

# Table of Contents

数据库课程	1.1
MySQL高级	1.2
数据库设计	1.2.1
命令行客户端	1.2.2
外键	1.2.3
索引	1.2.4

传智播客-黑马程序员

## 为什么学习数据库？

1、90%以上的软件都需要操作数据，比如游戏、社交、新闻、商城、财务等，这些软件都在不停的展示、存储数据，它们的数据都存储在数据库，数据库是软件的基础。

2、测试工程师在测试软件的过程中，不仅需要在界面上操作，还需要检查数据库中的数据是否正确。从而在软件出问题，测出更深层的问题。

例如：测试注册登录功能，在输入了注册信息后，提示注册成功，但是使用刚才注册的信息登录不成功。这时需要检查数据库中是否保存了正确的注册信息，如果数据库中没有保存数据，那么使用刚才注册的账号肯定登录不了，这样就能定位问题：注册时，没有把数据存储起来。

登录 · 注册

你的昵称

手机号

设置密码

注册

## 数据库阶段知识点

### 1. 数据库的基本概念

2

黑马程序员-软件测试

2. Navicat 操作数据库
  3. SQL 语言（重点）
  4. MySQL 高级（了解）
  5. Redis
- 

## 学习目标

- 熟练编写 SQL 语言中的查询语句

## MySQL高级

本节课程学习 MySQL 数据库的高级操作，不需要熟练操作，这部分内容作为了解，对于未来的工作是加分项

佐智播客-黑马程序员

# 数据库设计

## 1. E-R模型

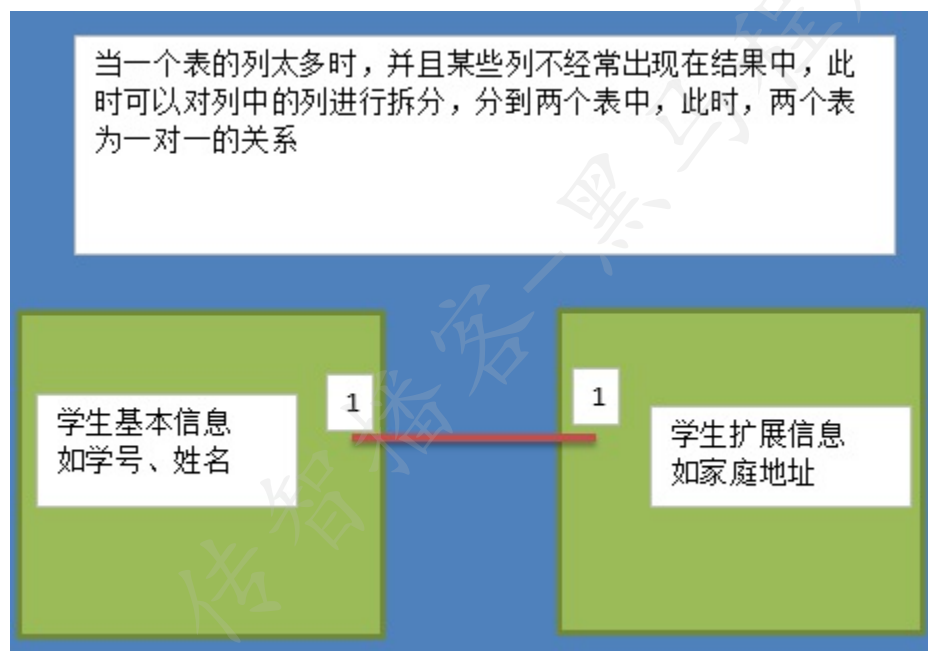
E-R模型的基本元素是：实体、联系和属性

- E 表示 entry，实体：一个数据对象，描述具有相同特征的事物
- R 表示 relationship，联系：表示一个或多个实体之间的关联关系，关系的类型包括包括一对一、一对多、多对多
- 属性：实体的某一特性称为属性

