



2026年春季学期

52

数据库系统概论

An Introduction to Database Systems

第一章 绪论

中国科学技术大学
人工智能与数据科学学院

黄振亚, huangzhy@ustc.edu.cn



第一章 绪论

53

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统的三级模式结构

1.4 数据库系统的组成

*1.5 数据库系统的体系结构

1.6 小结



数据库的地位

54

- 数据库技术产生于六十年代末，是**数据管理**的最新技术，是计算机科学的重要分支。
- 数据库技术是信息系统的核心和基础，它的出现极大地促进了计算机应用向各行各业的渗透。
- 数据库的建设规模、数据库信息量的大小和使用频度已成为衡量一个国家信息化程度的重要标志。



第一章 绪论

55

1.1 数据库系统概述

1.1.1 四个基本概念

1.1.2 数据管理技术的产生和发展



1.1.1 四个基本概念

56

- 数据(Data)
- 数据库(Database)
- 数据库管理系统(Database Management System, DBMS)
- 数据库系统(Database System, DBS)



一、数据

57

- **数据(Data)**是数据库中存储的基本对象
- 数据的定义
 - 描述事物的符号记录
- 数据的种类
 - **数字**，文本、图形、图像、音频、视频、学生的档案记录、个人的网购记录等
- 数据的特点
 - 数据的表现形式不能完全表达内容：**44**是什么？
 - 数据与其语义是不可分的



数据举例

58

- 数据的含义称为数据的语义，数据与其语义是不可分的。
 - 例如 93是一个数据
 - 语义1：学生某门课的成绩
 - 语义2：某人的体重
 - 语义3：计算机系2018级学生人数
 - 语义4：。。。。。



数据举例

59

- 描述事物：自然语言 vs 计算机
- 学生档案中的学生记录
(20180002, 刘晨, 女, 1999-9-1, 计算机科学与技术)
 - 语义：学生学号、姓名、性别、出生日期、主修专业
 - 解释：学号为20180002，姓名为刘晨的女生，1999年9月1日出生，计算机科学与技术专业
 - 数据是有结构的：“记录”是计算机中表示和存储数据的一种格式或一种方法



二、数据库

60

- 数据库的定义
 - 数据库(Database, DB)是长期储存在计算机内、有组织的、可共享的大量数据的集合
- 数据库的基本特征
 - 数据按一定的数据模型组织、描述和储存
 - 为各种用户共享
 - 冗余度较小
 - 数据独立性较高
 - 易扩展



三、数据库管理系统

- 数据库管理系统(DBMS)
 - 位于用户与操作系统之间的一层**数据管理软件**
 - 是基础软件，是一个大型复杂的软件系统
- DBMS的用途
 - 科学地组织和存储数据、高效地获取和维护数据





DBMS的主要功能

62

□ 1. 数据定义功能

- 提供数据定义语言(Data Definition Language, DDL)
- 定义数据库中的数据对象

□ 2. 数据组织、存储和管理

- 分类组织、存储和管理各种数据
- 确定组织数据的文件结构和存取方式
- 实现数据之间的联系
- 提供多种存取方法提高存取效率



DBMS的主要功能

63

□ 3. 数据操纵功能

- 提供数据操纵语言(Data Manipulation Language, DML)
- 实现对数据库的基本操作 (查询、插入、删除和修改)

□ 4. 数据库的事务管理和运行管理

- 数据库在建立、运行和维护时由DBMS统一管理和控制
- 保证数据的**安全性**、**完整性**、多用户对**数据的并发使用**
- 发生故障后的**系统恢复**



DBMS的主要功能

64

□ 5. 数据库的建立和维护功能(实用程序)

- 数据库初始数据装载转换，批量装载等
- 数据库转储，介质故障恢复
- 数据库的重组织
- 性能监视、分析等

□ 6. 其它功能

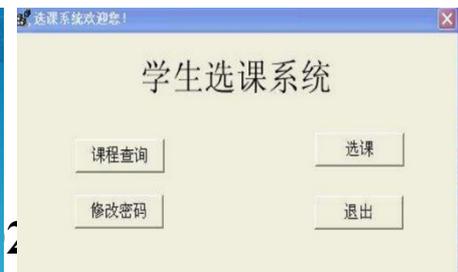
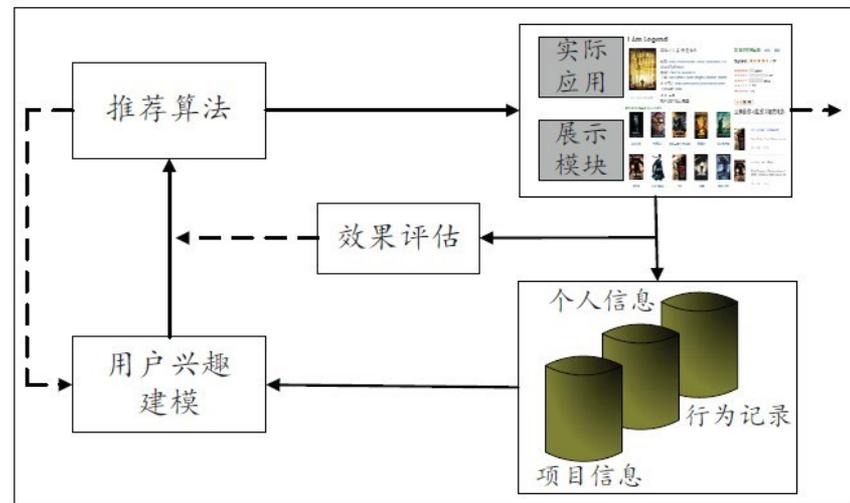
- DBMS与网络中其它软件系统的通信
- 两个DBMS系统的数据转换
- 异构数据库之间的互访和互操作



