数据库系统原理

教程：数据库系统概论（第5版）

结合：CMU 15-445/645 INTRO TO DATABASE SYSTEMS

华中科技大学 计算机学院

左琼

第二章 关系数据库

Principles of Database Systems

关系数据库简介

- 提出关系模型的是美国IBM公司的E.F.Codd
  - 1970年首次提出关系数据模型
    - 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

关系模型的特征：

数据结构简单

表达能力强大

数据独立性好

关系模型的组成

- 数据结构：
  - 二维表（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：

  读：查询(query)；写：增加(insert)、删除(delete)、修改(update)
  - 例如：选择(select)、投影(project)、连接(join)、除(divide)、并(union)、交(intersection)、差(difference)……
  - 理论基础：关系代数、元组关系演算、域关系演算
  - 具体实现：关系数据库语言——ALPHA、QBE、SQL……
- 数据的约束条件:

  三类完整性约束：实体完整性、参照完整性、用户自定义完整性

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

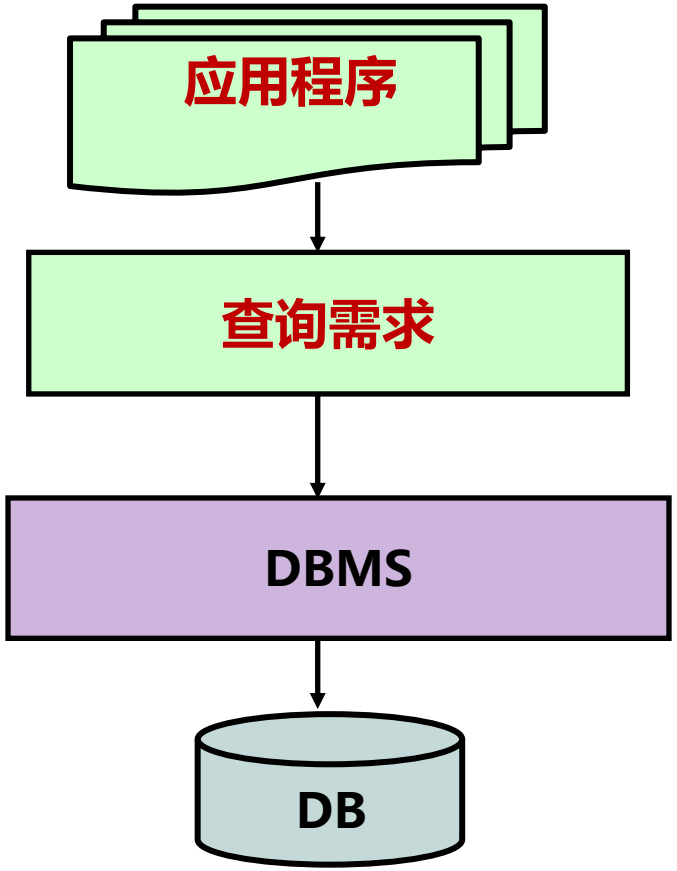
2.1.1 关系
2.1.2 关系模式
2.1.3 关系数据库

如何用离散数学中的代数系统来定义关系数据库？

如何定义代数运算，用数学语言描述查询需求？

关系代数系统的引入动机

关系代数产生动机



问题：为什么需要代数语言?

如何描述查询需求?

查询需求描述自动化
查询需求描述自然化

- 关系数据库是表集
- 定义完备的表操作
- 用表的运算表达式描述查询需求
- 系统解决表达式与查询语言的转化、执行

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。
- 关系模型建立在集合代数的基础上，关系数据结构的形式化定义：
  - 域（Domain）
  - 笛卡儿积（Cartesian Product）
  - 关系（Relation）

1. 域（Domain）

- 域是一组具有相同数据类型的值的集合。又称为值域。（一般用D表示）

  如：整数的集合、字符串的集合、{0,1}、……

- 域中所包含的值的个数称为域的基数（用m表示）。
- 关系中用域表示属性的取值范围。
- 例如：
  - D1 = { 李力，王平，刘伟 }　　m1=3
  - D2 = { 男，女 }　　m2=2
  - D3 = { 47，28，30 }　　m3=3

其中，D1，D2，D3为域名，分别表示教师关系中姓名、性别、年龄的取值集合。

2. 笛卡儿积（Cartesian Product）

- 给定一组域D1，D2，…，Dn，这些域中可以有相同的，则D1，D2，…，Dn的笛卡儿积为：

  $D1 \times D2 \times \ldots \times Dn = \{(d1,d2,\ldots, dn) \mid di \in Di，i = 1,2,\ldots,n\}$

- 笛卡儿积中每一个元素(d1,d2,…, dn)叫作一个n元组（n-tuple）或简称元组(Tuple)。
- 笛卡儿积元素(d1,d2,…, dn)中的每一个值di叫作一个分量(Component)。
- 若Di（i = 1,2,…,n）为有限集，其基数为mi（i = 1,2,…,n），则D1×D2×…×Dn的基数M 为：

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

2. 笛卡儿积

- 笛卡儿积是一个集合；是所有域的所有取值的一个组合。
- 不能重复。
- 例：设D1为学生集合（T）={张群，徐晶，王刚}

  D2为性别集合（S）={男，女}

  则D1×D2是个二元组集合，元组个数为3×2，是所有可能的（学生，性别）元组集合。

$D_1 \times D_2 =$

T	S
张群	男
张群	女
徐晶	男
徐晶	女
王刚	男
王刚	女

2. 笛卡儿积

- 如果 D1为教师集合 ={张清玫，刘逸}, D2为专业集合= {计算机专业，信息专业}, D3为研究生集合= {李勇，刘晨，王敏}

  则D1×D2×D3 =?

笛卡儿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。

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）

- 关系（relation）——笛卡儿积

  D1×D2×…×Dn的子集叫做在域D1，D2，…，Dn上的关系，用R (D1，D2，…，Dn)表示。

  - R是关系的名字，n是关系的度或目（Degree）；
  - n=1时，单元关系（Unary relation）/一元关系
  - n=2时，二元关系（Binary relation）
  - 关系中的每个元素称为元组，通常用t表示；
  - 关系也可以表示为二维表的形式，如：学生（姓名，性别）

一般来说，

取笛卡儿积上有意义的子集作一个关系。

姓名	性别
张群	男
徐晶	女
王刚	男

属性

元组

- **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。**

3. 关系

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，若从属性组中去掉任一属性，它就不再具有这一性质，则称该属性组为候选码。

- 简单情况：候选码只包含一个属性。
- 最极端情况：关系模式的所有属性组成它的候选码，称为全码（All-key）。
- 主码（Primary key）：若一个关系有多个候选码，则选定其中一个为主码。

  每个关系必定有且仅有一个主码，通常用较小的属性组合作为主码。选定以后，不能随意改变。
- 主属性（Prime attribute）：候选码的诸属性。
- 非主属性（Non-Prime attribute）或 非码属性（Non-key attribute）：不包含在任何候选码中的属性。

3. 关系

- 例：选课关系（学号，课程号，课程名，成绩），假设课程名不能重复，则（学号，课程号）和（学号，课程名）都可作为候选码。

提问1：（学号，课程号，课程名）是选课关系的候选码吗？ **No**

提问2：若关系中所有属性均为主属性，则其候选码必为全码吗？ **No**

3. 关系

- 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。

3. 关系

- 关系与二维表的区别：严格地说，关系是规范化了的二维表中行的集合，为了使相应的数据操作简化，在关系模型中对关系作了种种限制。
- 基本关系具有6大特性：
  - 列是同质的（Homogeneous）；
  - 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；
  - 列的顺序无所谓，列的次序可以任意交换；
  - 任意两个元组的候选码不能相同；
  - 行的顺序无所谓，行的次序可以任意交换；
  - 分量必须取原子值，即每个分量是都不可分的数据项。（规范化）

