

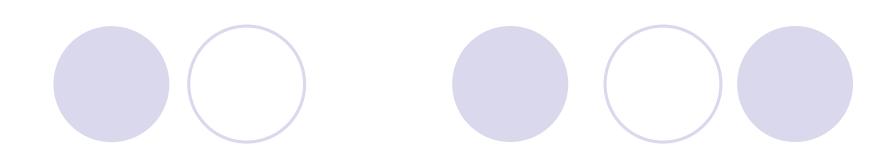
数据库系统

Database System

主讲: 张仲楠 教授

Email: zhongnan_zhang@xmu.edu.cn

Office: 海韵A416



数据库系统 Database System

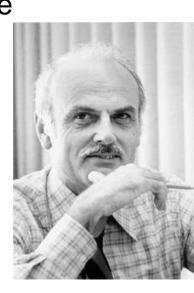
第二章 关系数据库

关系数据库简介

- 关系数据库系统是支持关系模型的数据库系统
- 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", 《Communications 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 关系的完整性
- 2.4 关系代数
- *2.5 关系演算
- 2.6 小结

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
- 2.1.4 关系模型的存储结构

- 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
 - 〇从用户角度,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
 - ○从集合论的角度给出关系数据结构的形式化定义

- 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 0≤n≤100
 - ▶长度指定长度的字符串集合 char(10)
 - >{'男','女'}

2. 笛卡尔积(Cartesian Product)

● 笛卡尔积 × : 一种域上的**集合运算** 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:

$$D_1 \times D_2 \times ... \times D_n =$$
{ $(d_1, d_2, ..., d_n) \mid d_i \in D_i, i=1, 2, ..., n$ }

- ○所有域的所有取值的全组合
- ○不能有重复的组合
- [例]: D_1 ={张清枚,刘逸}, D_2 ={计算机专业,信息专业}, D_3 ={李勇,刘晨,王敏}

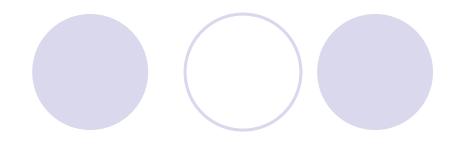
表 2.1 $D_1 imes D_2 imes D_3$

D_1	D_2	D_3
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

笛卡尔积 (续)

- 元组(Tuple)
 - 〇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)
 - 〇笛卡尔积元素(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

笛卡尔积(续)



- 基数(Cardinal number)
 - 一个域允许的**不同取值**的个数称为这个域的<mark>基数</mark>
 - 〇 若 D_i (i=1, 2, ..., n) 为<mark>有限集</mark>,其基数为 m_i (i=1, 2, ..., n) ,则 $D_1 \times D_2 \times ... \times D_n$ 的基数M为:

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

- \bigcirc [例]: D_1 基数 m_1 =2, D_2 基数 m_2 =2, D_3 基数 m_3 =3 $D_1 \times D_2 \times ... \times D_n$ 的基数M=12
- 笛卡尔积的表示方法
 - 笛卡尔积可表示为一个二维表
 - 表中的每行对应一个元组,表中的每列对应一个域

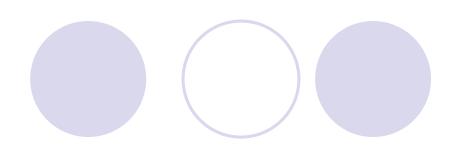
3. 关系(Relation)

1) 关系

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

$$R (D_1, D_2, ..., D_n)$$

- 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<mark>单元</mark>关系(Unary relation)

或一元关系

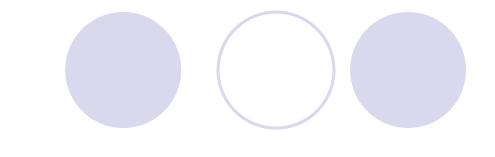
当*n*=2时,称该关系为二元关系(Binary relation)



- ○关系是笛卡儿积的有限子集,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



候选码(Candidate key)

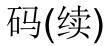
- 若关系中的某一**属性组**的值**能** 唯一地标识一个元组,而其子集 不能,则称该属性组为候选码
- 简单的情况:候选码只包含一 个属性

<u>全码(All-key)</u>

○ 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)

StudentName	Course	Grade
李勇	软件工程	90
刘晨	软件工程	80
王敏	软件工程	85
李勇	数据库	92
刘晨	数据库	95
王敏	数据库	80
李勇	操作系统	75
刘晨	操作系统	80
王敏	操作系统	88
李勇	信息安全	85
刘晨	信息安全	89
王敏	信息安全	95





主码

○若一个关系有多个候选码,则<mark>选定</mark>其中一个为**主码** (Primary key)