四、数据库系统

□ 什么是数据库系统（Database System，简称DBS）
在计算机系统中引入数据库后的系统构成

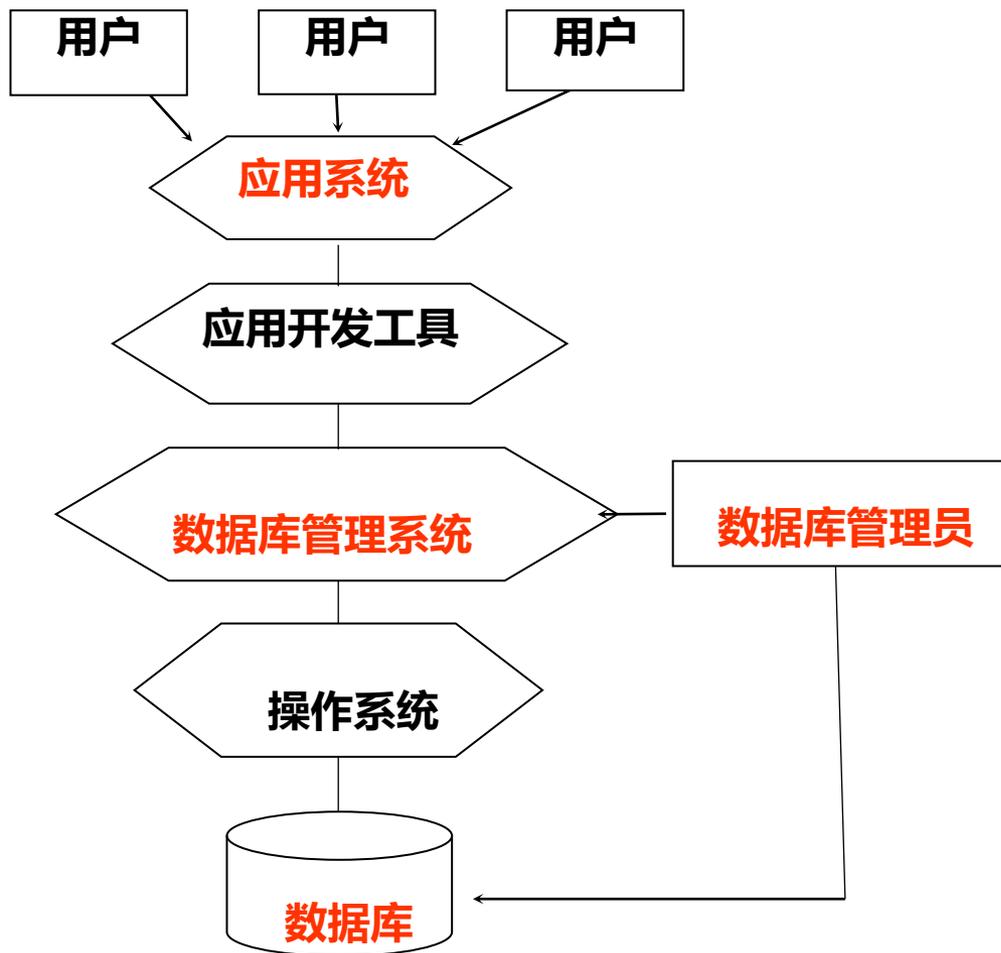
- 数据库系统的构成
 - 数据库
 - 数据库管理系统（及其开发工具）
 - 应用程序，应用系统
 - 数据库管理员





数据库系统

数据库系统
人-机系统





1.1 数据库系统概述

67

1.1.1 四个基本概念

1.1.2 数据管理技术的产生和发展



数据管理技术的产生和发展

68

- 什么是数据管理
 - 对数据进行分类、组织、编码、存储、检索和维护
 - 数据处理的中心问题
 - 对数据收集、存储、加工、传播、计算

- 数据管理技术的发展过程
 - 人工管理阶段(20世纪40年代中--50年代中)
 - 文件系统阶段(20世纪50年代末--60年代中)
 - 数据库系统阶段(20世纪60年代末--现在)



数据管理技术的产生和发展(续)

69

- 数据管理技术的发展动力
 - 应用需求的推动（很重要）
 - 计算机硬件的发展
 - 计算机软件的发展
- 目前，Bigdata的需求，推动了新一代数据库发展
 - NoSql的发展
 - 大模型



一、人工管理阶段

70

□ 时期

□ 20世纪40年代中--50年代中

□ 产生的背景

□ 应用需求

科学计算

□ 硬件水平

无直接存取存储设备 (磁带, 卡片, 纸带)

□ 软件水平

没有操作系统

□ 处理方式

批处理





人工管理阶段(续)

71

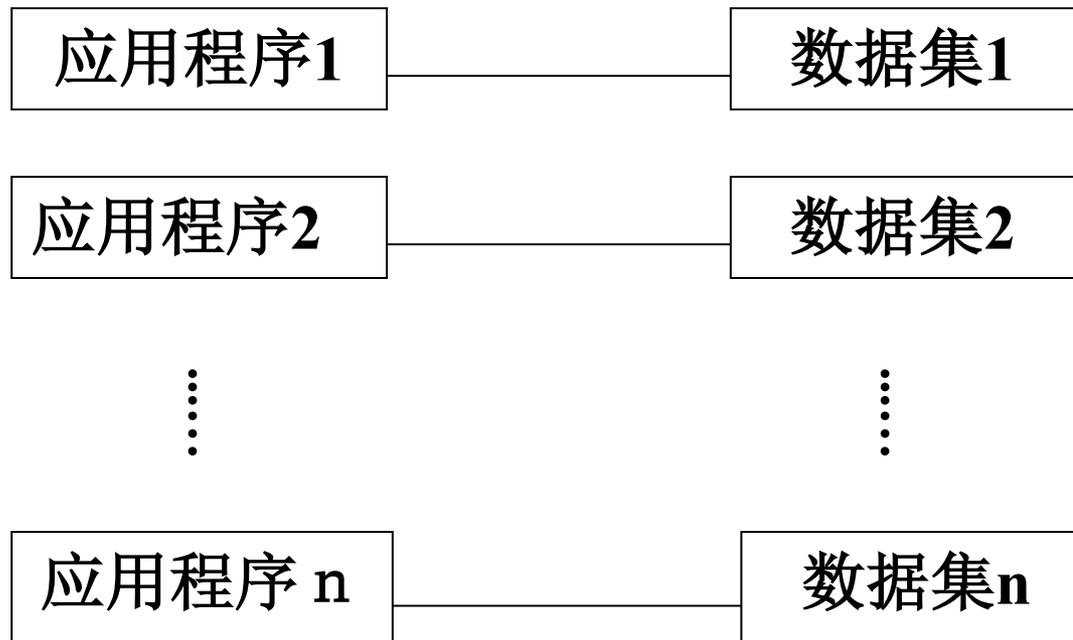
□ 特点

- 数据的管理者：用户（程序员），数据不保存
- 数据面向的对象：某一应用程序
- 数据的共享程度：无共享、冗余度极大
- 数据的独立性：不独立，完全依赖于程序
- **数据的结构化：无结构**
- 数据控制能力：应用程序自己控制



