



# 第一篇 基础篇

## 第一章 绪论(3)

福州大学数计学院

程 烨

chengye@fzu.edu.cn

2013年3月4日

## 层次数据模型的数据结构(续)

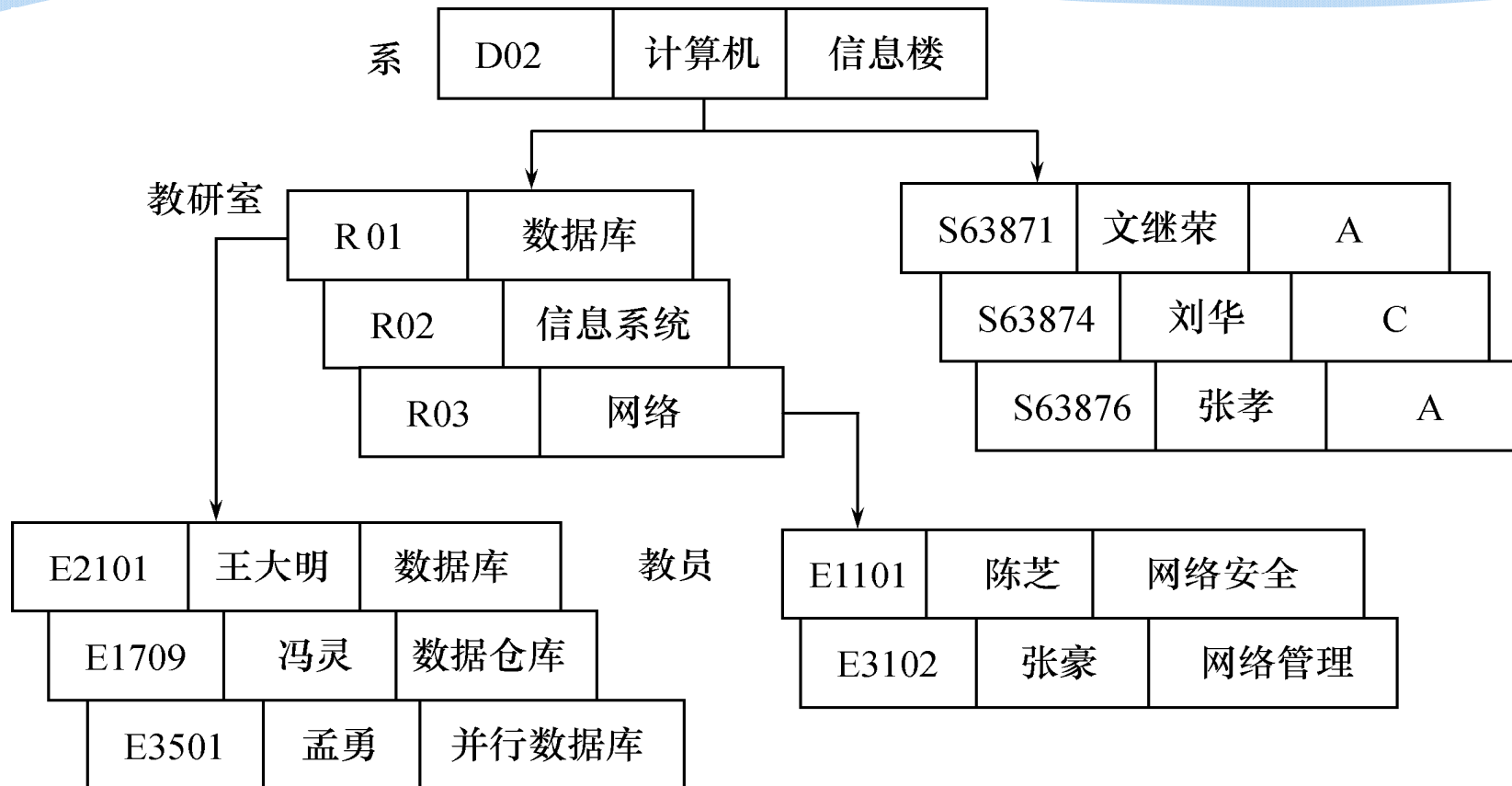
