表所属的数据库

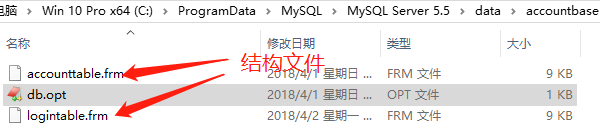
Create table 数据库名.表名();

方案2：隐式的指定表所属数据库：先进入到某个数据库环境，然后这样创建的表自动归属到某个指定的数据库.

进入数据库环境：use 数据库名字;

当创建数据表的SQL指令执行之后，到底发生了什么？

1. 指定数据库下已经存在对应的表
2. 在数据库对应的文件夹下，会产生对应表的结构文件（跟存储引擎有关系）



## 查看数据表

数据库可以查看的方式，表都可以查看.

1. 查看所有表：show tables;
2. 查看部分表：show tables like ‘pattern’;
3. 查看表创建语句

Show create table 表名\g --\g等价于;

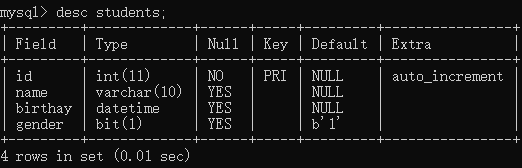
Show create table 表名\G --\G将查到的结构旋转90度变为纵向

1. 查看表结构：查看表中的字段信息

Desc 表名;

Describe 表名;

Show columns from 表名;



## 修改数据表

表的修改分为两个部分：修改表本身和修改字段

### 修改表本身

表本身可以修改：表名和表选项

修改表名:

rename table 老表名 to 新表名;

修改表选项：字符集，校对集 和 存储引擎

Alter table 表名 表选项 [=] 值;

### 修改字段

字段操作很多：新增，修改，重命名，删除

新增字段

alter table 表名 add [column] 字段名 字段类型 [列属性][位置];

位置：字段名可以插入到表中的任何位置

First:第一个位置

After:在哪个字段之后：after 字段名；默认的是在最后一个字段之后

修改字段：通常是修改属性或者数据类型

Alter table 表名 modify 字段名 数据类型 [属性][位置];

重命名字段

Alter table 表名 change 旧字段名 新字段名 数据类型 [属性][位置];

删除字段

Alter table 表名 drop 字段名;

小心：如果表中已经存在数据，那么删除字段会清空该字段的所有数据(不可逆)

### 删除数据表

Drop table 表名1,表名2.....; -- 可以一次性删除多张表

注意：删除有危险，操作需谨慎(不可逆)

## 数据操作

### 新增数据

有两种方案

方案1、给全表字段插入数据，不需要指定字段列表：要求数据的值出现的顺序必须与表中设计的字段出现的顺序一致：凡是非数值数据，都需要使用引号（建议是单引号）包裹

Insert into 表名 values(值列表)[,(值列表)]; -- 可以一次性插入多条数据

方案2、给部分字段插入数据，需要选定字段列表：字段列表出现的顺序与字段的顺序无关；但是值列表的顺序必须与选定的字段的顺序一致

Insert into 表名(字段列表) values(值列表)[,(值列表)];

### 查看数据

Select \* from 表名 [where条件];

Select 字段列表 from 表名 [where条件];

### 更新数据

Update 表名 set 字段 = 值 [where条件]; -- 建议都有where,否则会更新全部

### 删除数据

Delete from 表名 [where条件];