应用程序与数据的对应关系(人工管理阶段)

72



人工管理阶段应用程序与数据之间的一一对应关系



二、文件系统阶段

73

- 时期
 - 20世纪50年代末--60年代中
- 产生的背景
 - 应用需求 科学计算、管理
 - 硬件水平 磁盘、磁鼓
 - 软件水平 有文件系统（操作系统）
 - 处理方式 联机实时处理、批处理



文件系统阶段(续)

74

❖ 特点

数据的管理者：文件系统，数据可长期保存

数据面向的对象：某一应用程序

数据的共享程度：共享性差、冗余度大

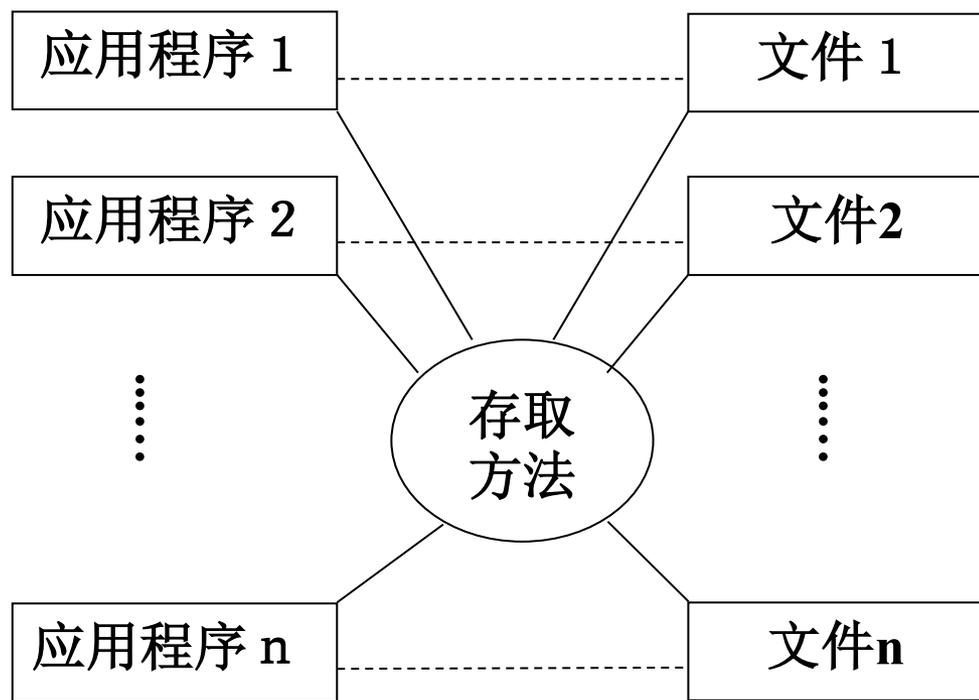
数据的结构化：记录内有结构,整体无结构

数据的独立性：独立性差，数据的逻辑结构改变必须
修改应用程序

数据控制能力：应用程序自己控制



应用程序与数据的对应关系(文件系统阶段)



文件系统阶段应用程序与数据之间的对应关系



文件系统中数据的结构

76

- 记录内有结构（定义字段，但是关系无法存储）。
- 数据的结构是靠程序定义和解释的。
- 数据只能是定长的。
 - 可以间接实现数据变长要求，但访问相应数据的应用程序复杂了。
- 文件间是独立的，因此数据整体无结构。
 - 可以间接实现数据整体的有结构，但必须在应用程序中对描述数据间的联系。
- 数据的最小存取单位是记录。



三、数据库系统阶段

77

- 时期
 - 20世纪60年代末以来
- 产生的背景
 - 应用背景 大规模管理
 - 硬件背景 大容量磁盘、磁盘阵列
 - 软件背景 有数据库管理系统
 - 处理方式 联机实时处理,分布处理,批处理
- 文件系统到数据库系统，标志着数据管理技术的飞跃



数据库系统的特点—例子

- 例子：学籍管理
 - 学生的信息包括学号、姓名、性别、年龄、专业和奖励
- 用**文件系统**实现学籍管理

“学生基本信息”文件的结构和内容

| 学号 | 姓名 | 性别 | 年龄 | 专业 | 位置 | 长度 |
|----------|-------|-------|-------|-------|-------|-------|
| 20180001 | 史玉明 | 女 | 20 | 计算机 | 0 | 30 |
| 20180100 | 李明虎 | 男 | 21 | 机械 | 30 | 15 |
| 20180234 | 张翔 | 男 | 21 | 化工 | 45 | 0 |
| | | | | | | |

奖励

2019校奖学金，2020国家奖学金
2020校优秀学生

“奖励”文件的结构和内容



数据库系统的特点—例子

79

- 用**文件系统**实现学籍管理
 - 数据存储
 - 定长记录 存储在“学生基本信息”文件中
 - 变长记录 存放在另一个“奖励”文件
 - “学生基本信息”表中的位置和长度描述“奖励”文件中记录的开始位置和长度
 - 查询数据
 - 编写应用程序，实现数据的录入和查找
 - fopen, fread, fwrite, fclose, etc
- **缺点：**程序员必须关注记录结构和不同文件中记录之间的联系，工作量大，编程复杂，开发速度慢



数据库系统的特点—例子

- 例子：学籍管理
 - 学生的信息包括学号、姓名、性别、年龄、专业和奖励
- 用数据库系统实现学籍管理

```
CREATE TABLE STUDENT(  
    Sno CHAR(8),  
    Sname CHAR(10),  
    Ssex CHAR(8),  
    Sage INT,  
    Smajor CHAR(20));
```

```
CREATE TABLE AWARD(  
    Sno CHAR(8),  
    Details VARCHAR(200)  
);
```

```
1 • INSERT  
2     INTO Student (Sno, Sname, Ssex, Sdept, Sage)  
3     VALUES ("200215129", "陈春", "男", "IS", 18);  
4 • SELECT * FROM student:
```

| Sno | Sname | Ssex | Sage | Sdept |
|-----------|-------|------|------|-------|
| 200215128 | 陈冬 | 男 | 18 | IS |
| 200215129 | 陈春 | 男 | 18 | IS |

```
SELECT *  
FROM Student A Join Award B  
    On A.Sno=B.Sno  
WHERE A.Sno='200215128';
```



