数据库技术与应用

北京邮电大学计算机学院 肖达

xiaoda99@gmail.com

数据库系统结构

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的更新而变动

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

- 2013年的一个学生数据库实例，包含：
  - 2013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
- 2012年度学生数据库模式对应的实例与2013年度学生数据库模式对应的实例是不同的

数据库系统结构（续）

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）

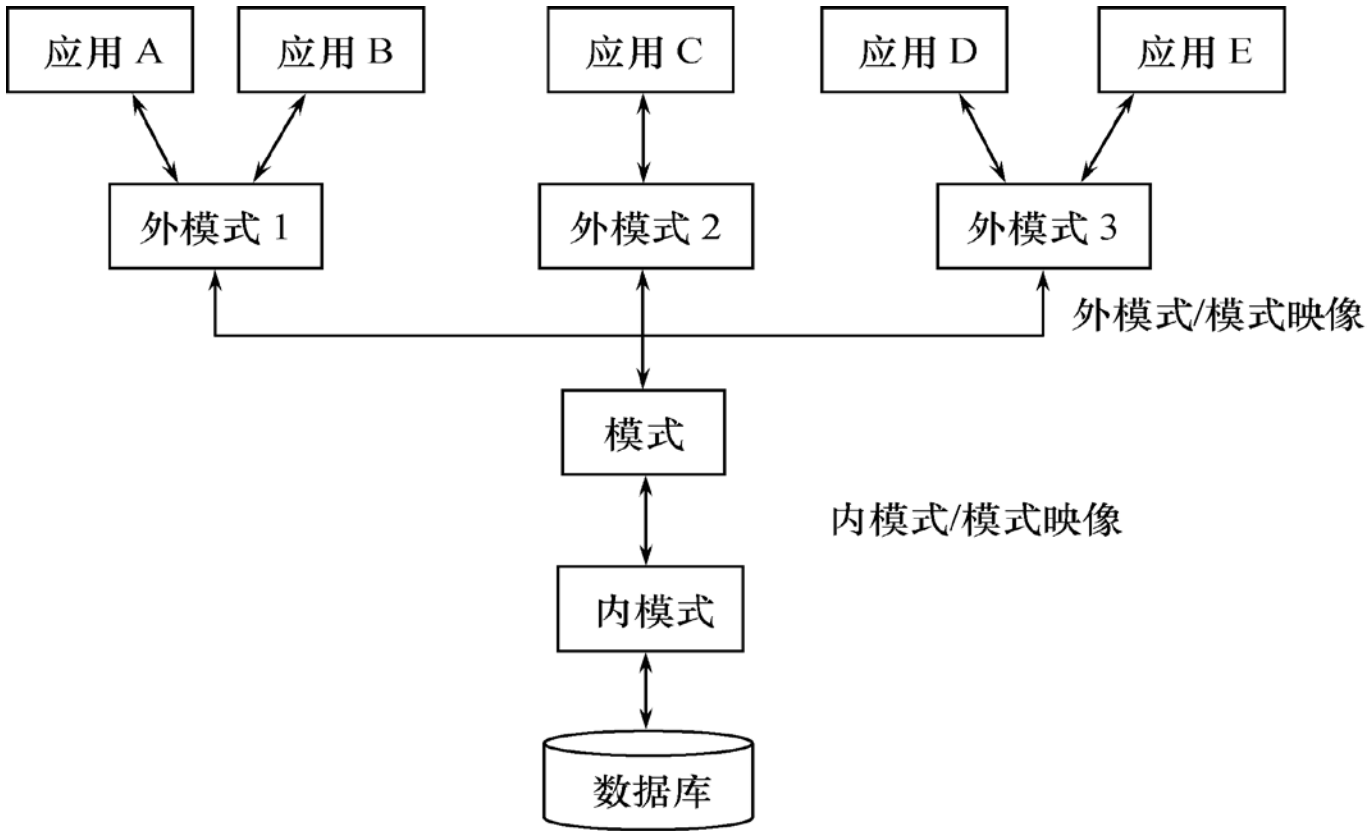


图1.28 数据库系统的三级模式结构

数据库系统的三级模式结构（续）

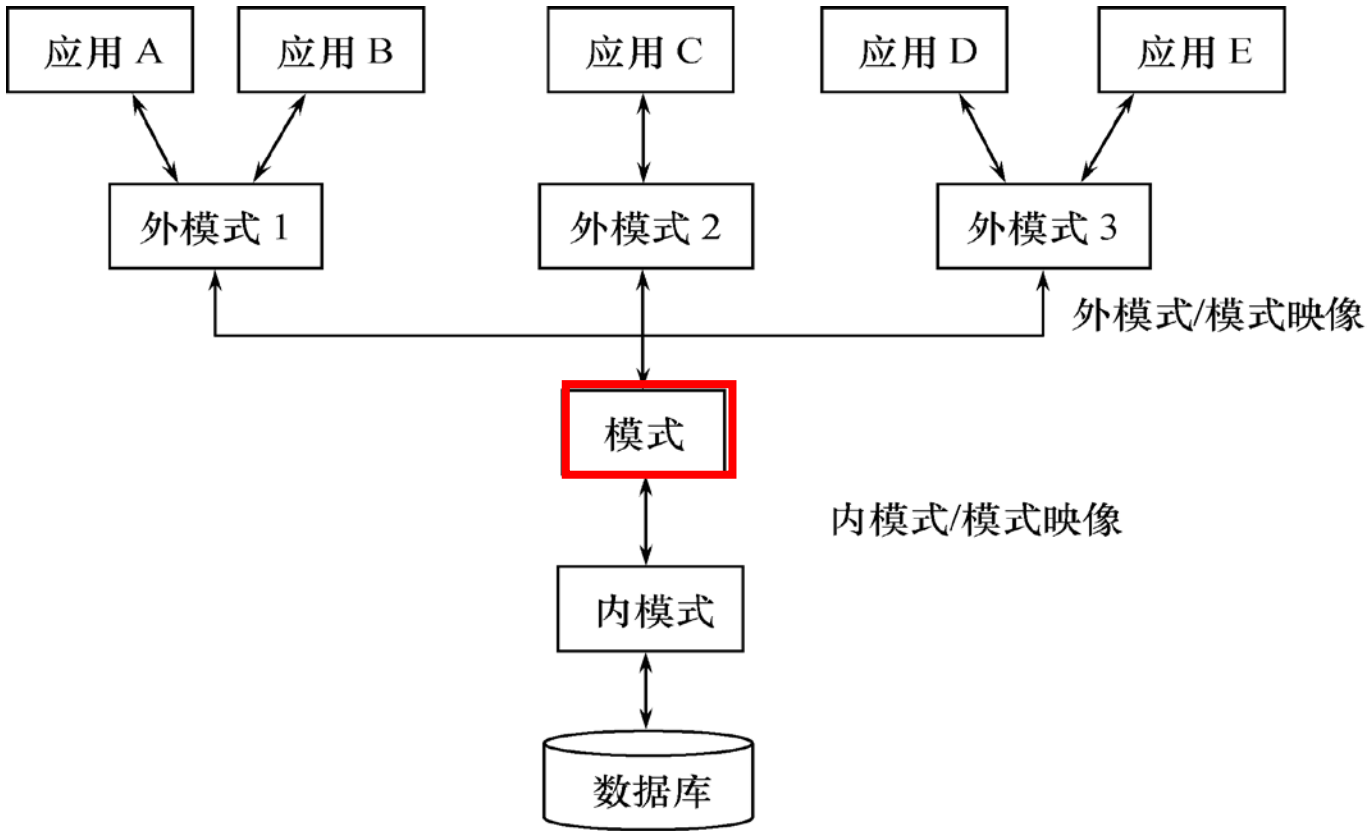


图1.28 数据库系统的三级模式结构