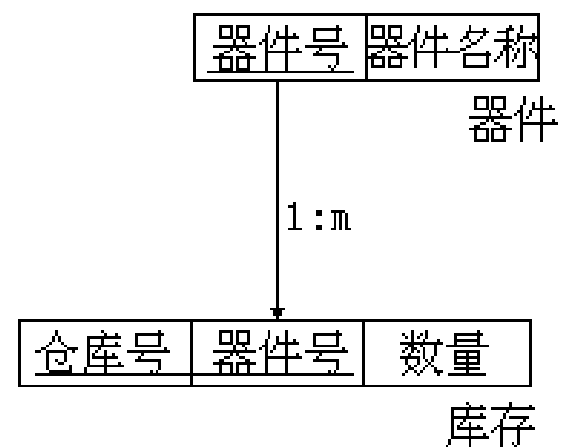
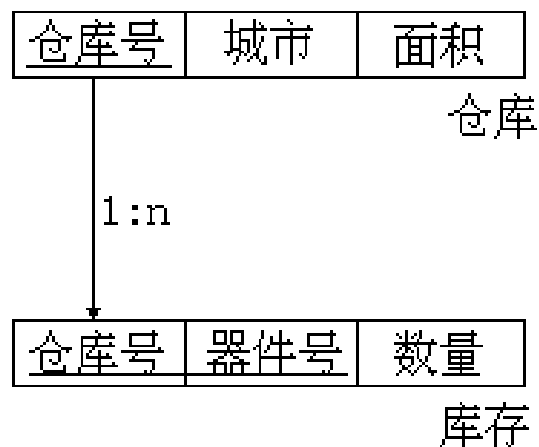
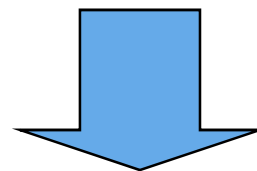
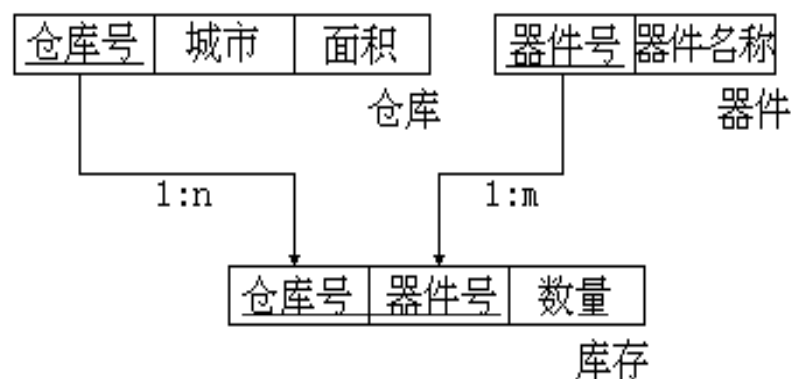