数据库系统的特点—例子

81

□ 用数据库系统管理

● 存储数据

➤ 建立两张表:

STUDENT表-存放学生的基本信息

AWARD表-存放学生的奖励情况

➤ 使用插入命令 完成学生基本信息和奖励情况的数据录入功能

● 查询功能

可以用一条查询语句实现



数据库系统的特点

82

- 数据结构化
- 数据的共享性高，冗余度低，易扩充
- 数据独立性高
- 数据由DBMS统一管理和控制



数据结构化

83

- 整体数据的结构化是数据库的主要特征之一
- 整体结构化
 - 不再仅仅针对某一个应用，而是面向全组织或企业
 - 不仅数据内部结构化，整体是结构化的，数据之间具有联系
- 数据库中实现的是数据的真正结构化
 - 数据的结构用数据模型描述，无需程序定义和解释
 - 数据可以变长
 - 数据的最小存取单位是数据项



数据结构化

□ 整体数据的结构化是数据库的主要特征之一

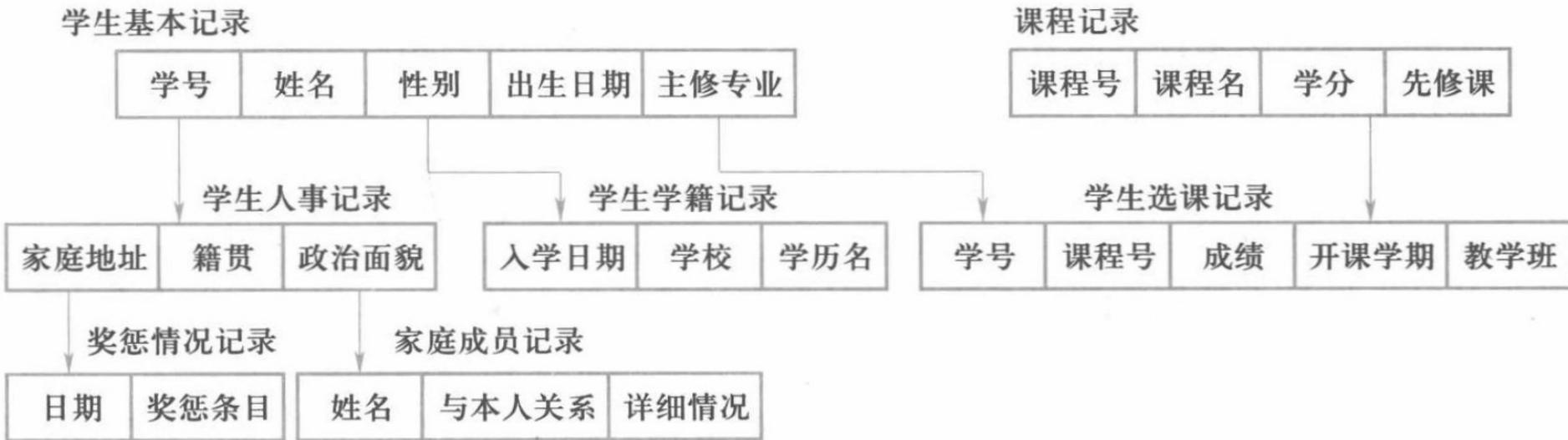


图 1.5 “高校本科教务管理”信息系统中与学生有关的数据结构



数据的共享性高，冗余度低，易扩充

85

- 数据库系统从整体角度看待和描述数据，数据面向整个系统，可以被**多个用户、多个应用**共享使用。
- 数据共享的好处
 - 减少数据冗余，节约存储空间
 - 避免数据之间的不相容性与不一致性
 - 使系统易于扩充



数据独立性高

86

□ 物理独立性（高度）

- 指用户的应用程序与存储在磁盘上的数据库中数据是相互独立的。当数据的物理存储改变了，应用程序不用改变
- 例如：数据按行存储，还是按列存储，应用程序不变

□ 逻辑独立性（一定程度）

- 指用户的应用程序与数据库的逻辑结构是相互独立的。数据的逻辑结构改变了，用户程序也可以不变
- 例如：增加了字段，有时可以不用改
- 数据独立性是由DBMS的二级映像功能来保证的



