

基础[1]

- 介绍
- 安装
- 基本使用

目录

基础[1]

一、数据库介绍

- 1、使用数据库
- 2、数据库与管理系统
 - 1、数据库相关概念
 - 2、常见的DBMS
 - 3、数据库与数据库管理系统的关系
- 3、MySQL介绍
- 4、RDBMS与非RDBMS
 - 1、关系型数据库(RDBMS)
 - 1、实质
 - 2、性质
 - 3、优点
 - 2、非关系型数据库(非RDBMS)
 - 1、分类
 - 1、键值型数据库
 - 2、文档型数据库
 - 3、搜索引擎数据库
 - 4、列式数据库
 - 5、图形数据库
- 5、关系型数据库的设计
 - 1、表、记录、字段
 - 2、表的关联关系
 - 1、一对一关联
 - 2、一对多关联
 - 3、多对多关联
 - 4、自我引用

二、MySQL安装

- 1、版本
- 2、下载
- 3、安装
- 4、MySQL登陆
 - 1、启动服务
 - 2、命令行登陆

三、MySQL基本使用

- 1、基本使用
 - 1、登陆
 - 2、查看所有数据库
 - 3、创建数据库
 - 4、使用已有的数据库
 - 5、查看某个数据库的所有表格
 - 6、创建新的表格
 - 7、查看某个表的数据
 - 8、添加一条记录

- 9、查看表的创建信息
 - 10、查看数据库的创建信息
 - 11、删除表格
 - 12、删除数据库
- 2、字符集设置
 - 3、图形化工具
 - 1、Workbench
 - 2、Navicat
 - 3、SQLyog
 - 4、dbeaver