## 中文数据问题

中文数据问题本质是字符集问题。

计算机只识别二进制，人类更多的是识别符号：需要有个二进制与字符的对应关系（字符集）

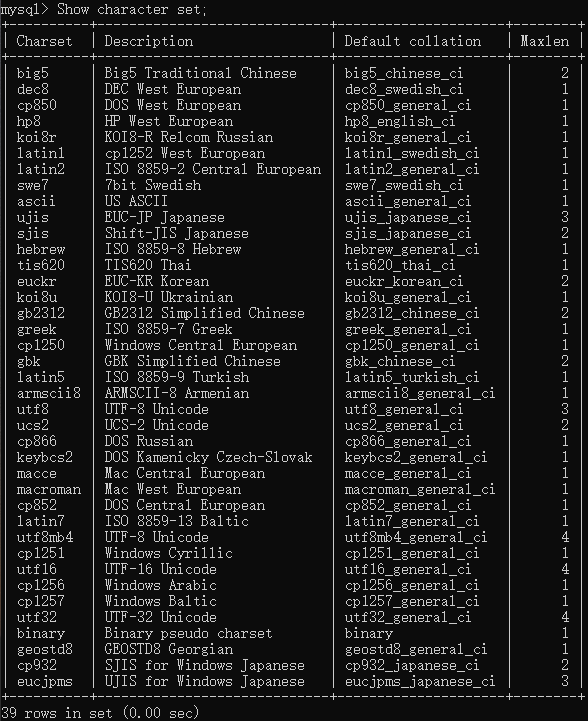


报错：服务器没有识别对应的四个字节：服务器认为数据是UTF8,一个汉字有三个字节：读取三个字节转换成汉子（失败），剩余的再读三个字节（不够）：最终失败。

所有的数据库服务器认为（表现）的一些特性都是通过服务器端的变量来保存：系统先读取自己的变量，看看应该怎样表现。

-- 查看服务器识别哪些字符集

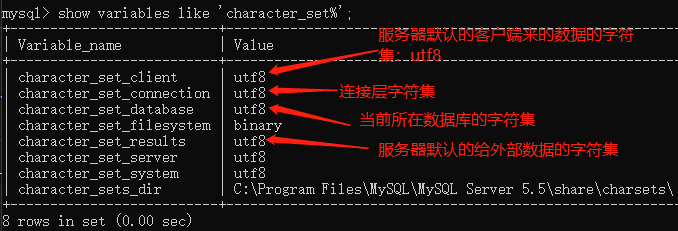
Show character set;



基本上：服务器是万能，什么字符集都支持

-- 既然服务器识别这么多：总有一种是服务器默认的跟客户端打交道的字符集

Show variables like ‘character\_set%’;



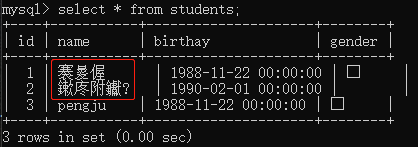
问题根源：客户端（cmd.exe）数据只能是GBK,而服务器认为是UTF8:矛盾产生

解决方案：

改变服务器，默认的接受字符为GBK;

Set character\_set\_client = gbk;

插入中文后查看数据，显示乱码：

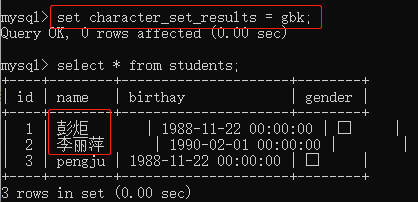


原因：数据来源是服务器，解析数据是客户端（客户端只识别GBK:只会两个字节一个汉字）：但是事实服务器给的数据却是UTF8,三个字节一个汉字：导致乱码

解决方案：

修改服务器给客户端的数据字符集为GBK;

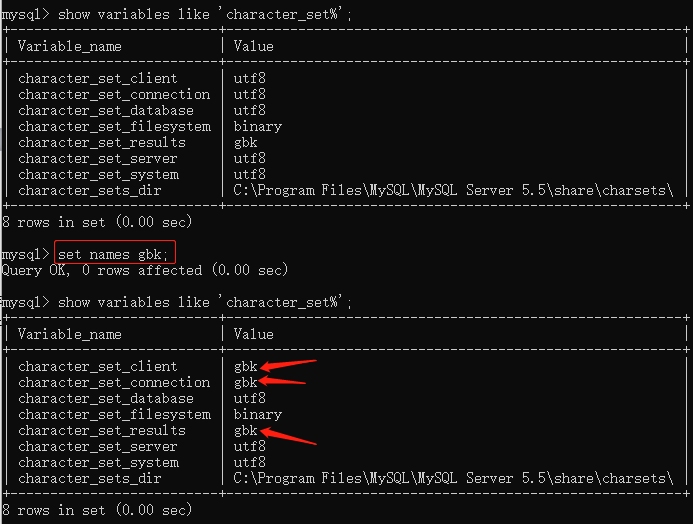
Set character\_set\_results = gbk;