数据由DBMS统一管理和控制

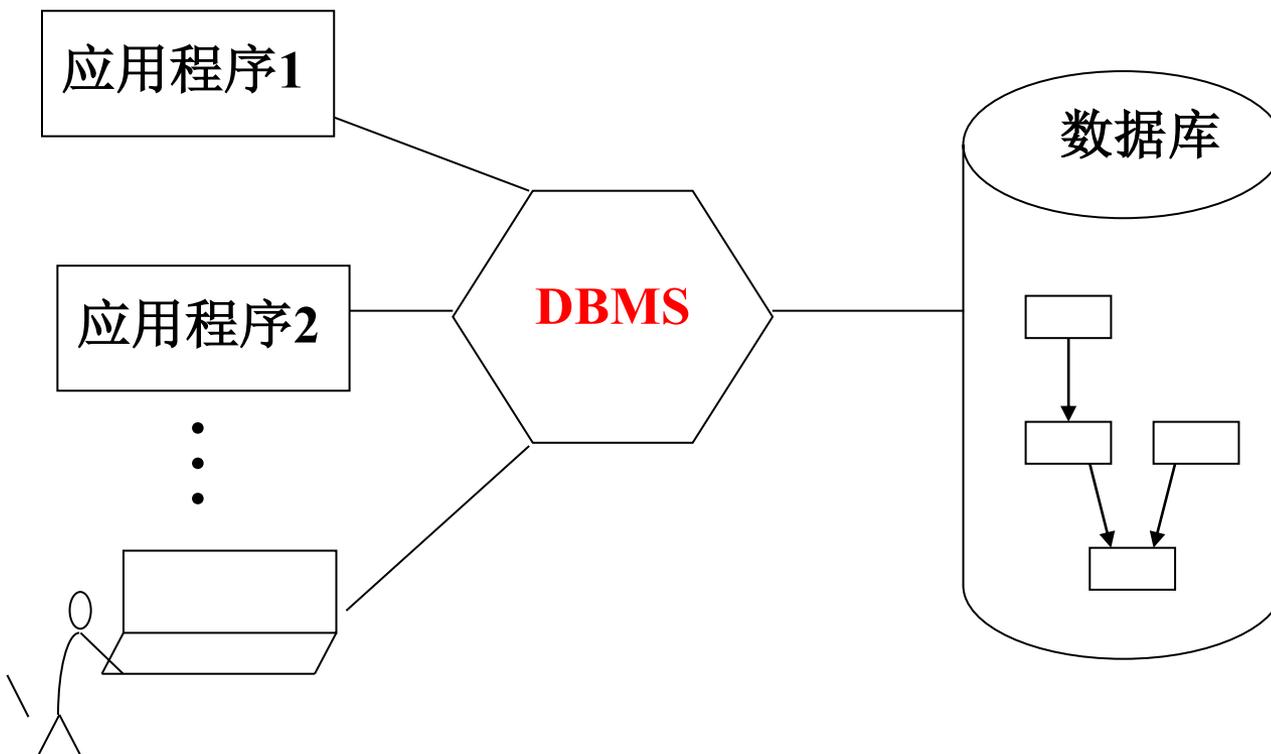
87

- 数据共享带来了数据的安全隐患
 - 用户的操作，相互干扰
- DBMS提供的数据控制功能
 - **数据的安全性 (Security) 保护 (第4章)**
保护数据，以防止不合法的使用造成的数据的泄密和破坏。
 - **数据的完整性 (Integrity) 检查 (第5章)**
将数据控制在有效的范围内，或保证数据之间满足一定的关系。
 - **并发 (Concurrency) 控制 (第12章)**
对多用户的并发操作加以控制和协调，防止相互干扰而得到错误的结果。
 - **数据库恢复 (Recovery) (第11章)**
将数据库从错误状态恢复到某一已知的正确状态。



应用程序与数据的对应关系(数据库系统)

88



数据库系统阶段应用程序与数据之间的对应关系



数据库系统概述

89

- 数据库是长期存储在计算机内有组织的大量的共享的数据集合
- 可以供各种用户共享，具有最小冗余度和较高的数据独立性。
- 数据库管理系统在数据库建立、运用和维护时对数据库进行统一控制，以保证数据的完整性、安全性，并在多用户同时使用数据库时进行并发控制，在发生故障后对数据库进行恢复。



第一章 绪论

90

1.1 数据库系统概述

1.2 数据模型（重点）

1.3 数据库系统的三级模式结构

1.4 数据库系统的组成

*1.5 数据库系统的体系结构

1.6 小结



数据模型

91

- 通俗地讲：**数据模型**就是**现实世界的模拟**
- 数据模型应满足三方面要求
 - 能比较**真实**地模拟现实世界
 - **容易**为人所理解
 - 便于在计算机上**实现**
- 在数据库中用数据模型这个工具来**描述、组织和操作**现实世界中的数据和信息。



1.2 数据模型

92

1.2.1 数据建模

1.2.2 概念模型

1.2.3 数据模型组成的三要素

1.2.4 层次模型

1.2.5 网状模型

1.2.6 关系模型

1.2.7* 数据库领域中不断涌现的数据模型



1.2.1 数据建模

93

- 数据模型分为两类（两个不同的层次）
 - 概念模型
 - 也称信息模型，它是按**用户的观点**来对数据和信息建模，用于数据库设计
 - 逻辑模型和物理模型
 - **计算机的视角**
 - **逻辑模型**主要包括网状模型、层次模型、**关系模型**、面向对象模型等，按计算机系统的观点对数据建模，用于**DBMS**实现。
 - **物理模型**是对数据最底层的抽象，描述数据在系统内部的表示方式和存取方法，在磁盘或磁带上的存储方式和存取方法。



数据建模 (续)

94

- 客观对象的抽象过程---**两步抽象**
 - 现实世界中的**客观对象**抽象为**概念模型**；
 - 将现实世界抽象为信息世界
 - 把**概念模型**转换为**某一DBMS支持的数据模型**
 - 将信息世界转化为机器世界



数据建模 (续)