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

数据库系统的三级模式结构（续）

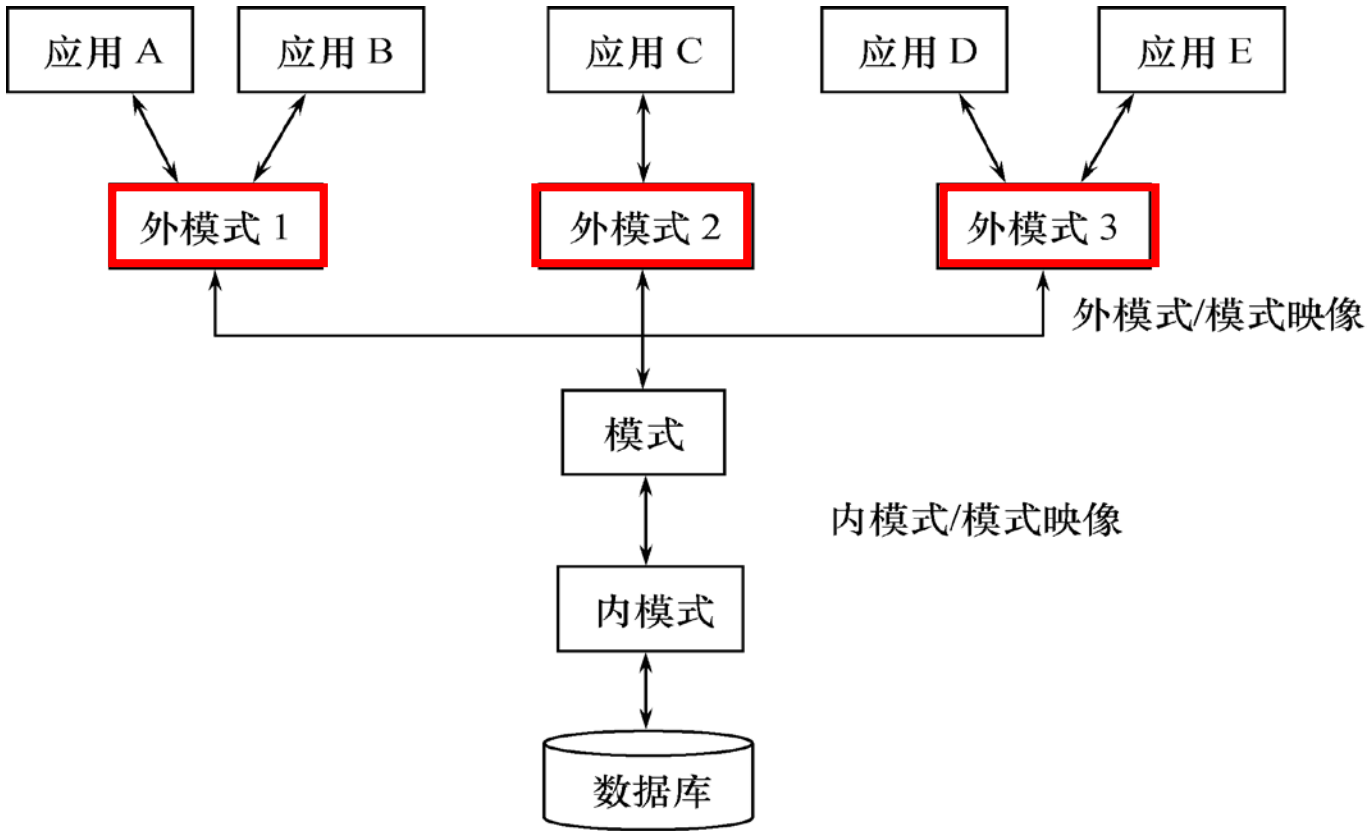


图1.28 数据库系统的三级模式结构

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。反映了不同的用户的应用需求、看待数据的方式、对数据保密的要求
  - 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

数据库系统的三级模式结构（续）

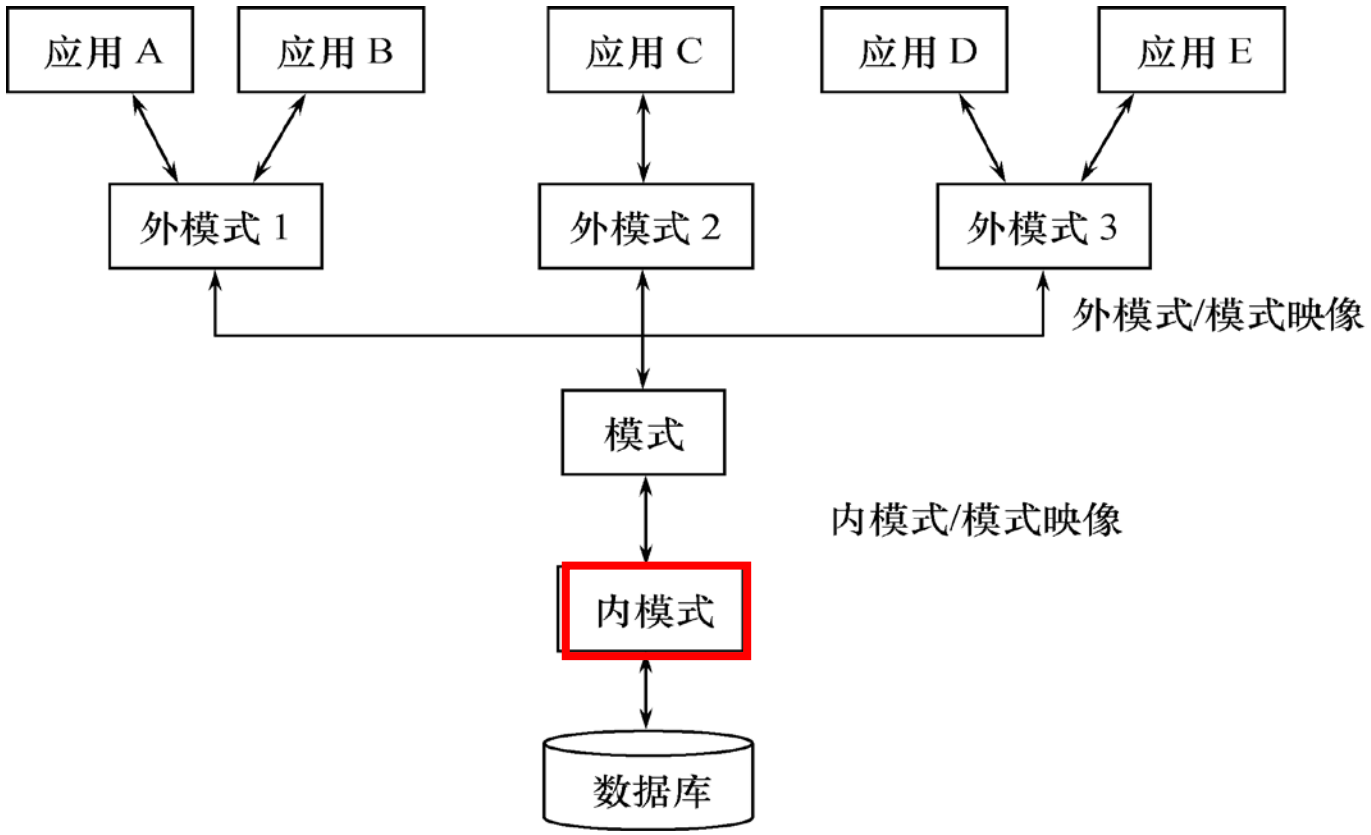


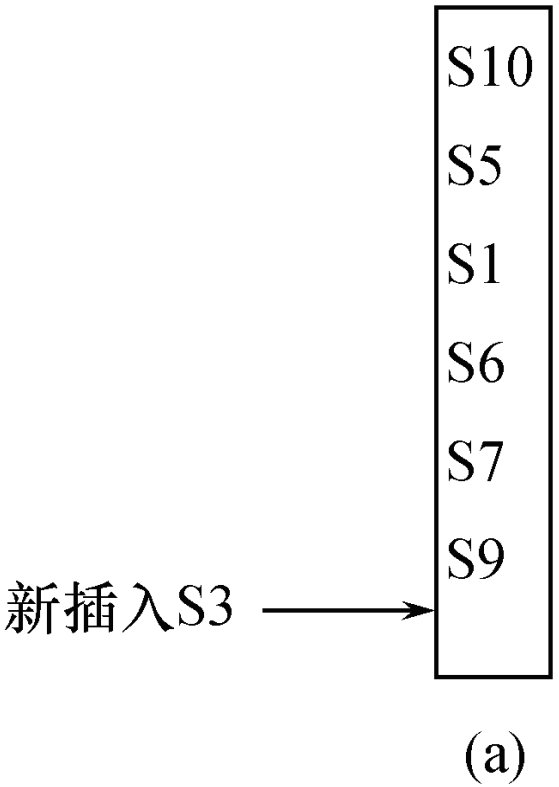
图1.28 数据库系统的三级模式结构

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（顺序存储，按照B树结构存储，按hash方法存储）
    - 索引的组织方式
    - 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

