

大学计算机基础——培养计算思维

第6章 数据库技术基础 (数据的组织与管理)

西交大 卫颜俊



西安交通大学
计算机教学实验中心



本章内容

■ 了解

- 数据库的发展
- 数据库、基本功能与类型
- 数据库的基本功能

■ 掌握

- 关系数据库
 - 数据库结构的设计
 - **SQLite**关系数据库管理系统
 - **SQL**语言基本语法
-



6.1 数据库技术的概念 (数据库的基础知识)



6.1.1 数据管理技术与发展

■ 什么是数据管理？

- 对数据的组织、分类、编码、存储、检索和维护等环节的操作。它是数据处理的核心。

■ 数据管理经历了三个发展阶段

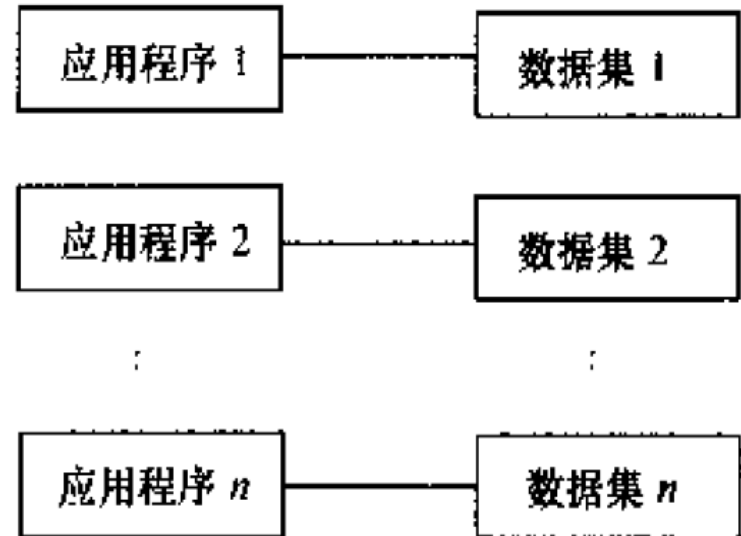
- 人工管理
- 文件系统
- 数据库系统



1. 人工管理阶段

■ 特征：

- 数据依赖于特定的应用程序，数据与程序不可分割。

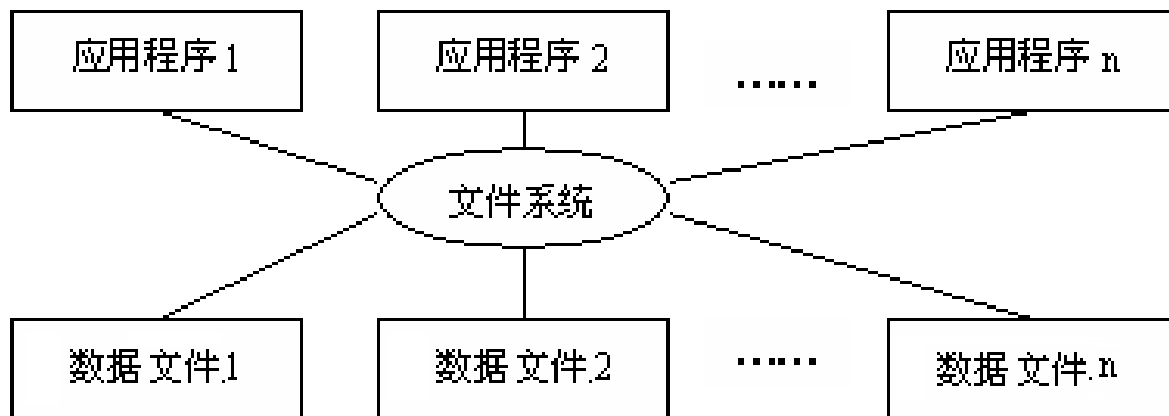


■ 特点：

- 数据和应用程序一一对应
- 数据不进行存储，不能实现共享。
- 各程序之间所用的数据彼此孤立，数据组织不具有逻辑性和独立性
- 各程序之间大量的数据冗余存在。