主属性

- ○组成**候选码的诸属性**称为<u>主属性</u>(Prime attribute)
- ○不包含在任何**侯选码**中的属性称为<u>非主属性</u> (Non-Prime attribute) 或<u>非码属性</u> (Non-key attribute)

 D_1 , D_2 , ..., D_n 的笛卡尔积的<u>某个子集</u>才有实际含义

例:表2.1的笛卡尔积中许多元组没有实际意义

取出有实际意义的元组来构造关系

关系: SAP(SUPERVISOR, SPECIALITY, POSTGRADUATE)

假设:专业与导师: 1:n, 导师与研究生: 1:n

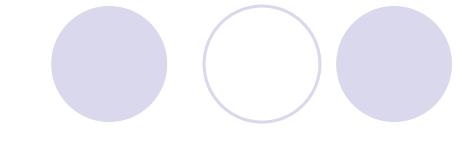
主码: POSTGRADUATE (假设研究生不会重名)

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) }



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



- ① 列是同质的(Homogeneous), 即均是同一类型数据
- 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

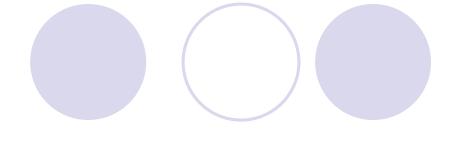
基本关系的性质(续)

⑥ 分量必须取原子值 这是规范条件中最基本的一条 (第一范式) 规范化的关系简称为范式(Normal Form, NF)

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
- 2.1.4 关系模型的存储结构

2.1.2 关系模式

- 1. 什么是关系模式
- 2. 定义关系模式
- 3. 关系模式与关系

1. 什么是关系模式

- 关系模式(Relation Schema)是型
- 关系是值,是元组的集合
- 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
 - 属性的取值范围, Ex: Grade∈[0, 100]
 - ●属性间的数据依赖关系, Ex: 属性"系名"、 "系主任名",后者函数依赖于前者

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

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

定义关系模式(续)

[例]关系Student:

学号	姓名	性别	系名
0101	张	男	CS
0102	李	女	CS
0203	赵	男	MA
• • •	•••	•••	•••

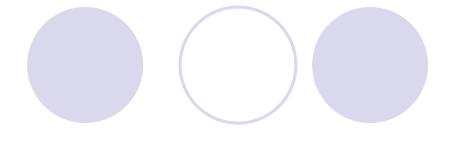
```
其中 D1={长度为4的、由'0'-'9'字符构成的字符串}
D2={长度不超过10的字符串}
D3={'男','女'}
D4={长度为2 的字符串}
U={学号,姓名,性别,系名} D={D1,D2,D3,D4}
Dom(学号)=D1, Dom(姓名)=D2, Dom(性别)=D3, Dom(系名)=D4
该关系简记为: Student(学号,姓名,性别,系名)
```

3. 关系模式与关系

- 关系模式
 - ■对关系的描述
 - ■静态的、稳定的
- · 关系
 - ■关系模式在某一时刻的状态或内容
 - ■动态的、随时间不断变化的
- 关系模式和关系往往统称为关系

通过上下文加以区别

2.1 关系数据结构

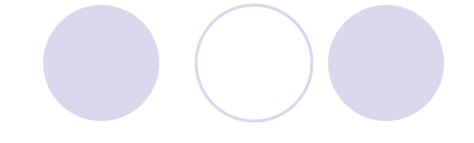


- 2.1.1 关系
- 2.1.2 关系模式
- 2.1.3 关系数据库
- 2.1.4 关系模型的存储结构

2.1.3 关系数据库

- 针对特定应用,所有关系的集合构成一个RDB
- <u>关系数据库的型:</u>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
- <u>关系数据库的值:</u>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

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
- 2.1.4 关系模型的存储结构

2.1.4 关系模型的存储结构

- 表是关系数据的逻辑模型
- 物理组织仍然要使用文件
 - ○一个表对应一个系统文件
 - 〇由若干个大文件存储,存储表、索引等

第二章 关系数据库

- 2.1 关系模型概述
- 2.2 关系操作
- 2.3 关系的完整性
- 2.4 关系代数
- *2.5 关系演算
- 2.6 小结

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、笛卡尔积是5种基本操作
- 关系操作的特点
 - ○集合操作:对象和结果都是集合,一次一集合的方式

2.2.2 关系数据语言的分类

- 关系代数语言
 - 用对关系的**运算**来表达查询要求,代数方式
 - ○代表: ISBL
- 关系演算语言:用谓词来表达查询要求,逻辑方式
 - ○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
- 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 关系数据结构及形式化定义
- 2.2 关系操作
- 2.3 关系的完整性
- 2.4 关系代数
- *2.5 关系演算
- 2.6 小结

2.3 关系的完整性

- 完整性规则是对关系的某种约束
- 任何关系的值在任何时刻都要满足这些语义约束
- 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<u>不变性</u>,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

