第二章　关系数据库（续）

第二章 关系数据库

2.4 关系代数

- 概述
- 传统的集合运算
- 专门的关系运算

概　述

1. 关系代数
2. 运算的三要素
3. 关系代数运算的三个要素
4. 关系代数运算的分类
5. 表示记号

概　述

1.关系代数

一种抽象的查询语言

用对关系的运算来表达查询

概　述(续)

2. 关系代数运算的三个要素

运算对象：关系

运算结果：关系

运算符：四类

概　述(续)

- 集合运算符
  - 将关系看成元组的集合
  - 运算是从关系的“水平”方向即行的角度来进行
- 专门的关系运算符
  - 不仅涉及行而且涉及列
- 算术比较符
  - 辅助专门的关系运算符进行操作
- 逻辑运算符
  - 辅助专门的关系运算符进行操作

概 述(续)

表2.4 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
	×	广义笛卡尔积		≤	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	
专门的关系运算符	σ	选择	逻辑运算符	¬	非
	Π	投影		∧	与
	⋈	连接		∨	或
	÷	除			

概　述(续)

4．关系代数运算的分类

传统的集合运算

并、差、交、广义笛卡尔积

专门的关系运算

选择、投影、连接、除

概　述(续)

5. 表示记号

（1） R，t∈R，t[A_i]

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 。

概　述(续)

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

概　述(续)

- （3） $\widehat{t_r t_s}$

  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。它是一个$n+m$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。

概　述(续)

◆ 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：

$$Z_x=\{t[Z]|t \in R，t[X]=x\}$$

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 关系代数

- 概述
- 传统的集合运算
- 专门的关系运算

2.4.1 传统的集合运算

- 并
- 差
- 交
- 广义笛卡尔积

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
- $R \cup S$
  - 仍为***n***目关系，由属于***R***或属于***S***的元组组成

$$R \cup S = \{ t | t \in R \vee t \in S \}$$

并(续)

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
a1	b3	c2
a2	b2	c1

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

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

差(续)

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

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

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

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

交 (续)

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

$R \cap S$

A	B	C
a1	b2	c2
a2	b2	c1

4. 广义笛卡尔积（Extended Cartesian Product）

- **R**
  - ***n***目关系，k_1个元组
- **S**
  - ***m***目关系，k_2个元组
- $R \times S$
  - 列：（***n+m***）列的元组的集合
    - **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：$k_1 \times k_2$个元组
    - $R \times S = \{t_r\ t_s \mid t_r \in R \wedge t_s \in S\}$

广义笛卡尔积 (续)

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

$R \times S$

A	B	C	A	B	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 关系代数

- 概述
- 传统的集合运算
- 专门的关系运算

2.4.2 专门的关系运算

- 选择
- 投影
- 连接
- 除

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

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

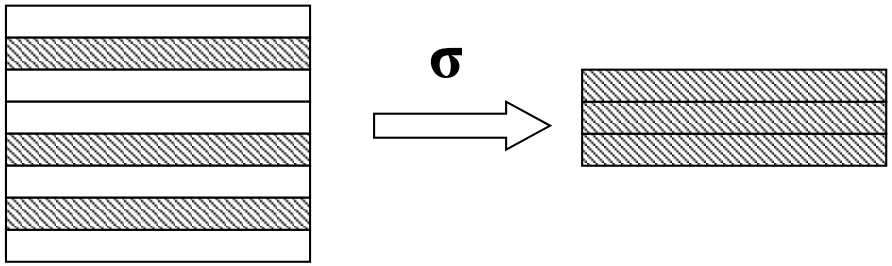
  - F：选择条件，是一个逻辑表达式，基本形式为：

$$[\neg(\ ]\ X_1 \theta Y_1\ [\ )][\varphi\ [\neg(\ ]\ X_2 \theta Y_2\ [\ )]]\ldots$$

    - θ：比较运算符（＞，≥，＜，≤，＝或<>）
    - X_1，Y_1等：属性名、常量、简单函数；属性名也可以用它的序号来代替；
    - φ：逻辑运算符（∧或∨）
    - [ ]：表示任选项
    - …：表示上述格式可以重复下去

选择（续）

- 3) 选择运算是从行的角度进行的运算



- 4) 举例

设有一个学生-课程数据库，包括学生关系**Student**、课程关系**Course**和选修关系**SC**。

选择（续）

学 号 Sno	姓 名 Sname	性 别 Ssex	年 龄 Sage	所在系 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