2. 文件系统阶段



■ 特征：

程序和数据分开存储，程序可以通过文件系统读写不同的文件；文件一个文件可以被不同的程序使用。

■ 优点

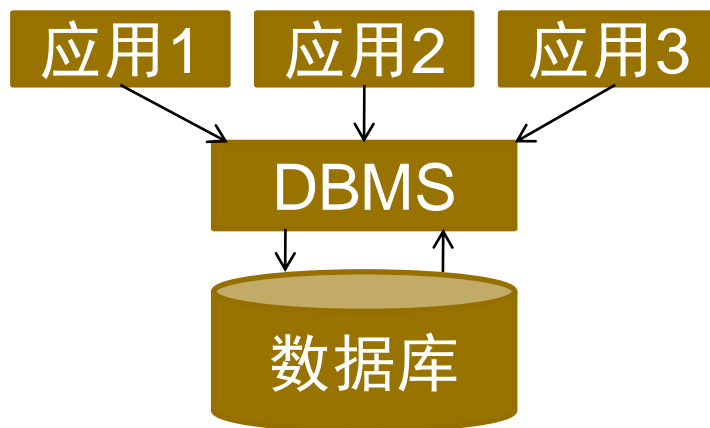
数据可以长期保存
数据具有一定的独立性。
数据具有共享性。

■ 缺点：

程序只能读写一定结构的数据，独立性和共享性不够。
数据文件间联系松散，数据存在冗余和不一致现象。



3. 数据库系统阶段



■ 特征：

- 数据按一定的数据模型进行组织，使用专门的软件管理数据，这就是**数据库管理系统**。

■ 特点：

- 数据结构化，数据间有联系
- 数据统一管理，数据的共享性高，冗余度低，易扩充
- 数据独立性高。可以被多个应用程序使用，数据的存储方式、结构改变不影响程序



6.1.2数据库管理系统

- 1.数据库（ Database ）
- 以一定方式储存在一起、能为多个用户共享、具有尽可能小的冗余度的、并与应用程序彼此独立的数据集合。
- 2.数据库的特点
 - 数据尽可能不重复，
 - 数据组织具有较优的方式，
 - 其数据结构独立于应用程序，
 - 对数据的增、删、改、查由统一的专门软件进行管理和控制。



■ 3.数据库管理系统

- 对数据库进行管理的系统软件，它以统一的方式管理和维护数据库，响应和完成用户提出的各种访问数据的请求。

■ 4.数据库管理系统的功能

- 数据定义：定义数据对象，数据库、数据表、索引
- 数据操纵：数据的维护，插入、修改、删除、查找
- 运行控制：安全性检查、完整性约束、多用户
- 建立和维护数据库：初始化文件、存储位置、备份、恢复等



5. 常见数据库管理系统

- Informix
- SYBASE
- Visual Foxpro
- ORACLE
- IBM DB2
- SQL Server
- MySQL
- Access
- SQLite





6.数据模型

- 数据模型是指对数据进行的特定的组织结构。

(1) 层次模型

用树形结构组织数据，可以表示数据之间的多级层次结构

IBM公司的IMS

(2) 网状模型

用图表示数据之间的关系，可以表示多对多的联系。

DBTG

(3) 关系模型

数据之间的联系用二维表格的形式来形象地表示。 **SQLite**



7.数据库系统

- 在计算机系统中引入数据库后的系统，一般由数据库（DB）、数据库管理系统（DBMS）、应用系统、数据库管理员（DBA）构成。
- 不混淆的情况下：
 - 数据库系统->数据库
 - 数据库管理系统->数据库



6.1.3 关系数据库

■ 采用关系模型的数据库

读者信息表

编号	姓名	年龄	性别
0800001	黄青山	18	男
0800002	李化	17	女
0800003	周得鑫	19	女
0800004	宋青	18	男

← 字段

← 记录

1. 关系模型的概念

采用二维表格的形式组织数据

每行数据是一个**记录**

每列是一个**字段**



关系模型的几个概念

字段名：列名

字段值：具体取值

域：每个字段的取值范围

表结构：表中所有字段的名称和属性的集合，是记录的组成形式。

关系模式：对关系结构的描述，简记为：

关系名（字段1，字段2，字段3，...，字段n）

读者信息表（编号，姓名，年龄，性别）



2.关系模型的特点

读者信息表

编号	姓名	年龄	性别
0800001	黄青山	18	男
0800002	李化	17	女
0800003	周得鑫	19	女
0800004	宋青	18	男

← 字段

← 记录

- (1) 关系中的每一列不可再分。
- (2) 同一个关系中不能出现相同的字段名。
- (3) 关系中一般不出现完全相同的两条记录。
- (4) 关系中任意交换两行位置不影响数据的实际含义。
- (5) 关系中任意交换两列位置不影响数据的实际含义