图1.18 教员学生层次数据库的一个值

## 层次型不能表示多对多的联系



# 层次模型的优缺点

## ❖ 优点

- 层次模型的数据结构比较简单清晰
- 查询效率高，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，应用程序的编写比较复杂
- 查询子女结点必须通过双亲结点
- 由于结构严密，层次命令趋于程序化

## 1.2 数据模型

1.2.1 两大类数据模型

1.2.2 数据模型的组成要素

1.2.3 概念模型

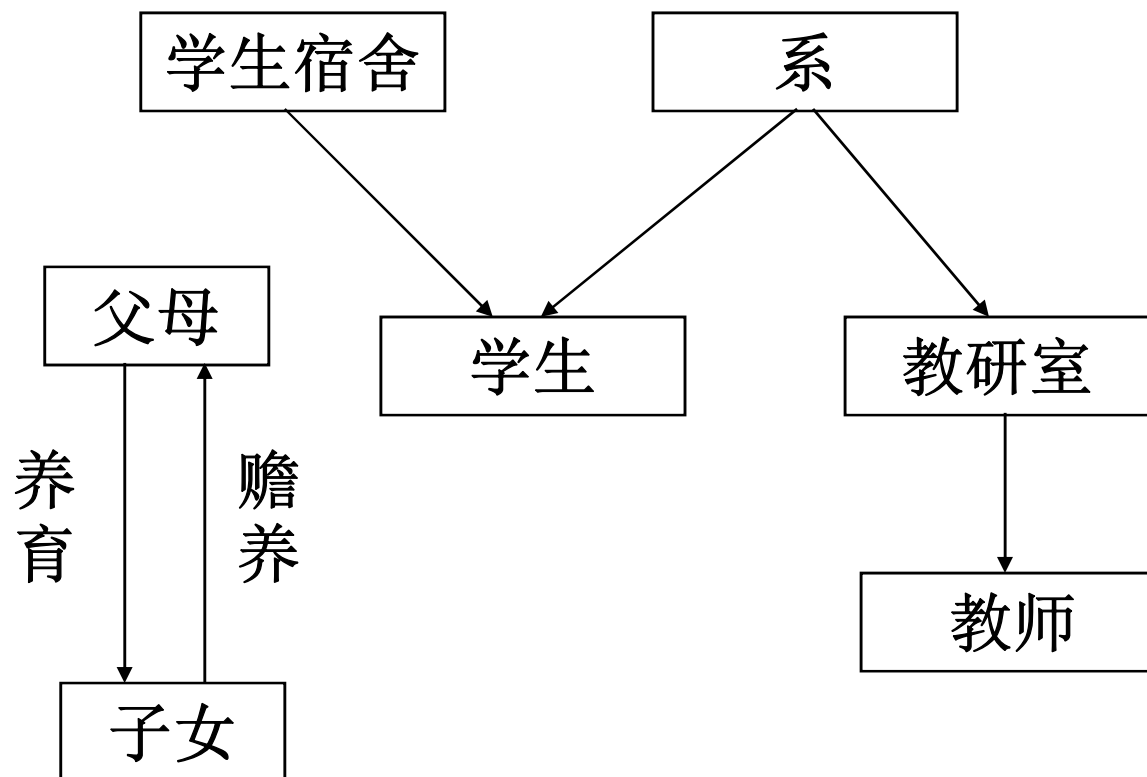
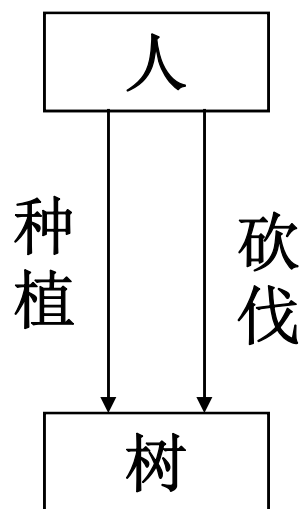
1.2.4 最常用的数据模型

1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型

## 1.2.6 网状模型



## 1.2.6 网状模型

- ❖ 网状数据库系统采用**网状模型**作为数据的组织方式
- ❖ 典型代表是**DBTG**系统(亦称**CODASYL**系统)
  - 70年代由**DBTG**提出的一个**系统方案**
  - 给出了网状数据库系统的基本概念、方法和技术
- ❖ 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**

## 网状模型（续）

- ❖ 网状数据模型的数据结构
- ❖ 网状数据模型的操纵与完整性约束
- ❖ 网状数据模型的存储结构
- ❖ 网状数据模型的优缺点



# 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：

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（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

## 1.2 数据模型

**1.2.1 两大类数据模型**

**1.2.2 数据模型的组成要素**

**1.2.3 概念模型**

**1.2.4 最常用的数据模型**

**1.2.5 层次模型**

**1.2.6 网状模型**

**1.2.7 关系模型**

## 1.2.7 关系模型

- ❖ 一、关系数据模型的数据结构
- ❖ 二、关系数据模型的操纵与完整性约束
- ❖ 三、关系数据模型的存储结构
- ❖ 四、关系数据模型的优缺点

## 1.2.7 关系模型

- ❖ 关系数据库系统采用关系模型作为数据的组织方式
- ❖ 1970年美国IBM公司San Jose研究室的研究员E.F.Codd首次提出了数据库系统的关系模型
- ❖ 计算机厂商推出的数据库管理系统几乎都支持关系模型

# 一、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(称为关系)，它由行和列组成。

学生登记表

学 号	姓 名	年 龄	性 别	系 名	年 级
2005004	王小明	19	女	社会学	2005
2005006	黄大鹏	20	男	商品学	2005
2005008	张文斌	18	女	法律	2005
...	...	...	...	...	...

