

第3章 关系数据模型

课程知识结构

Chp.1 数据库系统概述

Chp.2 数据库系统体系结构

Chp.3 关系数据模型

Chp.9 完整性

Chp.4 SQL

Chp.6 关系数据库模式设计

Chp.10 安全性

Chp.5 PL/SQL

Chp.7 数据库设计

Chp.11 事务与恢复

Chp.8 数据库应用系统设计

Chp.12 并发控制

Chp.13 高级主题

本章主要内容

- 数据模型的概念
- 关系数据模型的概念
- 关系数据模型的形式化定义
- 关系模型的三类完整性规则
- 关系代数

一、数据模型

- 使用数据库技术，首先必须把现实世界中的事物表示为计算机能够处理的数据
- 模型是对现实世界特征的抽象
 - 如建筑模型、沙盘模型等
- 数据模型是对现实世界**数据特征**的抽象
- 数据模型的定义
 - 描述现实世界**实体、实体间联系以及数据语义和一致性约束**的模型

1、数据模型的分类

■ 根据模型应用的不同目的

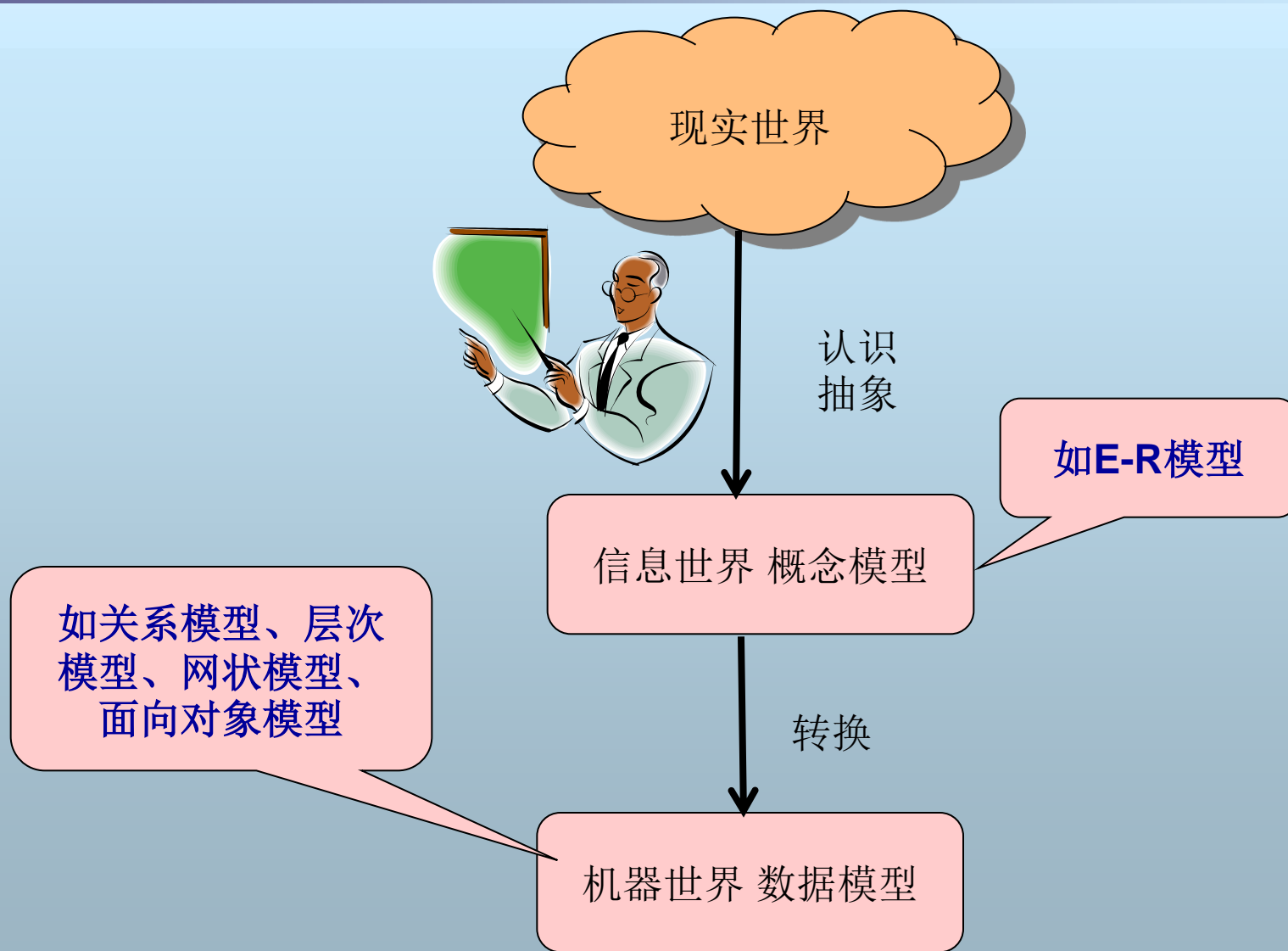
● 概念数据模型（概念模型）

- ◆ 按用户的观点对数据进行建模，强调语义表达功能
- ◆ 独立于计算机系统和DBMS
- ◆ 主要用于数据库的概念设计

● 结构数据模型（数据模型）

- ◆ 按计算机系统的观点对数据进行建模，直接面向数据库的逻辑结构
- ◆ 与计算机系统和DBMS相关
 - 回顾：DBMS支持某种数据模型
- ◆ 有严格的形式化定义，以便于在计算机系统中实现