关系是有限集合

2.1.2 关系模式

- 关系模式（Relation Schema）是型，相对静态、稳定
- 关系是值，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

- 关系模式和关系往往统称为关系；
- 通过上下文加以区别

2.1.2 关系模式

- 关系模式——关系的描述称作关系模式，包括关系名、关系中的属性名、属性向域的映象、属性间的数据依赖关系等，记作 R (U，D，DOM，F)。
  其中：
  - R：关系名
  - U：R中的属性名集合
  - D：属性组U中属性所来自的域（取值范围）
  - DOM：属性到域的映象集（属性类型、长度）
  - F：属性间数据的依赖关系集合。
  - 也可简记为 R(U) 或 R(A1, A2, …, An)。
- 关系——某一时刻对应某个关系模式的内容。

例：
DOM (SUPERVISOR-PERSON)
= DOM (POSTGRADUATE-PERSON)
= PERSON

//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

2.1.3 关系数据库

- 关系数据库模式——关系数据库的型
  - 基于某一应用领域所定义的所有关系模式的集合。
  - 包括若干域的定义，及在这些域上定义的若干关系模式。
- 关系数据库——关系数据库的值
  - 关系数据库模式在某一时刻所对应的关系的集合。
- 从系统思维理解：

  数据字典中存放数据库模式，数据文件中存放数据库值。

2.2 关系操作

2.2.1 基本关系操作

关系操作：对关系表进行运算的运算符，这些运算符描述了应用对数据库的数据处理需求。

- 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：
  - 集合操作方式：操作的对象和结果都是集合，一次一集合的方式

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：ALPHA, 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）：高度非过程化语言

2.3 关系的完整性

- 实体完整性机制——这条机制要求关系中元组在组成主码的属性上不能有空值和重复值。
- 参照完整性机制——这条规则要求“不引用不存在的实体”。

前2者称为关系的两个不变性，由关系系统自动支持。

- 用户定义的完整性机制——这是针对某一具体数据的约束条件，由应用环境决定。体现了具体领域中的语义约束。

2.3.1 实体完整性（Entity Integrity）

- 在关系中的所有元组在码上的取值满足以下条件，则说该关系具有实体完整性：
  - 主属性非空——若属性A是基本关系R的主属性，则属性A不能取空值。
  - 主码各不相同——不会出现主码相同的两个记录。
- 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) 主码中的属性不能取空值。

2.3.1 实体完整性

- 关系的主码中的属性值不能为空值
- 空值：不知道或无意义
- 意义：关系对应到现实世界中的实体集，元组对应到实体，实体是相互可区分的，通过主码来唯一标识，若主码为空，则出现不可标识的实体，这是不容许的

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

向关系中插入新行，下列哪些行能够插入？

A. （~~'0203'~~，'张'，男, null)

B. (~~null,~~ '吴'，女，'IS'）

C. （'0301'，null, null, null)

D. （'0105'，'张'，男，'MA'）

2.3.2 参照完整性（Reference Integrity）

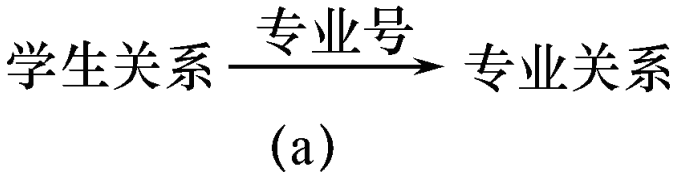
- 关系间的引用

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（学号，姓名，性别，专业号，年龄）

专业（专业号，专业名）

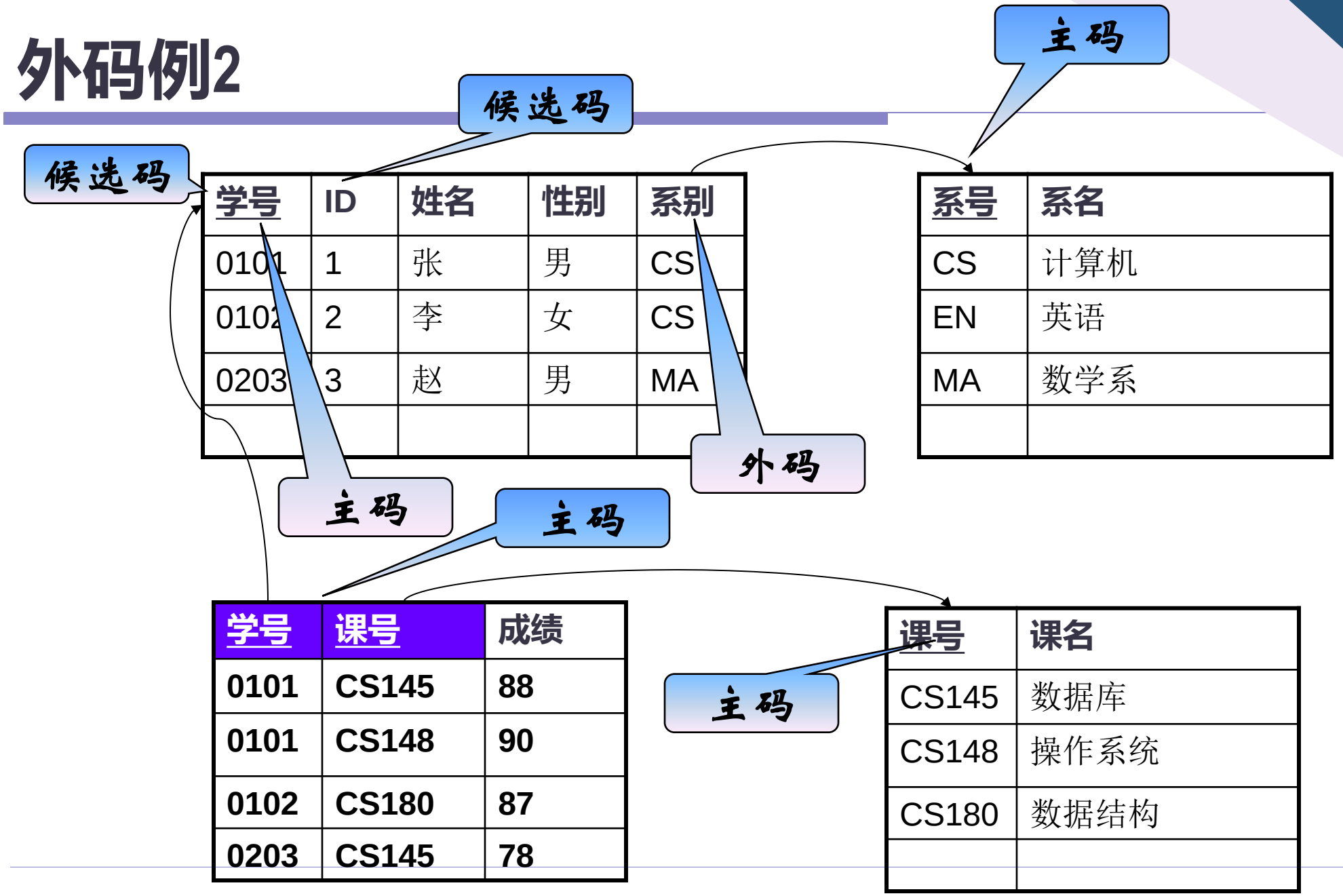


- 设F 是基本关系R 的一个或一组属性，但不是关系 R 的码。如果 F 与基本关系S 的主码 Ks 相对应，则称 F 是基本关系R 的外码。

R 称为参照关系（Referencing Relation）

S 称为被参照关系（Referenced Relation）或 目标关系（Target Relation）

外码例2


候选码
候选码
主码
主码
外码
主码
主码
学号
ID
姓名
性别
系别
0101
1
张
男
CS
0102
2
李
女
CS
0203
3
赵
男
MA
系号
系名
CS
计算机
EN
英语
MA
数学系
学号
课号
成绩
0101
CS145
88
0101
CS148
90
0102
CS180
87
0203
CS145
78
课号
课名
CS145
数据库
CS148
操作系统
CS180
数据结构