属性

元组

## 关系数据模型的数据结构（续）

每个实体型对应一张二维表（关系）

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）

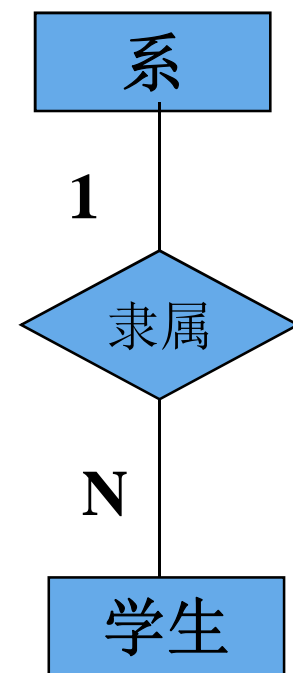
系（系号，系名，办公地点）

例2

系、系主任、系与系主任间的一对一联系

系（系号，系名，办公地点）

系主任（系主任号，姓名，年龄，性别，系号）



1:N联系



## 关系数据模型的数据结构（续）

实体型之间多对多联系也用**二维表**表示

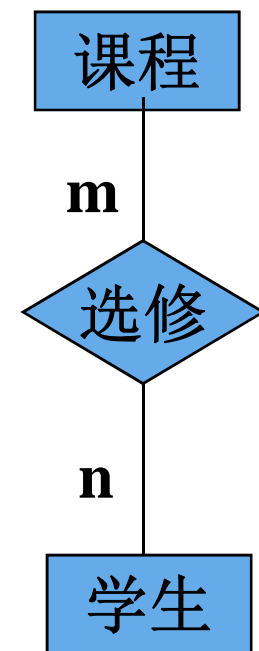
### 例3

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）



**m:n**联系

## 关系数据模型的数据结构（续）

❖ 关系必须是规范化的，满足一定的规范条件

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，不允许表中还有表

图1.27中工资和扣除是可分的数据项，不符合关系模型要求

职工号	姓名	职 称	工 资			扣 除		实 发
			基 本	津 贴	职 务	房 租	水 电	
86051	陈 平	讲 师	1305	1200	50	160	112	2283
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

图1.27 一个工资表(表中有表)实例

## 关系数据模型的数据结构（续）

表1.2 术语对比

关系术语	一般表格的术语
关系名	表名
关系模式	表头（表格的描述）
关系	（一张）二维表
元组	记录或行
属性	列
属性名	列名
属性值	列值
分量	一条记录中的一个列值
非规范关系	表中有表（大表中嵌有小表）

## 二、关系数据模型的操纵与完整性约束

### ❖ 数据操作:

- 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“做什么”，不必详细说明“怎么做”

## 关系数据模型的操纵与完整性约束（续）

### ❖ 关系的完整性约束条件

- 实体完整性
- 参照完整性
- 用户定义的完整性

### 三、关系数据模型的存储结构

#### ❖ 表以文件形式存储（由**DBMS**创建与管理）

- 有的**DBMS**一个二维表对应一个操作系统文件
- 有的**DBMS**从操作系统获得若干大的文件,自己设计二维表、索引等存储结构

## 四、关系数据模型的优缺点

### ❖ 优点

- 建立在严格的数学概念的基础上
- 概念单一
  - 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

## 关系数据模型的优缺点（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DBMS的难度



# 各种模型的比较

	层次模型	网状模型	关系模型	面向对象模型
创始	1968年IBM公司的IMS系统	1969年CODASYL的DBTG报告(71年通过)	1970年F. Codd提出关系模型	20世纪80年代
数据结构	复杂 (树结构)	复杂 (有向图结构)	简单 (二维表)	复杂 (嵌套递归)
数据联系	通过指针	通过指针	通过表间的公共属性	通过对象标识
查询语言	过程性语言	过程性语言	非过程性语言	面向对象语言
典型产品	IMS	IDS/ II IMAGE/3000 IDMS TOTAL	Oracle Sybase DB2 SQL Server Informix	ONTOS DB
盛行期	20世纪70年代	70年代至80年代中期	80年代至现在	90年代至现在

# 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(重点与难点)

## 1.4 数据库系统的组成

## 1.5 小结

## 1.3 数据库系统结构

### ■ 一. 数据库系统的模式结构

从**数据库管理系统**角度看, 数据库系统通常采用三级模式结构, 是数据库系统**内部**的系统结构。

### ■ 二. 数据库系统的体系结构

从**数据库用户**角度看, 数据库系统**外部**的体系结构。

# 一、数据库系统的模式结构

## 1.3.1 数据库系统模式的概念

## 1.3.2 数据库系统的三级模式结构

## 1.3.3 数据库的二级映像功能与数据独立性

## 1.3.1 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**型**:

(学号, 姓名, 性别, 系别, 年龄, 籍贯)

一个记录**值**:

(**900201**, 李明, 男, 计算机, **22**, 江苏)

