南京大学 NANJING UNIVERSITY

奋进行动 团结奋斗 争先进位 坐言起行 应势而动

数据管理基础

第2章 关系数据库

（关系代数的扩充运算）

智能软件与工程学院

- 回顾：关系模型 与 关系运算符
- 关系代数的基本运算与扩充运算
- 关系代数的应用

review: 关系数据结构 1

- 域（domain）是一组具有相同数据类型的值的集合

- 笛卡尔积（Cartesian Product）
  - 给定一组域$D_1, D_2, ..., D_n$，它们之间的笛卡尔积$D_1×D_2×…×D_n$是一个具有如下形式的'n元组'的集合：

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

  - 笛卡尔积中的元素$(d_1, d_2, ..., d_n)$被称为是一个'**n元组**'，也被称为'**元组**'或'**n元有序组**'

  - 元素$(d_1, d_2, ..., d_n)$中的每一个值d_i被称为是一个'**分量**'

review: 关系数据结构 2

❑关系（Relation）

- 给定一个域的序列 $D_1, D_2, \ldots, D_n$（其中可能存在相同的域），笛卡尔积 $D_1 \times D_2 \times \ldots \times D_n$ 的子集叫做在域 $D_1, D_2, \ldots, D_n$ 上的关系，关系中的元素被称为是关系中的元组。
- 关系是元组的集合，关系也可以被表示成一张‘二维表’。表中的一行(不包括第一行)对应关系中的一个元组，表中的一列对应一个域。
- 为了区分一个关系中的不同列，在关系模型中，关系中的一列被称为是关系中的一个‘属性’；在同一个关系中，不同的列具有不同的属性名。
- 关系表示：在域 $D_1, D_2, \ldots, D_n$ 上的 n 目关系 R 可以被表示如下：
  - 关系名：R
  - 属性名：$A_1, A_2, \ldots, A_n$
  - 关系模式：$R(A_1, A_2, \ldots, A_n)$
  - 元组：$t \in R$
  - 属性值/分量：$t[A_i] \in D_i$ 或者 $t[i] \in D_i$ $(i = 1,2,\cdots,n)$
  - 关系中的元组集合：R（也可用关系名来表示一个关系对应的元组集合）

review: 关系数据结构 3

- **关系的性质**

　　在关系数据模型中，‘关系’是属性域的笛卡尔积的一个有限子集，且必须满足以下性质：

① 列是同质的；
② 不同的列具有不同的属性名，不同的列可出自同一个域；
③ 列的无序性（属性的无序性）
④ 行的唯一性（元组的唯一性）
⑤ 行的无序性（元组的无序性）
⑥ 分量必须取原子值。

- 每个关系都可以被称为一张二维表，但只有同时满足上述六条性质的二维表才能被称为是‘关系’。

review: 关系数据结构 4

- **码(Key)、候选码 (Candidate key)**
  - 若关系R中的某一属性组K的值能唯一地标识关系R中的一个元组，并且K的所有真子集都不能，则称该属性组K为关系R的‘候选码’，简称‘码’。
  - 每一个关系中都存在‘候选码’；在一个关系中，也可能存在多个‘候选码’。
  - 如果‘候选码’是由关系中的所有属性构成的，这称为‘全码’（All-key）
- **主属性 与 非主属性/非码属性**
  - 候选码中的诸属性称为该关系的‘主属性’（Prime attribute）
  - 不包含在任何候选码中的属性称为该关系的‘非主属性’（Non-Prime attribute）或‘非码属性’（Non-key attribute）
- **主码 (Primary key)**
  - 在一个关系中，可以选择一个候选码作为该关系的‘主码’来定义。
  - ‘主码’是关系数据库管理系统（SQL）中才有的概念，在关系模型理论中，不需要为关系定义‘主码’。
  - 当我们在创建关系对应的‘基表’时，可以为基表定义‘主码’也可以不定义‘主码’。在一张基表中，最多只能定义一个主码。