3.关系中的候选键和主键

■ (1) 候选键 (Candidate Key)

- 一个关系中可以用来唯一地标识一个记录的字段或字段的集合，称为**候选键**。
- **一个关系中，可以有多个候选键**
- 候选键可以是单个字段，也可以是**字段的组合**。

■ 下表是某校学生信息表，哪些字段可以作为候选键？

班级	学号	姓名	身份证号	联系电话
建环01	1001	张建	620***	
建环01	1002	张建	420***	
建环02	1102	孙浩	350***	



【思考题】分析以下的借阅关系中的候选键

编号	书号	借阅日期
0899001	C01	09/10/2008
0899001	C02	08/09/2008
0899002	C02	10/11/2008



■ (2) 主键 (Primary Key)

- 从多个候选键中指定其中的一个作为数据元素的唯一标识，该候选键称为**主键**。
- 设置主键的目的是为了实现**实体完整性**的约束规则
- **实体完整性**，关系（表）中的记录在主键上**不允许重复，也不允许为空**。

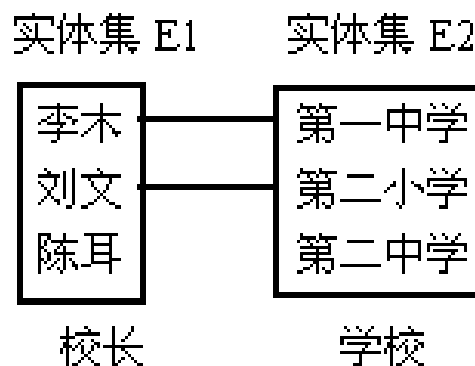
学号	姓名	成绩
0899001	张浩	90
0899001	李薇	93
	王菲	91



