

2020/08/19_数据库_第1课_数据库表的设计、降低数据冗余率

笔记本： 数据库

创建时间： 2020/8/19 星期三 10:20

作者： ileemi

- [数据库](#)
- [数据管理技术的发展](#)
- [数据库的基本概念](#)
 - [数据描述](#)
 - [数据模型](#)
- [关系型数据库相关术语](#)
- [数据库的设计](#)
 - [学生管理系统的设计](#)

数据库

相关软件：

mysql：开源、免费（关系型数据库）

sql server、**access**、**sqlite**

使用数据库的好处：管理大数据效率较高

数据管理存在两个问题：冗余（需要自己处理）、效率（数据库软件系统可以帮助解决）

管理系统：生活中很常见，需求大，大量数据需要进行高效的管理

数据库系统本身可以看成是一个服务器

数据管理技术的发展

1. 数据与信息

数据是现实世界的物质表示形式。如人的身高、体重等。

信息是经过加工和处理过的数据。如求一个班的总分和平均及格率等。

通过计算可以算出来的数据不需要进行存储。

2. 数据管理

数据管理就是对数据进行存储、更新和检索。

3. 数据管理的三个发展阶段：人工管理、文件系统、数据库。

人工管理阶段特点：

- 数据不长期保存在计算机里

- 应用程序管理数据，数据与程序结合在一起
- 数据不共享、仅面向某个应用
- 数据不具有独立性

文件管理阶段：

- 数据以文件的形式长期保存在计算机里
- 操作系统的文件管理提供了对数据的输入输出的管理
- 数据可以共享
- 数据文件之间彼此孤立，不能反映数据之间的联系
- 数据冗余、数据可能不一致

数据库管理阶段特点：

- 数据具有结构化
- 数据的共享性高、冗余低
- 数据的独立性高
- 数据由 DBMS（数据库管理系统）统一管理和控制

数据库的文件格式定义比较复杂，个人信息不用存储，数据库可以帮助我们计算所需的信息数据。

数据库的基本概念

数据描述

- 现实世界：存在于人脑之外的客观世界
实体：客观存在，可相互区分的事物
特征：实体的特性
- 信息世界：客观事物在人们头脑中的反映
实体记录：表示实体
实体集：性质相同的同类实体集合
属性：实体具有的特性
- 机器世界：信息在机器世界中的数据存储形式
字段：标识实体属性的命名单位
记录：字段的有序集合（一个记录描述一个实体）
关键字：唯一标识每个记录的字段或字段集

