



第一章 绪 论



知识点回顾

- 1、数据库相关概念
- 2、数据管理技术的产生与发展
- 3、概念模型及E-R图画法



补充知识点---弱实体集

□ 弱实体集

如果一个实体集的**所有属性都不足以形成主码**，则称这样的实体集为**弱实体集**

- ◆ 贷款（贷款号，金额），还款（还款号，还款日期，金额），每个“贷款”的各个“还款”不同，但不同“贷款”之间的“还款”却可能相同，因此**“还款”是一个弱实体集**。
- ◆ 产品（名称，价格），公司（名称，地址，联系电话），“产品”与“公司”之间有“制造”联系，**“产品”是一个弱实体集**。



弱实体集

- ◆ 弱实体集与其拥有者之间的联系称作**标识性联系**
- ◆ 弱实体集与强实体集之间是**多对一**的联系
- ◆ 弱实体集**必然存在依赖于**强实体集
- ◆ **存在依赖并不总会导致一个弱实体集，从属实体集可以有自己的主码**
- ◆ 如实体集信用卡（信用卡号，客户帐号，金额），它存在依赖于客户帐号实体集，但**信用卡有自己的主码信用卡号**。



弱实体集

- 分辨符
- 弱实体集中用于区别依赖于某个特定强实体集的属性集合。也称作部分码
 - ◆ 如“还款”中的还款号
 - ◆ 弱实体集的主码由该弱实体集所存在依赖的强实体集的主码和该弱实体集的分辨符组成
 - 如“还款”主码 = 贷款号 + 还款号
 - “产品”主码 = 产品名称 + 公司名称。



弱实体集

□ 为什么使用弱实体集？

通过为弱实体集加上合适的属性，可转变为强实体集，
为什么还要使用弱实体集？

- ◆ 避免数据冗余（强实体集码重复），以及因此带来的数据的不一致性
- ◆ 弱实体集反映了一个实体对其它实体依赖的逻辑结构
- ◆ 弱实体集可以随它们的强实体集的删除而自动删除
- ◆ 弱实体集可以物理地随它们的强实体集存储



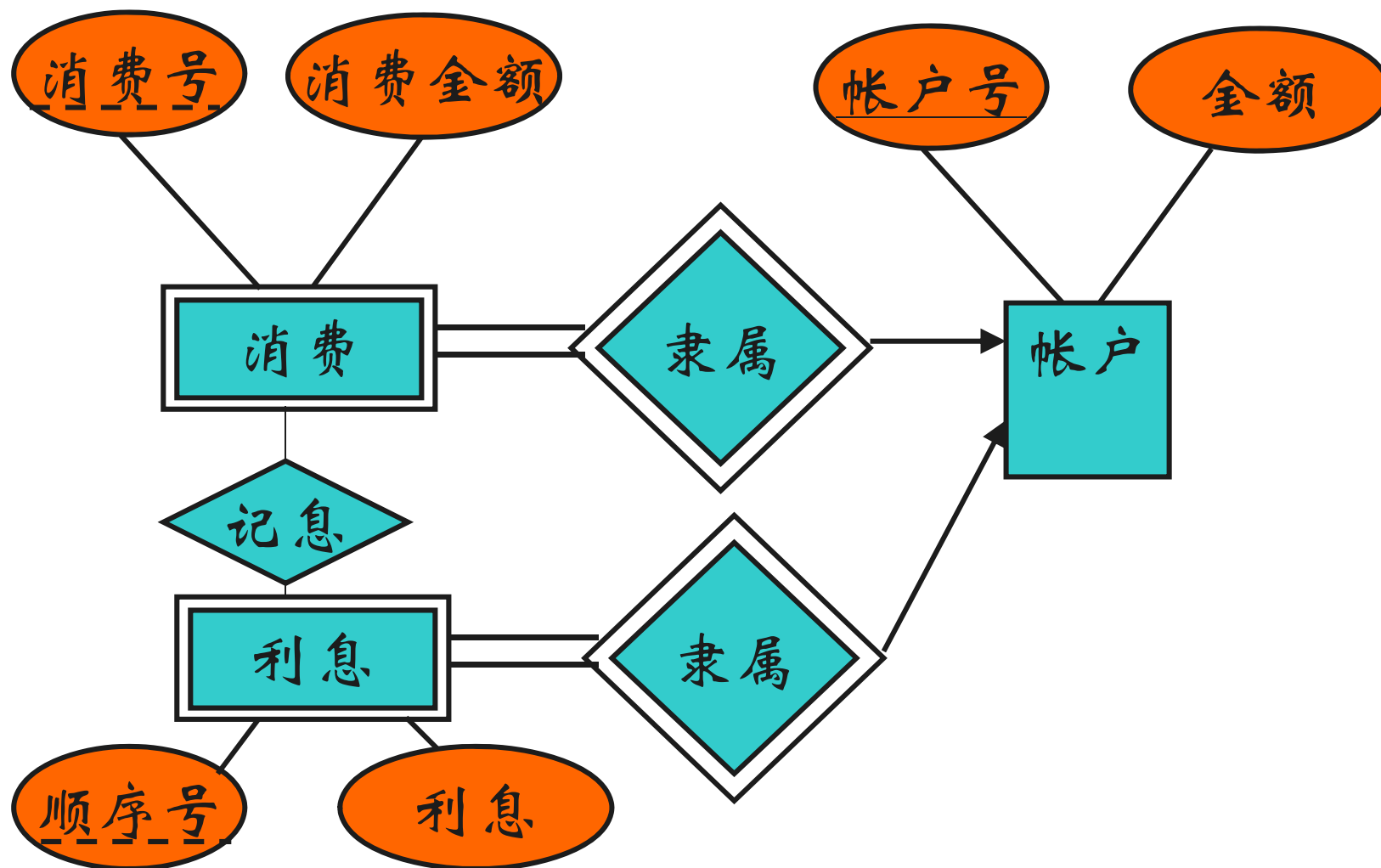
弱实体集

□ 弱实体集的引入

- ◆ 作为层次结构的一部分
- ◆ 实体集的一些多值、复合属性可以抽取出来作为弱实体集
- ◆ 如果弱实体集不但参与和强实体集之间的标识性联系，而且参与和其它实体集的联系，或者弱实体集本身含有很多属性，则将其表述为弱实体集
- ◆ 如果弱实体集只参与和强实体集之间的标识性联系，或者弱实体集本身属性不多，则将其表述为属性



弱实体集参与其它联系



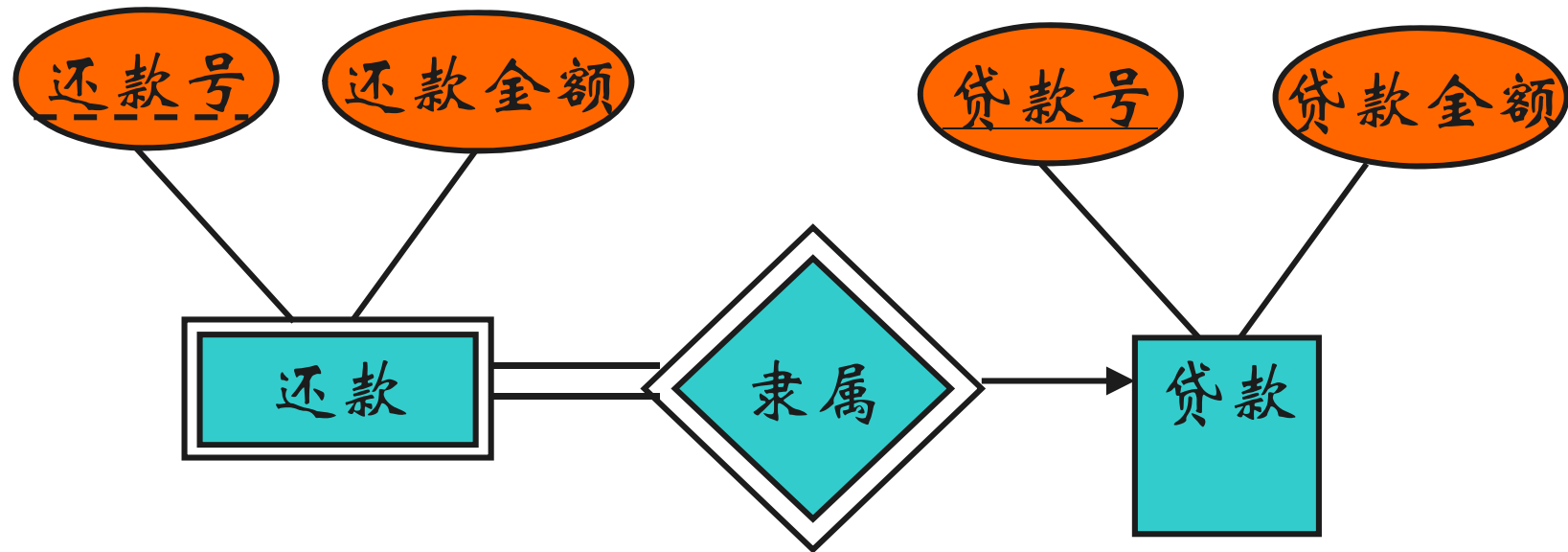


弱实体集

- 弱实体集在 E-R 图中的表示
 - ◆ 弱实体集以**双边框**的矩形表示
 - ◆ 标识性联系以**双边框**的菱形表示
 - ◆ 从联系集用**双线**（全部参与）连接弱实体集，用**箭头**（一对多联系）指向强实体集
 - ◆ 弱实体集的分辩符用**下划虚线**标明