Student

(a)

例1 例2 例3 例4 例9

选择（续）

课程号	课程名	先行课	学分
Cno	Cname	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
7	PASCAL语言	6	4

Course

例9

(b)

选择（续）

学 号	课程号	成 绩
Sno	Cno	Grade
95001	1	92
95001	2	85
95001	3	88
95002	2	90
95002	3	80

SC

(c)

例7　例9

选择（续）

[例1] 查询信息系（IS系）全体学生

$$\sigma_{Sdept = 'IS'}(Student)$$

或 $$\sigma_{5 ='IS'}(Student)$$

结果：

Sno	Sname	Ssex	Sage	Sdept
95002	刘晨	女	19	IS
95004	张立	男	19	IS

选择（续）

[例2] 查询年龄小于20岁的学生

$\sigma_{Sage < 20}(Student)$

或 $\sigma_{4 < 20}(Student)$

结果：

Sno	Sname	Ssex	Sage	Sdept
95002	刘晨	女	19	IS
95003	王敏	女	18	MA
95004	张立	男	19	IS

2. 投影（Projection）

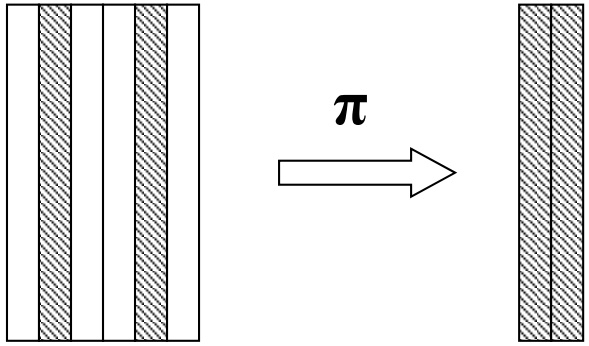
- 1）投影运算符的含义
  - 从***R***中选择出若干属性列组成新的关系

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

***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）

- 3) 举例

[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

$\pi_{Sname，Sdept}(Student)$

或 $\pi_{2，5}(Student)$

结果：

投影（续）

Sname	Sdept
李勇	CS
刘晨	IS
王敏	MA
张立	IS

投影（续）

[例4] 查询学生关系Student中都有哪些系

$\pi_{Sdept}(Student)$

结果：

Sdept
CS
IS
MA

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

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

    - ***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*×*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的元组。

连接(续)

- 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：

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

连接(续)

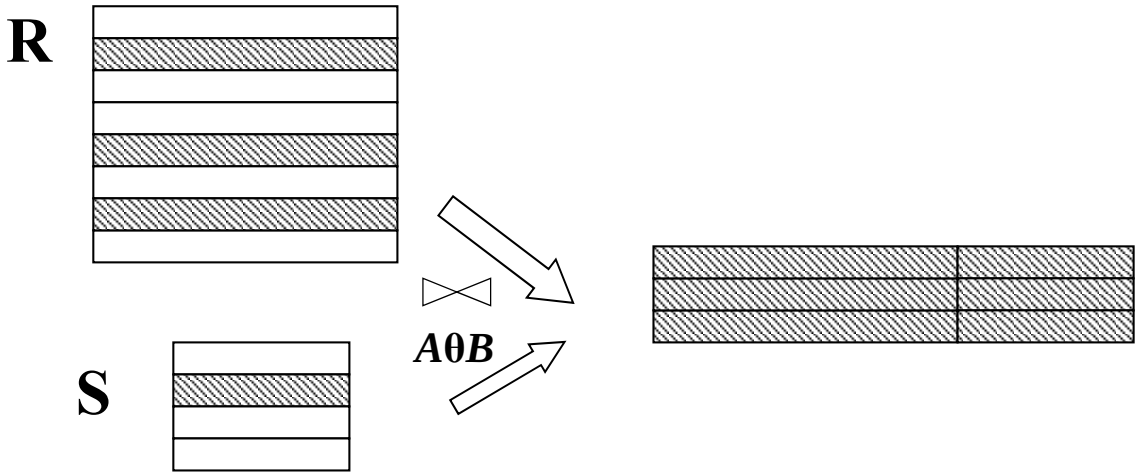
- 自然连接（Natural join）
  - 什么是自然连接
    - 自然连接是一种特殊的等值连接
      - ✓ 两个关系中进行比较的分量必须是相同的属性组
      - ✓ 在结果中把重复的属性列去掉
  - 自然连接的含义

