1. 什么是数据库？

数据库是按照一定数据结构组织、存放和使用的数据仓库。

1. 为什么要使用数据库？

使用数据库便于存储、使用、分享数据，同时能保证数据相对安全。

1. 关系型数据库与非关系型数据库？

关系型数据库：mysql、oracle、sqlserver、

非关系型数据库：redis、mongodb

1. 数据库的基础概念：

数据库 --> 表 --> 字段

**主键**：能唯一代表表中某条记录的列称为主键。

**外键**：表中某一列的值是另外表中主键的值，那么这一列称为外键，外键能保证数据的一致性、完整性，当然我们通过应用程序也可以保证数据一致性与完整性。

**注意**：表中每一行叫**记录**、表中每一列叫**字段**

1. 创建mysql数据库、表、字段？

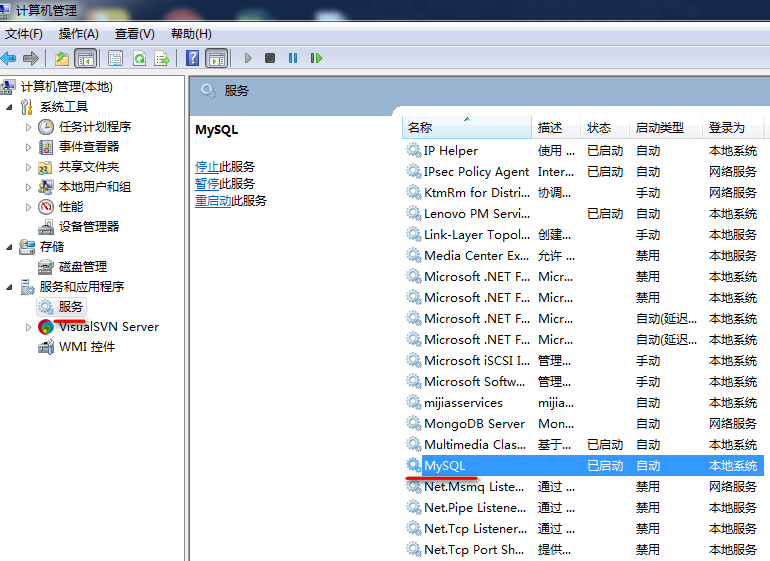
1. windows下启动/停止mysql服务：
2. 在dos小黑窗中使用下面命令：