内模式（续）

- 例如学生记录，如果按**堆**存储，则插入一条新记录总是放在学生记录存储的**最后**，如右图所示



内模式（续）

- 如果按学号升序存储，则插入一条记录就要找到它应在的位置插入，如图1.29（b）所示
- 如果按照学生年龄聚簇存放，假如新插入的S3是16岁，则应插入的位置如图1.29（c）所示

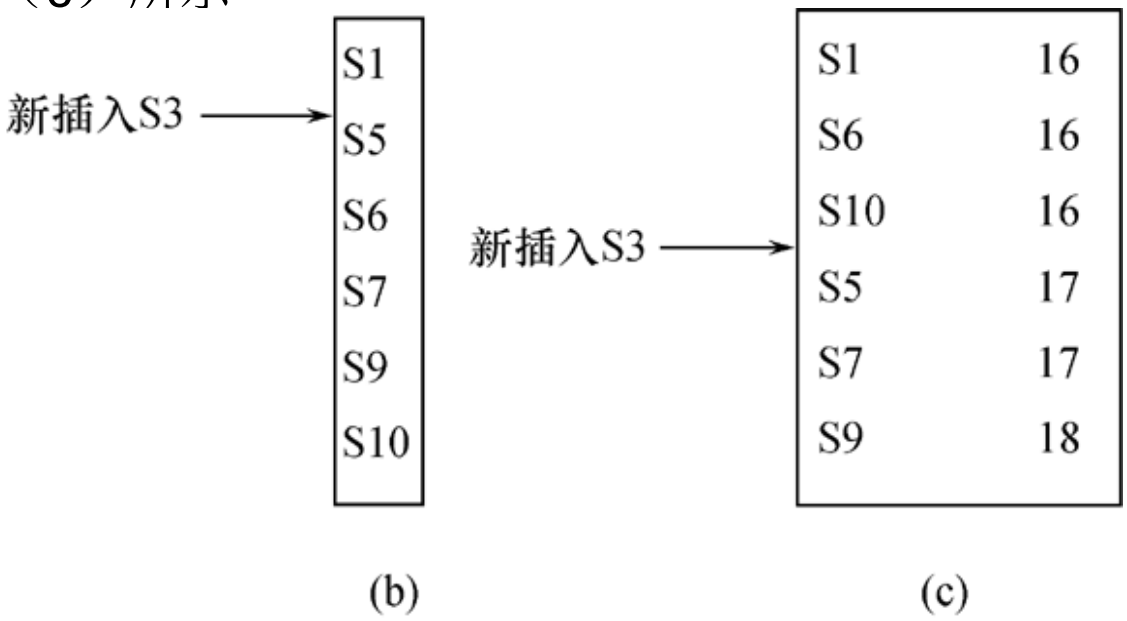


图1.29 记录不同的存储方式示意图

数据库系统的三级模式结构（续）

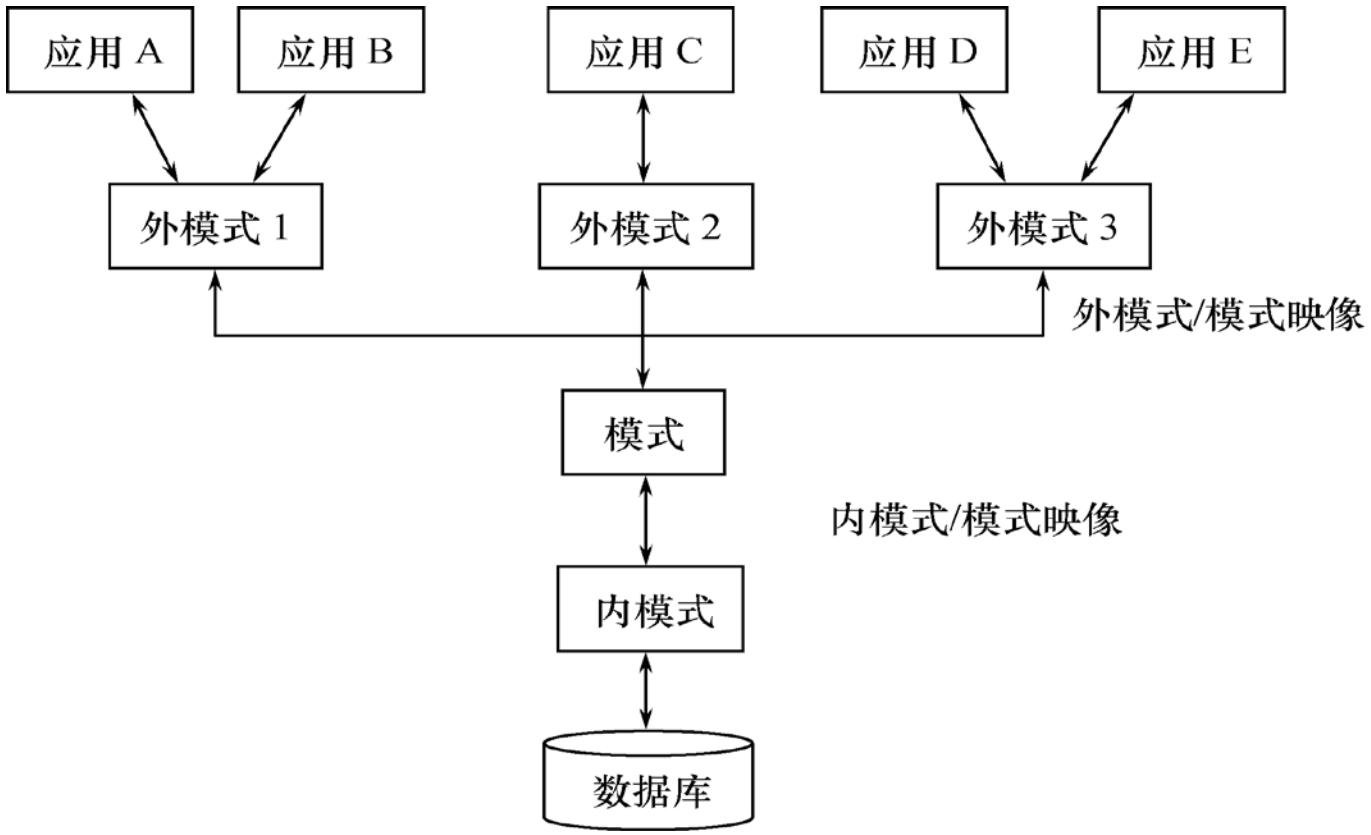
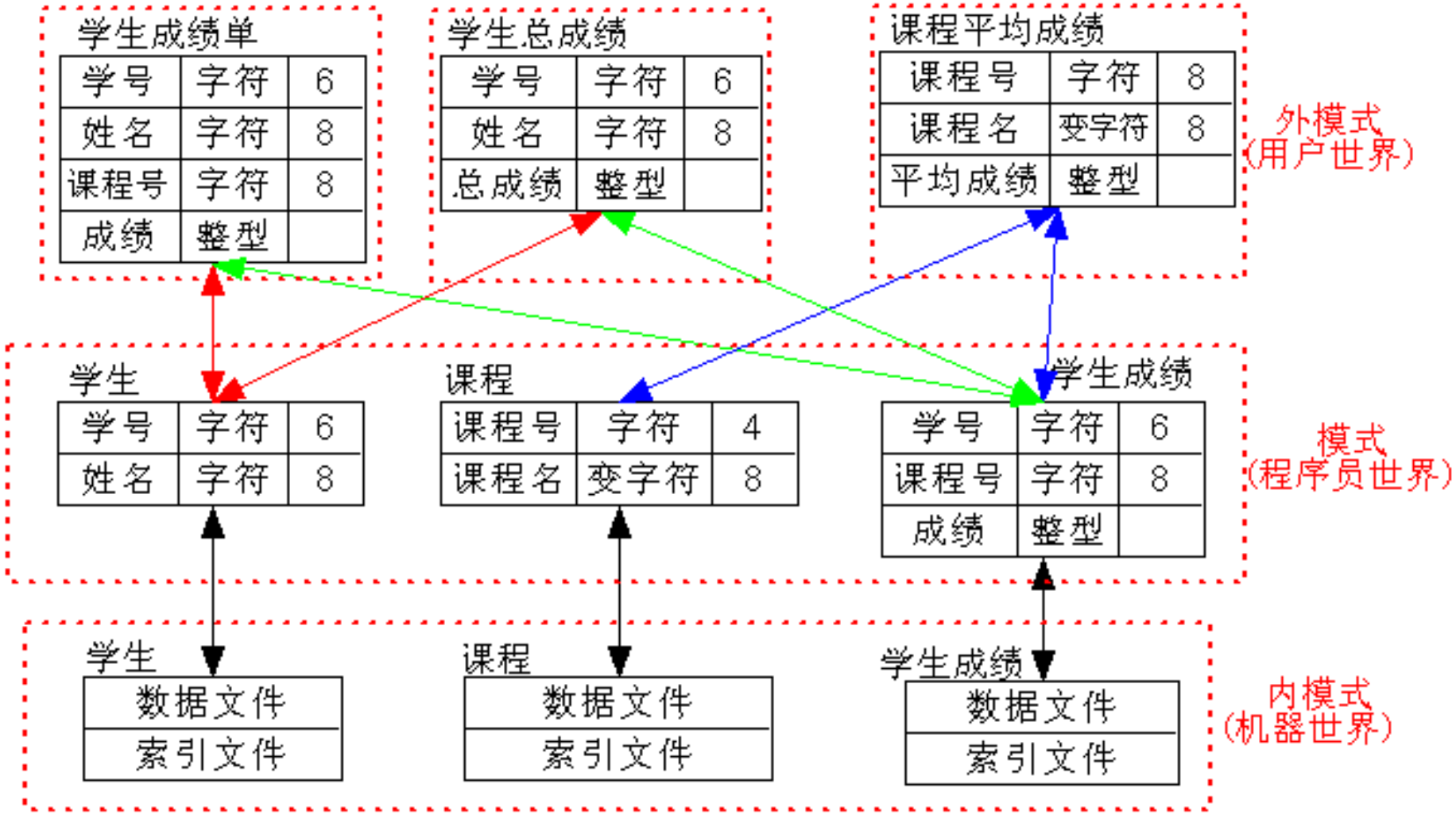