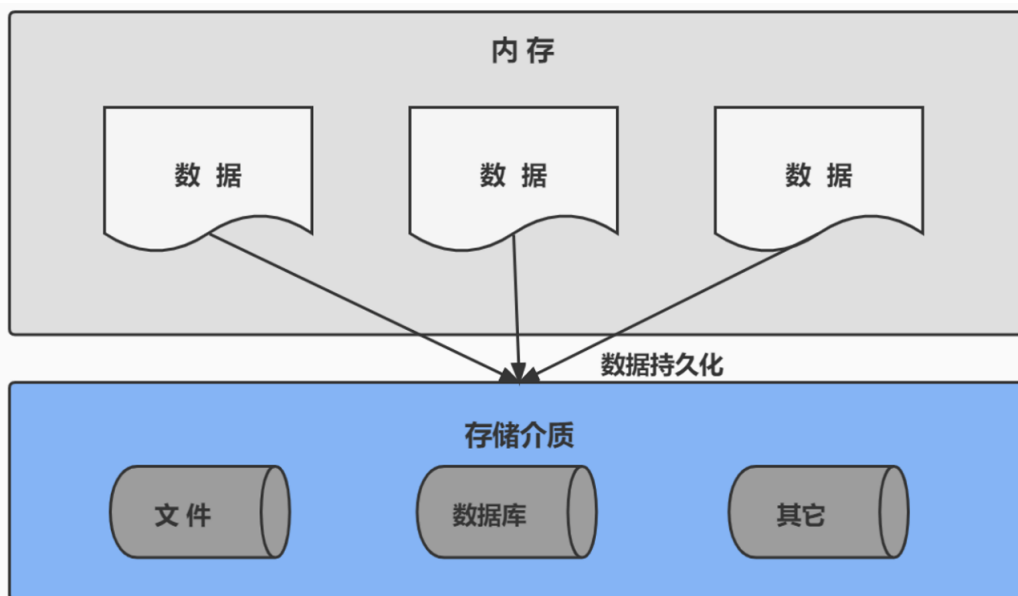
一、数据库介绍

存放数据的一个容器，

是长期储存在计算机内、有组织的、可共享的大量数据的集合。

1、使用数据库

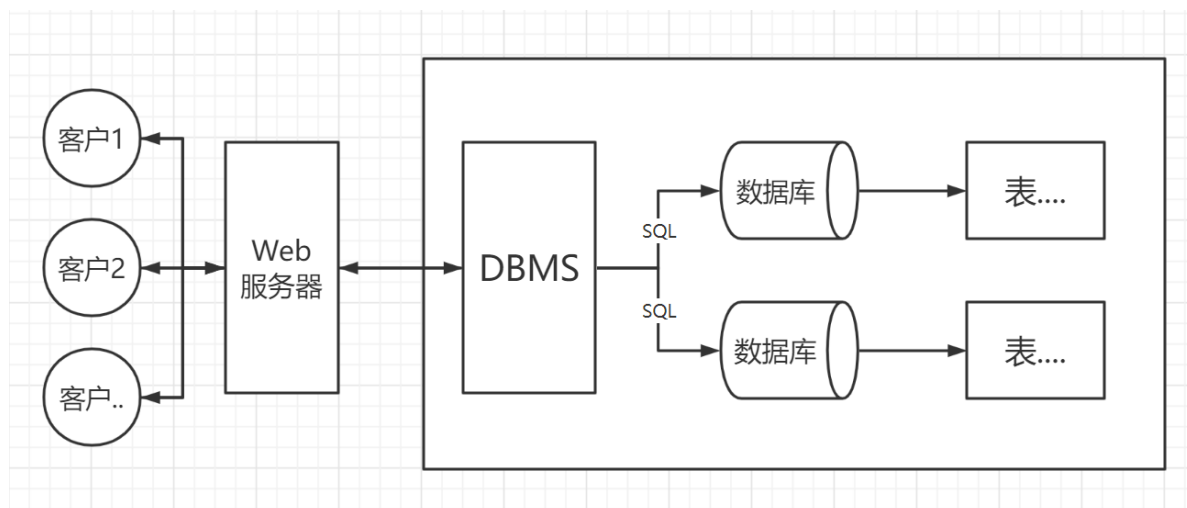
- 持久化(persistence): **把数据保存到可掉电式存储设备中**以供之后使用。大多数情况下，特别是企业级应用，数据持久化意味着将内存中的数据保存到硬盘上加以“固化”，而持久化的实现过程大多通过各种关系数据库来完成。
- 持久化的主要作用是将**内存中的数据存储在关系型数据库**中，当然也可以存储在磁盘文件、XML数据文件中。



2、数据库与管理系统

1、数据库相关概念

名称	简称	作用
数据库	DataBase(DB)	存储数据的仓库，数据是有组织的进行存储
数据库管理系统	DBMS	操纵和管理数据库的大型软件
结构化查询语句	SQL	操作关系型数据库的编程语言，与数据库通信，定义了一套操作关系型数据库统一标准