    R*和*S*具有相同的属性组*B

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

连接(续)

- 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。

连接(续)

- 5）举例

[例5]

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

R

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

S

连接(续)

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

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(续)

等值连接 $R \underset{R.B=S.B}{\bowtie} S$

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(续)

自然连接 $R \bowtie S$

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

连接运算的类型

内连接(inner join)： 将公共属性值相等的记录连接起来组成新关系，且不去除重复属性。

自然连接(natural join)：将公共属性相等的记录连成新表，且去除重复属性。

左外连接(left outer join)：新表中包括公共属性相等的记录和左表中其它记录 。

右外连接(right outer join)：新表中包括公共属性相等的记录和右表中其它记录。

全外连接(full outer join)：左联与右联的全集。

内连接

例：将班级关系R1与学生关系R2进行内连接运算，构成新关系R3。

R1

班级编号	班级名称
10102	01工商
10101	00工商
10201	01经济

R2

学号	姓名	班级编号
012134	李长江	10102
012133	江利利	10102
012136	何光明	10201
002321	方虹	10101

R3

有重复列

班级编号	班级名称	学号	姓名	班级编号
10102	01工商	012134	李长江	10102
10102	01工商	012133	江利利	10102
10101	00工商	002321	方虹	10101
10201	01经济	012136	何光明	10201

内连接：根据公共属性班级编号

将R1与R2连接起来

新表中只有两表共同班级号的记录

自然连接

例：将班级关系R1与学生关系R2进行自然连接运算，构成新关系R3。

R1

班级编号	班级名称
10102	01工商
10101	00工商
10201	01经济

R3

班级编号	班级名称	学号	姓名
10102	01工商	012134	李长江
10102	01工商	012133	江利利
10101	00工商	002321	方虹
10201	01经济	012136	何光明

无重复列

R2

学号	姓名	班级编号
012134	李长江	10102
012133	江利利	10102
012136	何光明	10201
002321	方虹	10101

自然连接：根据公共属性班级编号将R1连接起来R2

1、只有两表中共同班级号的记录

2、无重复列

左外连接

以左表为主线，依次从左表中取一个记录，在右表中找与公共属性之相匹配的记录，若找到，则在新表中连接成一条新记录，若找不到，则只将左表中数据形成新记录。因此，在新关系中，除公共属性相等的记录以外，还保留左边关系与连接条件不匹配的记录。

左表中不匹配的保留

R1

班级编号	班级名称
10102	01工商
11111	00工商
10201	01经济

R2

学号	姓名	班级编号
012134	李长江	10102
012133	江利利	10102
012136	何光明	10201
002333	李芳	12121

右表中不匹配的舍去

不匹配记录

R3

班级编	班级名称	学号	姓名
1010	01工商	012134	李长江
10102	01工商	012133	江利利
11111	00工商		
10201	01经济	012136	何光明

左外连接：

将左表R1与右表R2连接

1、有两表中共同班级号的记录

2、还有左表中不匹配的记录

右外连接

在新关系中，除公共属性相等的记录以外，还保留右边关系与连接条件不匹配的记录。

左表中不匹配的舍去

R1

班级编号	班级名称
10102	01工商
11111	00工商
10201	01经济

R2

学号	姓名	班级编号
012134	李长江	10102
012133	江利利	10102
012136	何光明	10201
002333	李芳	12121

以右表为主线，将右表中每一记录与左表中所有记录进行比较，相同则连接成新记录。

右表中不匹配的保留

不匹配记录

R3

班级编号	班级名称	学号	姓名
10102	01工商	012134	李长江
10102	01工商	012133	江利利
10201	01经济	012136	何光明
12121	null	002333	李芳

右外连接：

将左表R1与右表R2连接

1、有两表中共同班级号的记录

2、还有右表中不匹配的记录

全外连接

在新关系中，除公共属性相等的记录以外，还保留右边关系和右边关系与连接条件不匹配的记录。

左表中不匹配的

R1

班级编号	班级名称
10102	01工商
11111	00工商
10201	01经济

R2

学号	姓名	班级编号
012134	李长江	10102
012133	江利利	10102
012136	何光明	10201
002333	李芳	12121

右表中不匹配的保留去

R3

班级编号	班级名称	学号	姓名
10102	01工商	012134	李长江
10102	01工商	012133	江利利
11111	00工商	null	null
10201	01经济	012136	何光明
12121	null	002333	李芳

