

数据库系统概论

An Introduction to Database System

第二章 关系数据库

中国人民大学信息学院

关系数据库简介

- IBM公司的E.F.Codd于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



第二章 关系数据库

1 关系模型（2个视频）

- 关系数据结构
- 关系完整性约束

2 关系代数（3个视频）

- 传统集合操作
- 关系特有操作
- 综合例子

3 关系演算（2个视频）



关系模型

(关系数据结构)



❖ 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



关系数据结构

学生

学号	姓名	专业	所在系
201501001	张三	计算机	计算机	



关系数据结构

域：取值范围

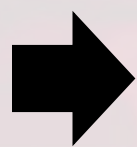
属性

值

学号	姓名	专业	所在系
201501001	张三	计算机	计算机	

关系型：模式

关系实例



关系



关系模式



关系数据库

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'}
-



笛卡尔积 (Cartesian Product)

❖ 域的笛卡尔积空间

给定一组域 D_1, D_2, \dots, D_n , 允许其中某些域是相同的。

D_1, D_2, \dots, D_n 的笛卡尔积空间为:

$$D_1 \times D_2 \times \dots \times D_n =$$

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

■ 所有域取值的任意组合

■ 笛卡尔积可以看着是关系的“域”。



笛卡尔积（续）

❖ 基数（Cardinal number）

- 若 D_i ($i=1, 2, \dots, n$) 为有限集，其基数为 m_i ($i=1, 2, \dots, n$)，则 $D_1 \times D_2 \times \dots \times D_n$ 的基数 M 为：

$$M = \prod_{i=1}^n m_i$$

❖ 笛卡尔积的表示方法

- 笛卡尔积可表示为一张二维表
- 表中的每行对应一个元组，表中的每列对应一个域



笛卡尔积（续）

例如，给出3个域：

- ❖ **D1=导师集合SUPERVISOR= { 张清玫, 刘逸 }**
- ❖ **D2=专业集合SPECIALITY= { 计算机专业, 信息专业 }**
- ❖ **D3=研究生集合POSTGRADUATE= { 李勇, 刘晨, 王敏 }**
- ❖ **D1, D2, D3的笛卡尔积为**



笛卡尔积（续）

❖ $D1 \times D2 \times D3 = \{$

(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) }

❖ 基数为 $2 \times 2 \times 3 = 12$



笛卡尔积（续）

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



1. 关系 (Relation)

(1) 关系

$D_1 \times D_2 \times \dots \times D_n$ 的子集叫作在域 D_1, D_2, \dots, D_n 上的关系, 表示为

$$R(D_1, D_2, \dots, D_n)$$

- R : 关系名
- n : 关系的目或度 (Degree)



关系（续）

（2）元组

关系中的每个元素（ d_1, d_2, \dots, d_n ）叫作一个 n 元组（**n-tuple**）或简称元组，通常用 t 表示。

（3）属性

- 关系中不同列称为属性（**Attribute**），每个属性有一个名字
- n 目关系必有 n 个属性



关系（续）

（4）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**）



关系（续）

码（续）

- 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**）



关系（续）

（5）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）

（6）基本关系的性质

- ① 列是同质的（**Homogeneous**），即来自同一个属性
- ② 不同的列可出自同一个域
- 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



⑥ 分量必须取原子值

这是规范条件中最基本的一条

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

小表

关系（续）

❖ D_1, D_2, \dots, 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）

SUPERVISOR	SPECIALITY	POSTGRADUATE
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



1.2 关系模式

1. 什么是关系模式？
2. 如何定义关系模式？
3. 关系模式与关系的关系



1. 什么是关系模式

❖ 关系模式（**Relation Schema**）是型、关系是值

❖ 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

■ 完整性约束条件



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 U 中属性所来自的域

DOM 属性向域的映象集合

F 属性间数据的依赖关系的集合



定义关系模式（续）

关系模式通常可以简记为

$R(U)$ 或 $R(A_1, A_2, \dots, A_n)$

- R : 关系名
- A_1, A_2, \dots, 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