图1.28 数据库系统的三级模式结构

数据库系统的三级模式结构（续）



作业：设计模式/外模式

- 将如下火车票订票系统的模式补充完整（增加必要的属性），并设计余票查询系统和验票系统外模式
  - 车次（车次号，出发站，到达站，出发时间，到达时间，总座位数）
  - 座位（车次号，出发日期，已订座位数）
  - 乘客（身份证号，姓名，电话）
  - 订票（身份证号，车次号，订单号）

数据库系统结构（续）

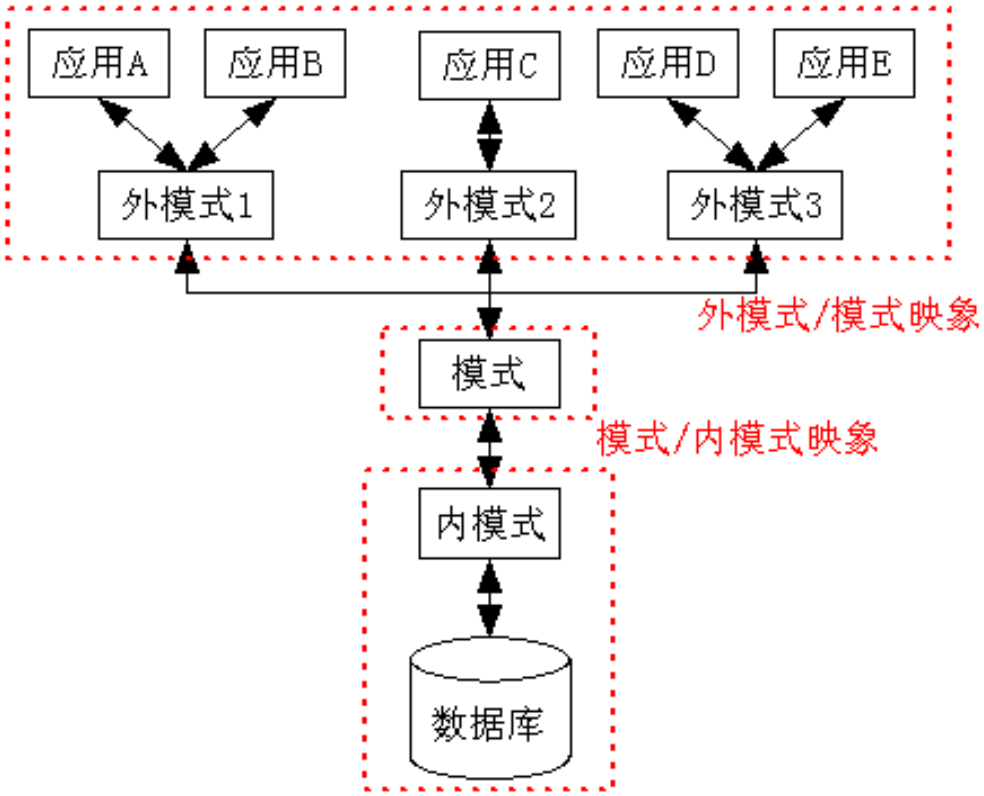
1.3.1 数据库系统模式的概念

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

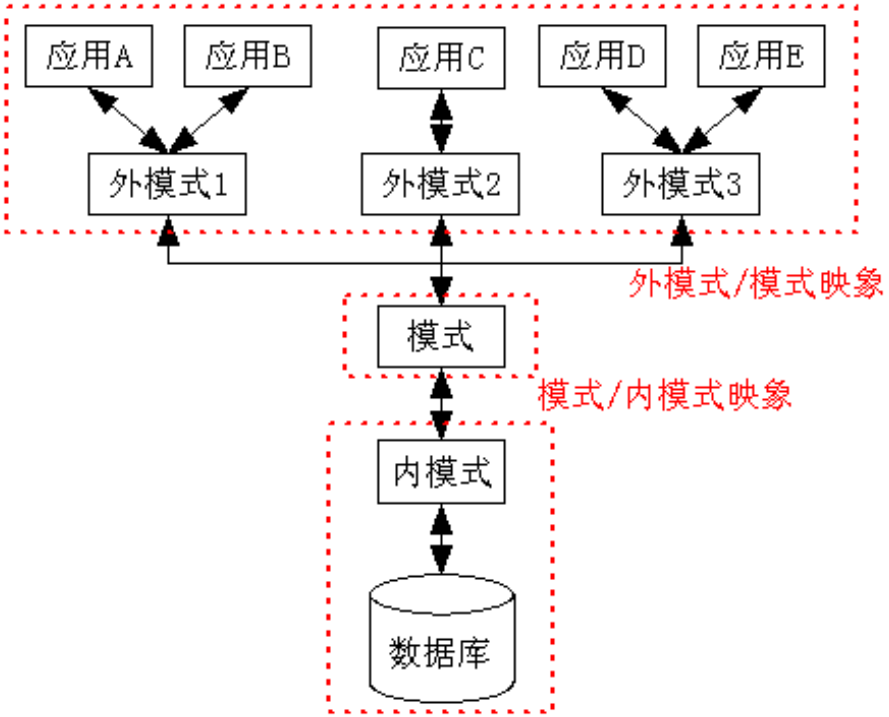
数据库的二级映像

- 三级模式是对数据的三个抽象级别
- 二级映象在DBMS内部实现这三个抽象层次的联系和转换



一、外模式 / 模式映象