# 数据库系统模式的概念

## ❖ 模式（Schema）

- 数据库逻辑结构和特征的描述
- 是型的描述
- 反映的是数据的结构及其联系
- 模式是相对稳定的

## ❖ 实例（Instance）

- 模式的一个具体值
- 反映数据库某一时刻的状态
- 同一个模式可以有很多实例
- 实例随数据库中的数据更新而变动

## 数据库系统模式的概念（续）

例如：在学生选课数据库模式中，包含学生记录、课程记录和学生选课记录

- **2010年**的一个学生数据库实例，包含：
  - **2010年**学校中所有学生的记录
  - 学校开设的所有课程的记录
  - 所有学生选课的记录
- **2009年度**学生数据库模式对应的实例与**2010年度**学生数据库模式对应的实例是**不同**的



# 一、数据库系统的模式结构

**1.3.1 数据库系统模式的概念**

**1.3.2 数据库系统的三级模式结构**

**1.3.3 数据库的二级映像功能与数据独立性**



## 1.3.2 数据库系统的三级模式结构

- ❖ 模式 (**Schema**)
- ❖ 外模式 (**External Schema**)
- ❖ 内模式 (**Internal Schema**)

## 数据库系统的三级模式结构（续）

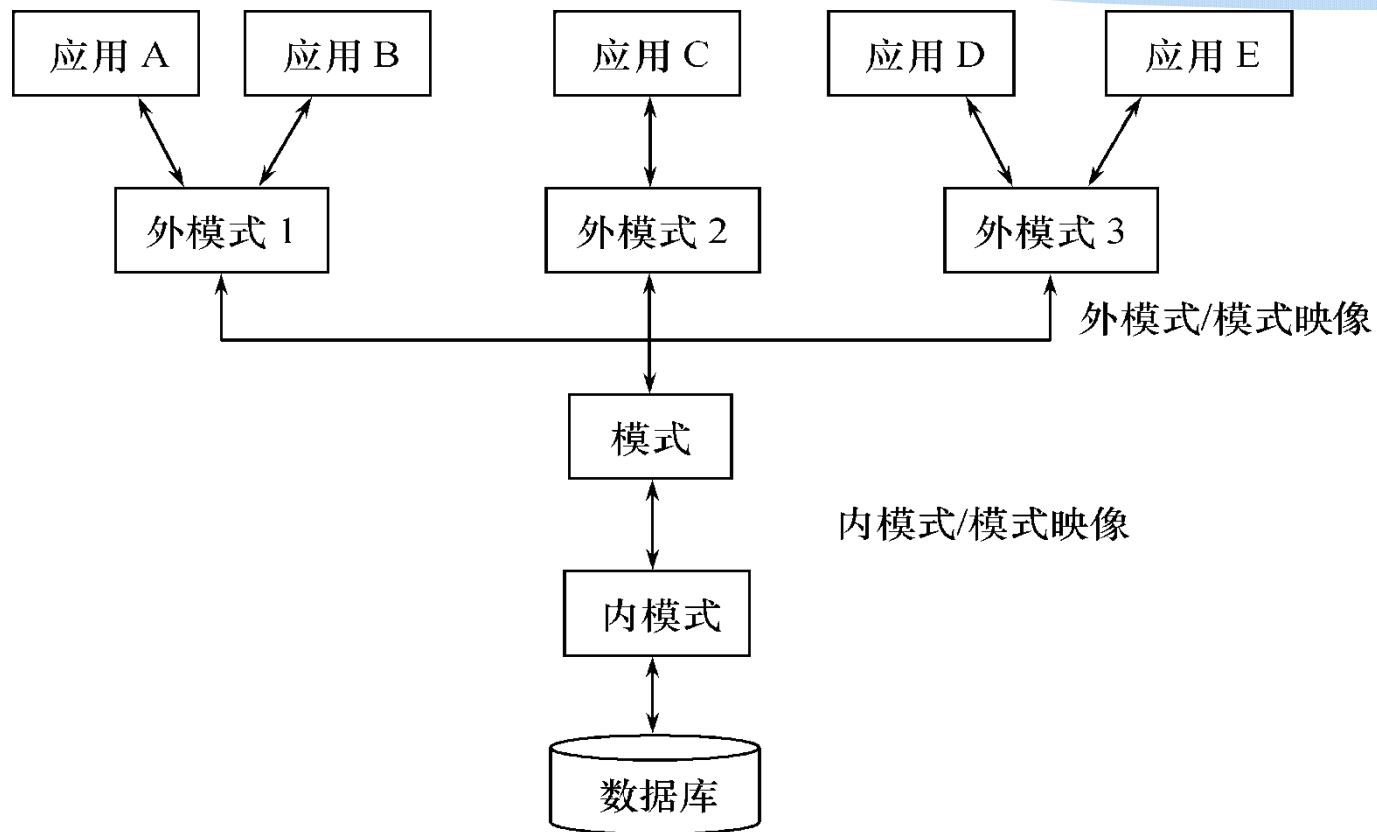


图1.28 数据库系统的三级模式结构

# 一、模式（Schema）

## ❖ 模式（也称逻辑模式）

- 数据库中全体数据的逻辑结构和特征的描述
- 所有用户的公共数据视图，综合了所有用户的需求

## ❖ 一个数据库只有一个模式，模式以某种数据模型为基础

## ❖ 模式的地位：是数据库系统模式结构的中间层

- 与数据的物理存储细节和硬件环境无关