2.3 关系的完整性



- 2.3.2 参照完整性
- 2.3.3 用户定义的完整性

2.3.1 实体完整性

每个元组应该是可区分的、唯一的,用实体完整 性保证

规则2.1 实体完整性规则(Entity Integrity)

若属性A是基本关系R的主属性,则属性A不能取空值例:

SC(Sno, Cno, Grade)

主码: (Sno, Cno) 不能取空值

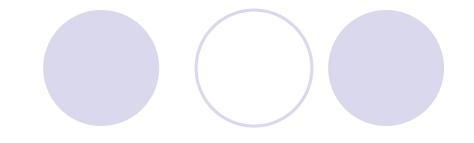
实体完整性(续)

实体完整性规则的说明

- (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) 主码中的属性是主属性,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

2.3关系的完整性



- 2.3.1 实体完整性
- 2.3.2 参照完整性
- 2.3.3 用户定义的完整性

2.3.2 参照完整性

- 1. 关系间的引用
- 2. 外码
- 3. 参照完整性规则

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, 因此可能存在着关系与关系间的引用。

例1 学生实体、专业实体

学生(<u>学号</u>,姓名,性别,专业号,年龄)

专业(<u>专业号</u>,专业名)

- *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,即专业关系中有该专业的记录。

关系间的引用(续)

例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	802
803	王五	男	01	20	802
804	赵六	女	02	20	805
805	钱七	男	02	19	805

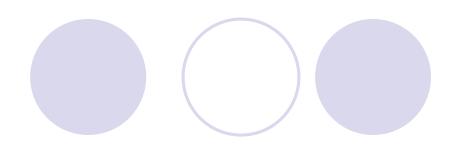
- ❖ "班长"引用了本关系的"学号"
- ❖"班长"必须是确实存在的学生的学号

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是基本关系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)

$$R(K_r, F, ...), S(K_s, ...)$$
 $R \xrightarrow{F} S$

- [例1]: 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

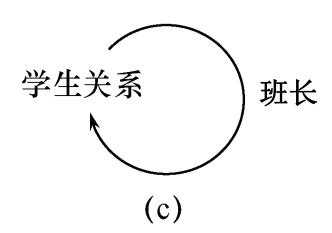


● [例2]:

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

- [例3]: "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



- 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
- 外码并不一定要与被参照的主码同名
- 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

规则2.2 参照完整性规则(Referential Integrity)

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:

- \blacksquare 或者 \mathbf{n} 空值(若 \mathbf{r} 是属性组, \mathbf{r} 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1]:

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]:

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3]:

学生(<u>学号</u>,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

关系的完整性(续)

- 2.3.1 实体完整性
- 2.3.2 参照完整性
- 2.3.3 用户定义的完整性

2.3.3 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, 以便用统一的系统的方法处理它们,而不要由 应用程序承担这一功能

用户定义的完整性(续)

例:

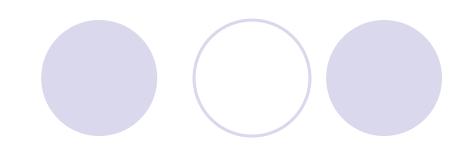
课程(课程号,课程名,学分)

- ○"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}

第二章 关系数据库

- 2.1 关系模型概述
- 2.2 关系数据结构
- 2.3 关系的完整性
- 2.4 关系代数
- *2.5 关系演算
- 2.6 小结

2.4 关系代数



2.4.1 概述

2.4.2 传统的集合运算

2.4.3 专门的关系运算

2.4.1 概述

- 属于关系操作的一种
- 关系代数是一种抽象的查询语言
- 通过对关系的运算来表达查询操作
- 运算
 - 运算符, 运算对象, 运算结果
 - 这三者称为运算的三大要素
 - 集合运算、专门的关系运算、比较运算、逻辑运算
- 运算对象、结果均为关系

概述 (续)



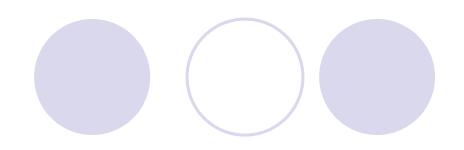
运算	符	含义	运算	符	含义
集合运算符	—	并 差 交 笛卡尔积	比较运算符	> ∧ı ∨ ∨ı ♦	大于 大于等于 小于等于 小于等于 等于 不等于

概 述(续)

表2.4 关系代数运算符(续)