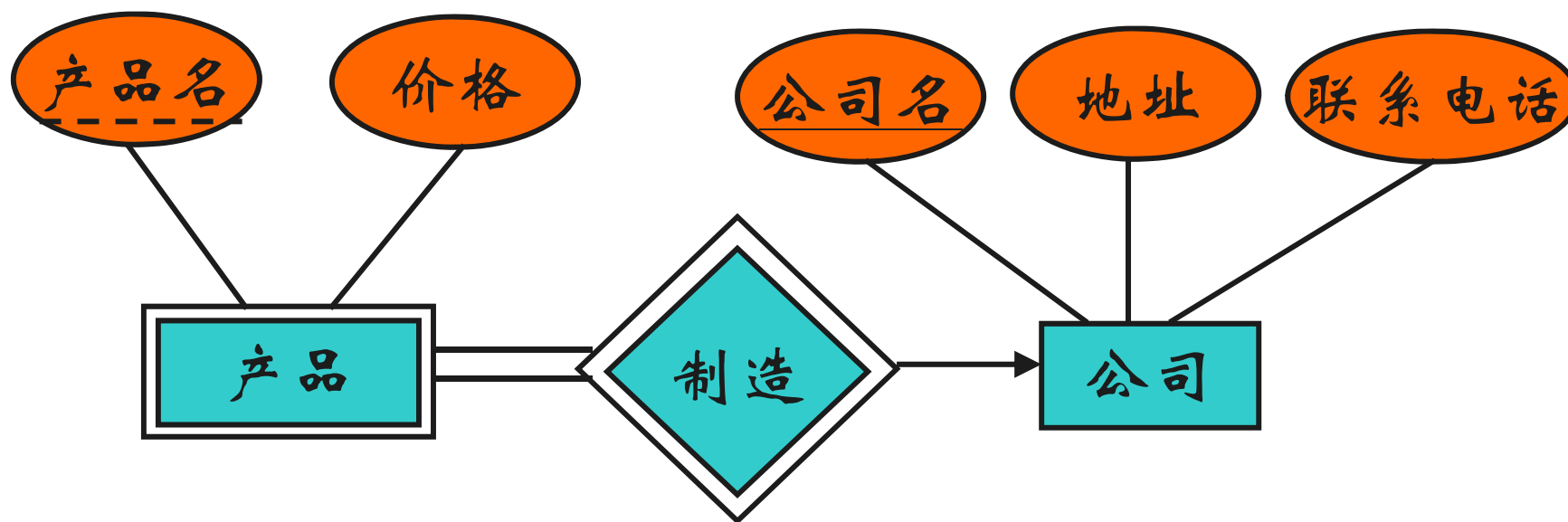


弱实体集





弱实体集





第一章 绪论

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统结构

1.4 数据库系统的组成

1.5 数据库技术的研究领域

1.6 小结



1.2.3 常用数据模型

□ 非关系模型

- ◆ 层次模型 (**Hierarchical Model**)

- ◆ 网状模型(**Network Model**)

- ◆ 数据结构：以基本层次联系为基本单位

基本层次联系：两个记录以及它们之间的一对多（包括一对一）的联系



常用数据模型(续)

- 关系模型(Relational Model)

- ◆ 数据结构：表

- 面向对象模型(Object Oriented Model)

- ◆ 数据结构：对象



1.2 数据模型

1.2.1 数据模型的组成要素

1.2.2 概念模型

1.2.3 常用数据模型

1.2.4 层次模型

1.2.5 网状模型

1.2.6 关系模型



1.2.4 层次模型

1. 层次数据模型的数据结构
2. 层次数据模型的数据操纵
3. 层次数据模型的完整性约束
4. 层次数据模型的存储结构
5. 层次数据模型的优缺点
6. 典型的层次数据库系统



1. 层次数据模型的数据结构

□ 层次模型

满足下面两个条件的基本层次联系的集合为层次模型。

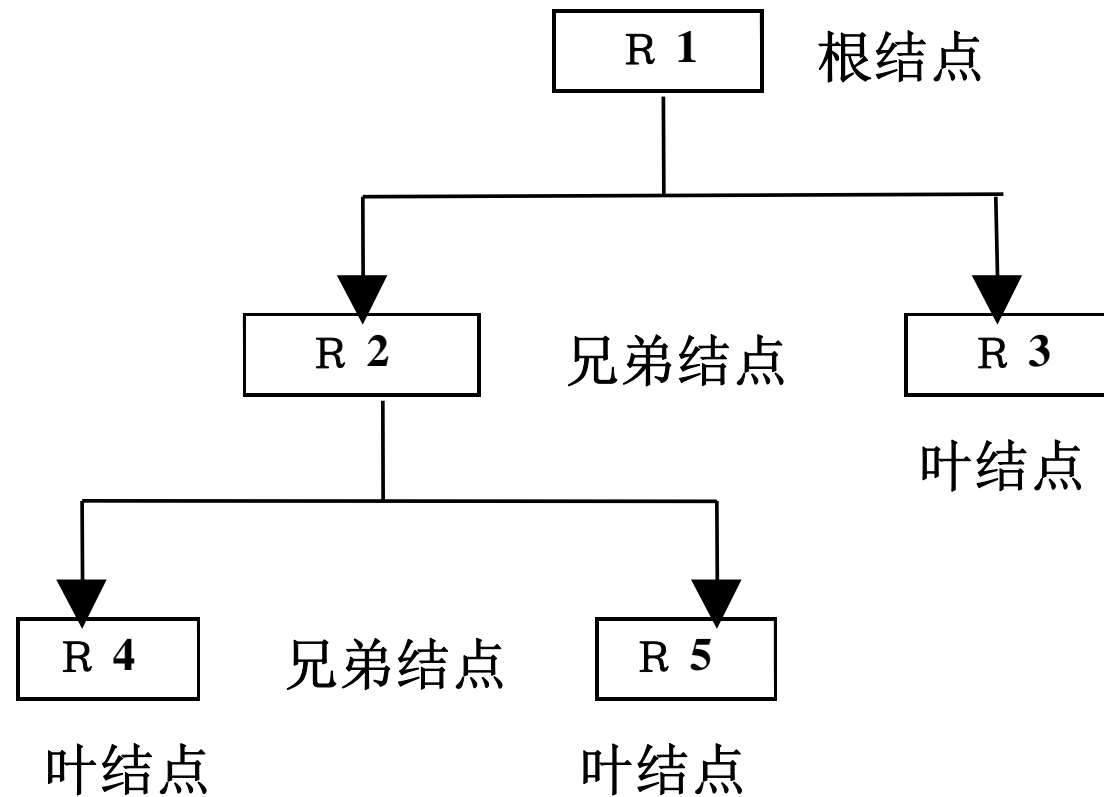
1. 有且只有一个结点没有双亲结点，这个结点称为根结点
2. 根以外的其它结点有且只有一个双亲结点

□ 层次模型中的几个术语

- ◆ 根结点，双亲结点，兄弟结点，叶结点



层次数据模型的数据结构(续)





层次数据模型的数据结构(续)