关系也是一种数据，需要通过一个字段存储在表中

### 1.1 一对一

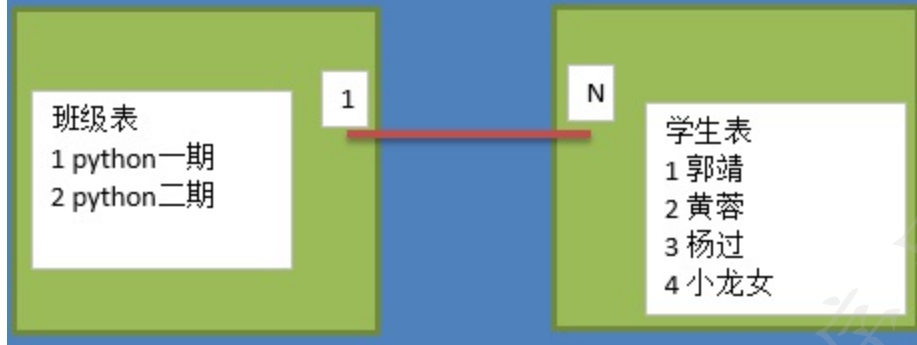
实体A对实体B为1对1，则在表A或表B中创建一个字段，存储另一个表的主键值



### 1.2 一对多

实体A对实体B为1对多：在表B中创建一个字段，存储表A的主键值

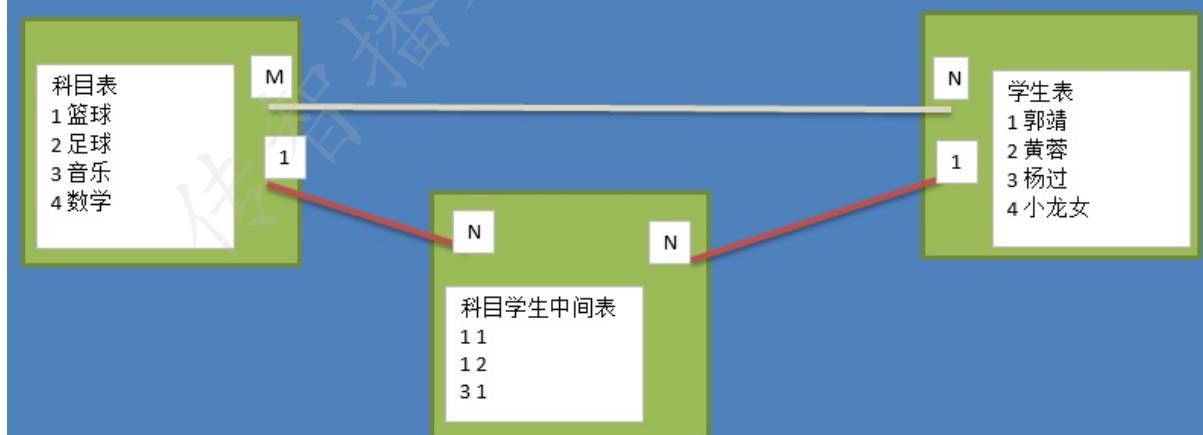
A表中的一条数据对应B表中的一条数据  
B表中的一条数据对应A表中的多条数据  
此时，A与B为多对一的关系  
可以举例来确定关系，如班级表、学生表



### 1.3 多对多

实体A对实体B为多对多：新建一张表C，这个表只有两个字段，一个用于存储A的主键值，一个用于存储B的主键值

A表中的一条数据对应B表中的多条数据  
B表中的一条数据对应A表中的多条数据  
此时，A与B为多对多的关系  
可以举例来确定关系，如学生表、科目表  
存储：将建中间表，存储表A和表B的主键



- 想一想：举些例子，满足一对一、一对多、多对多的对应关系

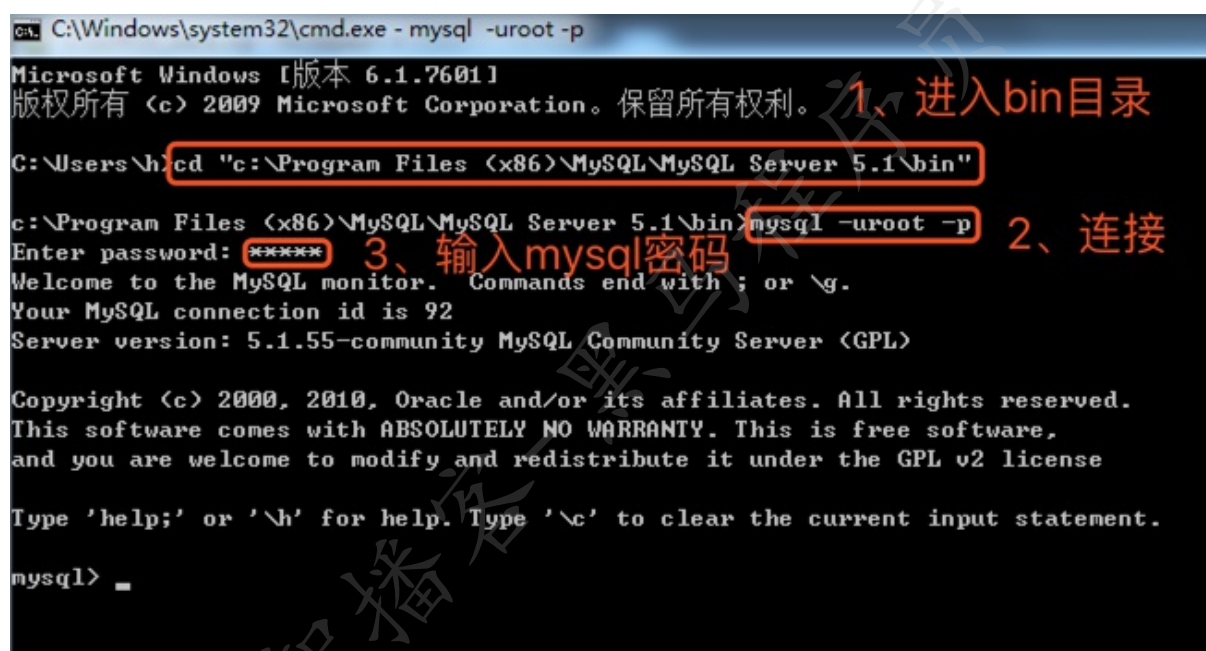
# 命令行客户端

## 1. 连接服务端

另一种方式，打开 cmd 程序，进入到 MySQL 安装目录的 bin 目录下

```
1、进入 MySQL 的 bin 目录
cd C:\Program Files (x86)\MySQL\MySQL Server 5.1\bin

2、连接 MySQL
mysql -uroot -p
```



```
C:\Windows\system32\cmd.exe - mysql -uroot -p
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。
C:\Users\h>cd "c:\Program Files (x86)\MySQL\MySQL Server 5.1\bin"
c:\Program Files (x86)\MySQL\MySQL Server 5.1\bin>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 92
Server version: 5.1.55-community MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```

## 2. 操作数据库

- 查看所有数据库