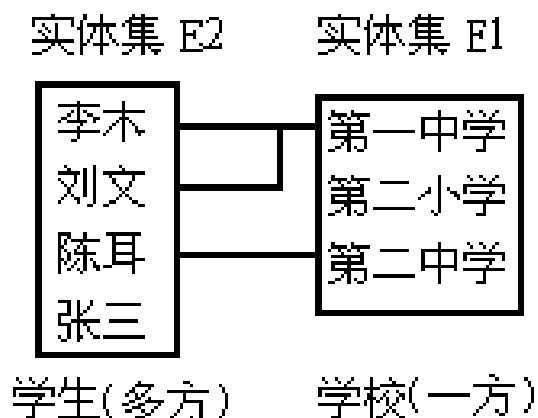
4. 实体之间的联系

表与表之间的联系

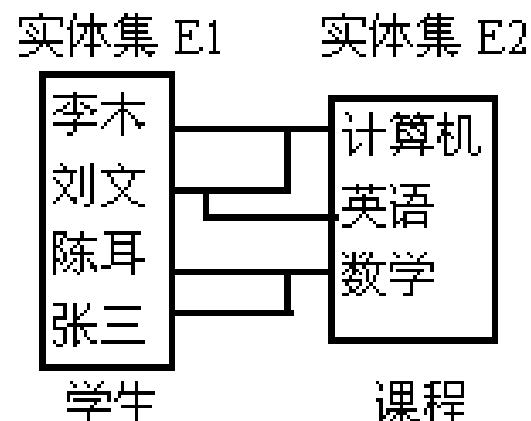
■ 一对一



■ 一对多



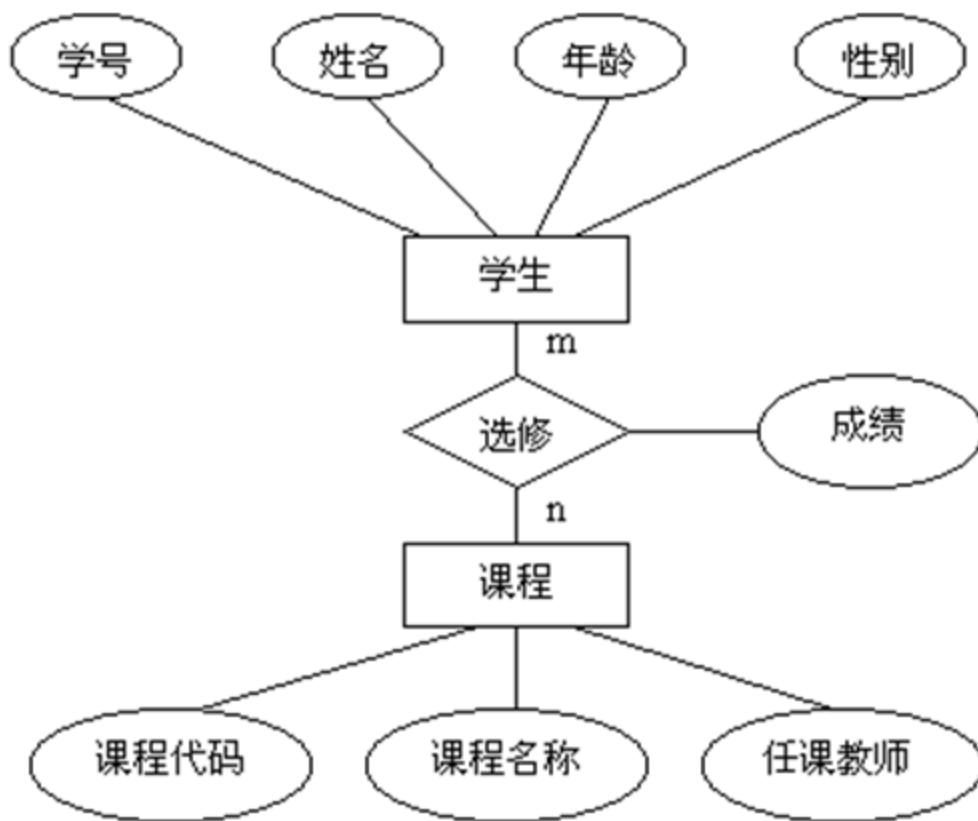
■ 多对多





5.E-R图

■ 实体、属性、实体集间的联系的表达





6.关系的三种完整性

关系的完整性约束确保数据的合理性

(1) 实体完整性

关系的主键不能取空值，不能重复。

比如学生表的主键通常是取学号为主键



学生信息

2.参照完整性

班级	学号	姓名	身份证号	联系电话
建环01	0899001	张建	620***	
建环01	0899002	张建	420***	
建环02	0899032	孙浩	350***	

借阅信息

外键、外键
主表，被参照关系
从表，参照关系

编号	书号	借阅日期
0899001	C01	09/10/2008
0899101	C02	08/09/2008
0899032	C02	10/11/2008

参照完整性约束： 参照关系中每个元素的外键要么为空，要么等于被参照关系中某个元素的主键。
参照关系也称为**外键表**，被参照关系也称为**主键表**。



关系的三种完整性

(3) 用户定义的完整性

对关系中每个域的取值作限制(或称约束)的具体定义。

比如

性别属性只能取“男”或“女”，

年龄的取值范围，可以取值**0-130**，但不能取负数，因为年龄不可能是负数。



7.表的设计

■ 读者登记表

借书证号	姓名	性别	年龄	专业

数据类型

表名：读者登记表

字段名称	字段类型	字段大小	是否主键	默认值	取值范围	允许空？
借书证号	文本	8	是			
姓名	文本	20				
性别	文本	2				
年龄	数值	整型				
专业	文本	20				



表的设计 (2)

- 读者登记表
- 图书信息表
- 借书信息表



6.1.4数据库管理系统SQLite简介

- 一种遵守ACID的关系型数据库管理系统，在其上可以建立轻型的数据库。
 - ACID
 - 指数据库事务正确执行的四个基本要素的缩写。
 - 事务
 - 数据库中数据操作的一个完整的过程，叫做一个（事务）
 - 如，转账，一个账户的转出和另一个账户的转入
 - 四个基本要素
 - 原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）
-