2. 外码

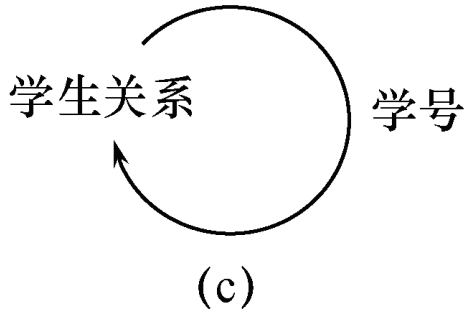
- 关系R 和S 不一定是不同的关系；
- 目标关系S 的主码Ks 和参照关系的外码F 必须定义在同一个（或一组）域上；
- 外码并不一定要与相应的主码同名：

  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。

例3：学生(学号，姓名，…，班长)

其中：学号是“主码”，“班长”是外码

学生关系既是参照关系也是被参照关系



(c)

3. 参照完整性规则

规则2.2 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 Ks 相对应（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

例4：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

例5：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

外码例6

供应商关系S（主码是“供应商号”）

供应商号	供应商名	所在城市
B01	红星	北京
S10	宇宙	上海
T20	黎明	天津
Z01	立新	重庆

零件关系P（主码是“零件号”，外码是“供应商号”）

零件号	颜色	供应商号
010	红	B01
312	白	S10
201	蓝	T20

现要向关系P中插入新的元组，下列元组中哪些能够成功的插入？

A．(037，‘绿’，null)

B．~~(null，~~‘黄’，‘T20’)

C．~~(201，~~‘红’，‘T20’)

D．(105，‘蓝’，‘B01’)

E．(101 ‘黄’，~~‘T11’~~)

2.3.3 用户定义的完整性

- 用户针对具体的应用环境定义的完整性约束条件。
- 意义：反映某一具体应用所涉及的数据必须满足的语义要求，以便用统一的、系统的方法处理它们，而不需要由应用程序来承担这一功能。
- 例如：
  - “成绩”不能为负数，
  - “学号”要求是8位整数，
  - “性别”要求取值为“男”或“女”，
  - “工龄”应该小于“年龄”等。

2.3 关系的完整性

- 问题：以上三类完整性是如何实现的?
- DBMS 提供接口，用户负责定义完整性约束；

  如：实体完整性约束通过说明关系的主码来定义，参照完整性通过说明关系的外码来定义
- DBMS 提供完整性约束的自动检查。

  如：向关系中插入新元组时自动检查主码属性值是否唯一和非空，否则拒绝插入

小结

- 关系数据结构
  - 关系
    - 域
    - 笛卡儿积
    - 关系
      - 关系，属性，元组
      - 候选码，主码，主属性
      - 基本关系的性质
  - 关系模式
  - 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
  - 参照完整性
    - 外码
  - 用户定义的完整性

数据库系统原理

教程：数据库系统概论（第5版）

结合：CMU 15-445/645 INTRO TO DATABASE SYSTEMS

华中科技大学 计算机学院

左琼

第二章 关系数据库

案例：教学数据库有三个关系

问题：如何表示对该数据库的各种操作？

- 查询选修了“数据库”课程的学生姓名。
- 查询学习了1号课程但没学5号课程的学生学号和姓名。
- 查询选修了全部课程的学生学号。

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

C

Cno	Cname	先行课号 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

SC

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

2.4 关系代数

- 关系代数——将关系作为运算单位(操作数)，用关系代数表达式表示的运算方法。

  运算对象：关系

  运算结果：关系

  关系操作：按运算符的不同主要分为两类：

  - 传统的集合运算：把关系看成元组的集合，从行的角度进行运算，包括并、差、交和笛卡尔积等。
  - 专门的关系运算：不仅从行的角度，也从列的角度进行运算，是为数据库的应用而引进的特殊运算，包括选择、投影、连接和除法等。

2.4.1 传统的集合运算

- 传统的集合运算是二目运算。
- 并不是任意的两个关系都能进行这种集合运算，除笛卡尔积外，要求参加运算的关系必须具备相容性。

定义：设给定两个关系R、S，若满足：

(1) 具有相同的度n，

(2) R中第i个属性和S中第i个属性来自同一个域，则说关系R、S是相容的。

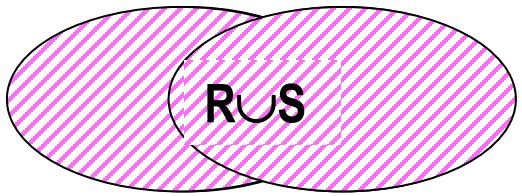
表 关系代数运算符

运算符		含义	运算符		含义
集合运算符	∪	并	比较运算符	>	大于
	-	差		≥	大于等于
	∩	交		<	小于
	×	笛卡尔积		≤	小于等于
				=	等于
				< >	不等于

1. 并（Union）

- 设两个关系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和S的并（记为：R∪S ）
  - 仍为n 目关系，由属于R 或属于S 的元组组成
  - R∪S = { t | t ∈ R ∨ t ∈S }



R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

S

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

R∪S

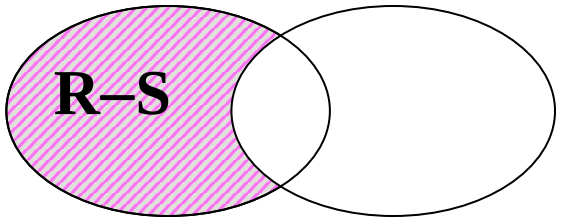
A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1
a1	b3	c2

实例：
- 选修了1号或者2号课程的学生选课记录。

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

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



思考题　如何用差运算求补集?

R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

S

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

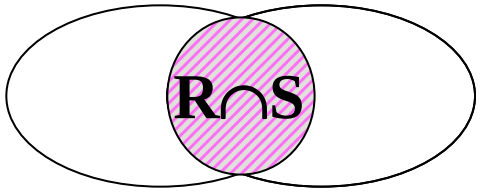
R - S

A	B	C
a1	b1	c1

实例：
——选修了1号课程但没选2号课程的学生选课记录。

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∩S
  - 仍为*n*目关系，由既属于*R* 又属于*S*的元组组成

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

  - 而交运算为非基本运算，不属于最小操作完备集中的操作，可用差运算来表示：

$$R \cap S = R - (R - S) \text{ 或 } R \cap S = S - (S - R)$$

R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

S

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

R∩S

A	B	C
a1	b2	c2
a2	b2	c1

实例：

——既选修了1号课程又选修了2号课程的学生选课记录。

课堂练习：

- 设R和S同为相容的k元关系，R有m个元组，S有n个元组，则关于R∩S，以下论述错误的是（ ）

A. 等于R-(R-S)

B. 等于S-(S-R)

C. √ 最多有m个元组

D. 最少有0个元组

4. 笛卡尔积（Cartesian Product）

- 应用需求：应用程序中需查询来自两张表的信息，系统如何解决？

  将两张表合并为一张表。
- 严格地讲是广义笛卡尔积（Extended Cartesian Product）
- R: m 目关系，k1 个元组；S: n 目关系，k2个元组
- R ×S：$R\times S = \{\widehat{trts} \mid tr\in R \wedge ts\in S\}$
  - 列：（m+n）列元组的集合
    - 元组的前 m 列是关系R的一个元组，后 n 列是关系S的一个元组
  - 行：k1×k2个元组
- 作用：将两个关系无条件的连接成一个新关系，可用于两关系的连接操作。