□ 表示方法

实体型：用记录类型描述。每个结点表示一个记录类型。

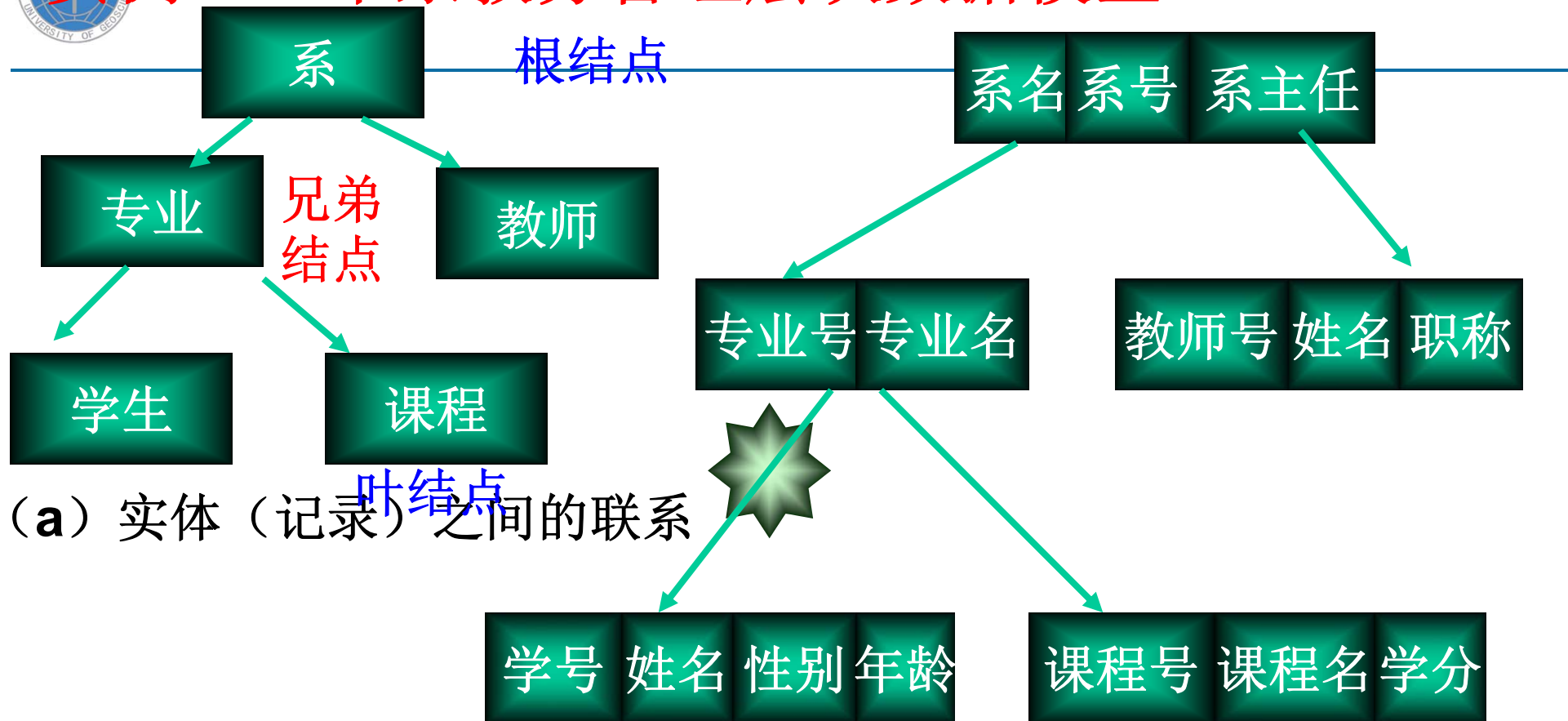
属性：用字段描述。每个记录类型可包含若干个字段。

联系：用结点之间的连线表示记录（类）型之间的 一对多的联系。

实例：教员-学生数据模型



实例：一个系教务管理层次数据模型。



(a) 实体（记录）之间的联系

(b) 实体型（记录型）之间的联系

图 教务管理层次模型

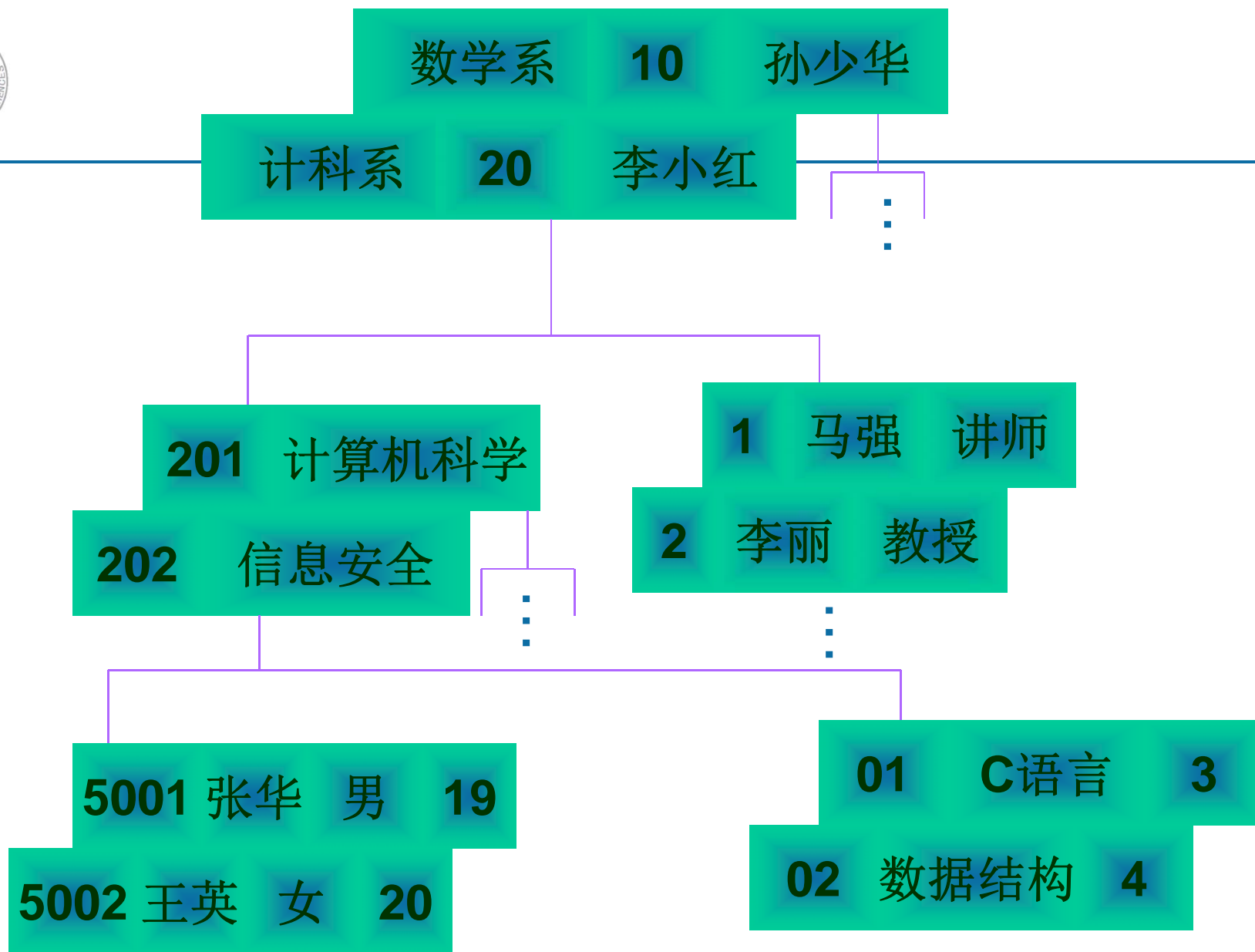


图 某系教务管理的实例（实体集之间的联系）



层次数据模型的数据结构(续)

□ 特点

- ◆ 结点的双亲是唯一的
- ◆ 只能直接处理一对多的实体联系
- ◆ 每个记录类型定义一个排序字段，也称为码字段
- ◆ 任何记录值只有按其路径查看时，才能显出它的全部意义
- ◆ 没有一个子女记录值能够脱离双亲记录值而独立存在



层次数据模型的数据结构(续)

- 多对多联系在层次模型中的表示
 - ◆ 用层次模型**间接**表示多对多联系
 - ◆ 方法
 - 将多对多联系分解成一对多联系**
 - ◆ 分解方法
 - **冗余结点法**
 - **虚拟结点法**

实例:

图1-10 冗余结点的方法

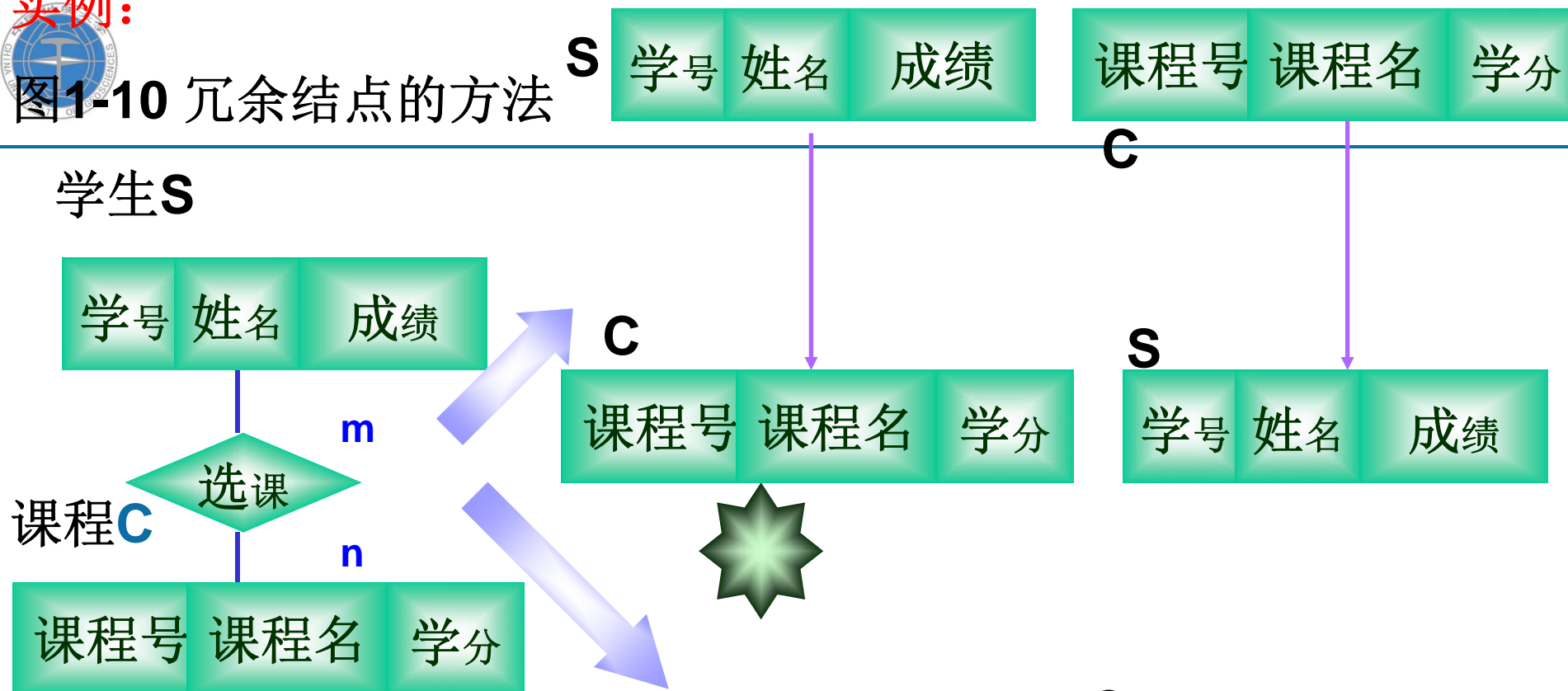
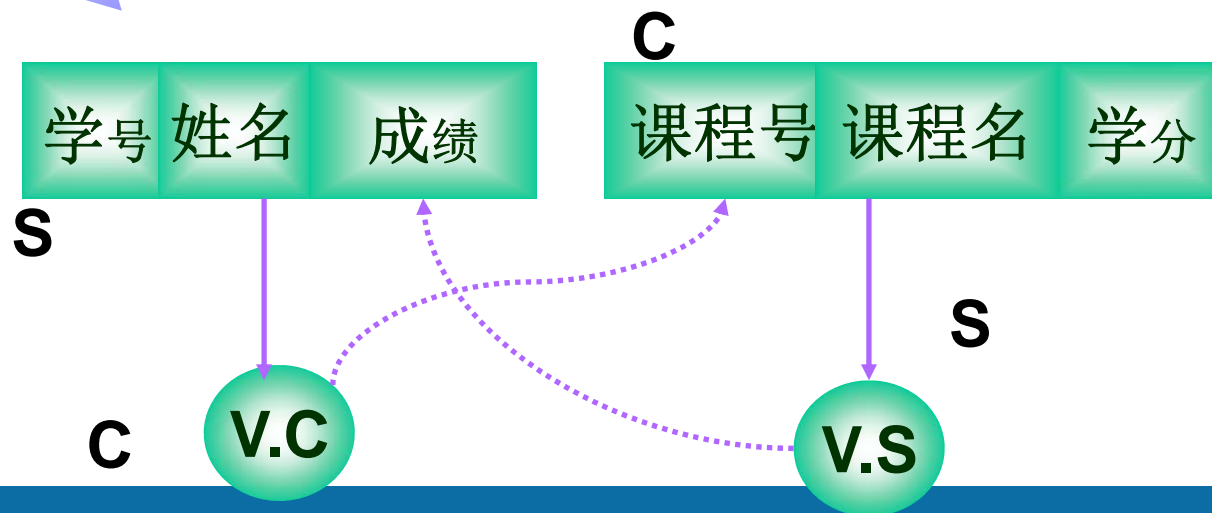


图1-11 虚拟结点的方法





2. 层次模型的数据操纵

- 查询
- 插入
- 删除
- 更新



3. 层次模型的完整性约束

- 无相应的双亲结点值就不能**插入**子女结点值
- 如果**删除**双亲结点值，则相应的子女结点值也被同时删除
- **更新**操作时，应更新所有相应记录，以保证数据的一致性



4.层次数据模型的存储结构

□ 邻接法

按照层次树前序遍历的顺序把所有记录值依次邻接存放，即通过物理空间的位置相邻来实现层次顺序

□ 链接法

用指引元来反映数据之间的层次联系

- ◆ 子女—兄弟链接法

- ◆ 层次序列链接法



5. 层次模型的优缺点

□ 优点

- ◆ 层次数据模型简单，对具有一对多的层次关系的部门描述自然、直观，容易理解
- ◆ 性能优于关系模型，不低于网状模型
- ◆ 层次数据模型提供了良好的完整性支持

□ 缺点

- ◆ 多对多联系表示不自然
- ◆ 对插入和删除操作的限制多
- ◆ 查询子女结点必须通过双亲结点
- ◆ 层次命令趋于程序化



6. 典型的层次数据库系统

IMS数据库管理系统

- ◆ 第一个大型商用**DBMS**
- ◆ **1968**年推出
- ◆ **IBM**公司研制



1.2 数据模型

1.2.1 概念模型

1.2.2 数据模型的组成要素

1.2.3 最常用的数据模型

1.2.4 层次模型

1.2.5 网状模型

1.2.6 关系模型



1.2.5 网状模型

1. 网状数据模型的数据结构
2. 网状数据模型的数据操纵
3. 网状数据模型的完整性约束
4. 网状数据模型的存储结构
5. 网状数据模型的优缺点
6. 典型的网状数据库系统



1. 网状数据模型的数据结构

□ 网状模型

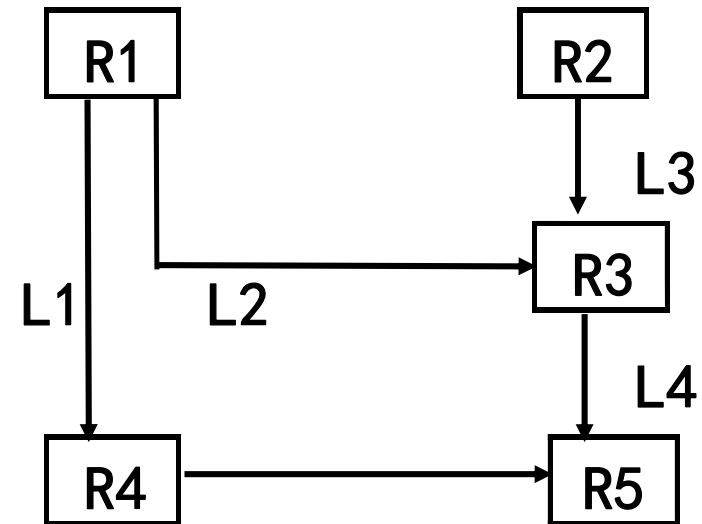
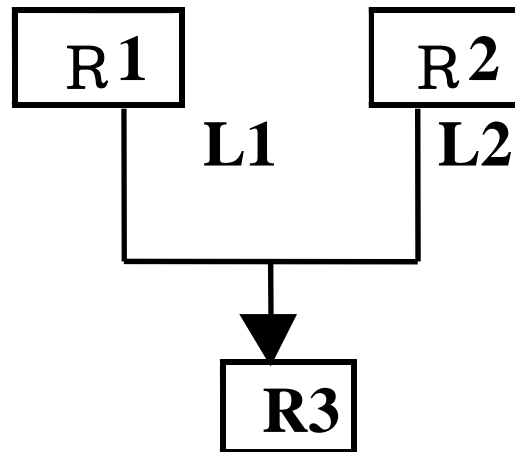
满足下面两个条件的基本层次联系的集合为网状模型

。

1. 允许一个以上的结点无双亲；
2. 一个结点可以有多个的双亲。



网状数据模型的数据结构





网状数据模型的数据结构(续)

□ 表示方法（与层次数据模型相同）

实体型：用记录类型描述。

每个结点表示一个记录类型。

属性：用字段描述。

每个记录类型可包含若干个字段。

联系：用结点之间的连线表示记录类型之间的一对多的父子联系。



网状数据模型的数据结构(续)




□ 特点

- ◆ 只能直接处理一对多的实体联系
- ◆ 每个记录类型定义一个排序字段，也称为码字段
- ◆ 任何记录值只有按其路径查看时，才能显出它的全部意义



网状数据模型的数据结构(续)