2、数据抽象的层次



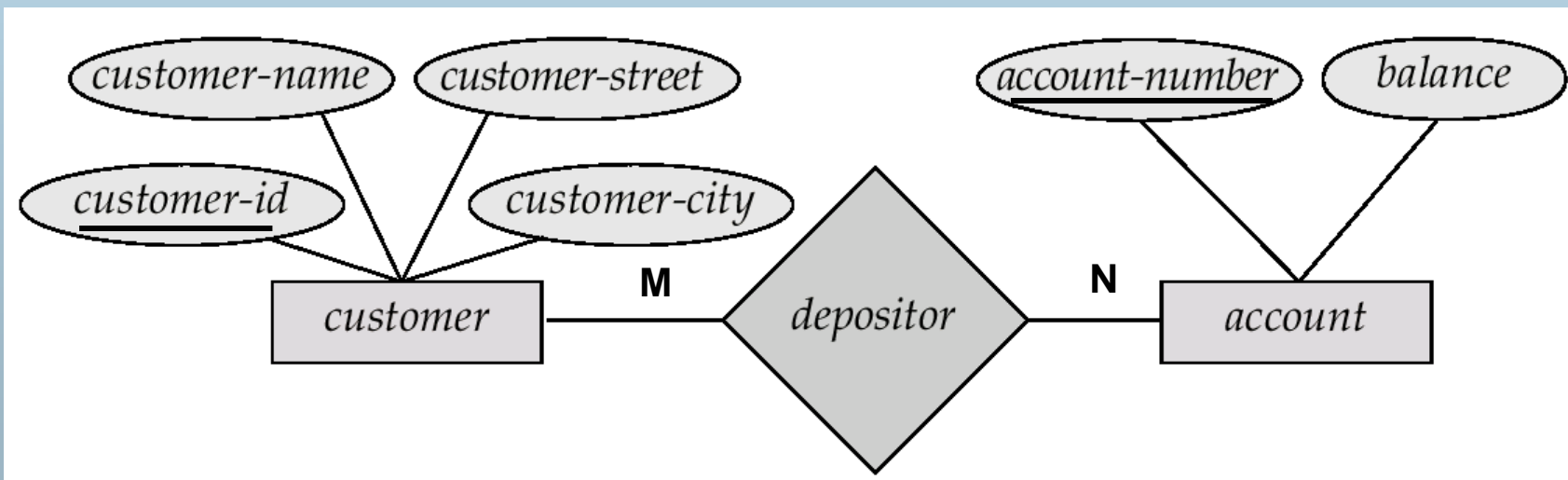
3、数据模型的例子

■ 现实世界

- 客户存款

■ 信息世界

- 概念模型（E-R模型）



3、数据模型的例子

■ 机器世界

● 数据模型 (关系模型)

<i>customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

<i>account-number</i>	<i>balance</i>
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

<i>customer-id</i>	<i>account-number</i>
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