- 与具体的应用程序、开发工具及高级程序设计语言无关

## 模式（续）

### ❖ 模式的定义

- 数据的逻辑结构（数据项的名字、类型、取值范围等）
- 数据之间的联系
- 数据有关的安全性、完整性要求

**DBMS**提供模式描述语言（**DDL**）定义模式。

## 二、外模式（External Schema）

❖ 外模式（也称子模式或用户模式）

- 数据库用户（包括应用程序员和最终用户）使用的局部数据的逻辑结构和特征的描述
- 数据库用户的数据视图，是与某一应用有关的数据的逻辑表示

## 外模式（续）

### ❖ 外模式的地位：介于模式与应用之间

#### ■ 模式与外模式的关系：一对多

- 外模式通常是模式的子集
- 一个数据库可以有多个外模式。反映了不同用户的应用需求、看待数据的方式、对数据保密的要求
- 对模式中同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同

#### ■ 外模式与应用的关系：一对多

- 同一外模式也可以为某一用户的多个应用系统所使用
- 但一个应用程序只能使用一个外模式

## 外模式（续）

### ❖ 外模式的用途

- 保证数据库安全性的一个有力措施
- 每个用户只能看见和访问所对应的外模式中的数据

### ❖ DBMS提供子模式描述语言（DDL）定义子模式。

### 三、内模式（Internal Schema）

#### ❖ 内模式（也称存储模式）

- 是数据物理结构和存储方式的描述
- 是数据在数据库内部的表示方式
  - 记录的存储方式（按属性值顺序存储或聚簇存储）
  - 索引的组织方式（是B+树索引，还是hash索引）
  - 数据是否压缩存储
  - 数据是否加密
  - 数据存储记录结构的规定