现实世界中客观对象的抽象过程

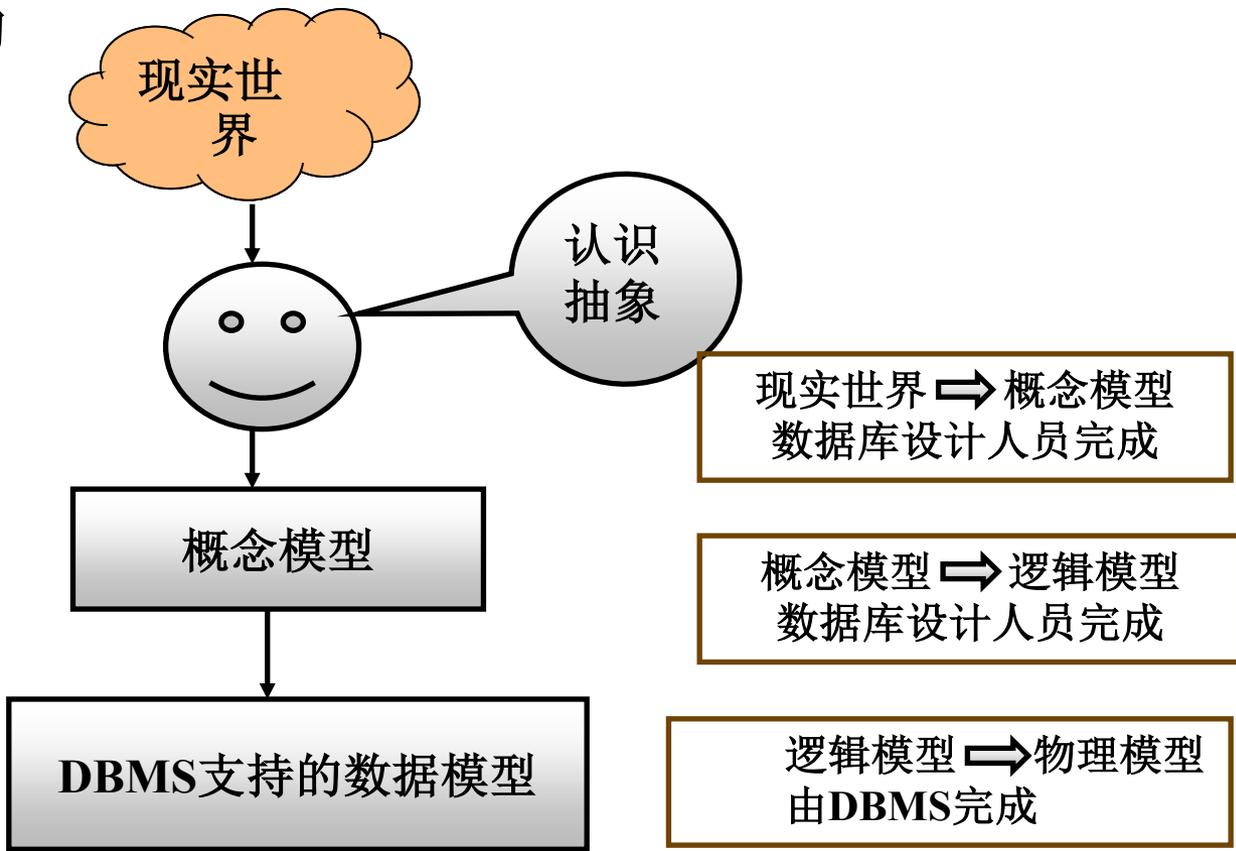
第一步

信息世界

第二步

机器世界

第三步 (自动)



看见一个动物 (现实世界), 命名为狗 (概念模型), 怎么在计算机里存 (逻辑模型), 存在哪里 (物理模型)



1.2 数据模型

96

1.2.1 数据建模

1.2.2 概念模型

1.2.3 数据模型组成的三要素

1.2.4 层次模型

1.2.5 网状模型

1.2.6 关系模型

1.2.7* 数据库领域中不断涌现的数据模型



概念模型

97

- 概念模型的用途
 - 概念模型用于信息世界的建模
 - 是现实世界到机器世界的一个中间层次
 - 是数据库设计的有力工具
 - 数据库设计人员和用户之间进行交流的语言
- 对概念模型的基本要求
 - 较强的语义表达能力
 - 能够方便、直接地表达应用中的各种语义知识
 - 简单、清晰、易于用户理解



1.2.2 什么是概念模型

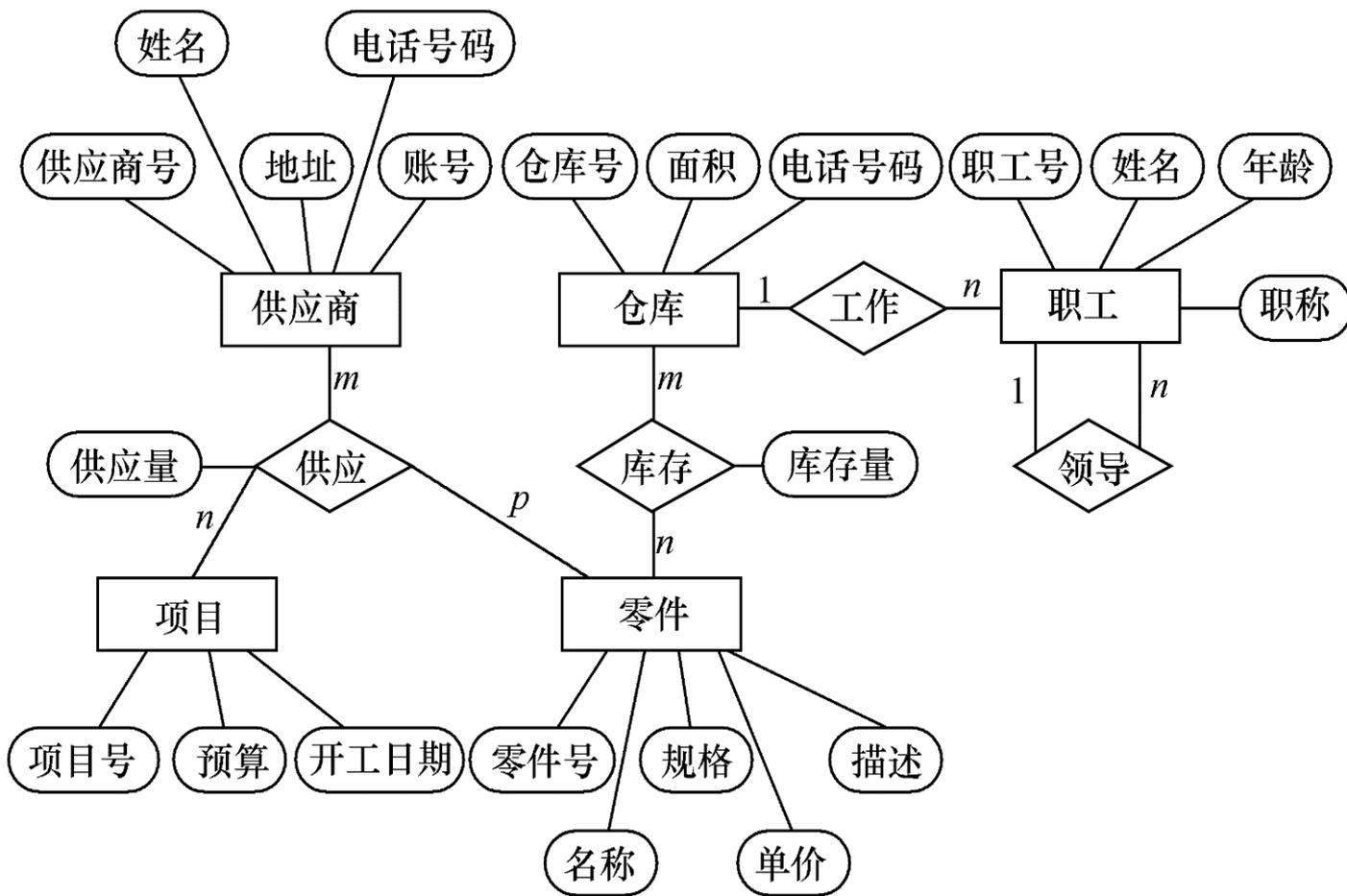
98

- 信息世界中的基本概念
- 两个实体型之间的联系
- 概念模型的一种表示方法：**E-R方法**
- 一个实例



1.2.2 什么是概念模型

□ 一个实例：工厂物资管理系统





一、信息世界中的基本概念

100

(1) 实体 (Entity)