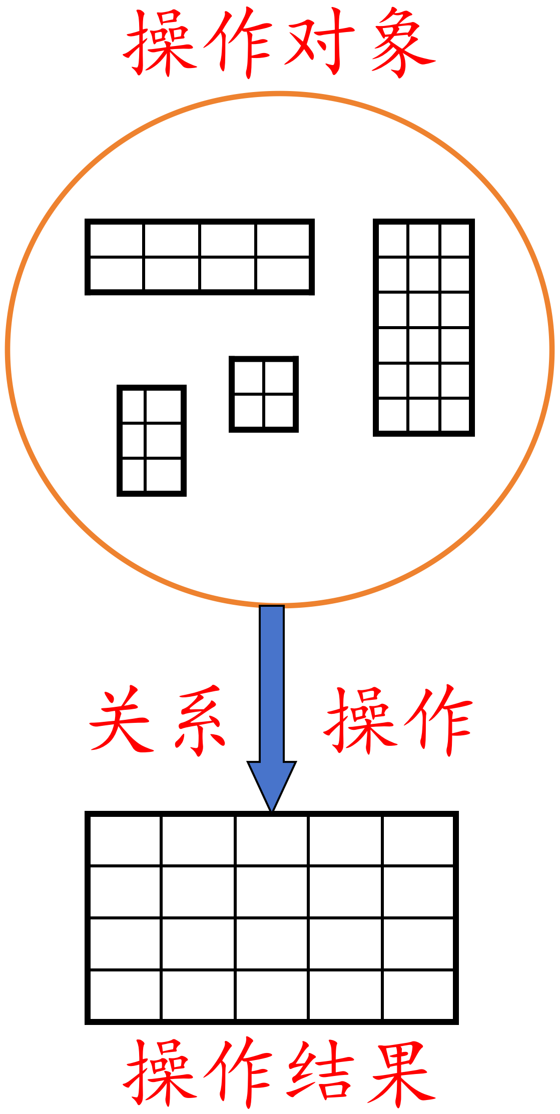
review: 关系数据结构 5

- 关系模式的形式化表示：$R\,(\,U, D, \mathrm{DOM}, F\,)$
  - **R** ：关系名
  - **U** ：组成该关系的属性名集合，$U = \{\,A_1, A_2, \cdots, A_n\,\}$
  - **D** ：U 中属性所来自的域，$D = \{D_1, D_2, \cdots, D_m\,\},\ m \le n$
  - **DOM** ：属性向域的映象集合（描述各个属性对应的域）

    $DOM(A_i) = D_j\ \ (1 \le i \le n, 1 \le j \le m)$
  - **F** ：属性间数据的依赖关系的集合（关系上的完整性约束条件）
- 关系模式通常可以简记为 $R\,(U)$ 或 $R\,(A_1,\ A_2,\ \cdots,\ A_n)$
- 关系模式是对关系组成结构的静态描述，是关系的‘型’，一般是稳定不变的；
- 关系是指关系模式在某一时刻的状态或内容，即关系中的元组集合，是关系的‘值’，是动态、随时间不断变化的。
- 用户在描述关系上的访问操作时，使用的是关系模式；访问结果取决于在执行时关系的‘值’。

review: 关系操作

- 操作对象是关系，操作结果也构成一个关系
- 关系模型上的数据操纵
  - 数据查询
  - （元组）插入、删除、修改
- 关系模型上的五种基本关系操作
  - （元组）选择
  - （属性）投影
  - （两个关系的）笛卡尔积
  - （两个关系的）并
  - （两个关系的）差



review: 关系的完整性

- **实体完整性**
  - 实体完整性（Entity Integrity）是指关系中元组的唯一性。
  - 在关系数据库管理系统中，实体完整性规则是指：若属性A是基本关系（基表）的主码中的属性，则属性A不能取‘空值’。
- **参照完整性**
  - 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），则关系R中每个元组在F上的取值必须是下列两种情况之一：
    - 空值（F的每个属性值均为空值）
    - 等于关系S中某个元组的主码值
- **用户定义的完整性**
  - 用户定义的完整性是特定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。

review: 关系代数

- 关系操作的特点
  - 操作对象和操作结果都是集合（关系）
- 关系操作的表示
  - 关系代数表达式：以关系为运算对象、以关系运算符为连接符
- 常用的关系运算符
  - 集合运算符：并、交、差、笛卡尔积
  - 专门的关系运算符
    - 选择、投影
    - 连接（包括 θ-连接，自然连接，外连接等）
    - 除
- 选择、投影、并、差、笛卡尔积是5种基本关系运算