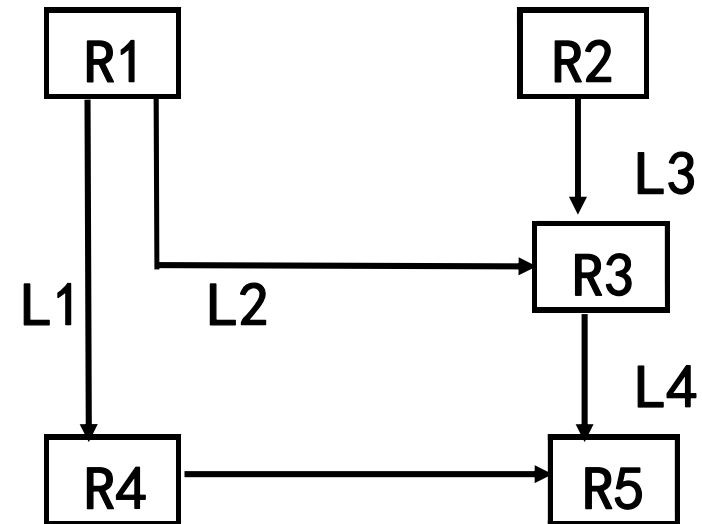
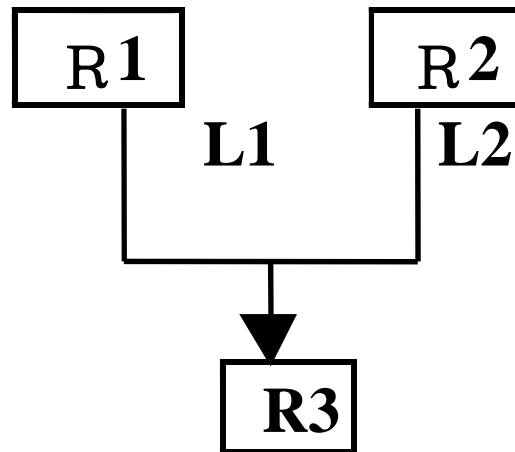
□ 网状模型与层次模型的区别

- ◆ 网状模型允许多个结点没有双亲结点 
- ◆ 网状模型允许结点有多个双亲结点 
- ◆ 网状模型允许两个结点之间有多种联系（复合联系） 
- ◆ 网状模型可以更直接地去描述现实世界
- ◆ 层次模型实际上是网状模型的一个特例



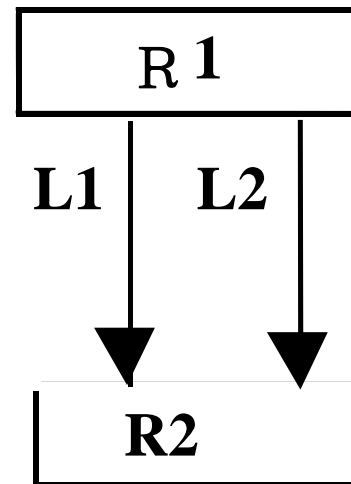


网状数据模型的数据结构(续)



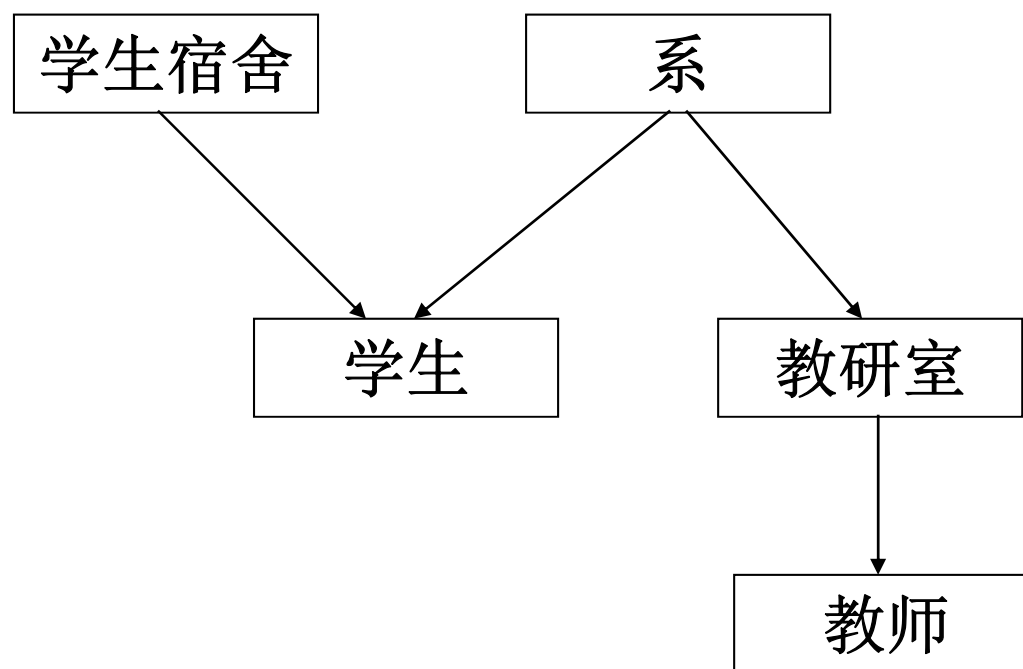


网状数据模型的数据结构(续)



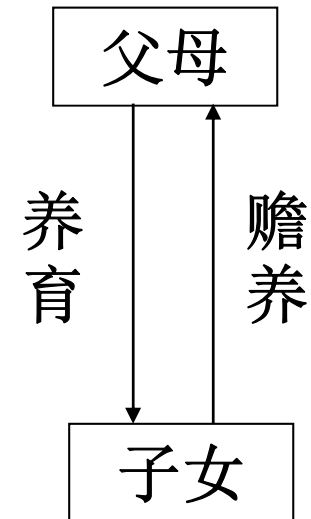
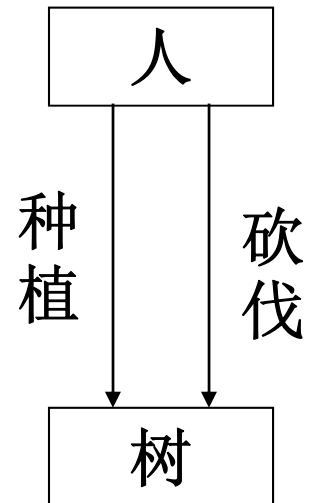


网状数据模型的数据结构(续)





网状数据模型的数据结构(续)





网状数据模型的数据结构(续)

多对多联系在网状模型中的表示

- ◆ 用网状模型**间接**表示多对多联系
- ◆ 方法

将多对多联系**直接**分解成一对多联系



2. 网状模型的数据操纵

- 查询
- 插入
- 删除
- 更新



3.网状数据模型的完整性约束

网状数据库系统（如**DBTG**）对数据操纵加了一些限制，提供了一定的完整性约束（**属籍类别**）

- ◆ 码
- ◆ 双亲结点与子女结点之间是一对多联系
- ◆ 不允许插入尚未确定双亲结点值的子女结点值
- ◆ 不允许只删除双亲结点值



4. 网状数据模型的存储结构

□ 关键

- ◆ 实现记录之间的联系

□ 常用方法

- ◆ 单向链接
- ◆ 双向链接
- ◆ 环状链接
- ◆ 向首链接



5.网状模型的优缺点

□ 优点

- ◆ 能够更为直接地描述现实世界，如一个结点可以有多个双亲
- ◆ 具有良好的性能，存取效率较高

□ 缺点

- ◆ 结构比较复杂，而且随着应用环境的扩大，数据库的结构就变得越来越复杂，不利于最终用户掌握
- ◆ **DDL、DML**语言复杂，用户不容易使用



6. 典型的网状数据库系统

- DBTG系统，亦称CODASYL系统
 - ◆ 由**DBTG**提出的一个系统方案
 - ◆ 奠定了数据库系统的基本概念、方法和技术
 - ◆ **70年代**推出
- 实际系统
 - ◆ **Cullinet Software Inc.**公司的 **IDMS**
 - ◆ **Univac**公司的 **DMS1100**
 - ◆ **Honeywell**公司的**IDS/2**
 - ◆ **HP**公司的**IMAGE**



1.2 数据模型

1.2.1 概念模型

1.2.2 数据模型的组成要素

1.2.3 最常用的数据模型

1.2.4 层次模型

1.2.5 网状模型

1.2.6 关系模型



1.2.6 关系模型

1. 关系数据模型的数据结构
2. 关系数据模型的操纵
3. 关系数据模型的完整性约束
4. 关系数据模型的存储结构
5. 关系数据模型的优缺点
6. 典型的关系数据库系统



关系模型

- 最重要的一种数据模型。也是目前主要采用的数据模型
- 1970年由美国IBM公司San Jose研究室的研究员E.F.Codd提出
- 本课程的重点



关系数据模型的数据结构

- 在用户观点下，关系模型中数据的逻辑结构是一张二维表，它由行和列组成。

学生登记表

学 号	姓 名	年 令	性 别	系 名	年 级
95004	王小明	19	女	社会学	95
95006	黄大鹏	20	男	商品学	95
95008	张文斌	18	女	法律学	95
...