ACID

■ 原子性

- 整个事务中的所有操作，要么全部完成，要么全部不完成，不可能停滞在中间某个环节。

■ 一致性

- 在事务开始之前和事务结束以后，数据库的完整性约束没有被破坏。

■ 隔离性

- 隔离状态执行事务，使它们好像是系统在给定时间内执行的唯一操作。

■ 持久性

- 在事务完成以后，该事务所对数据库所作的更改便持久的保存在数据库之中。



SQLite简介

- 创始人是D.RichardHipp。
 - 设计目标是嵌入式应用，目前已经在很多嵌入式产品中得到应用，它占用资源非常的低，在嵌入式设备中，可能只需要几百K的内存就够了。
 - 支持Windows/Linux/Unix等主流的操作系统，同时能够跟很多程序语言相结合，比如Java、C/C++、C#、Python、PHP等，还有ODBC接口
 - 处理速度比Mysql、PostgreSQL这两款开源的世界著名数据库管理系统都快。
 - 第一版诞生于2000年5月。目前推出SQLite 3版。
 - 官网：www.sqlite.org version 3.9.1
-



SQLite应用举例

- SQLite大量用于手机，PDA，MP3播放器以及机顶盒设备中。
 - Mozilla Firefox使用SQLite作为数据库。
 - Mac计算机中包含了多份SQLite的拷贝，用于不同的应用。
 - PHP将SQLite作为内置的数据库。
 - Skype客户端软件在内部使用SQLite。
 - Symbian OS(智能手机操作平台的领航)内置SQLite。
 - AOL邮件客户端绑定了SQLite。
 - Solaris 10在启动过程中需要使用SQLite。
 - McAfee杀毒软件使用SQLite。
 - iPhones使用SQLite。
 - Symbian和Apple以外的很多手机生产厂商使用SQLite。
-



SQLite使用

■ 下载

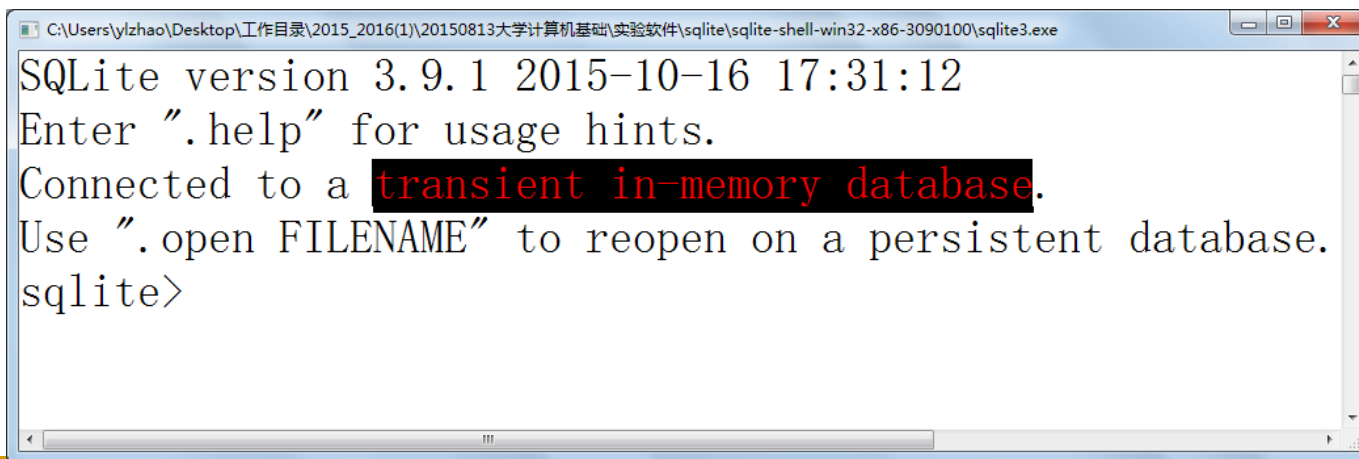
- <http://www.sqlite.org/2015/sqlite-shell-win32-x86-3081101.zip> ([sqlite-shell-win32-x86-3090100.zip](#))