3. 关系模式与关系的“关系”

❖ “型”与“值”的关系

- 关系模式：是对关系的描述，是静态的、稳定的
- 关系：是关系模式在某一时刻的状态或内容，是动态的、随时间不断变化的

❖ 关系模式和关系往往笼统称为关系

通过上下文加以区别



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，是对关系数据库的描述
- 关系数据库的值: 关系模式在某一时刻对应的关系的集合，通常称为关系数据库





关系模型

(关系完整性约束)



关系的三类完整性约束

❖ 实体完整性和参照完整性

- 关系模型必须满足的完整性约束条件称为关系的两个
不变性，应该由关系系统自动支持

❖ 用户定义的完整性

- 应用领域需要遵循的约束条件，体现了具体领域中的
语义约束



关系的三类完整性约束

1 实体完整性

2 参照完整性

3 用户定义的完整性



1 实体完整性

❖ 实体完整性 (Entity Integrity)

- 关系的主属性不能取空值
- 空值就是“不知道”或“不存在”或“无意义”的值

例：

选修 (学号, 课程号, 成绩)

“学号、课程号”为主码

“学号”和“课程号”两个属性都不能取空值



实体完整性（续）

❖ 说明

（1）实体完整性规则是针对基本关系而言的。

一个基本表通常对应现实世界的一个实体集。

（2）现实世界中的实体是可区分的，即它们具有某种唯一性标识。

（3）关系模型中以主码作为唯一性标识。

（4）主码中的属性即主属性不能取空值。

主属性取空值，就说明存在某个不可标识的实体，即存在不可区分的实体，这与第（2）点相矛盾，因此这个规则称为实体完整性



2 关系的完整性

1 实体完整性

2 参照完整性

3 用户定义的完整性



1. 关系间的引用

- ❖ 在关系模型中实体及实体间的联系都是用关系来描述的，自然存在着关系与关系间的引用。

[例2.1] 学生实体、专业实体

主码

学生（学号，姓名，性别，专业号，年龄）

专业（专业号，专业名）

主码

- ❖ 学生关系引用了专业关系的主码“专业号”。
- ❖ 学生关系中的“专业号”值必须是确实存在的专业的专业号

关系间的引用（续）

例[2.2] 学生、课程、学生与课程之间的多对多联系

学生（学号，姓名，性别，专业号，年龄）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）



关系间的引用（续）

例[2.3] 学生实体及其内部的一对多联系

学生（学号，姓名，性别，专业号，年龄，班长）

学号	姓名	性别	专业号	年龄	班长
801	张三	女	01	19	802
802	李四	男	01	20	
803	王五	男	01	20	802
804	赵六	女	02	20	805
805	钱七	男	02	19	

- ❖ “学号”是主码，“班长”是外码，它引用了本关系的“学号”
- ❖ “班长”必须是确实存在的学生的学号



2. 外码 (Foreign Key)

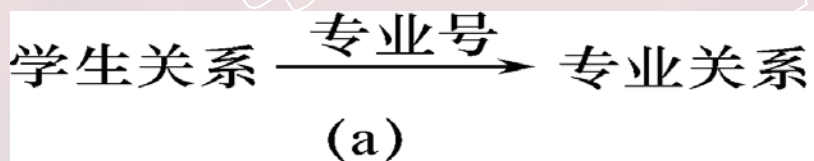
- ❖ 设 F 是基本关系 R 的一个或一组属性，但不是关系 R 的码。
如果 F 与基本关系 S 的主码 K_s 相对应，则称 F 是 R 的**外码**
- ❖ 基本关系 R 称为**参照关系** (Referencing Relation)
- ❖ 基本关系 S 称为**被参照关系** (Referenced Relation)
或**目标关系** (Target Relation)