关系模型的基本概念

◆ 关系 (Relation)

一个关系对应通常说的一张表。

◆ 元组 (Tuple)

表中的一行即为一个元组。

◆ 属性 (Attribute)

表中的一列即为一个属性，给每一个属性起一个名称即属性名。



关系模型的基本概念

◆ 主码 (Key)

表中的某个属性组，它可以唯一确定一个元组。

◆ 域 (Domain)

属性的取值范围。

◆ 分量

元组中的一个属性值。

◆ 关系模式

对关系的描述

关系名 (属性1, 属性2, ..., 属性n)

学生 (学号, 姓名, 年龄, 性别, 系, 年级)



关系数据模型的数据结构(续)

□ 实体及实体间的联系的表示方法

- ◆ **实体型**：直接用关系（表）表示。
- ◆ **属性**：用属性名表示。
- ◆ **一对一联系**：隐含在实体对应的关系中。
- ◆ **一对多联系**：隐含在实体对应的关系中。
- ◆ **多对多联系**：直接用关系表示。



关系数据模型的数据结构(续)

例1

学生、系、系与学生之间的一对多联系：

学生（学号，姓名，年龄，性别，系号，年级）

系（系号，系名，办公地点）

例2

系、系主任、系与系主任间的一对一联系



关系数据模型的数据结构(续)

例3

学生、课程、学生与课程之间的多对多联系：

学生（学号，姓名，年龄，性别，系号，年级）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）



最基本的规范条件：关系的每一个分量必须是一个不可分的数据项。

[illegible]



2.关系模型的数据操纵

- 查询、插入、删除、更新
- 数据操作是集合操作，操作对象和操作结果都是关系，即若干元组的集合
- 存取路径对用户隐蔽，用户只要指出“干什么”，不必详细说明“怎么干”



3.关系模型的完整性约束

- 实体完整性
- 参照完整性
- 用户定义的完整性



4. 关系数据模型的存储结构

- 表以文件形式存储
- 有的DBMS一个表对应一个操作系统文件
- 有的DBMS自己设计文件结构



5.关系模型的优缺点

□ 优点

- ◆ 建立在严格的数学概念的基础上
- ◆ 概念单一。数据结构简单、清晰，用户易懂易用
 - 实体和各类联系都用关系来表示。
 - 对数据的检索结果也是关系。
- ◆ 关系模型的存取路径对用户透明
 - 具有更高的数据独立性，更好的安全保密性
 - 简化了程序员的工作和数据库开发建立的工作



关系模型的优缺点（续）

□ 缺点

存取路径对用户透明导致查询效率往往不如非关系数据模型

为提高性能，必须对用户的[查询请求进行优化](#)
增加了开发数据库管理系统的难度



6. 典型的关系数据库系统

- ◆ ORACLE
- ◆ SYBASE
- ◆ INFORMIX
- ◆ DB/2
- ◆ COBASE
- ◆ PBASE
- ◆ EasyBase
- ◆ DM/2
- ◆ OpenBase



第一章 绪论

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统结构

1.4 数据库系统的组成

1.5 数据库技术的研究领域

1.6 小结



1.3 数据库系统结构

1.3.1 数据库系统内部的模式结构

从数据库管理系统角度看

1.3.2 数据库系统外部的体系结构

从数据库最终用户角度看



1.3.1 数据库系统的模式结构

- 数据库系统模式的概念
- 数据库系统的三级模式结构
- 数据库的二级映象功能与数据独立性
- 小结



数据库系统模式的概念

□ “型” 和 “值” 的概念

◆ 型 (Type)

对某一类数据的结构和属性的说明

◆ 值 (Value)

是型的一个具体赋值

例如：学生记录

记录型：

(学号，姓名，性别，系别，年龄，籍贯)

该记录型的一个记录值：

(**900201**，李明，男，计算机，**22**，江苏)



模式 (Schema)

数据库中全体数据的**逻辑结构和特征的描述**。（不涉及到具体的值）

例：学生选课数据库模式：

学生（学号，姓名，性别，专业班级，出生年月）

课程（课程号，课程名，学分）

选课（学号，课程号，成绩）

注意：模式中不包含具体的记录数据

模式的实例：模式的**具体值**称为**模式的实例**。



模式与实例的关系

- 1) 模式相对稳定。
- 2) 同一个模式，可以有很多实例，实例是相对变动的。
- 3) 模式反映的是数据结构及其联系，而实例反映的是数据库某一时刻的状态值。

数据库中的模式：用数据描述语言精确定义数据模型的全部语句。



数据库系统的三级模式两级映射结构

- 1) 外模式 (**External Schema**) (子模式 **Subschema**、用户模式)：用子模式数据描述语言 (子模式**DDL**) 来定义外模型的全部语句。
- 2) 概念模式 (简称模式, **Schema**) 又称逻辑模式：用模式数据描述语言 (模式**DDL**) 来定义概念模型的全部语句。
- 3) 内模式 (**Internal Schema**) 也称物理模式、存储模式 (**Storage Schema**) :用设备介质语言 (内模式**DDL**) 来定义内模型的模式。(是对数据物理结构和存储方式的描述)



教务处：
教务管理系统

学生处：
学生信息管理系统

现代化教学中心：
一卡通管理系统

...

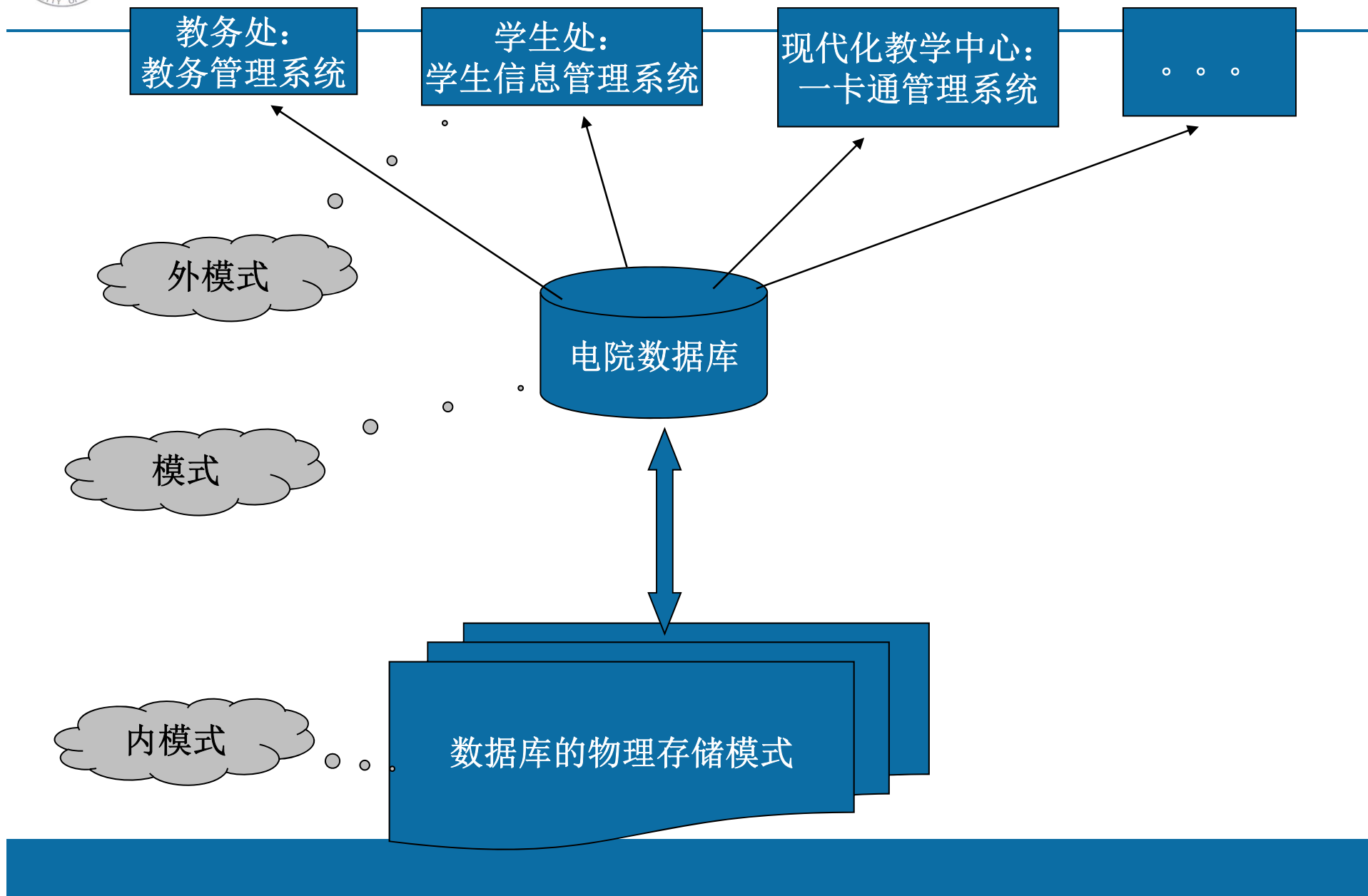
外模式

模式

内模式

电院数据库

数据库的物理存储模式





数据库管理系统三级模式之间的**两级映射**

- 1) **子模式** ↔ **模式映射**：保证了**逻辑数据独立性**：用户数据独立于全局逻辑数据的特性（模式变，子模式可以不变）。
- 2) **模式** ↔ **物理模式映射**：保证了**物理数据独立性**：全局逻辑数据独立于物理数据的特性。