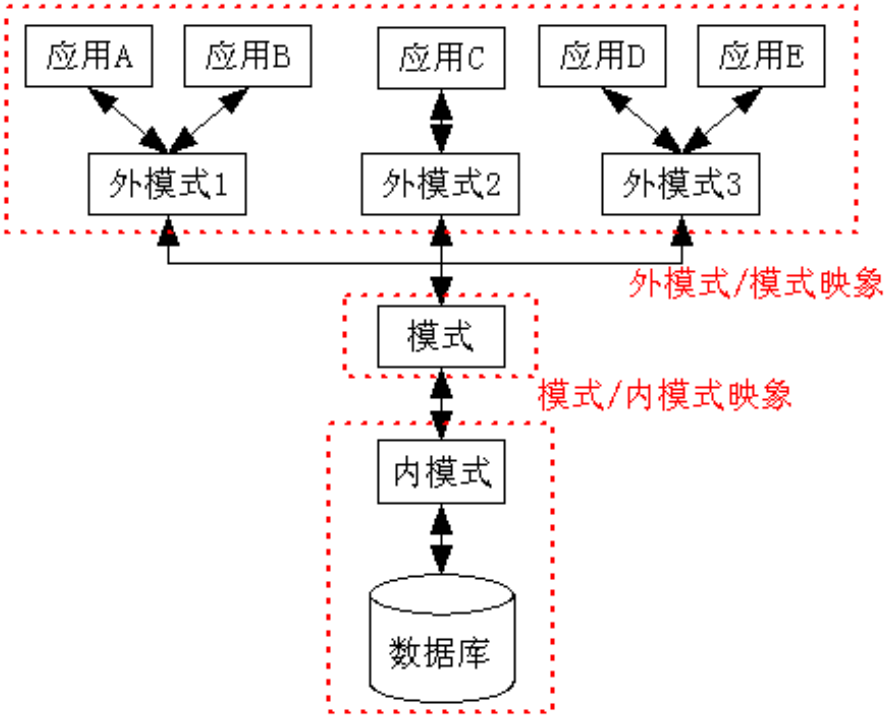
- 同一个模式可以有任意多个外模式
- 每一个外模式，数据库系统都有一个外模式 / 模式映象，定义外模式与模式之间的对应关系
- 映象定义通常包含在各自外模式的描述中



外模式 / 模式映象（续）

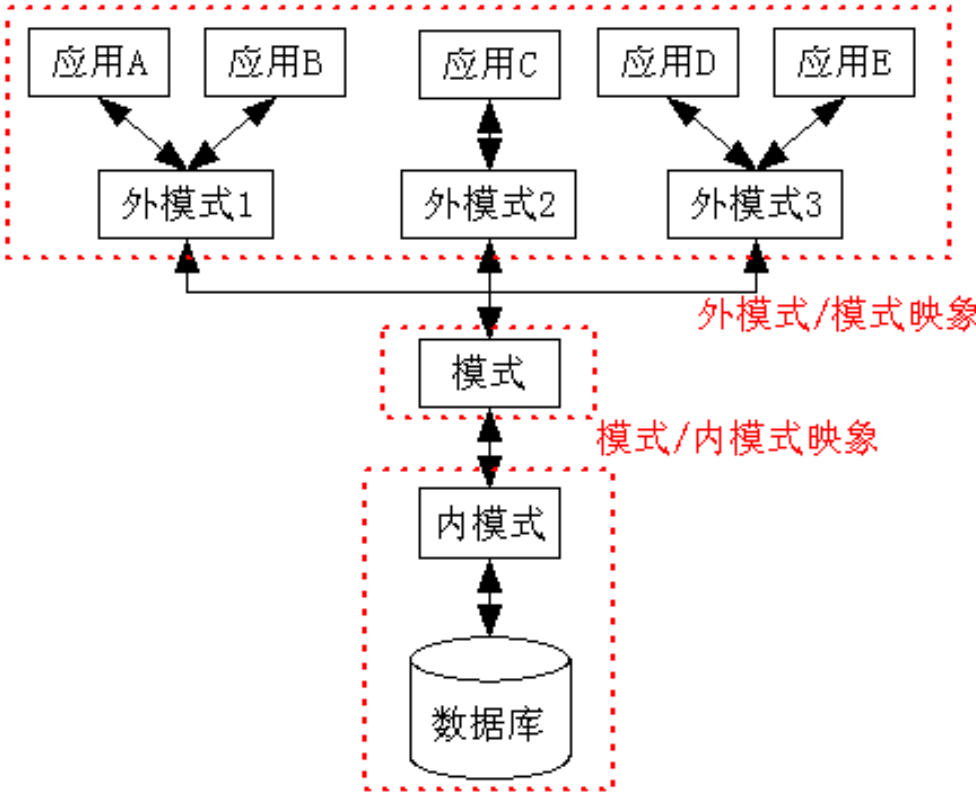
保证数据的逻辑独立性

- 当模式改变时，数据库管理员修改有关的外模式 / 模式映象，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 / 内模式映象

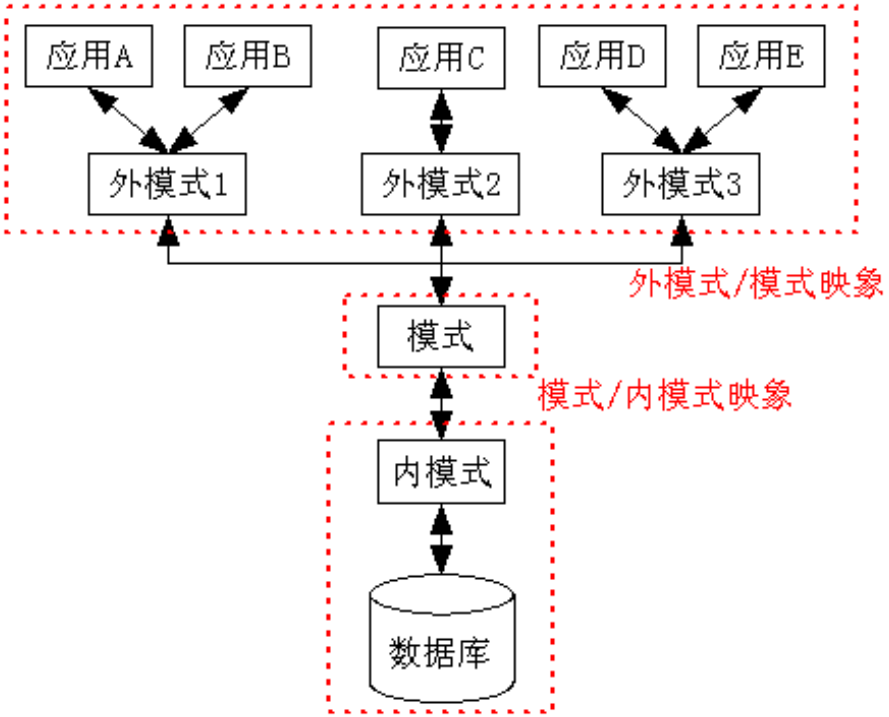
- 模式 / 内模式映象定义了数据全局逻辑结构与存储结构之间的对应关系。
  - 例如，说明逻辑记录和字段在内部是如何表示的
- 数据库中模式 / 内模式映象是唯一的
- 该映象定义通常包含在模式描述中