4.笛卡尔积

R

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c1

S

A	B	C
a1	b2	c2
a1	b3	c2
a2	b2	c1

R × S

R.A	R.B	R.C	S.A	S.B	S.C
a1	b1	c1	a1	b2	c2
a1	b1	c1	a1	b3	c2
a1	b1	c1	a2	b2	c1
a1	b2	c2	a1	b2	c2
a1	b2	c2	a1	b3	c2
a1	b2	c2	a2	b2	c1
a2	b2	c1	a1	b2	c2
a2	b2	c1	a1	b3	c2
a2	b2	c1	a2	b2	c1

2.4.2 专门的关系运算

先引入几个记号：

(1) R，t∈R，t[Ai]

设关系模式为R(A1，A2，…，An)，它的一个关系设为R，

t∈R 表示t 是R 的一个元组，t[Ai] 则表示元组t 中相应于属性Ai 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。

$\overline{A}$ 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。

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

2.4.2 专门的关系运算

（3）元组的连接 $\widehat{tr\ ts}$

R 为m目关系，S 为n目关系，tr ∈R，ts∈S， $\widehat{tr\ ts}$ 称为元组的连接。

$\widehat{tr\ ts}$ 是一个m + n 列的元组，前m个分量为R 中的一个m元组，后n 个分量为S 中的一个n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：

$$Z_x = \{ t[Z] \mid t \in R,\ t[X] = x \}$$

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。

2.4.2 专门的关系运算

R

x1	Z1
x1	Z2
x1	Z3
x2	Z2
x2	Z3
x3	Z1
x3	Z3

象集举例

x1在R 中的象集：

Z_{x1} ={Z1，Z2，Z3}，

x2在R 中的象集：

Z_{x2} ={Z2，Z3}，

x3在R 中的象集：

Z_{x3}={Z1，Z3}

从R中选出在X上取值为x的元组，去掉X上的分量，只留Z上的分量

象集例1

关系模式: 学生(学号,姓名,性别,系别)

元组t: (0102,李,女,CS)

属性列X: {性别,系别}

t[性别,系别]: (女,CS)

属性组Z: {学号,姓名}

t[X] = (女,CS)

Zx = ?

CS系全部女生的学号,姓名

Z X

学号	姓名	性别	系别
0101	张	男	CS
0102	李	女	CS
0203	赵	男	MA
0103	吴	女	CS

例2

X Z

姓名	课程
张蕊	物理
王红	数学
张蕊	数学

x=张蕊 → Z_x

课程
数学
物理

张蕊同学所选修的全部课程

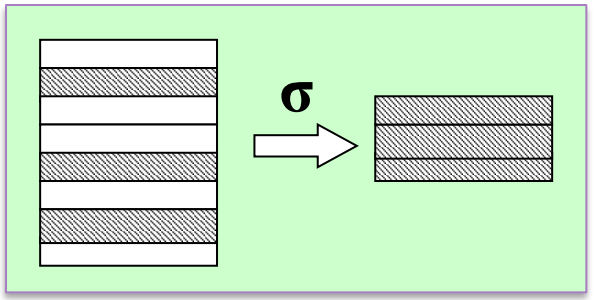
2.4.2 专门的关系运算

表 关系代数运算符

运算符	含义		运算符	含义	
专门的关系运算符	σ	选择	逻辑运算符	¬	非
	π	投影		∧	与
	⋈	连接		∨	或
	÷	除			

1. 选择（Selection）

- 选择操作是根据某些条件对关系做水平分割，即选取符合条件的元组构成结果关系，又称为限制（Restriction）。



- 关系R关于公式F的选择记作：
  - $\sigma_F(R) = \{t \mid t \in R \wedge F(t) = \text{'真'}\}$
  - 其中：σ为选择运算符，F为条件表达式，

    $\sigma_F(R)$表示从R中挑选满足公式 F 的元组所构成的关系。

F：是一个逻辑表达式，基本形式为：$X_1 \theta Y_1$

F的组成：

- 运算对象：属性，常数(如数字)
- 运算符：算术运算符（>，≥，=，<，≤，≠），
  逻辑运算符（∧，∨，┐）

1. 选择

[例1]

R

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

$\sigma_{A>1 \wedge B=2}(R)$

A	B	C
2	2	3

上式也可写作：$\sigma_{[1]>1 \wedge [2]=2}(R)$

[例2] 在S(Sno,Sname,Ssex,Sage,Sdept)上查询年龄小于20岁的学生。

$\sigma_{Sage < 20}(S)$ 或 $\sigma_{[4] < 20}(S)$

Sno	Sname	Ssex	Sage	Sdept
201415122	刘晨	女	19	IS
201415123	王敏	女	18	MA
201415125	张立	男	19	IS

1. 选择

- 例3：用关系表达式表达下列查询：

  找前页关系S中计算机系(代号:‘CS’)全部的男生。

  $\sigma_{Sdept='CS' \wedge Ssex='男'}(S)$

Sno	Sname	Ssex	Sage	Sdept
201515121	张晨	男	19	CS
201515123	李敏	男	18	CS
201515127	何立	男	19	CS

2. 投影（Projection）

- 关系R上的投影是从R中选择出若干属性列组成新的关系。

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

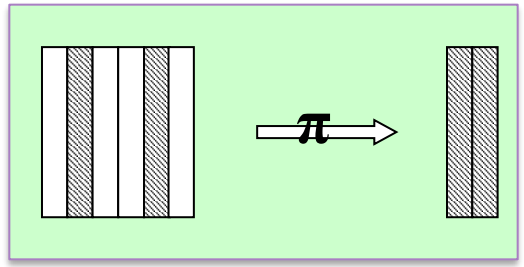
  A为R中的属性列，可用列的属性名或列在关系中的序号表示。

- 特征：

  1) 在单个关系上进行

  2) 从列的角度进行运算

  3) 投影的列可按自己的要求的顺序排列



- 作用：在关系中选择某些需要的列，并按要求组成一个新关系。
- 投影操作主要是从列的角度进行运算。
- 投影的结果中要去掉相同的行（避免重复行）。Why?

  - 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。

2. 投影

- 例1：求关系 R 在 A、C 两列上的投影。

解：关系代数表达式为： $\pi_{A,C}(R)$ 或 $\pi_{[1],[3]}(R)$

R

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

$\pi_{A,C}(R)$

A	C
1	3
4	6
2	3

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

B	C
2	3
5	6

- 例2：
- 给出所有学生的姓名和年龄

  $\Pi_{Sname,\ Sage}(S)$
- 找001号学生所选修的课程号

  $\Pi_{C\#}(\ \sigma_{S\#='001'}\ (SC))$

S(S#,Sname,Sage)
Course(C#,Cname)
SC(C#,S#,Score)

投影与选择

- **复合运用投影、选择、笛卡尔积运算**，可从任意n张表中截取满足条件的子表

- 例3:

  列出CS系和MA系学生的学号和姓名。

- 方案1:

  $\Pi_{SNO,SNA}(\sigma_{DEPT = 'CS' \vee DEPT = 'MA'}(S))$

- 方案2:

  $\Pi_{SNO,SNA}(\sigma_{DEPT = 'CS'}(S)) \cup \Pi_{SNO,SNA}(\sigma_{DEPT = 'MA'}(S))$

S

SNO	SNA	SEX	DEPT
0101	张	男	CS
0102	李	女	CS
0203	赵	男	MA
0103	吴	女	CS

3. 连接（Join）

- 连接也称为θ连接；
- 连接运算是从两个关系的笛卡尔积中选取满足连接条件的元组，记作：

$$\mathbf{R} \underset{A\theta B}{\bowtie} \mathbf{S} = \left\{ t_r \frown t_s \mid t_r \in R \wedge t_s \in S \wedge t_r[A] \theta t_s[B] \right\}$$

- 其中：
  - A和B分别为R和S上度数相等且可比的属性组。
  - θ是比较运算符（> ≥ = < ≤ ≠）。
  - 连接运算从R和S的广义笛卡尔积$R \times S$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. 连接