例:

学生（学号，姓名，性别，专业班级，出生年月，籍贯，家庭地址，联系方式）
奖惩信息（学号，奖惩记录）

学生（学号，姓名）
一卡通（。。。。。）

教务处：
教务管理系统

学生处：
学生信息管理系统

现代化教学中心：
一卡通管理系统

。。。

学生（学号，姓名，性别，专业班级）
课程（课程号，课程名，学分，开课单位）
选课（学号，课程号，成绩）

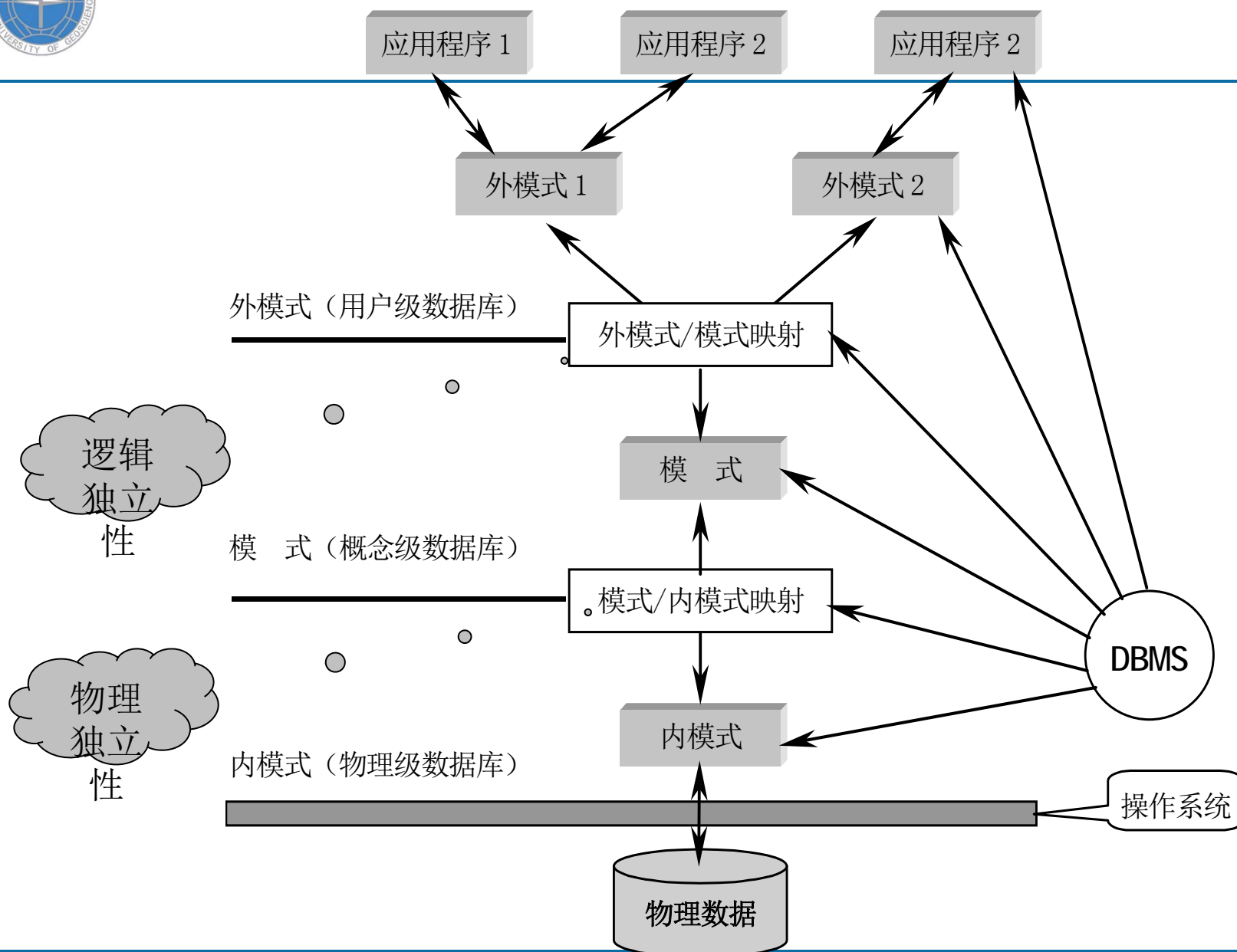
电院数据库

学生（学号，姓名，性别，专业班级，出生年月，籍贯，家庭地址，联系方式）
课程（课程号，课程名，学分，开课单位）
奖惩信息（学号，奖惩记录）
选课（学号，课程号，成绩）
一卡通（。。。。。）

思考：1、学生表中增加一个字段，对3个系统有影响吗？
2、将字段名“专业班级”改名为“班级”，对3个系统有影响吗？

理存储模式

文件名，
索引等。



数据库系统的三级模式结构图



三类模式之间的关系与区别

- 1) 子模式是概念模式的子集。
- 2) 一个数据库只有一个概念模式，只有一个内模式，可以有多个外模式。
- 3) 外模式是保证数据库安全性的一个有力措施。
- 4) 一个外模式可以对应多个应用，但一个应用只能对应一个外模式。



第一层

序号 姓名 性别 年龄 职称 工资

人事登记表

工资卡片

科研登记表

序号 姓名 职称 年龄 单位 项目 经费

序号 姓名 单位 工资 工龄 房租 水电

外模式
(子模式)

面向用户

第二层

基本文件

序号 姓名 性别 年龄 职称 工资 单位 项目 经费 工龄 房租 水电

职工信息表



序号 姓名 性别 年龄 职称 单位

项目表

序号 项目 经费

工资表

序号 工龄 工资 房租 水电

(概念) 模式

(面向系统整体)

第三层

物理组织最佳的形式

主数据文件
扩展名为.mdf

内模式

(物理模式)

(面向计算机存储)