运算符	含义		运算符	含义	
专门的	σ		逻辑运		非
关系运 算符	π	投影	算符	٨	
介 77 	M	连接		V	或
	•	除			

2.4 关系代数



2.4.1 概述

2.4.2 传统的集合运算

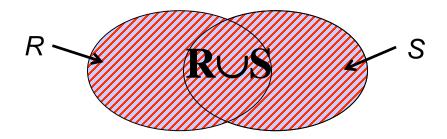
2.4.3 专门的关系运算

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

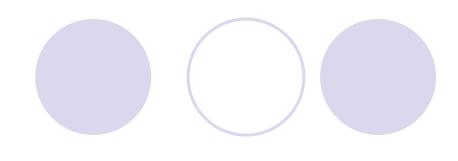
RUS

○仍为n目关系,由属于R或属于S的元组组成 $R \cup S = \{ t | t \in R \lor t \in S \}$



并(续)

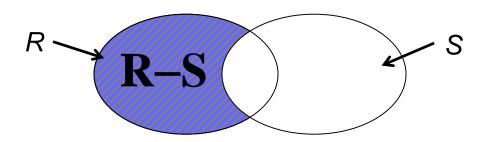
R		
A	В	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
S		
	В	С
S	ı	C
S A	В	



$R \cup S$		
A	В	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

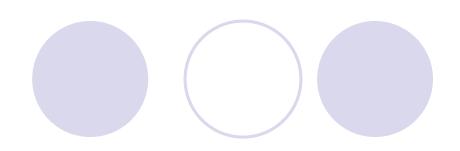
2. 差 (Difference)

- 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 $R-S=\{t|t\in R\land t\notin S\}$



差(续)

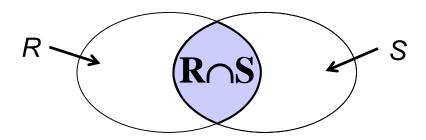
R		
A	В	С
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
S		
	В	С
S	1	C
S A	В	



R-S		
A	В	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
- \bullet $R \cap S$
 - 仍为n目关系,由既属于R又属于S的元组组成 $R \cap S = \{ t | t \in R \land t \in S \}$ $R \cap S = R (R S)$



交 (续)

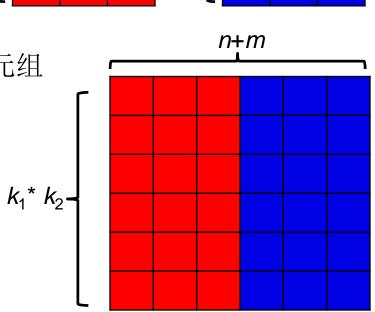
R		
A	В	C
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
S		
	В	С
S		C
S A	В	

$R \cap S$

A	В	С
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
- \bullet R \times S
 - \bigcirc 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×*k*₂个元组
 - $R \times S = \{t_r \ t_s \ | t_r \in R \land t_s \in S \}$



 k_2

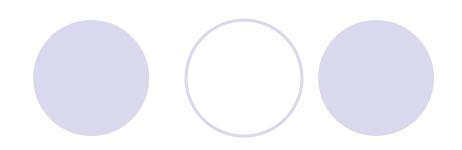
笛卡尔积(续)

R		
A	В	С
a_1	b_1	c_1
a_1	b_2	c_2
a_2	b_2	c_1
S		
A	В	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

2.4 关系代数



2.4.1 概述

2.4.2 传统的集合运算

2.4.3 专门的关系运算

2.4.2 专门的关系运算

先引入几个记号

 $(1) R, t \in R, t[A_i]$

设关系模式为 $R(A_1, A_2, ..., A_n)$

它的一个关系设为R

 $t \in R$ 表示t是R的一个元组

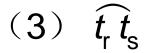
 $t[A_i]$ 则表示元组t中相应于属性 A_i 的一个分量

(2) A, t[A], \overline{A}

若 $A=\{A_{i1}, A_{i2}, ..., A_{ik}\}$, 其中 $A_{i1}, A_{i2}, ..., A_{ik}$ 是 $A_1, A_2, ..., A_n$ 中的一部分,则A称为属性列或属性组。

 $t[A]=(t[A_{i1}], t[A_{i2}], ..., t[A_{ik}])$ 表示元组t在属性列A上诸分量的集合。

A则表示 $\{A_1, A_2, ..., A_n\}$ 中去掉 $\{A_{i1}, A_{i2}, ..., A_{ik}\}$ 后剩余的属性组。



R为n目关系,S为m目关系。

 $t_r \in R$, $t_s \in S$, $t_r t_s$ 称为元组的连接。

 $\hat{t}_r t_s$ 是一个n + m列的元组,前n个分量为R中的一个n元组,后m个分量为S中的一个m元组。

(4) 象集**Z**_x

给定一个关系R(X, Z),X和Z为属性组。

当t[X]=x时,x在R中的象集(Images Set)为:

$$\mathbf{Z}_{\mathbf{x}} = \{t[\mathbf{Z}] | t \in \mathbb{R}, t[\mathbf{X}] = \mathbf{x}\}$$

它表示R中属性组X上值为x的诸元组在Z上分量的集

合

 $x=\{x_1, x_2, ...\}$,则 Z_x 可以用如下SQL获得:

Select Z₁, **Z**₂, ...