review: 关系代数运算符

运 算 符		含 义
集合运算符	∪	并
	−	差
	∩	交
	×	笛卡尔积
专门的关系运算符	σ	选择
	π	投影
	⋈	连接
	÷	除

review: 使用的记号 1

- 设关系模式为 $R(A_1, A_2, \ldots, A_n)$
  - 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（属性值）
  - 若$A = \{A_{i1}, A_{i2}, \ldots, A_{ik}\}$，其中 $A_{i1}, A_{i2}, \ldots, A_{ik}$ 是 $A_1, A_2, \ldots, A_n$ 中的一部分，则A称为‘属性列’或‘属性组’或‘属性集’。
  - $t[A] = (t[A_{i1}], t[A_{i2}], \ldots, t[A_{ik}])$表示元组$t$在属性列$A$上诸分量的集合。
  - $\overline{A}$则表示$\{A_1, A_2, \ldots, A_n\}$中去掉$\{A_{i1}, A_{i2}, \ldots, A_{ik}\}$后剩余的属性组。

review: 使用的记号 2

- R为n目关系，S为m目关系。
  - $t_r \in R$，$t_s \in S$，$\widehat{t_r t_s}$ 称为元组的连接。
  - $\widehat{t_r t_s}$ 是一个 $n+m$ 列的元组，前n个分量来自于R中的一个元组，后m个分量来自于S中的一个元组。
  - 为方便表示，以后用 (t_r, t_s) 来表示元组的连接。
- 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
  - 用关系代数来表示，象集的定义是：$Z_x = \boldsymbol{\pi}_Z(\boldsymbol{\sigma}_{X=x}(R))$

review: 集合运算符 1

- 并(Union), 差(Difference), 交(Intersection)
  - 运算前提：R和S具有相同的关系模式（相同的目n、相应的属性取自同一个域）
  - 结果关系的关系模式：关系模式不变（仍为n目关系）
  - 结果关系的元组集合：
    - 由属于R或属于S的元组组成：$R \cup S = \{t \mid t \in R \vee t \in S\}$
    - 由属于R而不属于S的所有元组组成：$R - S = \{t \mid t \in R \wedge t \notin S\}$
    - 由既属于R又属于S的元组组成：$R \cap S = \{t \mid t \in R \wedge t \in S\}$
- 并运算和交运算满足交换律

  $R \cup S = S \cup R, \quad R \cap S = S \cap R$
- 交运算不满足交换律：$R - S \neq S - R$

- **the set of tuples in the relation** $T = R \cup S$

```
/* 首先将关系R中的元组加入结果关系T */
T := R
for each tuple t_s ∈ S
{
    /* 确保结果关系T中元组的唯一性 */
    if (t_s ∉ R)
    then  add tuple t_s into T
}
return T
```

review: 差

- the set of tuples in the relation $T = R - S$

```
/* 将结果关系T初始化为空集 */
T := { }
for each tuple t ∈ R
{
    /* 仅由属于关系R但不属于关系S的元组组成结果关系T */
    if (t ∉ S)
    then  add tuple t into T
}
return T
```

review: 交

- **the set of tuples in the relation $T = R \cap S$**

```
T := { }
for each tuple t ∈ R
{
    /* 由既属于关系R、又属于关系S的元组组成结果关系T */
    if (t ∈ S)
    then  add tuple t into T
}
return T
```

review: 集合运算符 2

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

$R - S$

A	B	C
a_1	b_1	c_1

$S - R$

A	B	C
a_1	b_3	c_2

$R \cap S$

A	B	C
a_1	b_2	c_2
a_2	b_2	c_1

review: 笛卡尔积 1

- 任意两个关系，都可以进行笛卡尔积运算
- 设：R是n目关系，S是m目关系
- 笛卡尔积 $R \times S$ 的计算结果是一个$(n+m)$ 目关系（假设为T）

```
T := { }
for each tuple t_r ∈ R
{
    for each tuple t_s ∈ S
    {
        add tuple (t_r, t_s) into T
    }
}
return  T
```

review: 笛卡尔积 2

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

❑选择（Selection）

根据给定的条件F从关系R中选出符合条件的元组

$$\sigma_F(R) = \{\, t \mid t \in R \wedge F(t) = \text{'真'} \,\}$$

选择条件F是一个逻辑表达式，基本形式为：$X_1\,\theta\,Y_1$，其中：

- ➢X_1和Y_1是关系R中的属性名或常量（至少一个是属性名）
- ➢ θ表示比较运算符，它可以是$>, \geq, <, \leq, =, <>$或$\neq$
- ➢在基本的选择条件上使用逻辑与、或、非运算，可以构造复杂的选择条件

```
T := { }
for each tuple t ∈ R
{
    /* 将元组t的属性值代入到条件表达式F中进行计算 */
    if  F(t) = true
    then  add tuple t into T
}
return  T
```