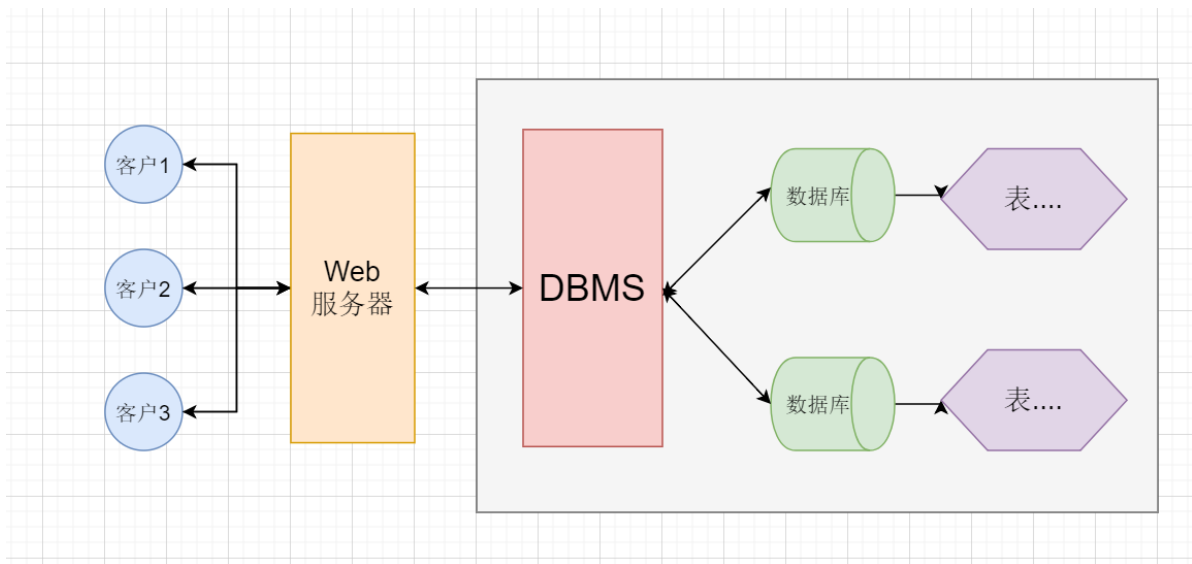


2、常见的DBMS

- Oracle
- **MySQL**
- SQL Server
- DB2
- sqlite
- Access
- **Redis**

3、数据库与数据库管理系统的关系

数据库管理系统(DBMS)可以管理多个数据库，一般开发人员会针对每一个应用创建一个数据库。为保存应用中实体的数据，一般会在数据库创建多个表，以保存程序中实体用户的数据。



3、MySQL介绍

- MySQL是一个 开放源代码的关系型数据库管理系统，由瑞典MySQL AB（创始人Michael Widenius）公司1995年开发，迅速成为开源数据库的 No.1。
- 2008被 Sun 收购（10亿美金），2009年Sun被 Oracle 收购。MariaDB 应运而生。
（MySQL 的创造者担心 MySQL 有闭源的风险，因此创建了MySQL 的分支项目 MariaDB）
- MySQL6.x 版本之后分为 **社区版** 和 **商业版**。MySQL是一种**关联数据库管理系统**，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**，所以你不需支付额外的费用。MySQL是可以定制的，采用了 GPL（GNU General Public License）协议，你可以修改源码来开发自己的MySQL系统。
- **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使用**标准的SQL数据语言形式**。
- **MySQL可以允许运行于多个系统上，并且支持多种语言**。这些编程语言包括C、C++、Python、Java、Perl、PHP和Ruby等。

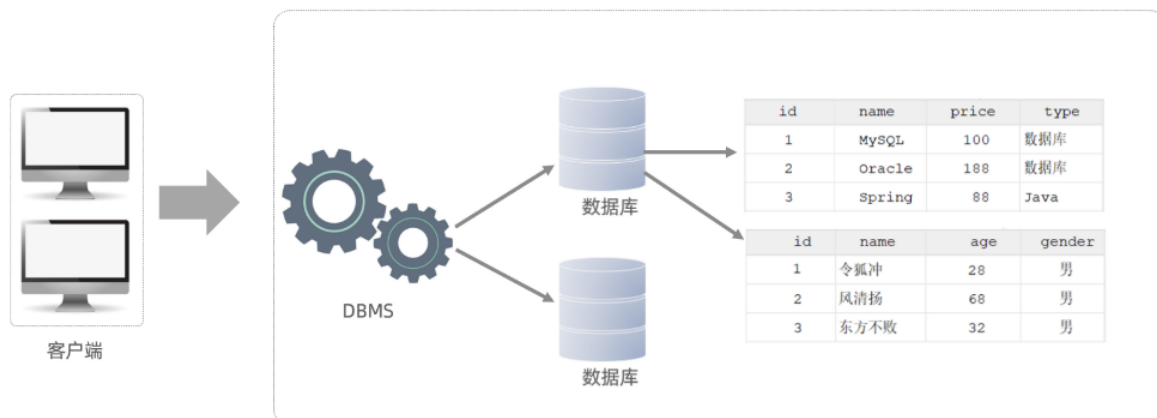
4、RDBMS与非RDBMS

RDBMS（Relational Database Management System），即**关系数据库管理系统**。是指包括相互联系的逻辑组织和存取这些数据的一套程序（数据库管理系统软件）。

关系型数据库绝对是 DBMS 的主流，其中使用最多的 DBMS 分别是 Oracle、MySQL 和 SQL Server。这些都是关系型数据库（RDBMS）。

1、关系型数据库(RDBMS)

关系模型指的就是二维表格模型，而一个关系型数据库就是由二维表及其之间的联系所组成的一个数据组织。



1、实质

- 这种类型的数据库是最古老的数据库类型，关系型数据库模型是把复杂的数据结构归结为简单的二元关系（即二维表格形式）

1	姓名	年龄	地址
2	Java		1 y
3	Python		2 h
4	C++		3 t
5	C		4 f
6	Rust		5 g
7	Go		6 f
8	Ruby		7 d
9	C#		8 a
10	JavaScript		9 u