4、数据模型的三要素

■ 数据结构

- 现实世界实体及实体间联系的表示和实现

■ 数据操作

- 数据检索和更新的实现

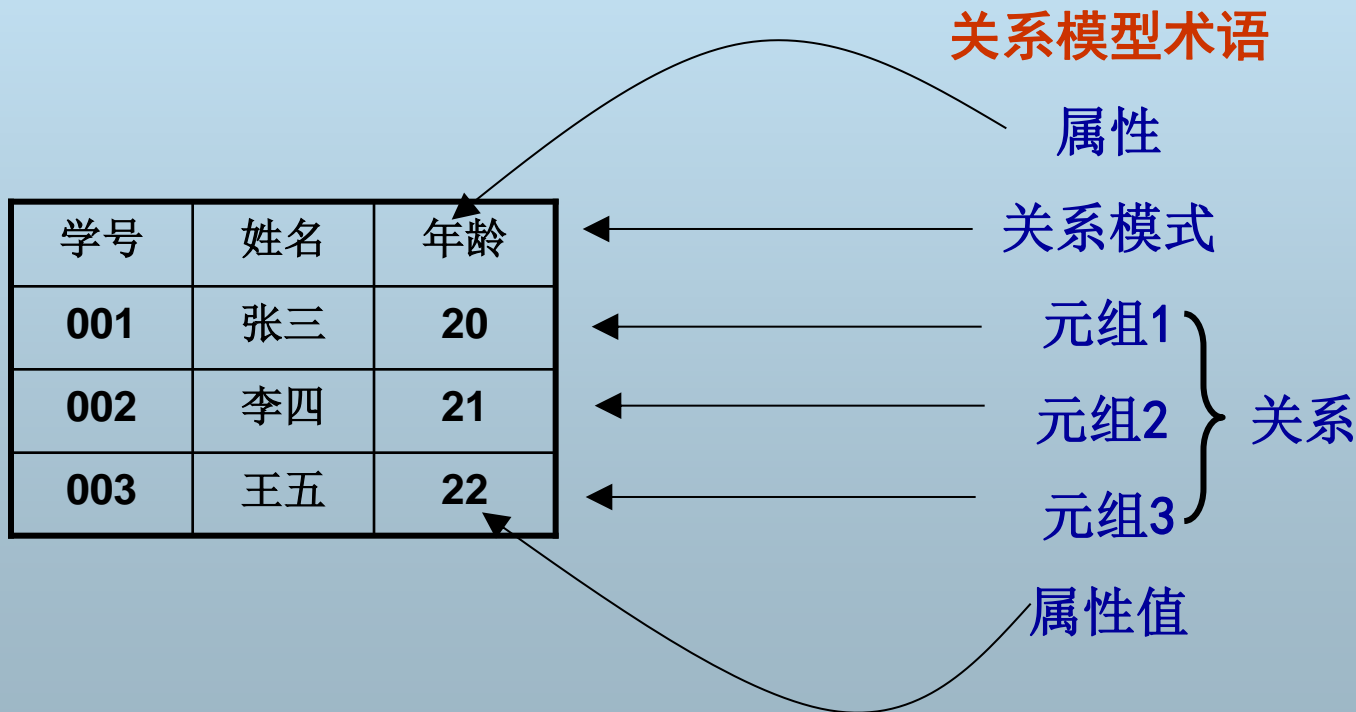
■ 数据的完整性约束

- 数据及数据间联系应具有制约和依赖规则
- 如：一个系可有多个学生，一个学生只属于一个系

二、关系模型概论

■ 关系模型

- 用二维表格结构表示**实体集**，外码表示实体间联系，三类完整性规则表示数据约束的数据模型



1、一些术语

■ 属性(Attribute)

- 二维表格的每一列称为关系的一个属性，列的数目称为度 (degree)

■ 元组(Tuple)

- 每一行称为关系的一个元组，元组的数目称为势 (cardinality)

■ 域(Domain)

- 一组具有相同数据类型的值的集合。每个属性有一个域

■ 关系(Relation)

- 元组的集合

2、关系、关系模式与关系数据库

■ 关系模式(Relation Schema)

- 关系的逻辑结构和特征的描述
- 对应于二维表格的表头
- 通常由属性集和各属性域表示，不关心域时可省略域
 - ◆ Student(Name, Age, Class)

■ 关系

- 关系模式的实例，即二维表（元组的集合）

■ 关系数据库模式(Relational Database Schema)

- 关系模式的集合

■ 关系数据库：关系数据库模式的实例

3、关系模式的形式化定义

■ 关系模式可以形式化定义为：

● $R(U, D, dom, F)$

- ◆ R 为关系模式名， U 是一个属性集， D 是 U 中属性的值所来自的域， Dom 是属性向域的映射集合， F 是属性间的依赖关系

■ 例：Student关系模式的定义

● $Student(U, D, dom, F)$

- ◆ $U = \{sno, name, age\}$
- ◆ $D = \{CHAR, INT\}$
- ◆ $Dom = \{dom(sno) = dom(name) = CHAR, dom(age) = INT\}$
- ◆ $F = \{sno \rightarrow name, sno \rightarrow age\}$

■ 关系模式通常简写为 $R(U)$ ，或 $R(A_1, A_2, \dots, A_n)$

4、超码、候选码和主码

■ 超码 (Super Key)

- 在关系模式中能唯一标识一个元组的属性集称为关系模式的超码

■ 候选码 (Candidate Key)

- 不含多余属性的超码
- 包含在任何一个候选码中的属性称为**主属性** (Primary Attribute)
- 不包含在任何一个候选码中的属性称为**非主属性** (Non-prime Attribute)

■ 主码 (Primary Key)

- 用户选作元组标识的一个候选码称为主码，其余的候选码称为**替换码** (Alternate Key)

4、超码、候选码和主码

■ Student(Sno, Name, Age, LibraryID)

● 超码

- ◆ (sno,name)

- ◆ (libraryID, name) ...