模式／内模式映象（续）

保证数据的物理独立性

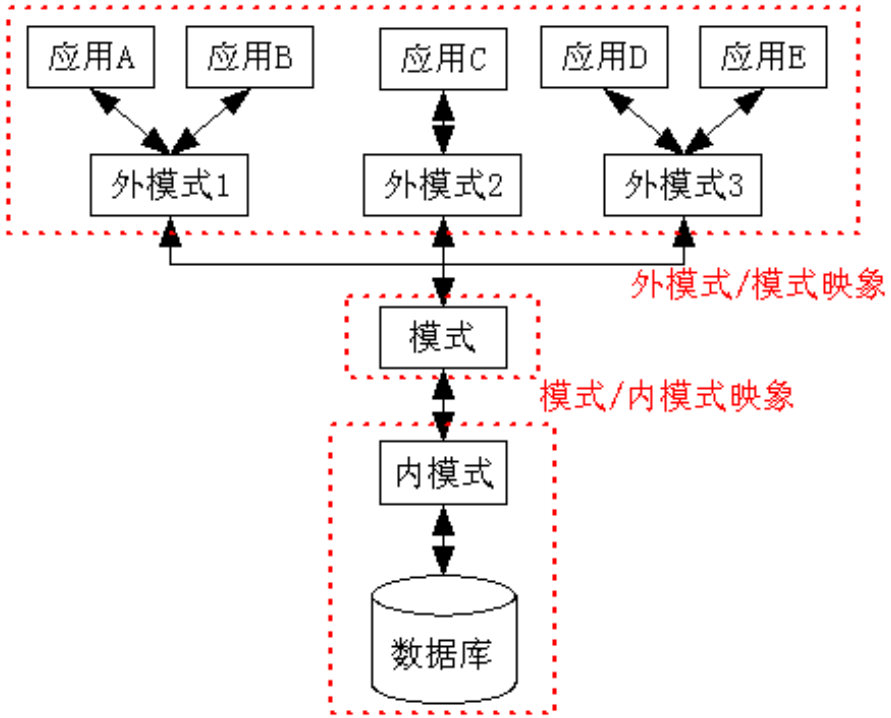
- 当数据库的存储结构改变了（例如选用了另一种存储结构），数据库管理员修改模式／内模式映象，使模式保持不变
- 应用程序不受影响。保证了数据与程序的物理独立性，简称数据的物理独立性。



模式 / 内模式映象（续）

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



关系数据库简介

- 1962年CODASYL发表“信息代数”
- 1968年David Child在IBM 7090机上实现的集合论数据结构

- 系统地、严格地提出关系模型的是美国**IBM**公司的**E. F. Codd**
  - 1970年提出关系数据模型

    E.F.Codd, “A Relational Model of Data for Large Shared Data Banks”, Communication of the ACM,1970
  - 之后，提出了关系代数和关系演算的概念
  - 1972年提出了关系的第一、第二、第三范式
  - 1974年提出了关系的**BC**范式

第二章 关系数据库

2.1 关系数据结构及形式化定义

2.2 关系操作

2.3 关系代数

2.4 关系的完整性

2.1 关系数据结构及形式化定义

- **2.1.1 关系**
- **2.1.2 关系模式**
- **2.1.3 关系数据库**

2.1.1 关系

- 单一的数据结构—关系

  现实世界的实体以及实体间的各种联系均用关系来表示

- 逻辑结构—二维表

  从用户角度，关系模型中数据的逻辑结构是一张二维表

- 建立在集合代数的基础上

关系(续)

1. 域（Domain）

2. 笛卡尔积（Cartesian Product）

3. 关系（Relation）

1. 域（Domain）

- **域**是一组具有相同数据类型的值的集合。例:
  - 整数
  - 实数
  - 介于某个取值范围的整数
  - 指定长度的字符串集合
  - {‘男’，‘女’}
  - ……

2. 笛卡尔积（Cartesian Product）

- 笛卡尔积

  给定一组域D_1，D_2，…，D_n，这些域中可以有相同的。

  D_1，D_2，…，D_n的笛卡尔积为：

  $$D_1 \times D_2 \times \ldots \times D_n =$$

  $$\{（d_1，d_2，\ldots，d_n） \mid d_i \in D_i，i=1，2，\ldots，n\}$$

  - 所有域的所有取值的一个组合
  - 不能重复

笛卡尔积（续)

- 元组（Tuple）
  - 笛卡尔积中每一个元素（d_1，d_2，…，d_n）叫作一个n元组（n-tuple）或简称元组(Tuple)
  - (张清玫，计算机专业，李勇)、(张清玫，计算机专业，刘晨)等都是元组
- 分量（Component）
  - 笛卡尔积元素（d_1，d_2，…，d_n）中的每一个值d_i叫作一个分量
  - 张清玫、计算机专业、李勇、刘晨等都是分量
- 基数（Cardinal number）
  - 若D_i（i=1，2，…，n）为有限集，其基数为m_i（i=1，2，…，n），则$D_1 \times D_2 \times \cdots \times D_n$的基数$M$为：$M = \prod_{i=1}^{n} m_i$

笛卡尔积（续)

- 例：设

  D_1为教师集合（T）= {t_1，t_2}

  D_2为专业集合（S）= {s_1，s_2，s_3}

  D_3为研究生集合（P）= {p_1，p_2}

则$D_1 \times D_2 \times D_3$是个三元组集合，元组个数为2×3×2，是所有可能的（教师，专业，研究生）元组集合

T	S	C
t_1	s_1	c_1
t_1	s_1	c_2
t_1	s_2	c_1
…	…	…
t_2	s_3	c_2

表 2.1　D_1，D_2，D_3 的笛卡尔积

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	计算机专业	李勇
张清玫	计算机专业	刘晨
张清玫	计算机专业	王敏
张清玫	信息专业	李勇
张清玫	信息专业	刘晨
张清玫	信息专业	王敏
刘逸	计算机专业	李勇
刘逸	计算机专业	刘晨
刘逸	计算机专业	王敏
刘逸	信息专业	李勇
刘逸	信息专业	刘晨
刘逸	信息专业	王敏

3. 关系（Relation）

- *D*1，*D*2，…，*D*n的笛卡尔积的某个子集才有实际含义

例：表2.1 的笛卡尔积没有实际意义

取出有实际意义的元组来构造关系

关系：SAP(SUPERVISOR，SPECIALITY，POSTGRADUATE)

假设：导师与专业：n:1，　导师与研究生：1:n

主码：POSTGRADUATE（假设研究生不会重名）

SAP关系可以包含三个元组

{ (张清玫，计算机专业，李勇)，

(张清玫，计算机专业，刘晨)，

(刘逸，信息专业，王敏) }

关系（续）

1) 关系

$D_1 \times D_2 \times \ldots \times D_n$的子集叫作在域$D_1$，$D_2$，…，$D_n$上的**关系**，表示为

$R（D_1，D_2，\ldots，D_n）$

- R：关系名
- n：关系的目或度（Degree）

2) 元组

关系中的每个元素是关系中的元组，通常用t表示。

3) 单元关系与二元关系

当n=1时，称该关系为**单元**关系（Unary relation）或**一元**关系

当n=2时，称该关系为**二元**关系（Binary relation）

关系（续）

4) 关系的表示

关系也是一个二维表，表的每行对应一个元组，表的每列对应一个域

表 2.2　SAP 关系

SUPERVISOR	SPECIALITY	POSTGRADUATE
张清玫	信息专业	李勇
张清玫	信息专业	刘晨
刘逸	信息专业	王敏

5)属性

关系中不同列可以对应相同的域

*为了加以区分，必须对每列起一个名字，称为属性（**Attribute**）*

***n**目关系必有**n**个属性*

关系（续）

6) 码

候选码（Candidate key）

若关系中的某一属性组的值能唯一地标识一个元组，则称该属性组为候选码

简单的情况：候选码只包含一个属性

全码（All-key）

最极端的情况：关系模式的所有属性组是这个关系模式的候选码，称为全码（All-key）

主码

若一个关系有多个候选码，则选定其中一个为**主码**（Primary key）

主属性

候选码的诸属性称为主属性（Prime attribute）

不包含在任何侯选码中的属性称为非主属性（ Non-Prime attribute）或非码属性（Non-key attribute）

思考题

- 哪些属性是候选码、主码、主/非主属性？（假设学生没有重名）
  - 学生（学号，姓名，性别，班号）
  - 选课（学号，课程号，成绩）

思考题

- 哪些属性是候选码、主码、主/非主属性？（假设学生没有重名）
  - 学生（学号，姓名，性别，班号）
  - 选课（学号，课程号，成绩）

关系（续）

7) 三类关系

基本关系（基本表或基表）

实际存在的表，是实际存储数据的逻辑表示

查询表

查询结果对应的表

视图表

由基本表或其他视图表导出的表，是虚表，不对应实际存储的数据

关系（续）

8)基本关系的性质

① 列是同质的（Homogeneous）

② 不同的列可出自同一个域

- 其中的每一列称为一个属性
- 不同的属性要给予不同的属性名

③ 列的顺序无所谓，列的次序可以任意交换

④ 任意两个元组的候选码不能相同

⑤ 行的顺序无所谓，行的次序可以任意交换

⑥ 分量必须取原子值

这是规范条件中最基本的一条

基本关系的性质(续)