2类常用连接运算：

两个关系参加运算，不一定有公共属性

- 等值连接（equijoin）：θ为“=”的连接运算。

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

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。

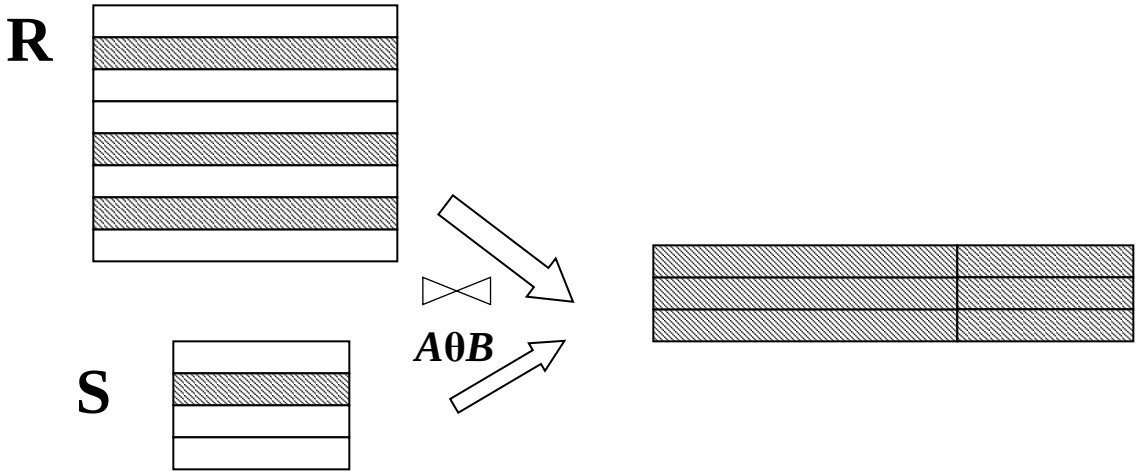
- 自然连接（natural join）：是一种特殊的等值连接：
  - 两个关系中进行比较的分量必须是相同的属性组；
  - 在结果中把重复的属性列去掉。

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

  - R 和S 具有相同的属性组B
  - 当R与S无相同属性时，$R \bowtie S = R \times S$

3. 连接

- 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。
- 自然连接的本质是将两张有关联的表，按照元组之间在属性B（外码）上的等值关系，合并为一张表。

连接示例

运算步骤：

1) 求笛卡尔积R×S

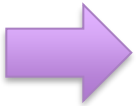
2) 选择其中满足AθB的元组

R

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

S

C	D
a	7
b	8



R ×S

A	B	R.C	S.C	D
1	2	3	a	7
1	2	3	b	8
4	5	6	a	7
4	5	6	b	8
7	8	9	a	7
7	8	9	b	8

R ⋈ S [3]>[2] 或 R ⋈ S C>D

A	B	R.C	S.C	D
7	8	9	a	7
7	8	9	b	8

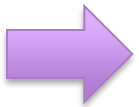
自然连接示例

R

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

S

B	E	F
b	c	f
g	h	i



1) 计算R×S

A	R. B	C	S.B	E	F
a	b	c	b	c	f
a	b	c	g	h	i
b	a	f	b	c	f
b	a	f	g	h	i
c	b	d	b	c	f
c	b	d	g	h	i

2) 选择R.B = S.B的元组

A	R. B	C	S.B	E	F
a	b	c	b	c	f
c	b	d	b	c	f

3) 去掉重复属性

A	B	C	E	F
a	b	c	c	f
c	b	d	c	f

3. 连接（续）

- **问题：自然连接会丢失信息，需引入新的连接运算**
  例如：student ⋈sc 会将一个未选课的学生丢失
- 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)。

连接例子

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

一般连接$R \underset{C<E}{\bowtie} S$的结果如下：

$R \underset{C<E}{\bowtie} S$

A	$R.B$	C	$S.B$	E
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

连接例子

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

等值连接 $R \underset{R.B=S.B}{\bowtie} S$ 的结果如下：

A	$R.B$	C	$S.B$	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_3	8	b_3	2

连接例子

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 $R \bowtie S$ 的结果如下：

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

连接例子

关系*R*和关系*S*的外连接、左外连接、右外连接：

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

外连接运算

- 外连接 外连接运算是扩展运算，可以用其它运算代数表达式表示。
- 例如： 左外连接可以写成：

  $(r \bowtie s) \cup (r - \pi_R(r \bowtie s)) \times \{(null,.....null)\}$

  分析：问题的关键是如何求r与s的自然连接后丢失的r中元组?

  解决方法：利用减法运算求补集

  说明：表达式 $r - \pi_R(r \bowtie s)$ 为所需丢失元组

- 思考题：写出右外连接和全连接的代数式

关系运算综合举例

常用的代数思维解决方法：

1）整体法

- 首先分析所需信息来自哪些表
- 其次用适当的连接运算合并表
- 再用选择运算σ_P行分解表，通过P去除无用元组
- 最后用投影运算π_A列分解表，通过A选择所需结果

2）分步法

- 首先将问题分解为多个简单步骤（可用单表解决）
- 其次对最里层的问题用一个代数表达式表示结果
- 再将结果作为已知值，代入上一层步骤中

注：分步法也可以是从外层向里层的迭代过程

案例：教学数据库有三个关系：

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

C

Cno	Cname	先行课号 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

SC

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

关系运算综合举例

例：查询选修了一门其直接先行课为5号课程的学生姓名。

- 解法1：整体法

问题分析：

1) 该查询是查找选过…课程的学生，所以先将学生表、选修表、课程表合并

$Course \bowtie SC \bowtie Student$

2) 再确定选择运算谓词P为cpno=5

$\sigma_{Cpno='5'}(Course \bowtie SC \bowtie Student))$

3) 最后投影所需的学生姓名即可

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

关系运算综合举例

- 解法2：分步法

  1) 找出先修课为5的课程：

  $\sigma_{Cpno='5'}(Course)$

  2) 找出选过上表达式结果的选课元组中学号：

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

  3) 从学生表中找出学号为上一步结果的学生：

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

- 提问：上述例子中我们通过自然连接实现了选择运算的功能能否用嵌套方法实现选择？ 例如: 第二步改为：

  $\sigma_{cno=\pi_{cno}(\sigma_{Cpno='5'}(Course))}(SC)$ ？

关系运算综合举例

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

[上例] 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。

解答：

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

或

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

或

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

回顾案例：教学数据库有三个关系：

C

Cno	Cname	先行课号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

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

SC

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

求解：如何表示对该数据库的各种操作？

- 查询选修了“数据库”课程的学生姓名。

$$\pi_{Sname}(\sigma_{Cname='数据库'}(Course \bowtie SC \bowtie Student))$$

- 查询学习了1号课程但没学5号课程的学生学号和姓名。

$$\pi_{Sno,\ Sname}((\pi_{Sno}(\sigma_{Cno='1'}(SC)) - \pi_{Sno}(\sigma_{Cno='5'}(SC))) \bowtie Student)$$

- 查询选修了全部课程的学生学号。

4. 除（Division）

- 引入动机：在查询中，经常需要查询包含短语“所有的”这样的查询。
- 例：找出选过学分为3分的所有课程的学生？

  解题思路：1）找出学分为3分的所有课程；

  2）从选课表中，找出学生，其所选课程包含1）中结果；

  解：1）令$S=\pi_{cno}(\sigma_{Ccredit=3}(Course))$

  2）对选课表按照Sno分组：$_{sno}G(SC)$，表示每个学生选了哪些课程为一组

  3）令S’= 分组以后的由Cno构成的表；若S’包含S，则将这样的学生放入结果集，记作：

  $$SC \div \pi_{cno}(\sigma_{Ccredit=3}(Course))$$

- 语义：R÷S是指从R中去除哪些不包含S的元组，即：从R中查找“选过所有的S”的查询。

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*属性列上的投影：

  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：

  $$R \div S = \{t_r[X] \mid t_r \in R \wedge \pi_Y(S) \subseteq Y_x\}$$

  Y_x：x在R中的象集，$x = t_r[X]$
- 除操作是同时从行和列角度进行运算

4. 除

为方便起见，我们假设 S 的属性为 R 中后 s 个属性。

R÷S 的具体计算过程如下：

1) $T = \pi_{1,2,..,r-s}(R)$

2) $W = (T \times S) - R$

3) $V = \pi_{1,2,...\ r-s}(W)$

4) $R \div S = T - V$

即 $R \div S = \pi_{1,2,..r-s}(R) - \pi_{1,2,..r\ -\ s}((\pi_{1,2,..r-s}(R) \times S) - R)$