外码（续）

❖ [例2.1]中学生关系的“专业号”与专业关系的主码“专业号”相对应

- “专业号”属性是学生关系的外码
- 专业关系是被参照关系，学生关系为参照关系



外码（续）

❖ [例2.2]中

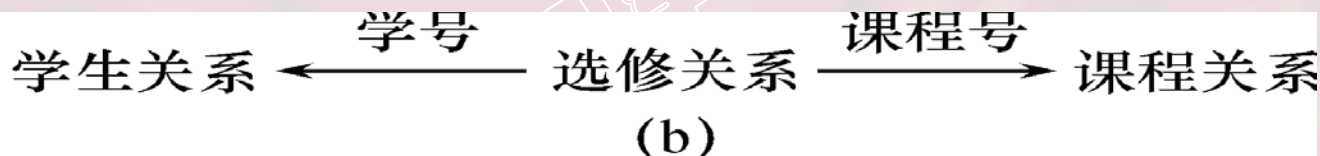
选修关系的“学号”与学生关系的主码“学号”相对应

选修关系的“课程号”与课程关系的主码“课程号”相对应

■ “学号”和“课程号”是选修关系的外码

■ 学生关系和课程关系均为被参照关系

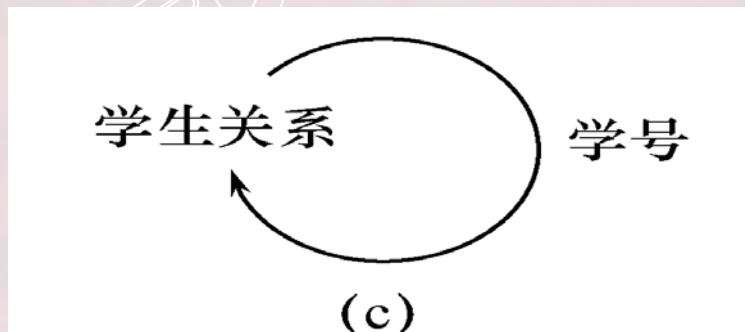
■ 选修关系为参照关系



外码（续）

❖ [例2.3]中“班长”与本身的主码“学号”相对应

- “班长”是外码
- 学生关系既是参照关系也是被参照关系



外码（续）

- ❖ 关系 R 和 S 不一定是不同的关系
- ❖ 目标关系 S 的主码 K_s 和参照关系的外码 F 必须定义在同一个（或一组）域上
- ❖ 外码并不一定要与相应的主码同名
当外码与相应的主码属于不同关系时，往往取相同的名字，以便于识别



3. 参照完整性规则

❖ 参照完整性规则

若属性（或属性组） F 是基本关系 R 的外码它与基本关系 S 的主码 K_s 相对应（基本关系 R 和 S 不一定是不同的关系），则对于 R 中每个元组在 F 上的值必须为：

- 或者取空值（ F 的每个属性值均为空值）
- 或者等于 S 中某个元组的主码值



参照完整性规则（续）

[例2.1]中

学生关系中每个元组的“专业号”属性只取两类值：

- （1）空值，表示尚未给该学生分配专业
- （2）非空值，这时该值必须是专业关系中某个元组的“专业号”值，表示该学生不可能分配一个不存在的专业



参照完整性规则（续）

[例2.2] 中

选修（学号，课程号，成绩）

“学号”和“课程号”可能的取值：

- （1）选修关系中的主属性，不能取空值
- （2）只能取相应被参照关系中已经存在的主码值



参照完整性规则（续）

[例2.3] 中

学生（学号，姓名，性别，专业号，年龄，班长）

“班长”属性值可以取两类值：

- （1）空值，表示该学生所在班级尚未选出班长
- （2）非空值，该值必须是本关系中某个元组的学号值



3 关系的完整性

1 实体完整性

2 参照完整性

3 用户定义的完整性



3 用户定义的完整性

- ❖ 针对某一具体关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的语义要求
- ❖ 关系模型应提供定义和检验这类完整性的机制，以便使用统一的系统的方法处理它们，而不需由应用程序承担这一功能



用户定义的完整性（续）

例:

课程（课程号，课程名，学分）

- “课程号” 属性必须取唯一值
- 非主属性 “课程名” 也不能取空值
- “学分” 属性只能取值{1, 2, 3, 4}