- 关系型数据库以行(row)和列(column)的形式存储数据，以便于用户理解。这一系列行和列被称为表(table)，一组表组成了一个库。
- 表与表之间数据记录有**关系(relationship)**。关系型数据库就是建立在**关系模型**基础上的数据库。

2、性质

- 原子性**：事务里的所有操作要么全部做完，要么都不做，事务成功的条件是事务里的所有操作都成功，只要有一个操作失败，整个事务就失败，需要回滚。
- 一致性**：数据库要一直处于一致的状态，事务的运行不会改变数据库原本的一致性约束。
- 独立性**：并发的事务之间不会互相影响，如果一个事务要访问的数据正在被另外一个事务修改，只要另外一个事务未提交，它所访问的数据就不受未提交事务的影响。
- 持久性**：一旦事务提交后，它所做的修改将会永久的保存在数据库上，即使出现宕机也不会丢失。

3、优点

- 复杂查询

可以用SQL语句方便的在一个表以及多个表之间做复杂的数据查询

- 事物支持

使得对于安全性能很高的数据访问要求得以实现

2、非关系型数据库(非RDBMS)

非关系型数据库，可看成传统关系型数据库的功能阉割版本，**基于键值对存储数据**，不需要经过SQL层的解析，性能非常高。同时，通过减少不常用的功能，进一步提高性能。

1、分类

1、键值型数据库

键值型数据库典型的使用场景是作为内存缓存 产品：Redis

键值型数据库通过 **key-value** 键值的方式来存储数据，其中 **key** 和 **value** 可以是简单的对象，也可以是复杂的对象。Key 作为唯一的标识符，优点是查找速度快，在这方面明显优于关系型数据库，缺点是无法像关系型数据库一样使用条件过滤（比如 WHERE），如果你不知道去哪里找数据，就要遍历所有的键，这就会消耗大量的计算。

2、文档型数据库

存放并获取文档 产品：MongoDB

此类数据库可存放并获取文档，可以是XML、JSON等格式。在数据库中文档作为处理信息的基本单位，一个文档就相当于一条记录。文档数据库所存放的文档，就相当于键值数据库所存放的“值”。MongoDB 是最流行的文档型数据库。此外，还有CouchDB等。

3、搜索引擎数据库

产品：Solr、Elasticsearch、Splunk

虽然关系型数据库采用了索引提升检索效率，但是针对全文索引效率却较低。搜索引擎数据库是应用在搜索引擎领域的数据存储形式，由于搜索引擎会爬取大量的数据，并以特定的格式进行存储，这样在检索的时候才能保证性能最优。**核心原理是“倒排索引”**。

4、列式数据库

产品：HBase

列式数据库是相对于行式存储的数据库，Oracle、MySQL、SQL Server 等数据库都是采用的**行式存储（Row-based）**，而列式数据库是将数据按照列存储到数据库中，这样做的好处是可以**大量降低系统的I/O**，适合于分布式文件系统，不足在于功能相对有限。典型产品：HBase等。

5、图形数据库

产品：Neo4j、InfoGrid

图形数据库顾名思义，就是一种存储图形关系的数据库。它利用了图这种数据结构存储了实体（对象）之间的关系。关系型数据用于存储明确关系的数据，但对于复杂关系的数据存储却有些力不从心。如社交网络中人物之间的关系，如果用关系型数据库则非常复杂，用图形数据库将非常简单。典型产品：Neo4j、InfoGrid等。

5、关系型数据库的设计

- 关系型数据库的典型**数据结构**就是**数据表**，这些数据表组成都是结构化的。
- 将数据放到表中，表在放到库种。
- 一个数据库中可以有多个表，每个表都有一个名字，用来标识自己。
- 表名具有唯一性。表具有一些特性，这些特性定义了数据在表中如何存储，类似Java和Python中“类”的设计。

1、表、记录、字段

- E-R (entity-relationship, 实体-联系) 模型中有三个主要概念是: 实体集、属性、联系集。
- 一个**实体集 (class)** 对应于数据库中的一个**表 (table)**, 一个**实体 (instance)** 则对应于数据库表中的一行 (**row**), 也称为一条**记录 (record)**。一个**属性 (attribute)** 对应于数据库表中的一列 (**column**), 也称为一个**字段 (field)**。