R

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

S

C	D
c	d
e	f

1) $T = \pi_{A,B}(R)$

A	B
a	b
b	c
e	d

2) $W = (T \times S) - R$

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

3) $V = \pi_{A,B}(W)$

A	B
b	c

4) $R \div S = T - V$

A	B
a	b
e	d

$$R \div S = \Pi_X(R) - \Pi_X(\Pi_X(R) \times \Pi_Y(S) - R)$$

例如：求选修了所有课程的学生的姓名 （R÷S公式理解）

课程表

课程
数学
物理

×

$\Pi_{姓名}$(选课表)

姓名
张军
王红

=

姓名	课程
张军	物理
王红	数学
张军	数学
王红	物理

所有学生选修全部课程

姓名	课程
张军	物理
王红	数学
张军	数学
王红	物理

−

选课表

姓名	课程
张军	物理
王红	数学
张军	数学

=

$\Pi_{姓名}$

姓名
王红

没有选修全部课程的学生

姓名
张军
王红

−

姓名
王红

=

姓名
张军

选修了全部课程的学生

除——例题

- 例：设关系R、S 分别为下图的(a)和(b)，R÷S 的结果为图(c)

R

A	B	C
a_1	b_1	c_2
a_2	b_3	c_7
a_3	b_4	c_6
a_1	b_2	c_3
a_4	b_6	c_6
a_2	b_2	c_3
a_1	b_2	c_1

(a)

S

B	C	D
b_1	c_2	d_1
b_2	c_1	d_1
b_2	c_3	d_2

(b)

$R \div S$

A
a_1

(c)

只有a_1的象集包含了S在$(B，C)$属性组上的投影

在关系R中，A可以取四个值{a1，a2，a3，a4}

a_1的象集为 $\{(b_1，c_2)，(b_2，c_3)，(b_2，c_1)\}$

a_2的象集为 $\{(b_3，c_7)，(b_2，c_3)\}$

a_3的象集为 $\{(b_4，c_6)\}$

a_4的象集为 $\{(b_6，c_6)\}$

除——分析

- 例题变形：求选修了所有课程的学生的姓名。

姓名	课程	成绩
张军	物理	88
王红	数学	80
张军	数学	90

÷

课程
数学
物理

=

姓名
张军

选修了全部课程的学生的姓名

对不对?

No

除　正确的写法是：

$\Pi_{姓名，课程}(R) \div S$

R

姓名	课程	成绩
张军	物理	88
王红	数学	80
张军	数学	90

÷

S

课程
数学
物理

$\Pi_y(S)=\{课程\}$，　$Y_x=\{课程\}$，
$X=\{姓名，成绩\}$

$R \div S =$ 

X	Y

Y	Z

$\Pi_y(S) \subseteq Y_x$

{张军，88}={物理}，

{王红，80}={数学}，

{张军，90}={数学}

没有哪个X的象集包含了{物理，数学}，所以答案应该是空集！

除运算示例

例：查询选修了全部课程的学生号码和姓名。

解： 与上题类似：

$$\pi_{Sno,\ Cno}(SC) \div \pi_{Cno}(Course)$$

是选过全部课程的学生号；

将其与Student自然合并，为所需结果：

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

数据库系统原理

教程：数据库系统概论（第5版）

结合：CMU 15-445/645 INTRO TO DATABASE SYSTEMS

华中科技大学 计算机学院

左琼

2.4 关系代数运算小结

课程回顾

- 5 种基本运算（并∪、差-、笛卡尔积×、选择σ、投影π）
- 其它运算（交∩、连接 ⋈、除÷）均可用 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：
  - $R \cap S = R - (R - S) = S - (S - R)$
  - $R \bowtie S = \pi_{i1,\ldots,im}(\sigma_{R.A1=S.A1 \wedge R.A2=S.A2 \ldots \wedge R.Ak=S.Ak}(R \times S))$
  - $R \div S = \pi_{1,2,..r-s}(R) - \pi_{1,2,..r-s}((\pi_{1,2,..r-s}(R) \times S) - R)$
- 关系代数中，这些运算经有限次复合后形成的式子称为关系代数表达式。利用这些表达式可以实现对关系数据库的各种操作（插入、删除、修改、查询）。

2.4.3 附加运算*（了解*）

- 问题　关系代数的基本运算足以表达任何查询，但使用不方便，写出的表达式太长。定义一些附加运算，不能增加关系代数功能，但能简化表达式。
- 更名运算

  $\rho_{x(A1,A2,.....An)}(E)$

  将E的结果取名字x，且将属性名改为A1,...An
- 为什么需要更名运算?

  一个关系代数表达式的结果关系没有名字；

  一个关系表可以多次参与到一个联系中，每次参与角色不同，如何区分?

更名运算示例*

- 例：只用基本关系运算符，找出学生中最大年龄？
- 求解：利用集合的补集运算，求出所有非最大年龄集合的补集即可，反证思维

    1）求出一个由非最大年龄构成的表

    2）求所有学生年龄与上一步结果的差

如何求非最大年龄集合？

令所有学生集合为A，所有年龄集合为B，显然，A与B都在Student表中；

对于A中每个学生t，将其与B中所有年龄元组配对，判断：该学生年龄是否小于所有学生年龄集中的某一个。若是，则将其放在结果集中。

更名运算示例*

分析知，Student表两次以不同角色参与运算，为区别起见，令代表年龄的学生表为d。

S1：首先将学生表与d表合并

$Student \times \rho_d(Student)$

S2：在合并表中选择哪些年龄比所有年龄中某一个小的学生

$\pi_{Student.age}(\sigma_{Student.age<d.age}(Student \times \rho_d(Student)))$

S3：求差（求问题的反面）

$\pi_{age}(Student) - \pi_{Student.age}(\sigma_{Student.age<d.age}(Student \times \rho_d(Student)))$

注：上面S1、S2步中，也可以用条件连接的选择功能，直接从学生表中将哪些年龄不是最大的学生选择出

$\rho_s(Student) \underset{s.age<d.age}{\bowtie} \rho_d(Student)$

思考题：求出与“张三”在同一个系的学生?

$$\rho_s(Student) \underset{s.dept=d.dept}{\bowtie} \sigma_{d.sname='张三'} \rho_d(Student)$$

聚集运算*

- 问题示例　选课表每个学生会有多个选课记录，若要查询每个学生所选课程的平均分如何办?
- 解决办法　1）将选课表按照学号分组
  2）对分组后的表中每个组再调用AVG（）函数

SNO	CNO　GRADE
200215121	(1,　92) (2,　85) (3,　88)
200215122	(2,　90) (3,　80)
……	……

聚集运算*

- 聚集运算　将表按照属性分组，即将元组按照属性值重新组合
- 定义　$_{A_1,\ldots A_n}G(E)$

  E是关系代数表达式，即是一张表

  $A_1,\ldots A_n$是用于分组的一组属性

  运算符G表示将表达式E按照$A_1,\ldots A_n$分组

  同一组中所有元组在$A_1,\ldots A_n$上值相同

  不同组元组在$A_1,\ldots A_n$上值不同
- 示例　$_{sno}G(SC)$结果为上一页中表

  $_{ssex}G(Student \bowtie SC)$的结果为多少?
- 问题　聚集运算的结果是由多个分组组成的表，对每个分组可以定义聚集函数

聚集函数*

- 聚集函数　输入为集合，输出为单一值的函数。

  sum(), avg(), max(), min(), count()