● 候选码

- ◆ Sno

- ◆ LibraryID

● 主码

- ◆ 若选sno, 则sno为主码, libraryID为替换码

- ◆ 若选sno, 则libraryID 为主码, sno为替换码

5、关系的性质

■ 一个关系是一个规范化的二维表格

- 属性值不可分解

- ◆ 不允许表中有表

- 元组不可重复

- ◆ 因此一个关系模式至少存在一个候选码

- 没有行序，即元组之间无序

- ◆ 关系是元组的集合

- 没有列序，即属性之间无序

- ◆ 关系模式是属性的集合

学号	课程
001	数据库
002	{数据库,C语言}

更新二义性：若001现也选了C语言，则DBMS在更新时面临二义性：

- 1.修改第1个元组的课程
- 2.修改第2个元组的学号

6、关系模型的形式化定义

■ 数据结构

- 关系：数据库中全部数据及数据间联系都以关系来表示

■ 数据操作

- 关系运算

- ◆ 关系代数

- ◆ *关系演算（元组关系演算、域关系演算）

■ 数据的完整性约束

- 关系模型的三类完整性规则


7、关系模型的三类完整性规则

- 关系数据库的数据和操作必须遵循的规则
 - 实体完整性(Entity Integrity)
 - 参照完整性(Referential Integrity)
 - 用户自定义完整性(User-Defined Integrity)

(1) 实体完整性

■ 关系模式R的主码不可为空

- 指组成主码的所有属性均不可取空值

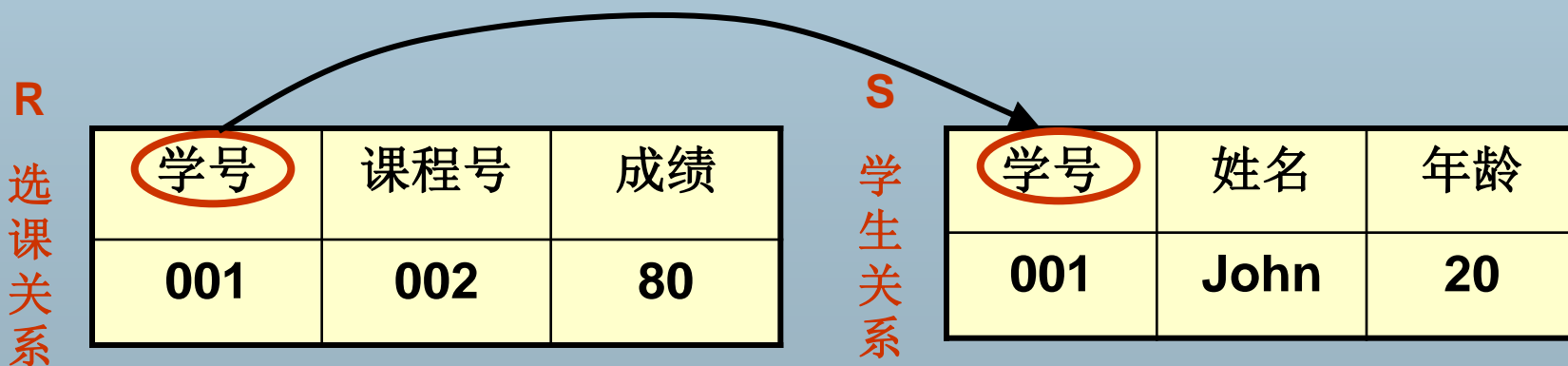


<u>学号</u>	<u>课程号</u>	成绩
S001	C001	80
S001		90
		80

(2) 参照完整性

■ 外码 (Foreign Key)

- 关系模式R的外码是它的一个属性集FK，满足：
 - ◆ 存在带有候选码CK的关系模式S，且
 - ◆ R的任一非空FK值都在S的CK中有一个相同的值
- S称为被参照关系 (Referenced Relation)，R称为参照关系 (Referential Relation)



(2) 参照完整性

■ 参照关系R的任一个外码值必须

- 等于被参照关系S中所参照的候选码的某个值
- 或者为空

R
选课关系

学号	课程号	成绩
001	002	80

002

S
学生关系

学号	姓名	专业号
001	John	

专业号	专业名	学科类别
0020	PHY	1

(3) 用户自定义完整性

- 针对某一具体数据的约束条件，反映某一具体应用所涉及的数据必须满足的特殊语义
- 由应用环境决定

学号	课程号	成绩
001	002	80

成绩 ≥ 0 and 成绩 ≤ 100



回顾：关系模型的形式化定义

■ 数据结构

- 关系：数据库中全部数据及数据间联系都以关系来表示

■ 数据的完整性约束

- 关系模型的三类完整性规则：实体完整性、参照完整性、用户自定义完整性

■ 数据操作

- 关系运算
 - ◆ 关系代数
 - ◆ 关系演算（元组关系演算、域关系演算）