Set 变量 = 值; 修改只是会话级别（当前客户端，当次连接有效：关闭失效）

设置服务器对客户端的字符集的认识：可以使用快捷方式：set names 字符集；

Set names gbk; ==>定价于 character\_set\_client,character\_set\_results,character\_set\_connection



connection连接层：是字符集转变的中间者，如果统一了效率更高，不同意也没问题。

## 校对集问题

校对集：数据比较的方式

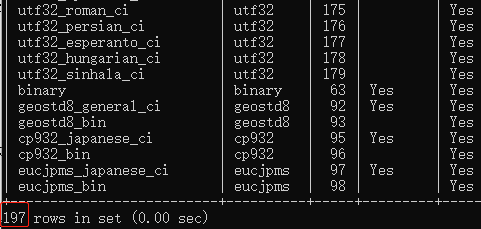
校对集有三种格式：

\_bin:binary,二进制比较，取出二进制位，一位一位的比较，区分大小写

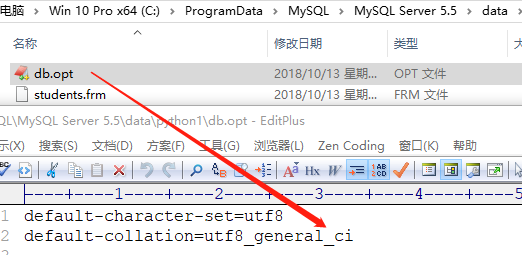
\_cs:case sensitive,大小写敏感，区分大小写

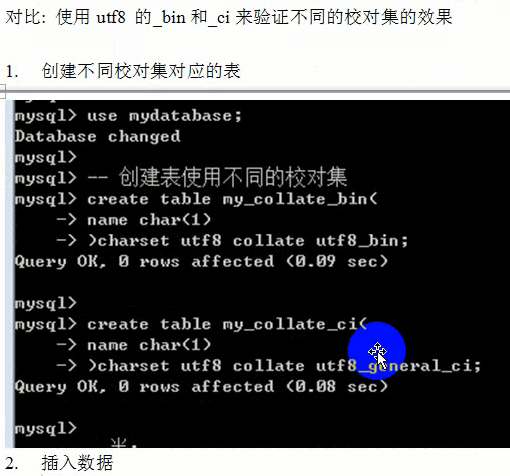
\_ci:case insensitice,大小写不敏感，不区分大小写

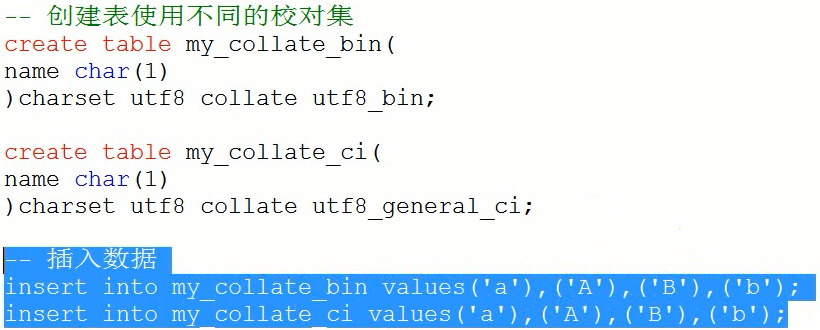
查看数据库所支持的校对集：show collation;

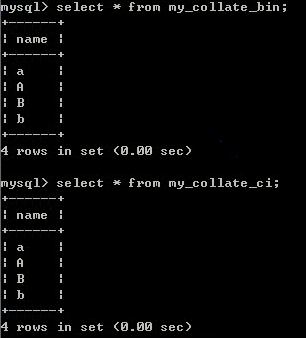


校对集应用：只有当数据产生比较的时候，校对集才会生效。

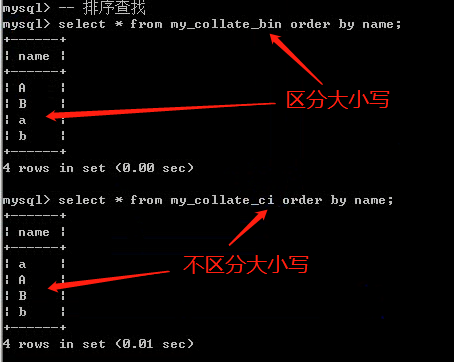








1. 比较：根据某个字段进行排序：order by 字段名 [asc|desc];asc升序，desc降序，默认是升序



校对集：必须在没有数据之前声明好，如果有了数据，再对校对集进行修改：那么修改无效

### Web乱码问题

动态网站由三部分构成：浏览器，web服务器，数据库服务器，三个部分都有自己的字符集（中文），数据需要在三个部分之间来回传递：很容易产生乱码。

如何解决乱码问题：统一编码（三码合一）

但是事实上不可能：浏览器是用户管理（根本不可能控制）

但是必须要解决这些问题：主要靠php来做

## 数据类型（列类型）

SQL中将数据类型分成了三大类：数值类型，字符串类型和时间日期类型