From R

Where $X_1=x_1$ and $X_2=x_2$ and ...;

7	D
I	I

x_1	Z_1
x_1	Z_2
x_1	Z_3
x_2	Z_2
x_2	Z_3
x_3	Z_1
x_3	Z_3

• x₁在R中的象集

$$Z_{x1} = \{Z1, Z2, Z3\},$$

• x,在R中的象集

$$Z_{x2} = \{Z2, Z3\},$$

 x_3 在R中的象集

$$Z_{x3} = \{Z1, Z3\}$$

象集举例

- 选择
- 投影
- 连接
- 除

4) 学生-课程数据库: 学生关系Student、课程关系Course和选修关系SC

Student

 学号	姓名	性别	年龄	所在系
Sno	Sname	Ssex	Sage	Sdept
201215121	李勇	男	20	cs
201215122	刘晨	女	19	cs
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

(a)

Course

课程号	课程名	先行课	学分
Cno	Cname	Cpno	Ccredit
1	数据库	5	4
2	数学	null	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	null	2
7	PASCAL语言	6	4

SC

	课程号	成绩
Sno	Cno	Grade
201215121	1	92
201215121	2	85
201215121	3	88
201215122	2	90
201215122	3	80

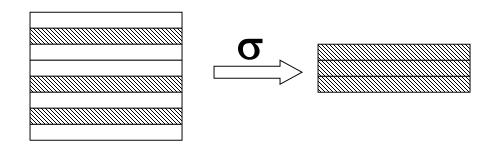
(c)

1. 选择(Selection)

- 1) 选择又称为限制(Restriction)
- 2) 选择运算符的含义
 - ○在关系R中选择满足给定条件的诸元组 $\sigma_F(R) = \{t | t \in R \land F(t) = '真'\}$
 - $\bigcirc F$: 选择条件,是一个**逻辑表达式**,基本形式为: $X_1\theta Y_1$
 - θ可以是>, ≥, <, ≤, =, <>
 - X_1, Y_1 可以是属性名、常量、简单函数
 - 可使用逻辑运算构建复杂表达式

选择(续)

● 3) 选择运算是从关系*R*中选取**使逻辑表达式** *F*为真的元组,是从行的角度进行的运算



选择(续)

[例1] 查询信息系(IS系)全体学生 $\sigma_{Sdept = 'IS'}$ (Student)

结果:

Sno	Sname	Ssex	Sage	Sdept
201215121	李勇	男	20	cs
201215122	刘晨	女	19	cs
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

Sno	Sname	Ssex	Sage	Sdept	
201215125	张立	男	19	IS	_ •

选择(续)

[例2] 查询年龄小于20岁的学生 $\sigma_{Sage < 20}$ (Student)

结果:

Sno	Sname	Ssex	Sage	Sdept
201215121	李勇	男	20	cs
201215122	刘晨	女	19	cs
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

Sno	Sname	Ssex	Sage	Sdept
201215122	刘晨	女	19	CS
201215123	王敏	女	18	MA
201215125	张立	男	19	IS



2. 投影(Projection)

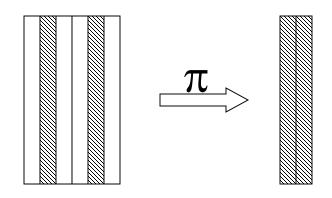
- •1)投影运算符的含义
 - ○从R中选择出若干属性列组成新的关系

$$\pi_A(R) = \{ t[A] \mid t \in R \}$$

A: R中的属性列

2. 投影(Projection)

•2)投影操作主要是从列的角度进行运算



○但投影之后不仅取消了原关系中的某些列,而且 还可能取消某些元组(避免重复行)

投影(续)

● [例3] 查询学生的姓名和所在系 即求Student关系上学生姓名和所在系两个属性上的投影

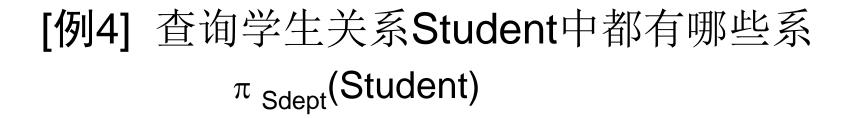
 $\pi_{Sname, Sdept}(Student)$

结果:

Sno	Sname	Ssex	Sage	Sdept
201215121	李勇	男	20	CS
201215122	刘晨	女	19	cs
201215123	王敏	女	18	MA
201215125	张立	男	19	IS

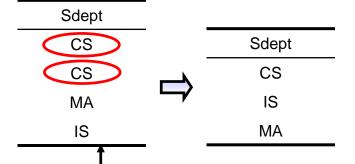
Sname	Sdept
李勇	CS
刘晨	cs
王敏	MA
张立	IS

投影(续)