```
show databases;
```

- 使用数据库

```
use 数据库名;
```

- 查看当前使用的数据库

```
select database();
```

- 创建数据库

```
create database 数据库名 charset=utf8;  
例:  
create database ceshi charset=utf8;
```

- 删除数据库

```
drop database 数据库名;  
例:  
drop database ceshi;
```

### 3. 操作表

- 查看当前数据库中所有表

```
show tables;
```

- 查看表结构

```
desc 表名;
```

- 查看表的创建语句

```
show create table 表名;  
例:  
show create table students;
```



# 外键

## 1. 外键（foreign key）

- 如果一个实体的某个字段指向另一个实体的主键，就称为外键。
- 被指向的实体，称之为主实体（主表），也叫父实体（父表）。
- 负责指向的实体，称之为从实体（从表），也叫子实体（子表）
- 对关系字段进行约束，当为从表中的关系字段填写值时，会到关联的主表中查询此值是否存在，如果存在则填写成功，如果不存在则填写失败并报错

### 1.1 语法

#### 设置外键约束

- 方式一：创建数据表的时候设置外键约束

```
create table class(  
    id int unsigned primary key auto_increment,  
    name varchar(10)  
);  
  
create table stu(  
    name varchar(10),  
    class_id int unsigned,  
    foreign key(class_id) references class(id)  
);  
  
foreign key(自己的字段) references 主表(主表字段)
```

- 方式二：对于已经存在的数据表设置外键约束

```
alter table 从表名 add foreign key (从表字段) references 主表名(主表字段);  
  
alter table stu add foreign key (class_id) references class(id);
```

#### 查看外键

```
show create table 表名
```

#### 删除外键

```
-- 需要先获取外键约束名称  
show create table stu;  
-- 获取名称之后就可以根据名称来删除外键约束  
alter table 表名 drop foreign key 外键名称;  
  
alter table stu drop foreign key stu_ibfk_1;
```

在实际开发中,很少会使用到外键约束,会极大的降低表更新的效率

佐智播客-黑马程序员

# 索引

- 思考：在字典中是如何找到一个汉字？
- 一般的应用系统对比数据库的读写比例在10:1左右，而且插入操作和更新操作很少出现性能问题，遇到最多的，也是最容易出问题的，还是一些复杂的查询操作，所以查询语句的优化显然是重中之重
- 当数据库中数据量很大时，查找数据会变得很慢
- 优化方案：索引

## 1. 索引演示

### 导入测试表test\_index

右键点击某个数据库->运行sql文件->选择test\_index.sql->点击开始

### 查询

- 开启运行时间监测：

```
set profiling=1;
```

- 查找第1万条数据test10000

```
select * from test_index where title='test10000';
```

- 查看执行的时间：

```
show profiles;
```

- 为表title\_index的title列创建索引：

```
create index title_index on test_index(title(10));
```

- 执行查询语句：

```
select * from test_index where title='test10000';
```

- 再次查看执行的时间

```
show profiles;
```

## 2. 索引语法

- 查看索引

```
show index from 表名;
```

- 创建索引

方式一：建表时创建索引

```
create table create_index(  
    id int primary key,  
    name varchar(10) unique,  
    age int,  
    key (age)  
);
```

方式二：对于已经存在的表，添加索引

如果指定字段是字符串，需要指定长度，建议长度与定义字段时的长度一致  
字段类型如果不是字符串，可以不填写长度部分

```
create index 索引名称 on 表名(字段名称(长度))  
例：  
create index age_index on create_index(age);  
create index name_index on create_index(name(10));
```

- 删除索引：

```
drop index 索引名称 on 表名;
```

### 3. 索引缺点

- 虽然索引大大提高了查询速度，同时却会降低更新表的速度，如对表进行INSERT、UPDATE和DELETE，因为更新表时，MySQL不仅要保存数据，还要保存一下索引文件
- 但是,在互联网应用中,查询的语句远远大于增删改的语句,甚至可以占到80%~90%,所以也不要太在意,只是在大数据导入时,可以先删除索引,再批量插入数据,最后再添加索引

### 4. 分析查询

```
explain  
select * from test_index where title='test10000'
```

查询创建工具

查询编辑器

```

1 explain
2 select * from test_index where title='test10000'

```

信息	解释	结果1	概况	状态					
id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	test_index	ref	title_index	title_index	33	const	1	Using where; Using index