	列					
字段	学号	姓名	年龄	性别	专业	属性
记录	161228001	张三	20	男	JavaEE	实体、对象
	161228002	李四	19	女	H5	
	161228003	王五	21	男	Android	
	161228004	赵六	20	女	PHP	
	161228005	钱七	23	男	JavaEE	
行	161228006	孙八	22	男	Android	

```
# ORM思想 (Object Relational Mapping)体现:
# 数据库中的一个表 <----> Java或Python中的一个类
# 表中的一条数据 <----> 类中的一个对象 (或实体)
# 表中的一个列 <-----> 类中的一个字段、属性(field)
```

2、表的关联关系

1. 表与表之间的数据记录有关系(relationship)。现实世界中的各种实体以及实体之间的各种联系均用 关系模型来表示。
2. 四种: 一对一关联、一对多关联、多对多关联、自我引用

1、一对一关联

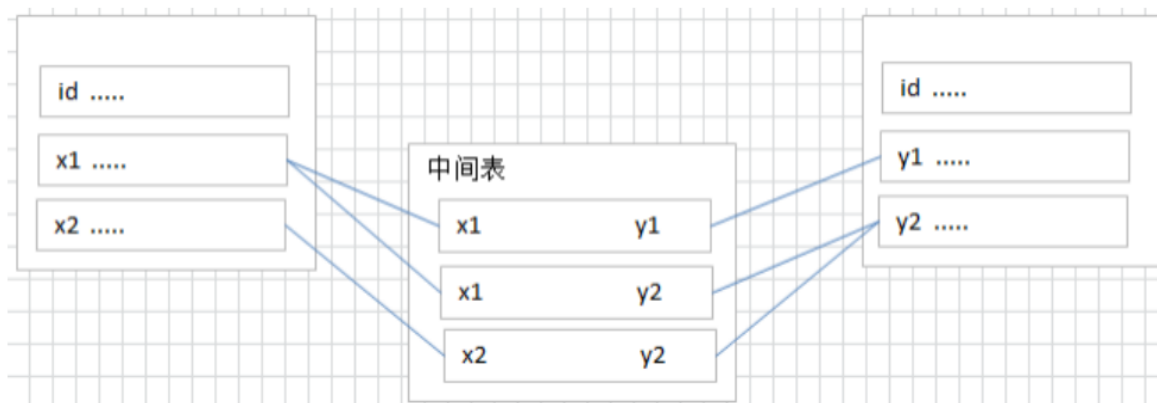
- 在实际的开发中应用不多, 因为一对一可以创建成一张表。
- 举例: 设计 学生表: 学号、姓名、手机号码、班级、系别、身份证号码、家庭住址、籍贯、紧急联系人、 ...
 - 拆为两个表: 两个表的记录是一一对应关系。
 - 基础信息表 (常用信息): 学号、姓名、手机号码、班级、系别
 - 档案信息表 (不常用信息): 学号、身份证号码、家庭住址、籍贯、紧急联系人、 ...
- 两种建表原则:
 - 外键唯一: 主表的主键和从表的外键 (唯一), 形成主外键关系, 外键唯一。
 - 外键是主键: 主表的主键和从表的主键, 形成主外键关系。

2、一对多关联

- 常见实例场景: 客户表和订单表, 分类表和商品表, 部门表和员工表。
- 举例:
 - 员工表: 编号、姓名、...、所属部门
 - 部门表: 编号、名称、简介
- 一对多建表原则: 在从表(多方)创建一个字段, 字段作为外键指向主表(一方)的主键

3、多对多关联

- 要表示多对多关系，必须创建第三个表，该表通常称为 联接表，它将多对多关系划分为两个一对多关系。将这两个表的主键都插入到第三个表中。
- 举例：学生-课程
 - 学生信息表：一行代表一个学生的信息（学号、姓名、手机号码、班级、系别...
 - 课程信息表：一行代表一个课程的信息（课程编号、授课老师、简介...）
 - 选课信息表：一个学生可以选多门课，一门课可以被多个学生选择



4、自我引用

员工编号	姓名	部门编号	主管编号
101	吴小亮	30	NULL
103	刘京生	30	101
104	李向明	30	103
105	张哲夫	30	103
210	黄 威	45	101
231	高大山	45	210