表2.3 非规范化关系

SUPERVISOR	SPECIALITY	POSTGRADUATE	
		PG1	PG2
张清玫	信息专业	李勇	刘晨
刘逸	信息专业	王敏	

小表

2.1 关系数据结构

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

1. 什么是关系模式

- 关系模式（Relation Schema）是型
  - 对关系的描述，是静态的、稳定的
- 关系是值
  - 关系模式在某一时刻的状态或内容
- 关系模式是对关系的描述
  - 元组集合的结构
    - 属性构成
    - 属性来自的域
    - 属性与域之间的映象关系
  - 元组语义以及完整性约束条件
  - 属性间的数据依赖关系集合

2. 定义关系模式

关系模式可以形式化地表示为：

R（U，D，DOM，F）

R　　关系名

U　　组成该关系的属性名集合

D　　属性组*U*中属性所来自的域

DOM　属性向域的映象集合

F　　属性间的数据依赖关系集合

例：导师和研究生出自同一个域——人，

取不同的属性名，并在模式中定义属性向域的映象，即说明它们分别出自哪个域：

DOM（SUPERVISOR）= DOM（POSTGRADUATE）=PERSON

定义关系模式 (续)

关系模式通常可以简记为

R (U)　或　R (A$_1$，A$_2$，…，A$_n$)

- *R:* 关系名
- A_1，A_2，…，A_n：属性名

注：域名及属性向域的映象常常直接说明为属性的类型、长度

2.1 关系数据结构

2.1.1 关系

2.1.2 关系模式

2.1.3 关系数据库

2.1.3 关系数据库

- 关系数据库
  - 在一个给定的应用领域中，所有关系的集合构成一个关系数据库
- 关系数据库的型与值
  - 关系数据库的型：关系数据库模式

    对关系数据库的描述。
  - 关系数据库模式包括
    - 若干域的定义
    - 在这些域上定义的若干关系模式
  - 关系数据库的值：关系模式在某一时刻对应的关系的集合，简称为关系数据库

第二章 关系数据库

2.1 关系数据结构

2.2 关系操作

2.3 关系代数

2.4 关系的完整性

2.2.1基本关系操作

- 常用的关系操作
  - 查询：选择、投影、连接、除、并、交、差
  - 数据更新：插入、删除、修改
  - 查询的表达能力是其中最主要的部分
  - 选择、投影、并、差、笛卡尔基是5种基本操作
- 关系操作的特点
  - 集合操作方式：操作的对象和结果都是集合，一次一集合的方式
  - 高度非过程化：只要指出“做什么”，不需要描述“怎么做”

2.2.2 关系数据库语言的分类

- 关系代数语言
  - 用对关系的运算来表达查询要求
  - 代表：ISBL
- 关系演算语言：用谓词来表达查询要求
  - 元组关系演算语言
    - 谓词变元的基本对象是元组变量
    - 代表：APLHA, QUEL
  - 域关系演算语言
    - 谓词变元的基本对象是域变量
    - 代表：QBE
- 关系代数和关系演算是相互等价的
- 具有关系代数和关系演算双重特点的语言
  - 代表：SQL（Structured Query Language）

第二章 关系数据库

2.1 关系数据结构

2.2 关系操作

2.3 关系代数

2.4 关系的完整性

概述

- 什么是关系代数？
  - 关系数据库的一种抽象的查询语言，用对关系的运算来表达查询。
  - 代数的要素：运算对象、运算符。
    - 运算符连接运算对象和/或表达式形成表达式
  - 关系代数
    - 运算对象：关系
    - 运算符：关系运算符
- 为什么要学关系代数？
  - 已不用作商用RDBMS的查询语言，但SQL语言以关系代数作为核心
  - RDBMS处理查询时，首先将SQL查询翻译成关系代数或相似的内部表示

概 述

表2.4 关系代数运算符

运算符		含义	运算符		含义
集合运算符	∪ - ∩ ×	并 差 交 笛卡尔积	比较运算符	＞ ≥ ＜ ≤ ＝ <>	大于 大于等于 小于 小于等于 等于 不等于
专门的关系运算符	σ π ⋈ ÷	选择 投影 连接 除	逻辑运算符	¬ ∧ ∨	非 与 或

2.3 关系代数

- 概述
- 传统的集合运算
- 专门的关系运算

1. 并（Union）

- R和S
  - 具有相同的目n（即两个关系都有n个属性）
  - 相应的属性取自同一个域
- $R \cup S$
  - 仍为n目关系，由属于R或属于S的元组组成

$$R \cup S = \{ t | t \in R \vee t \in S \}$$

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \cup S$

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
a_1	b_3	c_2

2. 差（Difference）

- R和S
  - 具有相同的目n
  - 相应的属性取自同一个域
- $R - S$
  - 仍为n目关系，由属于R而不属于S的所有元组组成

$$R - S = \{ t \mid t \in R \wedge t \notin S \}$$

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R-S$

A	B	C
a_1	b_1	c_1

3. 交（Intersection）

- R和S
  - 具有相同的目n
  - 相应的属性取自同一个域
- $R \cap S$
  - 仍为n目关系，由既属于R又属于S的元组组成

$$R \cap S = \{ t \mid t \in R \wedge t \in S \}$$

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \cap S$

A	B	C
a_1	b_2	c_2
a_2	b_2	c_1

4. 笛卡尔积（Cartesian Product）

- 严格地讲应该是广义的笛卡尔积（Extended Cartesian Product）
- R: n目关系，k_1个元组
- S: m目关系，k_2个元组
- $R \times S$
  - 列：（$n+m$）列元组的集合
    - 元组的前n列是关系R的一个元组
    - 后m列是关系S的一个元组
  - 行：$k_1 \times k_2$个元组
    - $R \times S = \{t_r t_s \mid t_r \in R \wedge t_s \in S\}$

笛卡尔积(续)

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1

S

A	B	C
a_1	b_2	c_2
a_1	b_3	c_2
a_2	b_2	c_1

$R \times S$

$R.A$	$R.B$	$R.C$	$S.A$	$S.B$	$S.C$
a_1	b_1	c_1	a_1	b_2	c_2
a_1	b_1	c_1	a_1	b_3	c_2
a_1	b_1	c_1	a_2	b_2	c_1
a_1	b_2	c_2	a_1	b_2	c_2
a_1	b_2	c_2	a_1	b_3	c_2
a_1	b_2	c_2	a_2	b_2	c_1
a_2	b_2	c_1	a_1	b_2	c_2
a_2	b_2	c_1	a_1	b_3	c_2
a_2	b_2	c_1	a_2	b_2	c_1

2.3 关系代数

- 概述
- 传统的集合运算
- 专门的关系运算

示例数据库

student

Sno	Sname	Ssex	Sage	Sdept
95001	李勇	男	20	CS
95002	刘晨	女	19	IS
95003	王敏	女	18	MA
95004	张立	男	19	IS

示例数据库

Course

Cno	Cname	Cpno	Ccredit
1	数据库	5	4
2	数学		2
3	信息系统	1	4
4	操作系统	6	3
5	数据结构	7	4
6	数据处理		2
7	PASCAL语言	6	4

示例数据库

SC

Sno	Cno	Grade
95001	1	92
95001	2	85
95001	3	88
95002	2	90
95002	3	80

选择运算

- 在关系R中选择满足给定条件的元组
  - $\sigma_F(R)=\{t \mid t \in R \wedge F(t) = \text{‘真’}\}$
- F是选择的条件，$\forall t \in R$， F(t)要么为真，要么为假
  - F的形式：由逻辑运算符（ $\wedge$，$\vee$，$\neg$）连接关系表达式而成
  - 关系表达式：X θ Y
    - X，Y是属性名、常量、或简单函数
    - θ是比较算符，$\theta \in \{>, \geq, <, \leq, =, \neq\}$