❑投影（Projection）

- 从R中选择出若干属性列组成新的关系（A是R中的属性列）

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

- 投影之后不仅过滤了原关系中的某些列，而且还可能消除某些元组（避免重复行），最后得到计算的结果关系

```
T := { }
for each tuple t ∈ R
{
    /* 投影过程中可能产生重复的结果元组 t[A] */
    if t[A] ∉ T
    then add tuple t into T
}
return T
```

- 在单个关系上，常常使用“选择+投影”来实现数据查询

[例] 查询2号课程的学生选修情况，结果返回选修学生的学号和成绩。

$$\pi_{Sno,\,Grade}\,(\sigma_{Cno\ =\ \text{‘2’}}\,(SC))$$

选修关系 **SC**

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

查询结果

学号 Sno	成绩 Grade
201215121	85
201215122	90

review: 选择 & 投影 4

❑结果元组的去重处理

[例] 查询年龄超过18岁的学生所在的系。

$$\pi_{Sdept}\left(\sigma_{Sage>18}(Student)\right)$$

学生关系 Student

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 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

查询结果

所在系 Sdept
CS
IS

θ-连接（θ-Join）

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

- 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$(R \times S)$中选取关系R在属性组A上的值与关系S在属性组B上的值满足比较关系θ的元组。

```
T := { }
for each tuple t_r ∈ R
{
    for each tuple t_s ∈ S
    {
        if (t_r[A] θ t_s[B]) is true then add tuple (t_r, t_s) into T
    }
}
return T
```

- **自然连接（Natural join）**
  - 关系R和S含有相同的属性组B，U是两个关系所有属性的并集（包括相同的属性组B）

$$R \bowtie S = \{ \widehat{t_r t_s}[U-B] \mid t_r \in R \wedge t_s \in S \wedge (t_r[B] = t_s[B]) \}$$

- **设关系R的属性被划分为A和B两个属性组，关系S的属性被划分为B和C两个属性组**

```
T := { }
for each tuple t_r ∈ R
{
    for each tuple t_s ∈ S
    {
        if (t_r[B] = t_s[B]) is true
        then add tuple (t_r[A], t_r[B], t_s[C]) into T
    }
}
return T
```

[例] 查询与张清玫老师同在一个院系的学生的学号和姓名。

❑分析：

- 本查询从找到的教师元组出发，查出与这些教师同在一个院系的学生，查询涉及到学生和教师两个关系，并根据他们所在系的名称实现跨关系的数据查询。
- 虽然都是系别名称，但在两个关系中采用了不同的属性名，所以采用θ-连接

$$\pi_{Sno,\, Sname}(\sigma_{Sname\ =\ \text{‘张清玫’}}(Teacher \underset{Tdept\ =\ Sdept}{\bowtie} Student))$$

学生关系 Student

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 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

教师关系 Teacher

工号 Tno	姓名 Tname	所在系 Tdept
907811	张清玫	CS
853609	刘逸	IS

[例2.12] 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

分析:

- 本查询从课程出发，查找选修过该课程的学生姓名，查询过程需要涉及到三个关系
- 需要根据课程号实现从课程到选修关系的查找，根据学号实现从选修到学生关系的查找，恰好在它们之间只有课程号和学号这一个同名属性，所以可使用自然连接

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

学生关系 Student

学号 Sno	姓名 Sname	性别 Ssex	年龄 Sage	所在系 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

课程关系 Course

课程号 Cno	课程名 Cname	先行课 Cpno	学分 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