1.3 数据库系统结构

□ 1.3.1 数据库系统内部的模式结构

◆ 从数据库管理系统角度看

□ 1.3.2 数据库系统外部的体系结构

◆ 从数据库最终用户角度看



1.3.2 数据库系统外部的体系结构

- 单用户结构
- 主从式结构
- 分布式结构
- 客户/服务器结构
- 浏览器/应用服务器/数据库服务器结构



1. 单用户数据库系统

- 整个数据库系统(应用程序、DBMS、数据)装在一台计算机上, 为一个用户独占, 不同机器之间不能共享数据。
- 早期的最简单的数据库系统

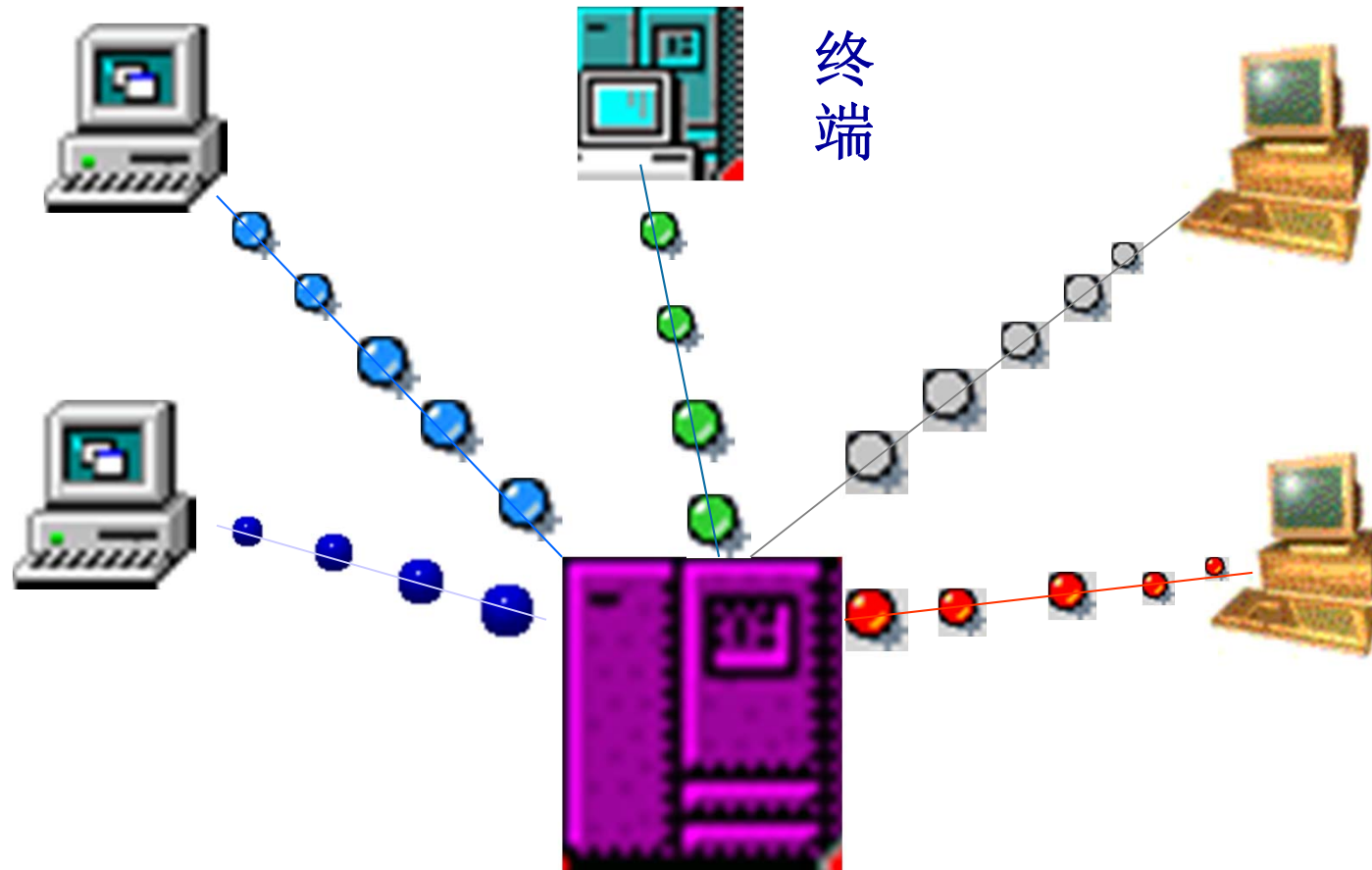


2. 主从式结构的数据库系统

- 一个主机带多个终端的多用户结构
 - ◆ 数据库系统，包括应用程序、DBMS、数据，都集中存放在主机上，所有处理任务都由主机来完成
 - ◆ 各个用户通过主机的终端并发地存取数据库，共享数据资源



主从式结构的数据库系统



主机



主从式结构的数据库系统(续)

□ 优点

- ◆ 易于管理、控制与维护。

□ 缺点

- ◆ 当终端用户数目增加到一定程度后，主机的任务会过分繁重，成为瓶颈，从而使系统性能下降。
- ◆ 系统的可靠性依赖主机，当主机出现故障时，整个系统都不能使用。



3. 分布式结构的数据库系统

- 数据库中的数据在逻辑上是一个整体，但物理地分布在计算机网络的不同结点上。
 - ◆ 网络中的每个结点都可以独立处理本地数据库中的数据，执行局部应用
 - ◆ 同时也可以同时存取和处理多个异地数据库中的数据，执行全局应用



分布式结构的数据库系统（续）

□ 优点

- ◆ 适应了地理上分散的公司、团体和组织对于数据库应用的需求。

□ 缺点

- ◆ 数据的分布存放给数据的处理、管理与维护带来困难。
- ◆ 当用户需要经常访问远程数据时，系统效率会明显地受到网络传输的制约。



4. 客户 / 服务器结构的数据库系统

□ 把DBMS功能和应用分开

- ◆ 网络中某个（些）结点上的计算机专门用于执行DBMS功能，称为数据库服务器，简称服务器
- ◆ 其他结点上的计算机安装DBMS的外围应用开发工具，用户的应用系统，称为客户机



客户 / 服务器数据库系统的种类

□ 集中的服务器结构

一台数据库服务器，多台客户机

□ 分布的服务器结构

- ◆ 在网络中有多台数据库服务器
- ◆ 分布的服务器结构是客户 / 服务器与
分布式数据库的结合



客户 / 服务器结构的优点

- ◆ 客户端的用户请求被传送到数据库服务器，数据库服务器进行处理后，只将结果返回给用户，从而显著减少了数据传输量
- ◆ 数据库更加开放
 - 客户与服务器一般都能在多种不同的硬件和软件平台上运行
 - 可以使用不同厂商的数据库应用开发工具



客户 / 服务器结构的缺点

“胖客户”问题：

- ◆ 系统安装复杂，工作量大。
- ◆ 应用维护困难，难于保密，造成安全性差。
- ◆ 相同的应用程序要重复安装在每一台客户机上，从系统总体来看，大大浪费了系统资源。

系统规模达到数百数千台客户机，它们的硬件配置、操作系统又常常不同，要为每一个客户机安装应用程序和相应的工具模块，其安装维护代价便不可接受了。



浏览器/应用服务器/数据库服务器结构

□ 客户端：

浏览器软件、用户界面

浏览器的界面统一，广大用户容易掌握

大大减少了培训时间与费用。

□ 服务器端分为两部分：

- ◆ Web服务器、应用服务器

- ◆ 数据库服务器等

大大减少了系统开发和维护代价

能够支持数万甚至更多的用户



第一章 绪论

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统结构

1.4 数据库系统的组成

1.5 数据库技术的研究领域

1.6 小结



1.4 数据库系统的组成

- 数据库
- 数据库管理系统（及其开发工具）
- 应用系统
- 数据库管理员
- （用户）



一、硬件平台及数据库

□ 数据库系统对硬件资源的要求

(1) 足够大的内存

- ◆ 操作系统
- ◆ **DBMS**的核心模块
- ◆ 数据缓冲区
- ◆ 应用程序



数据库系统对硬件资源的要求

(2) 足够大的外存

◆ 磁盘

- 操作系统
- **DBMS**
- 应用程序
- 数据库及其备份

◆ 光盘、磁带、软盘

- 数据备份

(3) 较高的通道能力，提高数据传送率



二、软件

- **DBMS**
- **操作系统**
- **与数据库接口的高级语言及其编译系统**
- **以DBMS为核心的应用开发工具**
- **为特定应用环境开发的数据库应用系统**



三、人员

- 数据库管理员
- 系统分析员
- 数据库设计人员
- 应用程序员
- (最终用户)



1. 数据库管理员(DBA)

- 决定数据库中的信息内容和结构
- 决定数据库的存储结构和存取策略
- 定义数据的安全性要求和完整性约束条件



数据库管理员(续)

□ 监控数据库的使用和运行

◆ 周期性转储数据库

- 数据文件
- 日志文件

◆ 系统故障恢复

◆ 介质故障恢复

◆ 监视审计文件



数据库管理员(续)

□ 数据库的改进和重组

- ◆ 性能监控和调优
- ◆ 数据重组

□ 数据库重构



2. 系统分析员

- 负责应用系统的需求分析和规范说明
- 与用户及DBA协商，确定系统的软硬件配置
- 参与数据库系统的概要设计



3. 数据库设计人员

- 参加用户需求调查和系统分析
- 确定数据库中的数据
- 设计数据库各级模式



4. 应用程序员

- 设计和编写应用系统的程序模块
- 进行调试和安装



5. 用户

□ 偶然用户

- ◆ 企业或组织机构的高中级管理人员

□ 简单用户

- ◆ 银行的职员、机票预定人员、旅馆总台服务员



用户（续）

□ 复杂用户

- ◆ 工程师、科学家、经济学家、科技工作者等
- ◆ 直接使用数据库语言访问数据库，甚至能够基于数据库管理系统的**API**编制自己的应用程序



第一章 绪论

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统结构

1.4 数据库系统的组成

1.5 数据库技术的研究领域

1.6 小结



1.5 数据库技术的研究领域

- 数据库管理系统软件的研制
- 数据库设计
- 数据库理论



数据库管理系统软件的研制

- DBMS核心
- 一组相互联系的软件系统
 - ◆ 工具软件
 - ◆ 中间件



数据库设计

- 数据库设计方法
- 设计工具
- 设计理论
- 数据模型和数据建模



数据库理论

- 关系的规范化理论
- 关系数据理论



第一章 绪论

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统结构

1.4 数据库系统的组成

1.5 数据库技术的研究领域

1.6 小结



小结

□ 数据库系统概述

- ◆ 数据库的基本概念
- ◆ 数据管理的发展过程
- ◆ 数据库系统的研究领域

□ 数据模型

- ◆ 数据模型的三要素
- ◆ 概念模型， **E-R** 模型
- ◆ 三种主要数据模型



小结(续)

□ 数据库系统的结构

- ◆ 数据库系统三级模式结构
- ◆ 数据库系统的体系结构

□ 数据库系统的组成



拓展学习

- 1、了解分布式数据库。
- 2、什么情况下需要使用分布式数据库？
- 3、了解开源数据库Hbase。
- 4、了解Hadoop的架构。