■ 解压

- `sqlite-shell-win32-x86-3081101.zip` -> `sqlite3.exe`

■ 交互式命令

- 在控制台输入**SQLie3**命令(双击**sqlite**)



```
C:\Users\ylzhao\Desktop\工作目录\2015_2016(1)\20150813大学计算机基础\实验软件\sqlite\sqlite-shell-win32-x86-3090100\sqlite3.exe

SQLite version 3.9.1 2015-10-16 17:31:12
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```



■ **sqlite>下输入下列命令：**

- **.open students.db**
- **create table student(cname varchar(20),number varchar(10),name varchar(40),score float);**
- **insert into student values("物理51","215001001","张三",91);**
- **insert into student values("物理51","215001002","李斯",92);**
- **insert into student values("物理51","215001003","王伟",88);**
- **insert into student values("物理52","215001061","孙犁",45);**



- **select * from student ;**
- **select * from student where score<60;**
- **select * from student where score>60;**
- **update student set cname="物理52" where number="215001001";**
- **select * from student where cname="物理51";**
- **.save student.db**
- **quit**
- **再次启动Sqlite**
- **.open student.db**
- **.tables**
- **select * from student;**



- `sqlite> create table tmp(name
varcahr(20),number varchar(20),score float);`
- `sqlite> .tables`
`student tmp`
- `sqlite> drop table tmp;`
- `sqlite> .tables`
`student`



- `sqlite`的命令有两类。
- 一类是关于数据库的**控制的命令**，象`.help`命令及使用它列出的命令，它们在使用时开头都有一个点号“.”。如果在使用中忘记加点号，系统会显示“...>”，实际是认为命令还没有输入完，这时请输入一个分号“;”，按回车，然后重新输入正确的命令

```
sqlite> help
```

```
...> ;
```

```
Error: near "help": syntax error
```

```
sqlite> .help
```

```
.backup ?DB? FILE          Backup DB (default "main") to FILE
```



- 另一类sqlite命令是SQL语句，使用SQL语句，前面不加点号，但后面要加分号，例如：
- `sqlite> select * from reader;`
- 如果忘记加分号，也会产生上面的情况，直接输入分号，回车，执行SQL语句。
- 有时一个命令会使用多次。如果想再使用已经用过的命令，或修改错误命令的个别字符，可以在提示符下按上箭头键，找到刚才的命令，使用左右箭头移动光标，可以修改这条命令。修改后按回车键执行。



常用命令

- 打开数据文件
 - **.open FILE**
- 保存数据文件
 - **.save FILE**
- 显示数据表
 - **.tables**
- 退出
 - **.quit**
 - **.exit**
- 显示创建表的语句
 - **.schema ?TABLE?**
- 设置数据分隔符
 - **separator "\t"**
- 执行文件中的sql语句
 - **.read FILENAME**
- 从文本文件中导入数据
 - **.import FILE TABLE**
- 显示数据库名
 - **.databases**
- 显示或关闭表头
 - **.header on/off**



SQLite常用数据类型

- 数据库中存储的每个值都具有一个数据类型
 - **NULL**: 空值
 - **INTEGER**: 整数,依据值的大小依次被存储为1,2,3,4,5,6,7,8个字节
 - **REAL**: 所有值都是浮动的数值,被存储为8字节的IEEE浮动标记序号
 - **TEXT**: 文本. 值为文本字符串,使用数据库编码存储(TUTF-8, UTF-16BE or UTF-16-LE)
 - **BLOB**: 原始大字节值,如何输入就如何存储,不改变格式
-



SQLite常用数据类型

■ 接受以下数据类型

- **smallint** 和 **integer** 分别为16位和32位的整数。
- **decimal(p,s)** 精确值 **p** 是指十进制位数, **s** 是指小数点后几位小数。如果没有特别指定, 则系统会默认为 **p=5 s=0**。
- **float** 32位的实数。
- **double** 64位的实数。
- **char(n)** **n** 长度的字符串, **n** 不能超过 254。
- **varchar(n)** 长度不固定且其最大长度为 **n** 的字符串, **n** 小于 4000。
- **graphic(n)** 和 **char(n)** 一样, 不过其单位是两个字节, **n** 不能超过 127。这个形态是为了支持两个字节长度的字体, 如中文字。
- **vargraphic(n)** 可变长度且其最大长度为 **n** 的双字节字符串, **n** 小于 2000。
- **date** 包含了年份、月份、日期。
- **time** 包含了小时、分钟、秒。
- **timestamp** 包含了年、月、日、时、分、秒、千分之一秒。



补充练习

- 1.设有“图书登记表”，其中的字段包括**ISBN**、书名、作者、出版社、出版年代等。
 - (1)写出创建该表的**SQL**语句
 - (2)写出查找作者“张帆”写的书的**SQL**语句。
- 2.设计两个表，指出其中可以作为主键的字段，指出外键字段，说出哪个是主表，哪个是从表，哪个是参照关系，哪个是被参照关系。