选修关系 SC

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

review: 外连接 1

- 悬浮元组（Dangling tuple）
  - 两个关系R和S在做自然连接时，关系R中某些元组有可能在S中不存在公共属性上值相等的元组，从而造成R中这些元组在操作时被舍弃了，这些被舍弃的元组称为悬浮元组。
- 外连接（Outer Join）
  - 如果把悬浮元组也保存在结果关系中，而在其他属性上填空值(Null)，就叫做外连接
  - 左外连接（ LEFT OUTER JOIN 或 LEFT JOIN ）
    - 只保留左边关系R中的悬浮元组
  - 右外连接（ RIGHT OUTER JOIN 或 RIGHT JOIN ）
    - 只保留右边关系S中的悬浮元组

review: 外连接 2

R

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

S

B	E
b_1	3
b_2	7
b_3	10
b_3	2
b_5	2

图(a) *R* outer join *S*

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

图（b） *R* left join *S*

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

图（c） *R* right join *S*

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

❑除运算（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：

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

  其中：Y_x是x在R中的象集，$x = t_r[X]$

review: 除运算 2

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

S

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

R÷S

A
a_1

- 在除法表达式 $R \div S$ 中，分子上的关系R被称为‘被除数关系’，分母上的关系S被称为‘除数’关系
- 在其他教材中，对于除法的描述是：**当除数关系S的关系模式是被除数关系R的一个真子集时，$R \div S$ 才是合法有意义的！**
  - 本题查询可表示为：$R \div \pi_{B,C}(S)$

- 回顾：关系模型 与 关系运算符
- 关系代数的基本运算与扩充运算
- 关系代数的应用

关系代数的基本运算与扩充运算

- 在关系代数中，并、差、笛卡尔积、选择、投影等五个运算符是基本运算，其他运算符都是扩充运算。
- 关系完备系统
  - 采用关系数据结构，并且支持并、差、笛卡尔积、选择、投影等五个关系代数的基本运算符的系统，被称为是关系完备的系统。
  - 一个关系完备的系统，能够支持所有的关系操作。
- 接下来
  - 赋值运算符
  - 扩充运算符的推导公式

关系模式的语义 1

- 在关系模型理论中
  - 一般通过属性名来确定一个属性的语义
    - 同名同义：‘属性名相同’表示具有‘相同的属性域和相同的语义’
    - 异名异义：‘属性名不同’表示具有‘不同的属性域或不同的语义’
  - 在后续的关系代数表示中，均采用“同名同义、异名异义”的约定
  - 但是，属性名也仅仅是用于标识关系中某一列的标识符。在不同关系之间，也有可能存在属性的‘同名异义’和‘同义异名’现象，这个时候可以通过赋值运算对关系中的属性进行重命名，使其符合“同名同义、异名异义”的要求。

关系模式的语义 2

- 有如下所示的五个关系，其中：
  - R与S_1具有相同的关系模式（相同的目n、相应的属性'同名同义'）
  - R与S_2具有不同的关系模式（属性A的'同名异义'）
  - R与S_3具有相同的关系模式（列的无序性）
  - R与S_4可以被认为是不同的关系模式（属性的'异名异义'），也可以被作为相同的关系模式（相同的目、相应列对应相同的域）

A	B	C
a_1	b_1	c_1
a_1	b_2	c_3
a_2	b_1	c_2

关系 R

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_3
a_3	b_2	c_3

关系 S_1

A	B	C
1	b_1	c_1
1	b_1	c_2
1	b_2	c_3
3	b_2	c_3

关系 S_2

A	C	B
a_1	c_1	b_1
a_1	c_2	b_1
a_1	c_3	b_2
a_3	c_3	b_2

关系 S_3

A	B	D
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_3
a_3	b_2	c_3

关系 S_4

关系代数中的 ‘赋值’ 运算

- 赋值运算的一般表示方法：$R(A_1, A_2, \ldots, A_n) := \langle expression \rangle$
  - $:=$ 是赋值运算符
  - 右边的 $\langle expression \rangle$ 是一个关系代数表达式（用于表示在关系数据库上的数据操纵请求），或者是一个当前存在的关系；
  - 左边的 $R(A_1, A_2, \ldots, A_n)$ 是 $\langle expression \rangle$ 的计算结果（即数据操纵返回的结果关系）

上述的赋值表达式可以理解为：

① 为右边的数据操纵请求定义了一个查询关系，用以保存其操纵结果；

② R是该查询关系的关系名，$A_1, A_2, \ldots, A_n$是查询关系的属性名；

③ 查询关系也可以被作为访问对象，参与后续关系代数表达式的表示。

赋值运算的使用方式

- 方式一：$R(A_1, A_2, \ldots, A_n) := \langle expression \rangle$
  - 计算右边的关系代数表达式$\langle expression \rangle$，并将其结果关系保存下来形成临时的查询关系$R$
  - 查询关系中的属性名，可以采用它们在右边表达式中的原属性名，也可以对查询关系中的部分属性进行重命名，但必须保证在查询关系R中不会出现属性同名现象。
- 方式二：$R := \langle expression \rangle$
  - 将右边的关系代数表达式$\langle expression \rangle$的计算结果保存下来形成临时的查询关系$R$
  - R是查询关系的关系名，其属性名直接沿用它们在右边表达式中的原属性名，但必须保证在查询关系R中不会出现属性同名现象。