举例：

实体：人 — 电视

字段：身高 — 尺寸

关键字（主键）：身份证 学号ID（唯一）

属性：性别

字段：姓名 身高

记录：张三 170

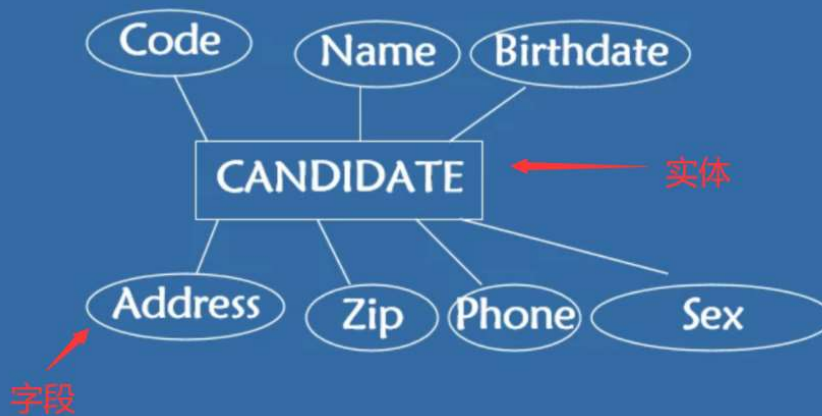
数据模型

描述数据结构和在数据库管理系统中的访问技术，是对现实世界的抽象。

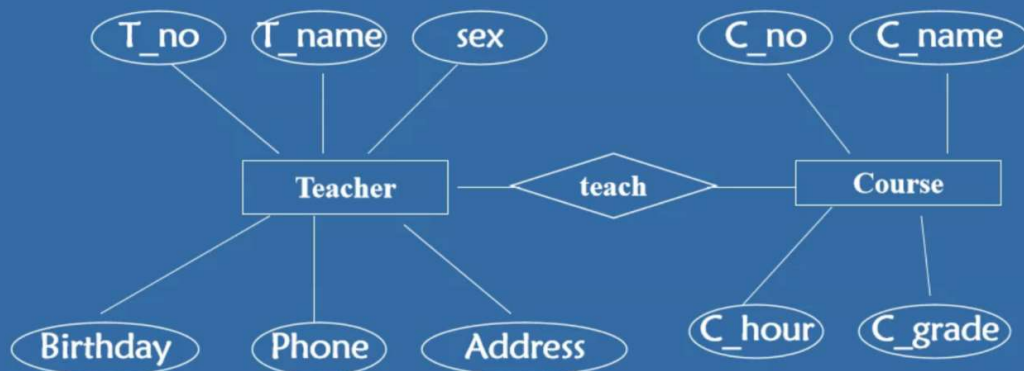
关系数据库管理系统

ER图（数据关系） -- 使用图形表示数据间的关系

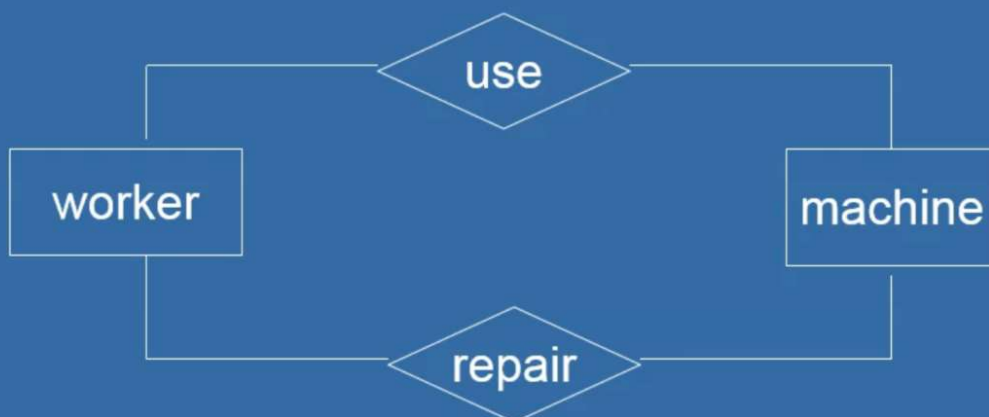
2、实体图示例



3、实体-关系图(E-R)示例



5、两实体间可存在多种关系



数据间的关系三种（对数据库做表影响较大）：

- 一对一关系 -- 大学与校长的关系
- 一对多关系 -- 部门与职员的关系
- 多对多关系 -- 顾客与商品的关系

关系型数据库相关术语

1. **数据库**: 数据库是一些关联表的集合。
2. **数据表**: 表是数据的矩阵。在一个数据库中的表看起来像一个简单的电子表格。
3. **列**: 一列(数据元素) 包含了相同类型的数据, 例如邮政编码的数据。
4. **行**: 一行 (=元组, 或记录) 是一组相关的数据, 例如一条用户订阅的数据。
5. **冗余**: 存储两倍数据, 冗余降低了性能, 但提高了数据的安全性。
6. **主键**: 主键是唯一的。一个数据表中只能包含一个主键。你可以使用主键来查询数据。
7. **外键**: 外键用于关联两个表。
8. **复合键**: 复合键 (组合键) 将多个列作为一个索引键, 一般用于复合索引。
9. **索引**: 使用索引可快速访问数据库表中的特定信息。索引是对数据库表中一列或多列的值进行排序的一种结构。类似于书籍的目录。
10. **参照完整性**: 参照的完整性要求关系中不允许引用不存在的实体。与实体完整性是关系模型必须满足的完整性约束条件, 目的是保证数据的一致性。

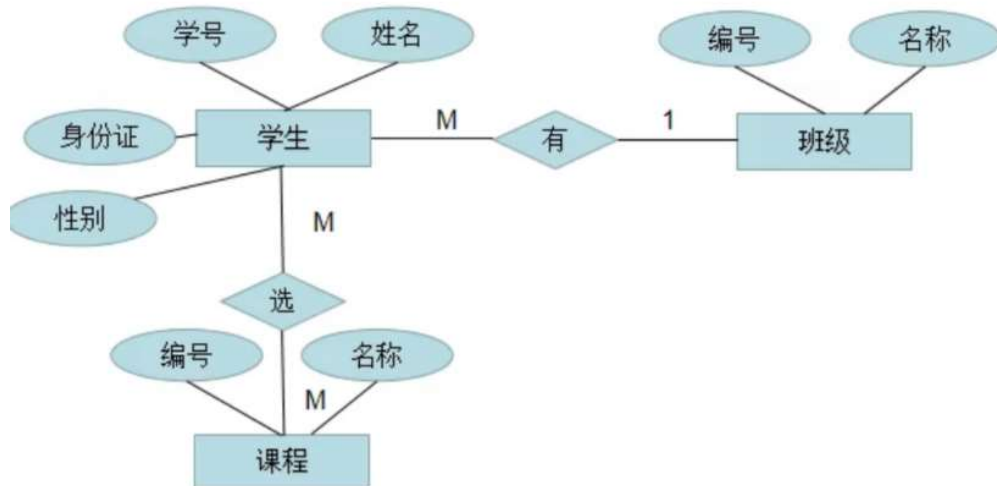
数据库的设计

学生管理系统的设计

绘制ER图前的步骤：

1. 找实体：学生、班级、课程（系统中的实体不能太多，不好管理，要找到系统设计的核心，根据核心找对应的系统）
2. 属性：学生（学号、姓名、性别、出生日期、身份证、联系方式）班级（编号，班级名称，开班时间）、课程（课程编号、课程名称）

3. 绘ER图（绘制实体） -- 可以直观的找出冗余较低的方案



设计数据库表的时候，需要考虑冗余度，首选冗余度低的方案。同时需要根据数据库表的需求设计出对应的实体，属性等。

解决冗余度较高的方法：

一对多关系：少的实体字段放在多的实体里

多对多关系：产生一个新表（降低重复字段）