二、MySQL安装

环境：Windows 版本：

卸载

- 停止MySQL服务
- 卸载安装好的软件包
 - 控制面板删除
 - 安装包自带删除
- 删除缓存数据库
- 删除关于MySQL的环境变量
- 清理注册表（5.7及之前版本需要）

1、版本

- MySQL Community Server 社区版本，开源免费，自由下载，但不提供官方技术支持，适用于大多数普通用户。
- MySQL Enterprise Edition 企业版本，需付费，不能在线下载，可以试用30天。提供了更多的功能和更完备的技术支持，更适合于对数据库的功能和可靠性要求较高的企业客户。
- MySQL Cluster 集群版，开源免费。用于架设集群服务器，可将几个MySQL Server封装成一个Server。需要在社区版或企业版的基础上使用。
- MySQL Cluster CGE 高级集群版，需付费。

2、下载

- 安装包下载
- 压缩包下载（需要自己配置）

3、安装

.....

4、MySQL登陆

1、启动服务

```
net start mysql #服务名根据实际更改
net stop mysql #服务名根据实际更改
```

2、命令行登陆

- 格式：

```
mysql -h 主机名 -P 端口号 -u 用户名 -p 密码
```

- 举例：

```
mysql -h localhost -P 3306 -u root -p abc123
# 这里我设置的root用户的密码是abc123
```

- 注意：
 - -p与密码之间不能有空格，其他参数名与参数值之间可以有空格也可以没有空格。
 - 密码建议在下一行输入，保证安全

```
mysql -h localhost -P 3306 -u root -p
Enter password:
```

- 客户端和服务在同一台机器上，所以输入localhost或者IP地址127.0.0.1。同时，因为是连接本机：-hlocalhost就可以省略，如果端口号没有修改：-P3306也可以省略。简写成：

```
mysql -u root -p
Enter password:
```

- 连接成功后，有关于MySQL Server服务版本的信息，还有第几次连接的id标识。

- 也可以在命令行通过以下方式获取MySQL Server服务版本的信息：

```
c:\> mysql -V  
c:\> mysql --version
```

- 或登录后，通过以下方式查看当前版本信息：

```
mysql> select version();
```

- 退出登录

```
exit  
#or  
quit
```

三、MySQL基本使用

1、基本使用

1、登陆

```
mysql -u root -p
```

2、查看所有数据库

```
show databases;
```

- "information_schema"是 MySQL 系统自带的数据库，主要保存 MySQL 数据库服务器的系统信息，比如数据库的名称、数据表的名称、字段名称、存取权限、数据文件 所在的文件夹和系统使用的 文件夹，等等
- "performance_schema"是 MySQL 系统自带的数据库，可以用来监控 MySQL 的各类性能指标。
- "sys"数据库是 MySQL 系统自带的数据库，主要作用是以一种更容易被理解的方式展示 MySQL 数据库服务器的各类性能指标，帮助系统管理员和开发人员监控 MySQL 的技术性能。
- "mysql"数据库保存了 MySQL 数据库服务器运行时需要系统信息，比如数据文件夹、当前使用的 字符集、约束检查信息，等等

3、创建数据库

```
create database [newName];  
create database if not exists [newName];  
create database [newName] default charset utf8mb4;
```

4、使用已有的数据库

```
use [databaseName];
```

5、查看某个数据库的所有表格

```
#1
use [database];
show tables;
#2
show tables from [databaseName];
```

6、创建新的表格

```
create table [tableName](
    [field] [type],
    [field] [type]
);
```

7、查看某个表的数据

```
select * from [tableName];
```

8、添加一条记录

```
insert into [tableName] values([value1],[value2]....);
```

9、查看表的创建信息

```
show create table [tableName];
```

10、查看数据库的创建信息

```
show create database [databaseName];
```

11、删除表格

```
drop table [tableName];
```

12、删除数据库

```
drop database [databaseName];
```

2、字符集设置

修改安装目录下的 my.ini

```
[mysql] #大概在63行左右，在其下添加
...
default-character-set=utf8 #默认字符集
[mysqld] # 大概在76行左右，在其下添加
...
character-set-server=utf8
collation-server=utf8_general_ci
```

重启服务

3、图形化工具

1、Workbench

2、Navicat

3、SQLyog

4、dbeaver