赋值运算的用途

- 对一个关系及关系中的属性进行重命名 (alias)，以方便‘关系的自联接’或者复杂查询的表示。
- 表的赋值：$S := R$
  - 简单的‘表的赋值’操作，可用于为关系R另起一个别名 (alias)，或者理解为：定义了一个与关系R‘完全相同’的中间关系S（关系名不同，但具有相同的关系模式和元组集合）
  - 也可以在赋值表达式中，对生成的中间关系S中的属性进行重命名（类似于前一页的‘方式一’）
  - 通常用于‘关系自联接’的表示，或者有目的地对关系中的某些属性进行重命名！
- 赋值运算也可以用来保存计算的‘中间结果’，方便某些复杂查询的表示。

关系代数中的扩充运算：交、θ-连接、自然连接

- 交运算推导公式

$$R \cap S = R - (R - S) = S - (S - R)$$

- θ-连接推导公式

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

$$= \sigma_{R.A=S.B}(R \times S)$$

- 自然连接推导公式
  - 设关系R的属性被划分为A和B两个属性组，关系S的属性被划分为B和C两个属性组

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

关系代数中的扩充运算：除 1

- 除运算：R ÷ S
- '除'运算是关系代数中的一个扩充运算，为什么要引入 '除' 运算？
  - 为了方便表示某类查询："选修过所有课程的学生"

选修关系 SC

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

- 请比较下述查询的区别：
  ① 选修过1号课程的学生学号；
  ② 查询至少选修1号课程和3号课程的学生学号；
  ③ 选修过课程关系中的所有课程的学生学号；

关系代数中的扩充运算：除 2

❑除运算： R ÷ S

- 用$Head(R)$表示关系R的关系模式
- 假设关系R和S的关系模式分别为：

$$Head(R) = \{A_1, A_2, ..., A_n, B_1, B_2, ..., B_m\}$$
$$Head(S) = \{B_1, B_2, ..., B_m\}$$

- 其中：
  - R 被称为'被除数关系'
  - S 被称为'除数关系'
  - $R \div S$的结果关系 被称为'商'

关系代数中的扩充运算：除 3

- 结果关系：$T = R \div S = \{ t_r[X] \mid t_r \in R \land \pi_Y(S) \subseteq Y_x \}$
  - 结果关系的关系模式
    $Head(T) = Head(R) - Head(S) = \{A_1, A_2, \dots, A_n\} = X$
  - 结果关系中的元组
    - 满足条件的元组t_r在 X 属性列上的投影：元组t_r在X上分量值 x 的象集Y_x包含S在Y上投影的集合
    - 根据前一页的假设，在除数关系S中只有属性组 Y 即 $head(S)$，"S在Y上投影的集合"就是关系S
    - 即：

      假设 $\boldsymbol{x}$ 是结果关系 $\boldsymbol{T}$ 中的一个元组 $(\boldsymbol{x} \in \boldsymbol{T})$，则对于除数关系 $\boldsymbol{S}$ 中的每一个元组 $\boldsymbol{y}$ 都有 $(\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{R}$
  - 由所有符合上述条件的元组 $\boldsymbol{x}$ 构成结果关系T的元组集合。

Result of T = R÷S

1. Assume a row x is in *T*, then:

```
for each row y in S
{
    we can find a row z in R, and
    {
```

$$z[A_1, A_2, \cdots A_n] = x[A_1, A_2, \cdots A_n]$$
$$z[B_1, B_2, \cdots B_m] = y[B_1, B_2, \cdots B_m]$$

```
    }
}
```

2. *T* contains <u>the largest possible set</u> of rows x

除运算的例子 1

关系 R

A	B	C
a_1	b_1	c_1
a_2	b_1	c_1
a_1	b_2	c_1
a_1	b_2	c_2
a_2	b_1	c_2
a_1	b_2	c_3
a_1	b_2	c_4
a_1	b_1	c_5