#### ❖ 一个数据库只有一个内模式

#### ❖ DBMS提供内模式定义语言（DDL）定义内模式



# 一、数据库系统的模式结构

**1.3.1 数据库系统模式的概念**

**1.3.2 数据库系统的三级模式结构**

**1.3.3 数据库的二级映像功能与数据独立性**

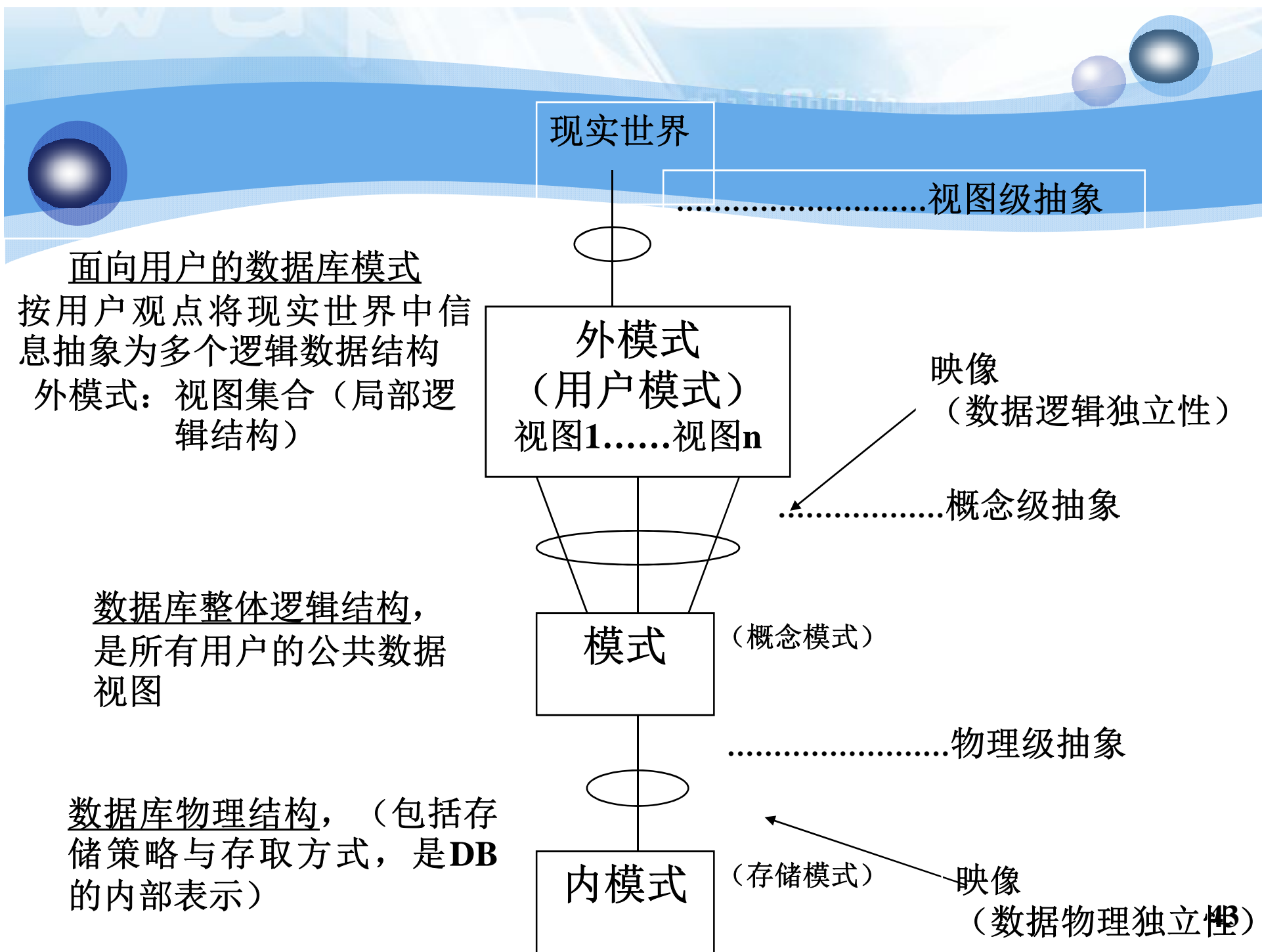
### 1.3.3 数据库的二级映像功能与数据独立性

❖ 三级模式是对数据的三个抽象级别

视图级抽象,概念级抽象,物理级抽象

二级映像**在DBMS内部实现三个抽象层次的联系和转换**

- 外模式 / 模式映像\_\_保证数据的**逻辑独立性**
- 模式 / 内模式映像\_\_保证数据的**物理独立性**



## 一、外模式 / 模式映像

- ❖ 模式：描述的是数据的**全局逻辑结构**
- ❖ 外模式：描述的是数据的**局部逻辑结构**
- ❖ 同一个模式可以有任意多个外模式
- ❖ 每一个外模式，数据库系统都有一个外模式 / 模式映像，定义外模式与模式之间的对应关系
- ❖ 外模式 / 模式映像的定义包含**外模式的描述**中

## 外模式 / 模式映像（续）

### 保证数据的逻辑独立性

- 当模式改变时，数据库管理员修改有关的外模式 / 模式映像，使外模式保持不变
- 应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。

## 二、模式 / 内模式映像

- ❖ 模式 / 内模式映像定义了数据全局逻辑结构与存储结构之间的对应关系。
  - 例如，说明逻辑记录和字段在内部是如何表示的
- ❖ 数据库中模式 / 内模式映像 is 唯一的
- ❖ 该映像定义通常包含在模式描述中

## 模式 / 内模式映像（续）

### 保证数据的物理独立性

- 当数据库的存储结构改变了（例如选用了另一种存储结构），数据库管理员修改模式 / 内模式映像，使模式保持不变
- 应用程序不受影响。保证了数据与程序的物理独立性，简称数据的物理独立性。

## 模式结构:小结

### ❖ 数据库模式

- 即全局逻辑结构是数据库的中心与关键
- 独立于数据库的其他层次
- 设计数据库模式结构时应首先确定数据库的逻辑模式



## 模式结构:小结

### ❖ 数据库的内模式

- 依赖于它的全局逻辑结构
- 独立于数据库的用户视图，即外模式
- 独立于具体的存储设备
- 作用：将全局逻辑结构中所定义的数据结构及其联系按照一定的物理存储策略进行组织，以达到较好的时间与空间效率