d:> net start mysql //启动mysql服务

d:> net stop mysql //停止mysql服务

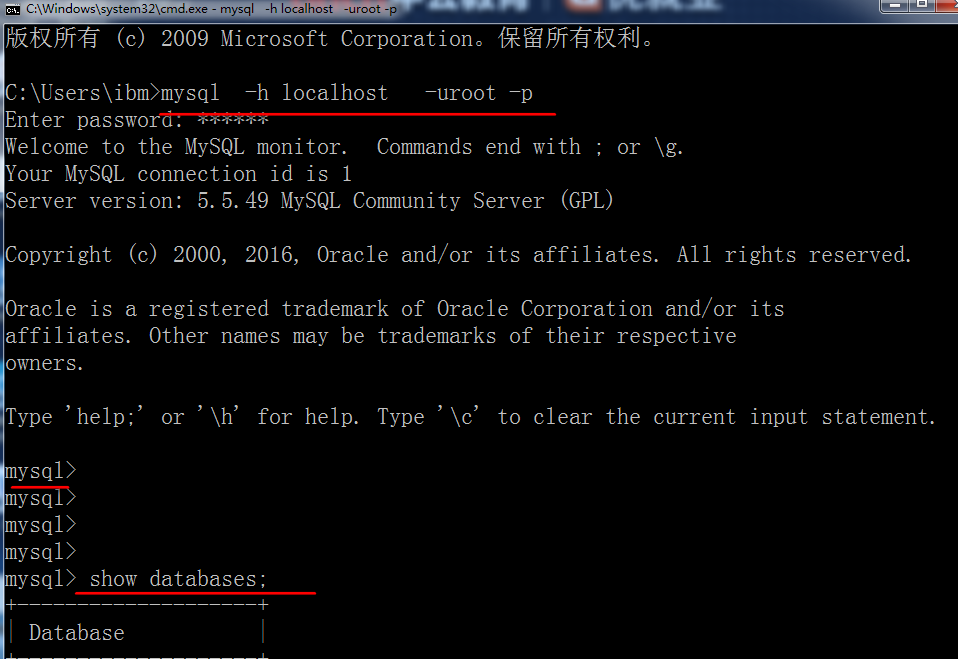
1. 在windows服务管理器中启动/停止mysql服务：

1)右单击桌面”计算机”图标-->”管理”，打开”windows服务管理器”，如下图：



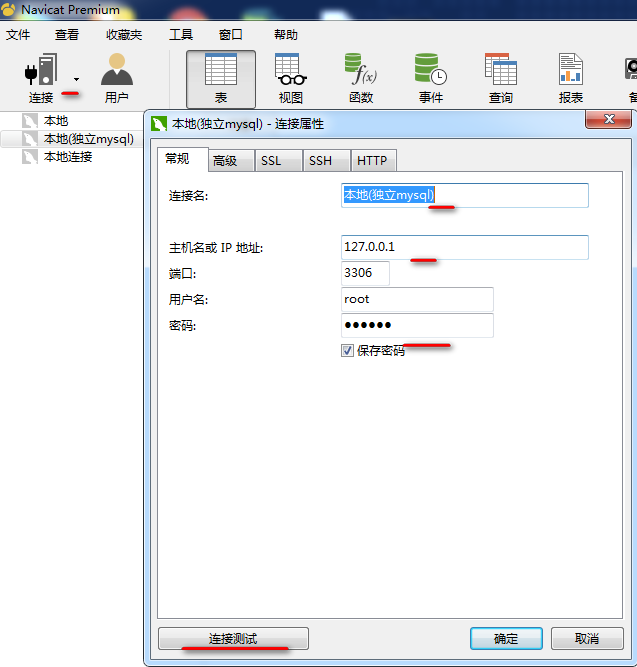
2)、右单击上图中的mysql服务即可停止/启动mysql服务。

1. 连接mysql服务器：
2. 使用mysql自带的客户端应用程序(mysql.exe)连接mysql服务器，如下：



**注**： -h表示要连接的mysql主机 -u表示使用的mysql用户 -p表示用户密码

1. 使用可视化客户端工具连接mysql数据库：



1. 常见的Mysql数据库**管理命令**：
2. show databases; //显示当前mysql主机中有哪些数据库
3. use 数据库名; //切换(打开)数据库
4. show tables; //显示当前库中有哪些表
5. create database 数据库名 //新建数据库
6. drop database 数据库名 //删除数据库 **注意：慎重使用**

1. 常用的字段类型有哪些？

1、数字类型：

整型： tinyint smallint int bigint

浮点型： decimal

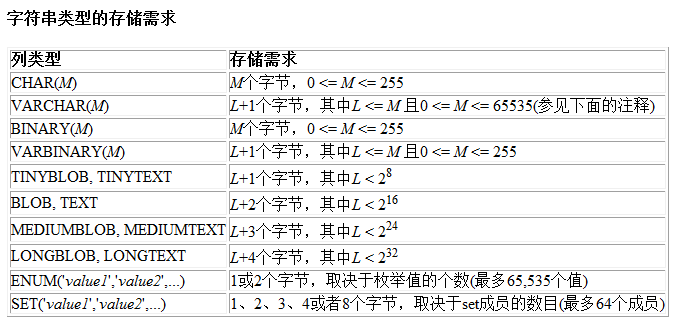


2、字符型：

char：通常用来存放值的长度相对是固定的，比如：性别，如果实际存储字符长度不够时mysql会自动使用空格来填充

varchar：通常用来存放值的长度不是固定的，按实际存储字符长度计算，存多少算多少

Text: 通常用来存放值的长度不是固定的、而且要存储的字符串比较多



3、日期时间类型：

date: 日期类型

datetime: 日期时间类型

**总结**：在选择字段类型时要根据实际存放值的类型来取舍，长度选择时既要满足当前存储的需要还要兼顾将来的扩展性。

1. 常见的mysql表类型：innodb、myisam
2. myisam表类型特点：强调速度、不支持事务、外键
3. Innodb表类型特点：强调稳定性、支持事务、外键

**事务**：比如给你朋友转2000元，那么要执行两个关键动作：你朋友帐户加2000、你自己帐户减2000，这个两个关键动作要么都成功、要么都失败。

1. 添加数据：

**insert into 表名 [(字段名1,字段名2...) ]values(值1,值2...)**

操作实例：

-- insert into banji(classname,teacher)values('python','郭老师')

-- insert into banji values(6,'web开发2','付老师');

insert into student(names,sex,age,tel,classid)values('小李','女',20,'12345',2)

1. 修改数据：

**update 表名 set 字段名1=’值1’,字段名2=’值2’... [where 条件]**

操作实例：

-- 将学生年龄减掉2岁

-- update student set age=age-2

-- 将班级id为2的学生性别改男、电话改为12345678

update student set sex='男',tel='12345678' where classid=2

1. 删除数据：

**delete from 表名 [where 条件]**

操作实例：

-- 删除班级id为2的学生信息

-- delete from student where classid=2

-- 删除年龄小于19岁的学生信息

delete from student where age < 19

1. 单表查询：

**select \*/字段名1,字段名2... from 表名 [where 条件] order by 字段名 asc[升序]/desc[降序] group by 字段名 limit m,n**

操作实例：

-- 查询所有学生信息

-- select \* from xsb

-- 查询计算机系中学生姓名、年龄、所在系

-- select xm,age,szx as '所在系' from xsb where szx='计算机系'

-- 查询计算机系和数学系的学生姓名、学号、所在系

-- select xm,xh,szx from xsb where szx='计算机系' or szx='数学系'

-- 上面的sql条件可以改用in()：

-- select \* from xsb where szx in( '计算机系', '数学系')

-- 查询计算机系的年龄大于等于19、小于等于21的学生姓名、学号、年龄

-- select \* from xsb where age >= 19 and age <= 21 and szx='计算机系'

-- 查询分数大于等于80、小于等95的分数信息

-- select \* from cjb where cj >= 80 and cj <= 95

-- 上面的查询条件改用between ... and

-- select \* from cjb where cj between 80 and 95

-- 查询姓王的学生信息: 模糊查询like

-- select \* from xsb where xm like '王%'

-- 查询姓王并且姓名为两个字的学生信息

-- select \* from xsb where xm like '王\_'

-- 查询计算机系的男生信息：

-- select \* from xsb where szx='计算机系' and sex='男'

-- 去重查询：查询有哪些系

-- select distinct szx from xsb

-- 查询计算机系的学生人数：

-- select count(\*) as total from xsb where szx='计算机系'

-- 排序：查询c01课程的分数信息并按分数降序排序

select \* from cjb where kch='c02' order by cj desc