结果:

201215121 李勇 男	20	cs
201215122 刘晨 女	19	cs
201215123 王敏 女	18	MA
201215125 张立 男	19	IS



3. 连接(Join)

- ●1)连接也称为θ连接
- 2) 连接运算的含义

从两个关系的**笛卡尔积中选取属性间满足一定条件的** 元组

$$R \bowtie_{A \theta B} S = \{ \widehat{t_r t_s} \mid t_r \in R \land t_s \in S \land t_r[A] \theta t_s[B] \}$$

- ►A和B: 分别为R和S上度数相等且可比的属性组
- ▶θ: 比较运算符
- ○连接运算从*R*和*S*的广义笛卡尔积*R*×*S*中选取(*R* 关系)在*A*属性组上的值与(*S*关系)在*B*属性组上 值满足比较关系θ的元组

- 3) 两类常用连接运算
 - ○等值连接(equijoin)
 -)什么是等值连接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:

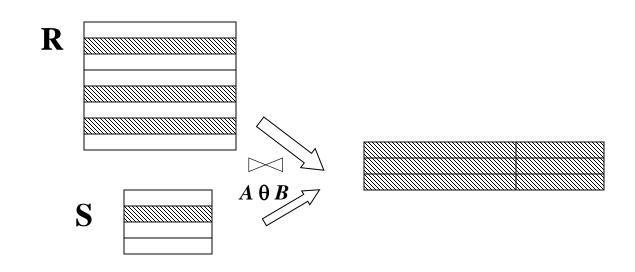
$$R \bowtie S = \{ \widehat{t_{\mathbf{r}}t_{\mathbf{s}}} \mid t_{\mathbf{r}} \in R \land t_{\mathbf{s}} \in S \land t_{\mathbf{r}}[A] = t_{\mathbf{s}}[B] \}$$

- ○自然连接(Natural join) ⋈
 - ●自然连接是一种**特殊的等值连接**
 - ▶两个关系中进行比较的分量必须是同名的属性组
 - ▶在结果中把重复的属性列去掉
 - ●自然连接的含义 *R*和*S*具有相同的属性组*B*

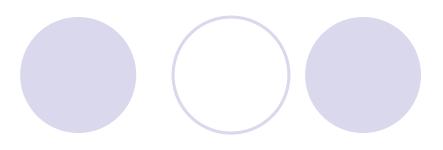
$$R \bowtie S = \{ \pi_{\mathsf{R-S,S}} (\widehat{t_{\mathbf{r}}t_{\mathsf{s}}}) | t_{\mathsf{r}} \in R \land t_{\mathsf{s}} \in S \land t_{\mathsf{r}}[B] = t_{\mathsf{s}}[B] \}$$

这里R-S表示: 出现在R中而不出现在S中的属性组

•4)一般的连接操作是从行的角度进行运算。



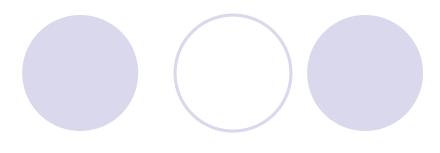
自然连接还需要取消重复列,所以是同时 从**行和列**的角度进行运算。



● [例5]关系*R*和关系*S*如下所示:

R		
A	В	C
a_1	b_1	5
a_1	b_2	6
a_2	b_3	8
a_2	b_4	12

<i>S</i>	
В	E
b_1	3
b_2	7
b_3	10
b_3	2
b_5	2



一般连接 $R \bowtie_{C \leq E} S$ 的结果如下:

S

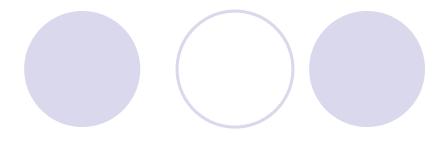
 b_5

R		
1	D	Γ

A	В	С	B	E
a_1	b_1	5	b_1	3
a_1	b_2	6 -	\dot{b}_2	7
a_2	b_3	8 -	b_3	1 0
a_2	b_4	12		2
			b_3	

R	M	S
- ($^{`}\leq I$	7

A	R.B	С	S.B	Е
a_1	b_1	5	b_2	7
a_1	b_1	5	b_3	10
a_1	b_2	6	b_2	7
a_1	b_2	6	b_3	10
a_2	b_3	8	b_3	10



等值连接 $R \bowtie S$ 的结果如下:

R				S	
\overline{A}	В	С	•	В	E
a_1	b ₁ -	5		$-b_1$	3
a_1	b ₂ -	6		$-b_2$	7
a_2	b_3	8		$-b_{3}$	10
a_2	b_4	12		$\sim b_3$	2
				<i>b</i> ₅	2