- 应用　　聚集函数往往和聚集运算组合使用

  对于聚集运算后的结果，对每个组再运用聚集函数处理

- 示例　　$_{sno}G_{count（cno）,min(grade)}(SC)$

  表示每个学生所选课的个数和最低分

- 如何去除相同元素?

  distinct操作符，其含义是消除相同元素。

  e.g　$_{sno}G_{count（distinct\ cno）,min(grade)}(SC)$

广义投影*

- 问题示例　学生表中只有年龄，若查询学生的出生年份如何办?
- 问题分析　出生年份可以通过年龄计算求得，因此，扩展投影运算，使投影属性可以是派生属性，即可以从表中经过运算得出。
- 广义投影　$\pi_{F1,...Fn}(E)$

  F1,...Fn中是可以涉及常量、系统函数及E中属性的算术表达式。

- 示例　$\pi_{sno,year()-age}(Student)$结果为由（sno，year()-age）两列组成的表；

  $\rho_{sno,\ birthday}(\pi_{sno,year()-age}(Student))$结果为（sno, birthday)两列组成的表。

数据库修改*

- 定义 对数据库的增、删、查的运算符，通过赋值完成。
- 删除 $r \leftarrow r - E$ 其中：E是关系代数查询表达式。

  例： $SC \leftarrow SC - \sigma_{sno=2}(SC)$

  表示从选课表中删除了2号学生的选课记录。
- 插入 $r \leftarrow r \cup E$
- 更新 $r \leftarrow \pi_{F1,...Fn}(r)$，使用广义投影可以改变元组中的值。

  $r \leftarrow \pi_{F1,...Fn}(\sigma_P(r)) \cup (r - \sigma_P(r))$，对r中的部分元组修改。
- 注 上述定义的所有运算符来自于应用语义需求，每个运算符在后面的SQL语言中都有相应语句。

视图操作*

- 视图操作　能够从已有的表集合中生成一个虚关系表的运算。
  出于安全性或出于方便性考虑，需要视图操作。
- 定义　creat view v as E,将E的结果作为视图v
- 示例　creat view cs-student
  as ($\sigma_{sdept='cs'}$ (SC))
- 注意　视图定义运算不同于关系赋值运算：
  执行赋值运算时，结果关系被计算并存储，
  执行视图定义运算时，结果关系未被计算，
  直到某个查询使用视图时才动态计算视图表。

第二章 关系数据库

2.5 关系演算（掌握与关系代数的差异*）

- 关系演算：以数理逻辑中的谓词演算为基础。
- 按谓词变元不同进行分类：

1.元组关系演算：

以元组变量作为谓词变元的基本对象

元组关系演算语言ALPHA

2.域关系演算：

以域变量作为谓词变元的基本对象

域关系演算语言QBE

关系代数——将关系作为运算单位(操作数)，用关系代数表达式表示的运算方法。

2.5.1 元组关系演算*

- 用元组作为谓词变量的一种谓词演算方法。元组关系演算表达式的一般形式为：

  $\{ t \mid P(t) \}$

  表示所有使 P(t) 为真的元组集合。

- 其中：
  - t —— 元组变量。表示一个元组，若 t 中有多个分量，表示为t[1]，t[2]，……；
  - P(t) —— 由原子公式和运算符组成的复合公式。

2.5.1 元组关系演算

原子公式有下列三种形式：

- R(s)　　R是关系名，s是元组变量。含义：s是关系R的一个元组。
- s[i] θ u[j]　s和u是元组变量，θ是算术比较运算符。含义：元组s的第i 个分量与元组u的第j个分量之间满足θ关系。
- s[i] θ a　　元组s的第i 个分量值与常量a之间满足θ关系。

❖**运算符包括四类：**

① 括号（）

② 算术运算符：＞　≥　＝　＜　≤　≠

③ 存在量词∃，全称量词∀

④ 逻辑运算符：¬　∧　∨

2.5.1 元组关系演算

公式的递归定义如下：

① 原子公式 P 是一个公式。其值为 P 的真、假值。

② 如果P1和P2是公式，那么¬P1、P1∧P2、P1∨P2也是公式。其真假值遵循逻辑运算的一般原则。

③ 如果P是公式，那么 (∃t)P(t) 也是公式。设元组变量的域集T = {t1, t2, …, tn}，(∃t)P(t) ⇔ P(t1)∨P(t2)∨…∨P(tn)，至少存在一个元组 ti 使得公式 P 为真，否则为假。

④ 如果P是公式，那么 (∀t)P(t) 也是公式。设元组变量的域集T = {t1, t2, …, tn}，(∀t)P(t) ⇔ P(t1)∧P(t2)∧…∧P(tn)，所有元组 ti 使得P为真时上式为真，否则为假。

t 在P中是自由变量，在 (∃t)P(t)和(∀t)P(t)中是约束变量，即：

自由元组变量——在一个公式中 t 未用∃ 、∀符号定义；

约束元组变量——在一个公式中 t 用∃ 、∀符号定义；

2.5.1 元组关系演算

- 例：设有两个关系 R 和 S，求表达式的值：

R

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

S

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

1） $R_1 = \{ t \mid S(t) \wedge t[1]>2 \}$

R_1

A	B	C
3	4	6
5	6	9

$\sigma_{[1]>2}(S)$

2） $R_2 = \{ t \mid R(t) \wedge \neg S(t) \}$

R_2

A	B	C
4	5	6
7	8	9

R - S

3） $R_3 = \{ t \mid (\exists u)(S(t) \wedge R(u) \wedge t[3]< u[1]) \}$

R_3

A	B	C
1	2	3
3	4	6

$\Pi_{R.*}(R \bowtie_{S.C<R.A} S)$

2.5.1 元组关系演算

R

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

S

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

4） $R_4=\{ t \mid (\forall u)(R(t) \wedge S(u) \wedge t[3]>u[1])\}$

R_4

A	B	C
4	5	6
7	8	9

$\sigma_{R.C > G_{max(A)}(S)}(R)$

5） $R_5=\{ t \mid (\exists u)(\exists v)(R(u) \wedge S(v) \wedge u[1]>v[2] \wedge t[1]=u[2] \wedge t[2]=v[3] \wedge t[3]=u[1])\}$

R_5

R.B	S.C	R.A
5	3	4
8	3	7
8	6	7
8	9	7

$\Pi_{R.B,S.C,R.A}(R \underset{R.A>S.B}{\bowtie} S)$

应用

S (Sno, Sname, Ssex, Sage, Sdept)
C (Cno, Cname, Cpno, Ccredit)
SC (sno, cno, grade)

例1：查询CS系的学生信息

$\{ t \mid S(t) \wedge t[5]='CS' \}$

例2：查询学习课程号为2的学生学号。

$\{ t \mid (\exists u)(SC(u) \wedge u[2]='2' \wedge t[1]=u[1] \}$

例3：查询选修了“数据库”课程的学生学号。

$\{ t \mid (\exists u)(\exists v)(SC(u) \wedge C(v) \wedge v[1]=u[2] \wedge v[2]='数据库' \wedge t[1]=u[1]) \}$

例4：查询选修了全部课程的学生学号。

$\{ t \mid (\forall u)(C(u) \wedge (\exists v)(SC(v) \wedge v[2]=u[1] \wedge t[1]=v[1])) \}$