## 模式结构:小结

### ❖ 数据库的**外模式**

- **面向**具体的应用程序
- 定义在逻辑模式（模式）之上
- **独立于**存储模式（内模式）和存储设备
- 当应用需求发生较大变化，相应外模式不能满足其视图要求时，该外模式就得做相应改动
- 设计外模式时应充分考虑到应用的扩充性

## 模式结构:小结

### ❖ 特定的应用程序

- 在外模式描述的数据结构上编制的
- 依赖于特定的外模式
- 与数据库的模式和存储结构独立
- 不同的应用程序有时可以共用同一个外模式

### ❖ 数据库的二级映像

- 保证了数据库外模式的稳定性
- 从底层保证了应用程序的稳定性，除非应用需求本身发生变化，否则应用程序一般不需要修改

## 模式结构:小结

- ❖ 数据与程序之间的独立性，使得数据的定义和描述可以从应用程序中分离出去
- ❖ 数据的存取由**DBMS**管理
  - 用户不必考虑存取路径等细节
  - 简化了应用程序的编制
  - 大大减少了应用程序的维护和修改

# 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 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,程序员,用户等

## 数据库系统的组成（续）

- ❖ 硬件平台
- ❖ 软件(软件支撑平台,应用软件与数据库)
- ❖ 人员

# 一、硬件平台

## ❖ 数据库系统对硬件资源的要求

- (1) 足够大的内存
- (2) 足够大的安全的数据存储设备（磁盘，磁盘阵列，磁带等）
- (3) 较高的通道能力，提高数据传送率



## 二、软件

### ❖ **DBMS**

- ❖ 支持**DBMS**运行的**操作系统**
- ❖ 基于**DBMS**创建的**数据库**
- ❖ 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(**DBA**)
- ❖ 系统分析员和数据库设计人员
- ❖ 应用程序员
- ❖ 用户

## 人员（续）

❖ 不同的人员涉及不同的数据抽象级别，具有不同的数据视图，如下图所示

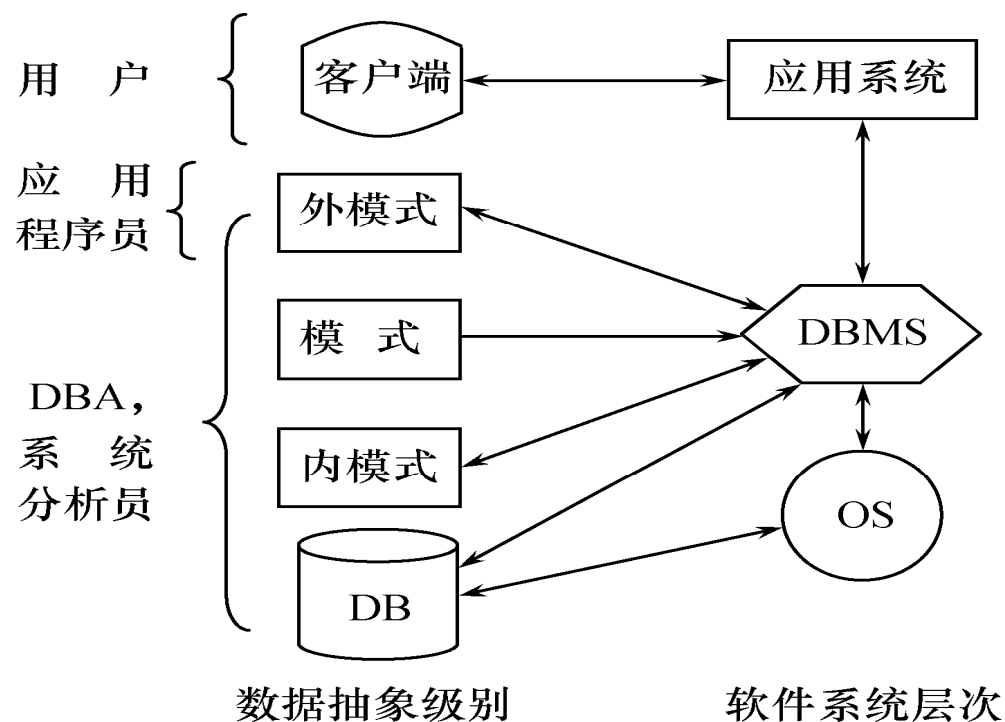


图1.30 各种人员的数据视图