A	R.B	С	S.B	Е
a_1	b_1	5	b_1	3
a_1	b_2	6	b_2	7
a_2	b_3	8	b_3	10
a_2	b_3	8	b_3	2



A	R.B	C	S.B	Е
a_1	b_1	5	b_1	3
a_1	b_2	6	b_2	7
a_2	b_3	8	b_3	10
a_2	b_3	8	b_3	2



A	В	С	Е
a_1	b_1	5	3
a_1	b_2	6	7
a_2	b_3	8	10
a_2	b_3	8	2



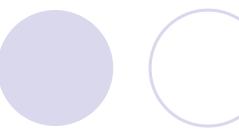
- 全外连接 →
 - ○如果把关系*R*和*S*舍弃的元组也保存在结果关系中,而在 其他属性上填空值(Null),这种连接就叫做全外连接 (FULL OUTER JOIN)。
- 左外连接 →
 - ○如果只把**左边关系R**中要舍弃的元组</u>保留就叫做左外连接 (LEFT OUTER JOIN或LEFT JOIN)
- - ○如果只把**右边关系S中要舍弃的元组**保留就叫做右外连接 (RIGHT OUTER JOIN或RIGHT JOIN)。



R				S	
A	В	С		В	E
a_1	<i>b</i> ₁ -	5		$-b_1$	3
a_1	b ₂ -	-6		- b ₂	7
a_2	<i>b</i> ₃	8		$-b_3$	10
a_2	b_4	12		b ₃	2
			<i></i>	b_5	2
	悬浮す	元组 /			

A	В	С	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) R **>**✓S





A	В	С	Е
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

A	В	С	Е
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

(b) R **>**⋈ S

(c) R **⋈** S

4. 除(Division)

给定关系R(X, Y)和S(Y, Z),其中X, Y,Z为属性组。

R中的Y与S中的Y可以有不同的属性名,但**必须出自相同的**域集。

R与S的除运算得到一个新的关系P(X)

P是R中满足下列条件的元组在X属性列上的<mark>投影</mark>:

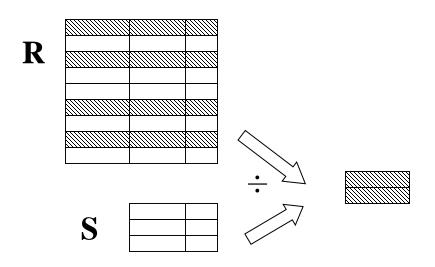
元组在X上分量值x的象集 Y_x 包含S在Y上投影的集合,记作:

$$R \div S = \{x \mid t_{r} \in R \land x = t_{r}[X] \land \pi_{Y}(S) \subseteq Y_{x} \}$$

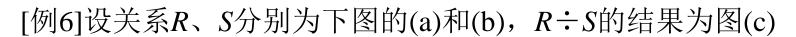
 Y_x : x在R中的象集

除(续)

• 除操作是同时从行和列角度进行运算



除(续)

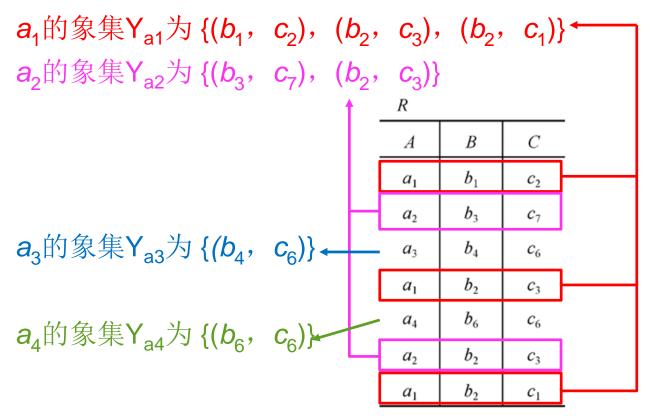


	R				S			1
	A	В	C		В	С	D	
	a_1	b_1	c_2		b_1	c_2	d_1	
	a_2	b_3	c_7	/	b_2	c_1	d_1	Z
	a_3	b_4	c_6	Y	b_2	c_3	d_2	
X	a_1	b_2	c_3			(b)		•
_ ^	a_4	b_6	c_6	_	$R \div S$			
	a_2	b_2	c_3	_	A			
	a_1	b_2	c_1	_	a_1	,		
		(a)		•	(c)			104

分析

求解第一步: Y_x : $x \in R$ 中的象集, $x = t_r[X]$

● 在关系R中,A可以取四个值{a1,a2,a3,a4}



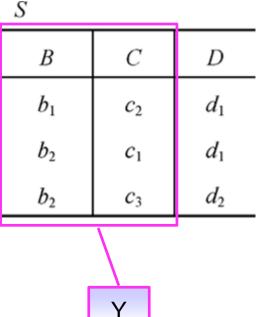
分析

- 求解第2步: S在Y上的投影 $\pi_{Y}(S)$ 为 $\{(b1, c2), (b2, c1), (b2, c3)\}$
- 求解第3步: 哪个x满足 $Y_x \supseteq \pi_Y(S)$?