2.5.2 域关系演算*

- 域关系演算类似于元组关系演算，不同处是用域变量代替元组变量的每一个分量，域变量的变化范围是某个值域而不是一个关系。域演算表达式形为：

  { t1…tk | P(t1,…,tk) }

  其中P(t1,….,tk)是关于自由域变量t1,….,tk的公式。

- 域关系演算的公式中也可使用∧、∨、¬等逻辑运算符，也可用 (∃ x) 和 (∀x) 形成新的公式，但变量 x 是域变量，不是元组变量。

- 域演算的原子公式：

  ① R(x1…xk)：R是一个k元关系，xi是常量或域变量。含义：由x1,…,xk组成的元组在关系R中。

  ② xθy：x，y是常量或域变量，但至少有一个是域变量，θ是算术比较符。含义：x 和 y 之间满足关系θ。

例：设有R、S和W三个关系，求表达式的值：

R

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

S

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

W

D	E
7	5
4	8

1）$R_1=\{xyz \mid R(xyz) \wedge x<5 \wedge y>3\}$

R_1

A	B	C
4	5	6

$\sigma_{[1]<5 \wedge [2]>3}(R)$

2）$R_2=\{xyz \mid R(xyz) \vee (S(xyz) \wedge y=4)\}$

R_2

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

…U…

3）$R_3=\{xyz \mid (\exists u)(\exists v)(R(zxu) \wedge W(yv) \wedge u>v)\}$

R_3

A	B	C
5	7	4
8	7	7
8	4	7

…⋈…

应用示例

S (Sno, Sname, Ssex, Sage, Sdept)
C (Cno, Cname, Cpno, Ccredit)
SC (sno, cno, grade)

例1：查询计算机系(IS)的全体学生。

{ abcde | S(abcde)∧e ='CS' }

例2：查询年龄小于20岁的学生。

{ abcde | S(abcde)∧d<20 }

例3：检索选修课程号为5的学生学号和姓名。

{ ab | (∃u)(∃v)(S(abcde)∧SC(uvw)∧a=u∧v='5') }

关系运算的安全性(★)

- 关系演算有可能会产生无限关系和无穷验证，这样的表达式是不安全的。例如：{ t | ¬R(t) }，(∀u)(ω(u))。
- 不产生无限关系和无穷验证的运算称为安全运算，其运算表达式称为安全表达式，所采取的措施称为安全限制。
- 在关系演算中，引入公式P的域概念，用**DOM(P)**表示。

DOM(P) = 显式出现在P中的值 + 在P中出现的关系的元组中出现的值（不必是最小集）

- 满足下列条件时，称元组演算表达式{ t | P(t) }是安全的：
  - 出现在表达式{ t | P(t) }结果中的所有值均来自DOM(P)；
  - 对P中的每个形如(∃u)(ω(u)) 的子式，若u使ω(u) 为真，则 u 的每个分量必属于DOM(P)。
  - 对P中的每个形如(∀u)(ω(u)) 的子式，若u使ω(u) 为假，则 u 的每个分量必属于DOM(P)。

关系运算的安全性(★)

R

A	B
a1	b1
a2	b2

S

A	B
1	d
5	b
6	c
7	d

$R_1=\{ t \mid \neg R(t) \}$

DOM(P)={{a1 , a2} , {b1 , b2}}

R_1

A	B
a2	b1
a1	b2

$R_2=\{ t \mid (\exists u)(S(u) \wedge u[1]>3 \wedge t[1]=u[2] \}$

DOM(P)={{1,5,6,7,3} , {d,b,c}}

R_2

B
b
c
d

元组演算对基本关系操作的表示

并：$R \cup S \equiv \{ t \mid R(t) \vee S(t) \}$

差：$R - S \equiv \{ t \mid R(t) \wedge \neg S(t) \}$

笛卡儿积：$R \times S \equiv \{ t(r+s) \mid (\exists u)(\exists v)(R(u) \wedge S(v) \wedge t[1]=u[1] \wedge \ldots \wedge t[r]=u[r] \wedge t[r+1]=v[1] \wedge \ldots \wedge t[r+s]=v[s]) \}$

$(\exists t)P(t)$
$(\forall t)P(t)$

投影：$\pi_{i1,\ldots,im}(R) \equiv \{t(m) \mid (\exists u)R(u) \wedge t[1]=u[i_1] \wedge t[2]=u[i_2] \wedge \ldots \wedge t[m]=u[i_m]\}$

选择：$\sigma_F(R) \equiv \{ t \mid R(t) \wedge F' \}$（F'是由F变化形成的谓词公式）

域演算对基本关系操作的表示

并：$R \cup S \equiv \{ x_1x_2\ldots x_n \mid R(x_1x_2\ldots x_n) \vee S(x_1x_2\ldots x_n) \}$

差：$R - S \equiv \{ x_1x_2\ldots x_n \mid R(x_1x_2\ldots x_n) \wedge \neg S(x_1x_2\ldots x_n) \}$

笛卡儿积：$R \times S \equiv \{ x_1x_2\ldots x_n y_1y_2\ldots y_m \mid R(x_1x_2\ldots x_n) \wedge S(y_1y_2\ldots y_m) \}$

投影：$\pi_{i1,\ldots,im}(R) \equiv \{ x_{i1}x_{i2}\ldots x_{im} \mid R(x_1x_2\ldots x_n) \}$

选择：$\sigma_F (R) \equiv \{ x_1x_2\ldots x_n \mid R(x_1x_2\ldots x_n) \wedge F' \}$

（F'是由F变化而形成的谓词公式）

$(\exists u)(\exists v)(R(zxu) \wedge W(yv) \wedge u>v$

2.6 小结

本章作业
P64 5，6（仅用关系代数完成）
发布一周内完成

- 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；
  - 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
- 关系模型概述
  - 关系数据结构及定义（关系，候选码，主码，外码，关系模式，关系数据库、关系数据库模式（型-值））。
  - 关系操作（查询、插入、删除、修改：集合操作）
  - 关系的完整性约束（实体完整性，参照完整性，用户定义的完整性）。
  - **关系代数**（5个基本运算（并，差，笛卡儿积，选择，投影）以及交，连接，自然连接，除）
  - 关系演算*（元组关系演算，域关系演算，关系运算安全性DOM）

关系代数随堂练习

考虑下面关系数据库：公司 com (cno, cname, city)

员工 emp (eno, ename, street, city, mno)

;mno 是外键，引用emp(eno)，表示该员工的直属上级经理

工作 works (eno, cno, salary)

请用关系代数表达式表示下面查询?

1）找出公司（编号cno为1001）的所有收入在10000元以上员工的姓名和居住城市。

2）找出与其直属上级经理居住在同一个城市的所有员工姓名和他的经理姓名。

3）找出比公司（ cno为1002）的所有员工收入都高的公司名和员工姓名。

4）假设公司可以在几个城市部署分部(cname相同，但cno不同)。找出在“DM”公司所在的各个城市都有分部的公司。

关系代数随堂练习

公司 com (cno, cname， city)
员工 emp (eno, ename, street, city, mno)
工作 works (eno, cno, salary)

1）找出公司（编号为1001）的所有收入在10000元以上员工的姓名和居住城市。

$$\pi_{ename,\ emp.city}\ (\sigma_{com.cno='1001' \wedge salary>10000}\ (com \bowtie works \bowtie emp))$$

2）找出与其经理居住在同一个城市的所有员工姓名和他的经理姓名。

$$\pi_{a.ename,\ b.ename}\ (\sigma_{a.city\ =\ b.city}\ (\rho_a\ (emp) \bowtie_{a.mno=b.eno} \rho_b\ (emp))\ )$$

关系代数随堂练习

公司 com (cno, cname， city)
员工 emp (eno, ename, street, city, mno)
工作 works (eno, cno, salary)

3）找出比公司（编号为1002）的所有员工收入都高的公司编号和员工编号。

$$\pi_{cno,eno}(\sigma_{salary > G_{max(salary)}(\sigma_{cno='1002'}(works))}(works))$$

4）假设公司可以在几个城市部署分部(cname相同，但cno不同)。找出在“DM”公司所在的各个城市都有分部的公司，输出公司名。

$$\pi_{cname,city}(com) \div \pi_{city}(\sigma_{cname='DM'}(com))$$