客观存在并可相互区别的事物称为实体。

可以是具体的人、事、物或抽象的概念。

(2) 属性 (Attribute)

实体所具有的某一特性称为属性。

一个实体可以由若干个属性来刻画。

(3) 码 (Key)

唯一标识实体的属性集称为码。



信息世界中的基本概念(续)

101

(4) 域 (Domain)

属性的取值范围称为该属性的域。

(5) 实体型 (Entity Type)

用实体名及其属性名集合来抽象和刻画同类实体称为实体型

(6) 实体集 (Entity Set)

同一类型实体的集合称为实体集



信息世界中的基本概念(续)

102

(7) 联系 (Relationship)

- 现实世界中事物内部以及事物之间的联系在信息世界中反映为实体内部的联系和实体之间的联系
- **实体内部**的联系通常是指组成实体的各属性之间的联系
- **实体之间**的联系通常是指不同实体集之间的联系
- 实体之间的联系有**一对一**、**一对多**和**多对多**等类型



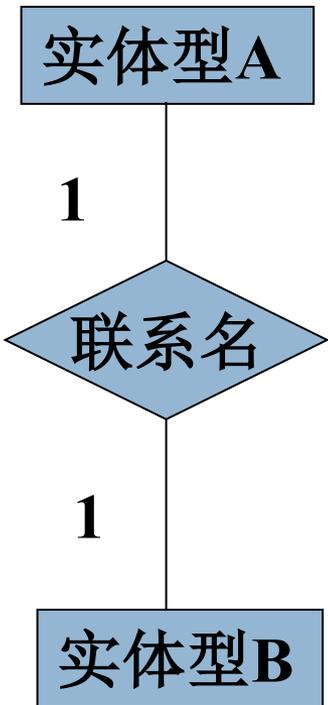
概念模型的一种表示方法

103

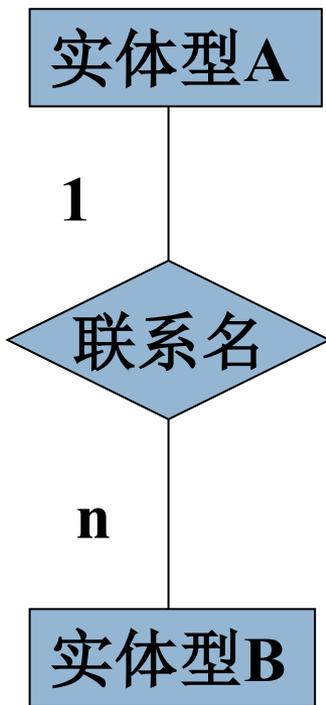
- 实体—联系方法(E-R方法) **第七章**
 - 用E-R图来描述现实世界的概念模型
 - E-R方法也称为E-R模型