选择运算

R

A	B	C
3	6	7
2	5	7
7	2	3
4	4	3

$\sigma_{A<5}(R)$

A	B	C
3	6	7
2	5	7
4	4	3

$\sigma_{A<5 \wedge C=7}(R)$

A	B	C
3	6	7
2	5	7

选择运算示例

- 找年龄不小于20的男学生

$\sigma_{AGE\geq 20 \wedge SEX='男'}$（Student）

Sno	Sname	Ssex	Sage	Sdept
95001	李勇	男	20	CS
95002	刘晨	女	19	IS
95003	王敏	女	18	MA
95004	张立	男	19	IS

查找结果

Sno	Sname	Ssex	Sage	Sdept
95001	李勇	男	20	CS

选择运算示例

- 查找信息系（IS系）的全体学生

$$\sigma_{Sdept='IS'}(Student)$$

Sno	Sname	Ssex	Sage	Sdept
95001	李勇	男	20	CS
95002	刘晨	女	19	IS
95003	王敏	女	18	MA
95004	张立	男	19	IS

查找结果

Sno	Sname	Ssex	Sage	Sdept
95002	刘晨	女	19	IS
95004	张立	男	19	IS

投影

- 从关系R中取若干列组成新的关系（从列的角度）

$$\Pi_A(R) = \{ t[A] \mid t \in R \}, A \subseteq R$$

其中A为R的属性列

  - 投影的结果中要去掉相同的行

R

A	B	C
a	b	c
d	e	f
c	b	c

$\Pi_{B,C}(R)$

B	C
b	c
e	f

投影示例

- 给出所有学生的姓名和年龄

$$\Pi_{Sname,\ Sage}(S)$$

Sno	Sname	Ssex	Sage	Sdept
95001	李勇	男	20	CS
95002	刘晨	女	19	IS
95003	王敏	女	18	MA
95004	张立	男	19	IS

sname	sage
李勇	20
刘晨	19
王敏	18
张立	19

投影示例

- 找95001号学生所选修的课程号

$$\Pi_{Cno}(\sigma_{Sno=95001}(SC))$$

Sno	Cno	Grade
95001	1	92
95001	2	85
95001	3	88
95002	2	90
95002	3	80

cno
1
2
3

连接

- 连接操作是从两个关系的广义笛卡尔积中选择属性间满足一定条件的元组。通常写为：

$$R \underset{A\,\theta\,B}{\bowtie} S = \sigma_{R.A\,\theta\,S.B}(R \times S)$$

- A,B为R和S上度数相等且可比的属性列，θ为比较运算符

连接

R

A	B	C
1	2	3
4	5	6
7	8	9

S

D	E
3	1
6	2

$R \bowtie S$
$B < D$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

等值连接

- θ为“＝”的连接运算称为等值连接
  - 等值连接的含义：从关系R与S的广义笛卡尔积中选取A、B属性值相等的那些元组，即等值连接为：

$$R \underset{A=B}{\bowtie} S = \{ \ \widehat{t_r t_s} \mid t_r \in R \wedge t_s \in S \wedge t_r[A] = t_s[B] \}$$

自然连接

- 若R和S具有相同的属性组（来自相同的域，表示相同的含义），且连接的运算符θ为“=”，并且在连接的结果中去掉重复的属性组，这种连接称为自然连接。
- 记为：

$$R \bowtie S$$

  - 当R与S无相同属性时，$R \bowtie S = R \times S$

自然连接

等值连接

A	B	C	B	E
a1	b1	5	b1	3
a1	b2	6	b2	7
a2	b3	8	b3	10
a2	b3	8	b3	2

r

A	B	C
a1	b1	5
a1	b2	6
a2	b3	8
a2	b4	12

s

B	E
b1	3
b2	7
b3	10
b3	2
b5	2

自然连接

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2

连接(续)

- 外连接
  - 如果把舍弃的元组也保存在结果关系中，而在其他属性上填空值(Null)，这种连接就叫做外连接（OUTER JOIN）。
- 左外连接
  - 如果只把左边关系*R*中要舍弃的元组保留就叫做左外连接(LEFT OUTER JOIN或LEFT JOIN)
- 右外连接
  - 如果只把右边关系*S*中要舍弃的元组保留就叫做右外连接(RIGHT OUTER JOIN或RIGHT JOIN)。

r

A	*B*	*C*
a1	*b1*	*5*
a1	*b2*	*6*
a2	*b3*	*8*
a2	*b4*	*12*

s

B	*E*
b1	*3*
b2	*7*
b3	*10*
b3	*2*
b5	*2*

A	*B*	*C*	*E*
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
a_2	b_4	12	NULL
NULL	b_5	NULL	2

(a) 外连接

A	*B*	*C*	*E*
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
a_2	b_4	12	NULL

(b) 左外连接

A	*B*	*C*	*E*
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2
NULL	b_5	NULL	2

(c) 右外连接

课堂练习：写出关系代数表达式和查询结果

- 查询每个同学的基本信息及选修的课程号和成绩

  Student ⋈ SC
- 查询李勇同学选修的课程号

  $\prod_{Cno}(\sigma_{Sname=李勇}(Student) \bowtie SC)$
- 查询李勇同学没选的课程号

  $\prod_{Cno}(Course) - \prod_{Cno}(\sigma_{Sname=李勇}(Student) \bowtie SC)$

Sno	Sname	Ssex	Sage	Sdept
95001	李勇	男	20	CS
95002	刘晨	女	19	IS
95003	王敏	女	18	MA
95004	张立	男	19	IS

Sno	Cno	Grade
95001	1	92
95001	2	85
95001	3	88
95002	2	90
95002	3	80

思考题

- 同样是对两个关系进行合并操作，**连接运算**和**集合交/并运算**有何区别？

综合举例：学生-课程数据库

- Student(Sno, Sname, Ssex, Sage, Sdept)
- Course(Cno, Cname, Cpno, Ccredit)
- SC(Sno, Cno, Grade)

[例8] 查询选修了2号课程的学生的学号。

π_{Sno}（$\sigma_{Cno='2'}$（SC））＝｛200215121，200215122｝

[例9] 查询至少选修了一门其直接先行课为5号课程的学生姓名

$\pi_{Sname}(\sigma_{Cpno='5'}(Course \triangleright\triangleleft SC \triangleright\triangleleft Student))$

或 $\pi_{Sname}(\sigma_{Cpno='5'}(Course) \triangleright\triangleleft SC \triangleright\triangleleft \pi_{Sno，Sname}(Student))$

或 $\pi_{Sname}(\pi_{Sno}(\sigma_{Cpno='5'}(Course) \triangleright\triangleleft SC) \triangleright\triangleleft \pi_{Sno，Sname}(Student))$

综合举例(续)

- 计算机产品数据库
  - Product(maker,model, type)
  - PC(model, speed, ram, hd, price)
  - Laptop(model, speed, ram, hd, screen, price)

Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649

Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500

Product

maker	model	type
A	1001	pc
A	1002	pc
A	2004	laptop
B	1003	pc
B	2005	laptop
C	1004	pc

PC

model	speed	ram	hd	price
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649

Laptop

model	speed	ram	hd	screen	price
2004	2.00	512	60	13.3	1150
2005	2.16	1024	120	17.0	2500

- 哪些PC型号的速度不小于2.8？

  $\pi_{model}(\sigma_{speed \geq 2.8}(PC))$
- 哪些制造商生产的笔记本的硬盘容量不小于100GB？

  $\pi_{maker}(Product \triangleright\triangleleft \sigma_{hd \geq 100}(Laptop))$
- 查询制造商B生产的所有类型产品的型号和价格。

  $\pi_{model,price}(\sigma_{maker=B}(Product) \triangleright\triangleleft (\pi_{model, price}(PC) \cup \pi_{model, price}(Laptop))$

作业

- 计算机产品数据库，写出关系代数表达式：
  - 查询所有速度在2.0以上的产品（包括PC和笔记本）的型号和价格
  - 查询所有只卖笔记本而不卖PC的制造商。