S_1

C
c_1

$R \div S_1$ → T_1

A	B
a_1	b_1
a_2	b_1
a_1	b_2

S_2

C
c_2

$R \div S_2$ → T_2

A	B
a_1	b_2
a_2	b_1

S_3

C
c_1
c_2

$R \div S_3$ → T_3

A	B
a_1	b_2
a_2	b_1

除运算的例子 2

关系 R_0

A	B	C
a_1	b_1	c_1
a_2	b_1	c_1
a_1	b_2	c_1
a_1	b_2	c_2
a_2	b_1	c_2
a_1	b_2	c_3
a_1	b_2	c_4
a_1	b_1	c_5
a_2	b_2	c_2

S_1

C
c_1

$R_0 \div S_1$

T_{01}

A	B
a_1	b_1
a_2	b_1
a_1	b_2

S_2

C
c_2

$R_0 \div S_2$

T_{02}

A	B
a_1	b_2
a_2	b_1
a_2	b_2

S_3

C
c_1
c_2

$R_0 \div S_3$

T_{03}

A	B
a_1	b_2
a_2	b_1

除运算的例子 3

关系 R

A	B	C
a_1	b_1	c_1
a_2	b_1	c_1
a_1	b_2	c_1
a_1	b_2	c_2
a_2	b_1	c_2
a_1	b_2	c_3
a_1	b_2	c_4
a_1	b_1	c_5

S_4

C
c_1
c_2
c_3
c_4

$$R \div S_4$$

T_4

A	B
a_1	b_2

S_5

C
c_1
c_2
c_3
c_4
c_5

$$R \div S_5$$

T_5

A	B

除运算的例子 4

关系 R

A	B	C
a_1	b_1	c_1
a_2	b_1	c_1
a_1	b_2	c_1
a_1	b_2	c_2
a_2	b_1	c_2
a_1	b_2	c_3
a_1	b_2	c_4
a_1	b_1	c_5

S_6

B	C
b_1	c_1

$R \div S_6$

T_6

A
a_1
a_2

S_7

B	C
b_1	c_1
b_2	c_1

$R \div S_7$

T_7

A
a_1

除运算的例子 5

R

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

S_1

C	D
3	4
5	6

$R \div S_1$

T_1

A	B
1	2
7	8

S_2

C	D
3	4

$R \div S_2$

T_2

A	B
1	2
7	8

S_3

C	D
3	4
5	6
4	2

$R \div S_3$

T_3

A	B
1	2

例：找出修读课程关系C中的所有课程的学生的学号

SC

sno	cno
s_1	c_1
s_1	c_2
s_2	c_1
s_2	c_2
s_2	c_3
s_3	c_2

C

cno
c_1
c_2
c_3

$SC \div C$ →

T

sno
s_2

【思考】下述查询分别可以有哪几种关系代数查询的表示方法?

① 找出选修了c_1和c_2这两门课程的学生的学号?

② 找出选修了c_2这门课程的学生的学号?

例：找出修读课程关系C中的所有课程的学生的学号（cont.）

SC

sno	cno	G
s_1	c_1	80
s_1	c_2	85
s_2	c_1	90
s_2	c_2	70
s_2	c_3	85
s_3	c_2	85

C

cno
c_1
c_2
c_3

【思考】在选课关系SC中增加一个属性‘成绩 G’，然后执行 SC÷C，那么：

① 结果关系的关系模式是什么？

② 结果关系中的元组又有哪些？

除运算与笛卡儿积的关系 1

- 在‘被除数关系’完全包含‘除数关系’中的所有属性的前提下，‘除’运算与‘笛卡儿积’运算存在如下的相关性

  - 如果 $R = T \times S$，那么有：

    $T = R \div S$

    $S = R \div T$

  - 如果 $T = R \div S$，那么有：

    $T \times S \subseteq R$

除运算与笛卡儿积的关系 2

T

A	B
a_1	b_1
a_2	b_2

S

C	D
c_1	d_1
c_2	d_2
c_3	d_3

$T \times S =$

R

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_1	b_1	c_3	d_3
a_2	b_2	c_1	d_1
a_2	b_2	c_2	d_2
a_2	b_2	c_3	d_3

$T = R \div S$ $\quad\quad$ $S = R \div T$

除运算与笛卡儿积的关系 3

R

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_1	b_1	c_3	d_3
a_1	b_2	c_1	d_1
a_2	b_2	c_1	d_1
a_2	b_2	c_2	d_2
a_2	b_2	c_3	d_3
a_2	b_1	c_2	d_2