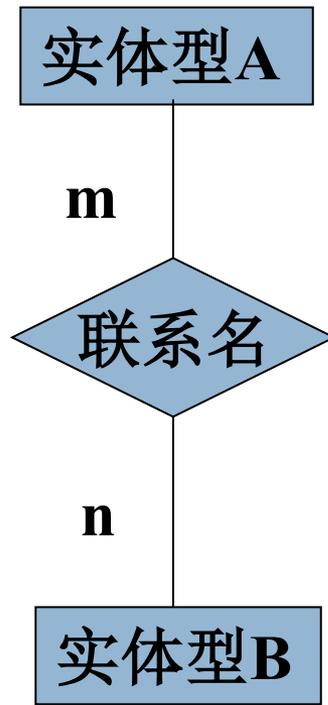
联系的表示方法



1:1联系



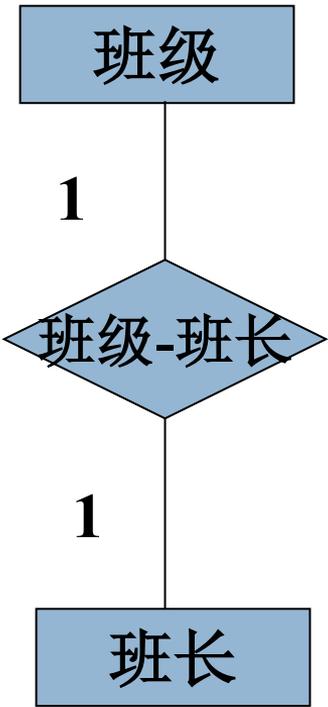
1:n联系



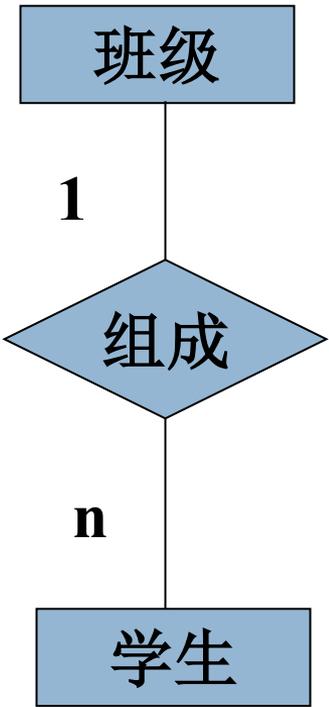
m:n联系



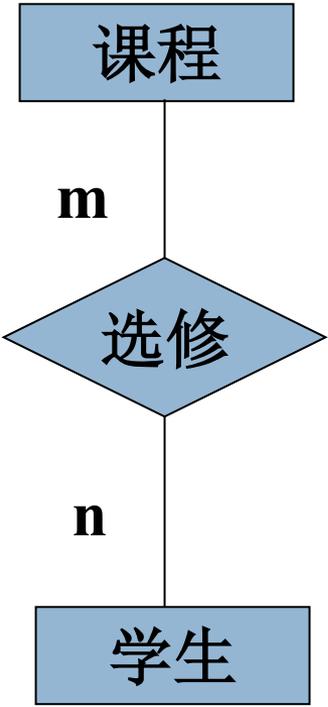
联系的表示方法示例



1:1联系



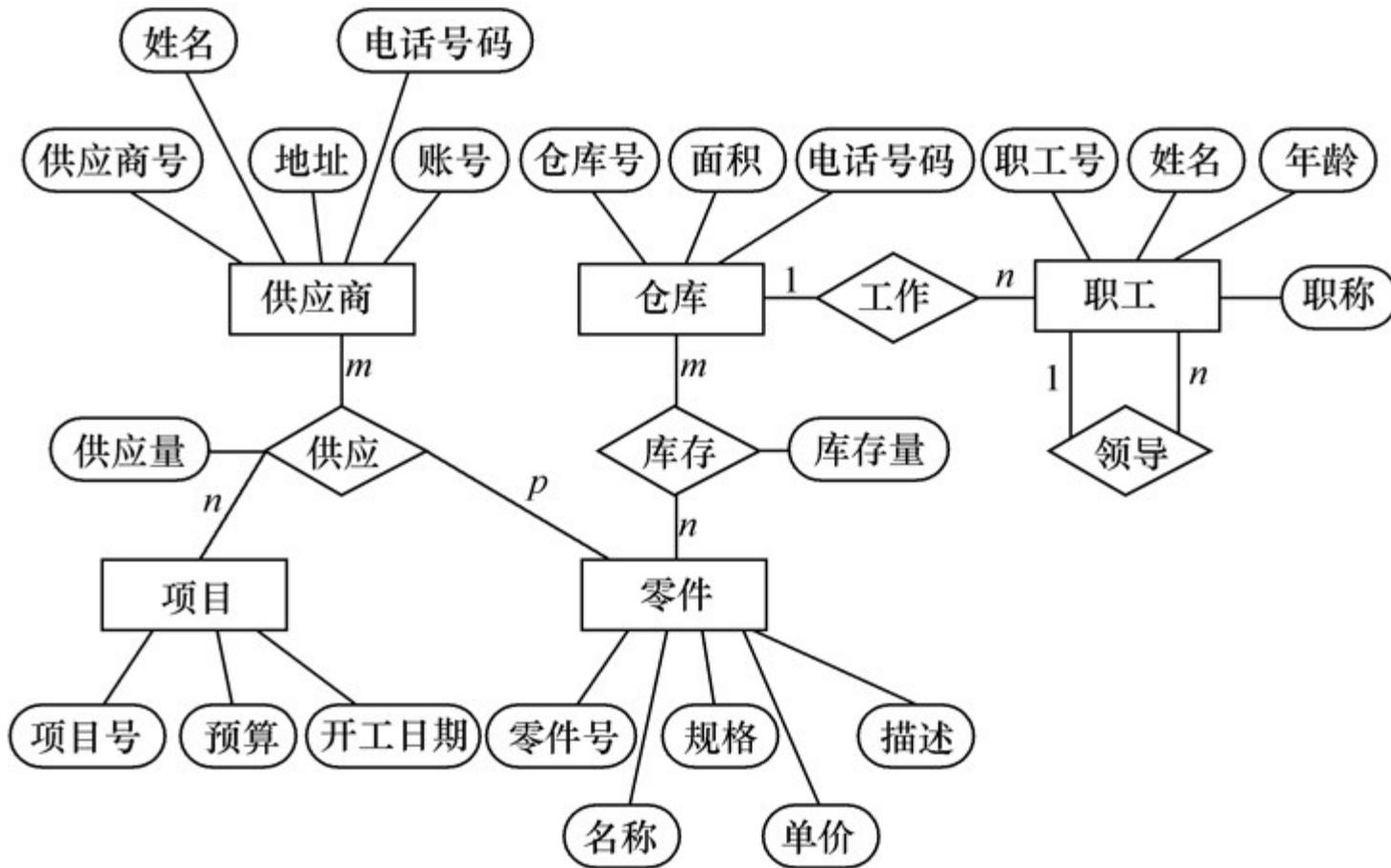
1:n联系



m:n联系



回顾E-R图的实例



(c) 完整的实体-联系图



1.2 数据模型

107

1.2.1 数据建模

1.2.2 概念模型

1.2.3 数据模型组成的三要素

1.2.4 层次模型

1.2.5 网状模型

1.2.6 关系模型

1.2.7* 数据库领域中不断涌现的数据模型



1.2.3 数据模型组成的三要素

108

- 数据结构
- 数据操作
- 完整性约束条件



一、数据结构

109

- 什么是数据结构
 - 描述数据库的**组成对象**，以及**对象之间的联系**
- 描述的内容
 - 与数据类型、内容、性质有关的对象
 - 如：关系模型中域，属性，关系
 - 与数据之间联系有关的对象
- 数据结构是对**系统静态特性的描述**



二、数据操作

110

- 数据操作
 - 对数据库中各种**对象(型)的实例(值)**允许执行的**操作集合**，包括**操作**及有关的**操作规则**
- 数据操作的类型
 - 查询
 - 更新(包括插入、删除、修改)



数据操作(续)

111

- 数据模型对操作的定义
 - 操作的确切含义（具体进行什么操作）
 - 操作符号（具体的符号是什么）
 - 操作规则（如优先级）（操作的顺序规则）
 - 实现操作的语言（软件的实现语言）
- 数据操作是对**系统动态特性**的描述
 - 数据库的状态会不断变化



三、数据的完整性约束条件

112

□ 数据的完整性约束条件

□ 一组完整性规则的集合

□ 完整性规则：给定的数据模型中数据及其联系所具有的
制约和依存规则

- 用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效、相容
- 保证数据的状态是从正确到正确的



数据的完整性约束条件(续)

113

- 数据模型对完整性约束条件的定义
 - 反映和规定数据模型必须遵守的基本的通用的完整性约束条件（系统规定的）
 - 例如，关系模型必须满足实体完整性和参照完整性（早期的关系模型有些不满足）
 - 学生必须有学号
 - 学生选的课（课程信息存在另外一个数据表里）必须已经开设
 - 提供定义完整性约束条件的机制，以反映具体应用所涉及的数据必须遵守的特定的语义约束条件（用户规定的）
 - 例如，关系模型中的用户定义完整性