$$Y_{a1}$$
为 $\{(b_1, c_2), (b_2, c_3), (b_2, c_1)\}$ Y_{a2} 为 $\{(b_3, c_7), (b_2, c_3)\}$ Y_{a3} 为 $\{(b_4, c_6)\}$ Y_{a4} 为 $\{(b_6, c_6)\}$

● 只有 $Y_{a1} \supseteq π_{Y}(S)$

所以 $R \div S = \{a_1\}$



5. 综合举例

以学生-课程数据库为例

Sno	Cno	Grade
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80

[例7] 查询至少选修1号课程和3号课程的学生号码

首先建立一个临时关系K:

然后求: $\pi_{\mathsf{Sno}.\mathsf{Cno}}(\mathsf{SC}) \div K$

● [例 7]续 π_{Sno,Cno}(SC)

200215122象集{2,3}

$$K=\{1, 3\}$$

于是:

 $\pi_{Sno,Cno}(SC) \div K = \{200215121\}$

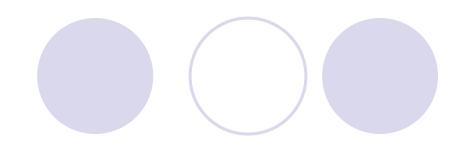
Sno	Cno
200215121	1
200215121	2
200215121	3
200215122	2
200215122	3

Sno
200215121

[例 8] 查询选修了2号课程的学生的学号。

Sno	Cno	Grade
200215121	1	92
200215121	2	85
200215121	3	88
200215122	2	90
200215122	3	80

$$\pi_{\text{Sno}}(\sigma_{\text{Cno}=2}^{\prime}) = \{200215121,200215122\}$$



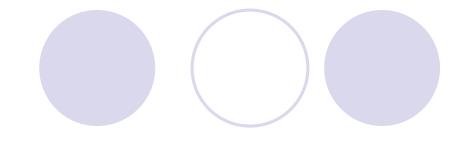
[例9] 查询至少选修了一门其直接先修课为5号课程的 的学生姓名

 $\pi_{\mathsf{Sname}}(\sigma_{\mathsf{Cpno}='5'}(\mathsf{Course} \Join \mathsf{SC} \Join \mathsf{Student}))$ 或

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

 $\pi_{\text{Sname}}(\pi_{\text{Sno}}(\sigma_{\text{Cpno}='5'}(\text{Course}) \bowtie \text{SC}) \bowtie \pi_{\text{Sno}, \text{Sname}}(\text{Student}))$

也可以利用**更名运算**而使用等值连接来实现 更名运算语法: $\rho_x(E)$, 其中 ρ 为运算符,E为表达式,x为关系名



[例10] 查询选修了全部课程的学生号码和姓名。

 $\pi_{\text{Sno, Cno}}$ (SC) $\div \pi_{\text{Cno}}$ (Course) $\bowtie \pi_{\text{Sno, Sname}}$ (Student)

小结

- 关系代数运算
 - 关系代数运算并、差、交、笛卡尔积、投影、选择、连接、除
 - 基本运算 并、差、笛卡尔积、投影、选择
 - 交、连接、除可以用5种基本运算来表达引进它们并不增加语言的能力,但可以简化表达
- 关系代数表达式
 - ○关系代数运算经有限次复合后形成的式子

第二章 关系数据库

- 2.1 关系模型概述
- 2.2 关系数据结构
- 2.3 关系的完整性
- 2.4 关系代数
- *2.5 关系演算
- 2.6 小结

2.6 小结

- 关系数据库系统是目前使用最广泛的数据库系统
- 关系数据库系统与非关系数据库系统的区别:
 - ○关系系统只有"表"这一种数据结构;
 - ○基本表、查询表、视图表
 - ○非关系数据库系统还有其他数据结构,以及对这些数据结 构的操作

- 关系数据结构
 - 关系
 - ●域
 - 笛卡尔积
 - 关系
 - 关系(域的子集),属性,元组
 - 候选码,主码,主属性,非主属性
 - 基本关系的性质(列同质,列可同域不同名,无序,行码唯一,原子性)
 - 关系模式(模式为型,关系为值)
 - 关系数据库

- 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

- 关系的完整性约束
 - ○实体完整性(主属性不能空)
 - ○参照完整性
 - 》外码(为空,或者等于被参照关系主码的某个值)
 - ○用户定义的完整性

- 关系数据语言
 - ○关系代数语言
 - ○关系演算语言
 - OSQL