S

C	D
c_1	d_1
c_2	d_2
c_3	d_3

$T = R \div S$

A	B
a_1	b_1
a_2	b_2

$R' := T \times S$

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_2	d_2
a_1	b_1	c_3	d_3
a_2	b_2	c_1	d_1
a_2	b_2	c_2	d_2
a_2	b_2	c_3	d_3

$R' \subseteq R$

除法的推导公式 1

- '除'运算是关系代数中的一个扩充运算，它的推导公式如下

- 设关系R的属性集为$\{A_1,...,A_n,B_1,...,B_m\}$，关系$S$的属性集为$\{B_1,...,B_m\}$，则：

$$R \div S = \boldsymbol{\pi_{A_1,\ldots,A_n}}(\mathrm{R}) - \boldsymbol{\pi_{A_1,\ldots,A_n}}((\boldsymbol{\pi_{A_1,\ldots,A_n}}(\mathrm{R}) \times \mathrm{S}) - \mathrm{R})$$

其推导过程如下：

除法的推导公式 2

① $T_{max} := \pi_{A_1,\ldots,A_n}(R)$

// T_{max}是最大可能的结果元组集合

② $R_{max} := T_{max} \times S$

// R_{max}与关系R是同类关系

③ $T_1 := R_{max} - R$

④ $T_2 := \pi_{A_1,\ldots,A_n}(T_1)$

// T_2是关系T_{max}中不满足除运算的结果要求的那些元组，即：对于关系T_2中的任一个元组q，至少能在关系S中找到一个元组s，使得由元组q和s所构成的元组(q,s)不在关系R中出现。

⑤ $R \div S := T_{max} - T_2$

除运算的推导过程（1） SC÷C

SC

sno	cno
s_1	c_1
s_1	c_2
s_2	c_1
s_2	c_2
s_2	c_3
s_3	c_2

C

cno
c_1
c_2
c_3

$\mathrm{T}_{\mathrm{max}}$

sno
s_1
s_2
s_3

SC_{max}

sno	cno
s_1	c_1
s_1	c_2
s_1	c_3
s_2	c_1
s_2	c_2
s_2	c_3
s_3	c_1
s_3	c_2
s_3	c_3

没有出现在SC中的元组，其中的sno不符合‘除’运算对结果元组的要求。

$\pi_{sno}(SC_{max} - SC)$

sno
s_1
s_3

结果关系T

sno
s_2

除运算的推导过程（2） R÷C

- 如果在被除数关系中增加一个属性G，那么除法的结果语义就会产生改变！

R

sno	cno	G
s_1	c_1	80
s_1	c_2	85
s_2	c_1	90
s_2	c_2	70
s_2	c_3	85
s_3	c_2	85

C

cno
c_1
c_2
c_3

$T_{max} = \pi_{sno,G}(R)$

sno	G
s_1	80
s_1	85
s_2	90
s_2	70
s_2	85
s_3	85

R

sno	cno	G
s_1	c_1	80
s_1	c_2	85
s_2	c_1	90
s_2	c_2	70
s_2	c_3	85
s_3	c_2	85

C

cno
c_1
c_2
c_3

$T_{max} = \pi_{sno,G}(R)$

sno	G
s_1	80
s_1	85
s_2	90
s_2	70
s_2	85
s_3	85

$R_{max} = T_{max} \times C$

sno	cno	G
s_1	c_1	80
s_1	c_2	80
s_1	c_3	80
s_1	c_1	85
s_1	c_2	85
s_1	c_3	85
s_2	c_1	90
s_2	c_2	90
s_2	c_3	90
s_2	c_1	70
s_2	c_2	70
s_2	c_3	70
s_2	c_1	85
s_2	c_2	85
s_2	c_3	85
s_3	c_1	85
s_3	c_2	85
s_3	c_3	85

- 从R_{max}中，我们无法判断哪些学生是选修过课程关系C中的所有三门课！
- 直接用关系R作为被除数，无法准确表示“选修过关系C中的所有课程”的查询语义，正确的表示方法是：$\pi_{sno,cno}(R) \div C$

- 回顾：关系模型 与 关系运算符
- 关系代数的基本运算与扩充运算
- 关系代数的应用