全外连接：

将左表R1与右表R2连接

1、有两表中共同班级号的记录

2、还有两表中不匹配的记录

4. 除（Division）

给定关系$R(X, Y)$和$S(Y, Z)$，其中X，Y，Z为属性组。

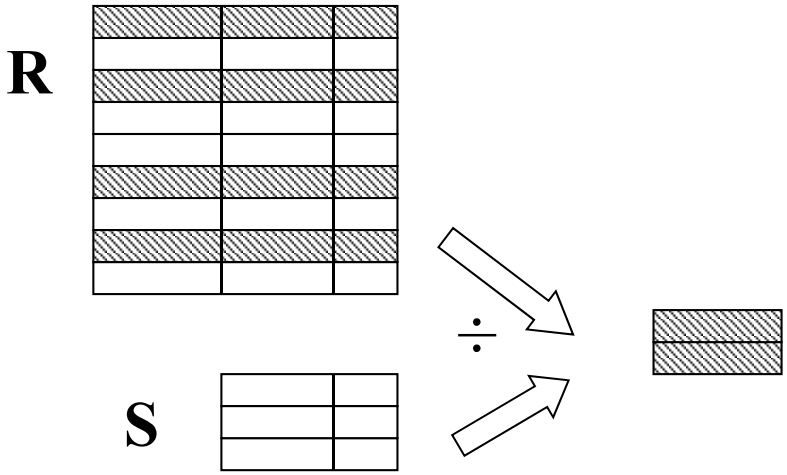
R中的Y与S中的Y可以有不同的属性名，但必须出自相同的域集。R与S的除运算得到一个新的关系$P(X)$，P是R中满足下列条件的元组在X属性列上的投影：元组在X上分量值x的象集Y_x包含S在Y上投影的集合。

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

Y_x：x在R中的象集，$x = t_r[X]$

除(续)

- 2）除操作是同时从行和列角度进行运算



- 3）举例

[例6]

除(续)

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

$R \div S$
A
a_1

分析：

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 $\{(b_6, c_6)\}$

S在(B，C)上的投影为

{(b1，c2)，(b2，c1)，(b2，c3) }

只有a_1的象集包含了S在(B，C)属性组上的投影

所以　$R \div S = \{a_1\}$

5. 综合举例

以学生-课程数据库为例 (P.59)

[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*：

Cno
1
3

然后求：$\pi_{Sno,Cno}(SC) \div K$

综合举例(续)

- 例 7续　$\pi_{Sno,Cno}(SC)$

  95001象集{1，2，3}

  95002象集{2，3}

  $\pi_{Cno}(K)$={1，3}

  于是：$\pi_{Sno,Cno}(SC) \div K$={95001}

Sno	Cno
95001	1
95001	2
95001	3
95002	2
95002	3

综合举例(续)

[例 8] 查询选修了2号课程的学生的学号。

$$\pi_{Sno}(\sigma_{Cno='2'}(SC)) = \{95001, 95002\}$$

综合举例(续)

[例9] 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。

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

或

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

或

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

综合举例(续)

[例10] 查询选修了全部课程的学生号码和姓名。

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

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

小结(续)

- 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
- 典型关系代数语言
  - **ISBL（Information System Base Language）**
    - **由IBM United Kingdom研究中心研制**
    - **用于PRTV（Peterlee Relational Test Vehicle）实验系统**

第二章 关系数据库

2.5 关系演算

- **关系演算**

  以数理逻辑中的谓词演算为基础

- **种类：按谓词变元不同分类**

  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**

2.5.1 元组关系演算语言ALPHA

- 由E.F.Codd提出

  INGRES所用的QUEL语言是参照ALPHA语言研制的

- 语句

  检索语句

  - **GET**

  更新语句

  - **PUT，HOLD，UPDATE，DELETE**

一、检索操作

- **语句格式：**

  GET　工作空间名 [（定额）]（表达式1）

  [：操作条件] [DOWN/UP 表达式2]

  定额：规定检索的元组个数

  - **格式：　　数字**

  表达式1：指定语句的操作对象

  - **格式：**

    关系名| 关系名. 属性名| 元组变量. 属性名| 集函数

    [，…]

  操作条件：将操作结果限定在满足条件的元组中

  - **格式：　　逻辑表达式**

  表达式2：指定排序方式

  - **格式：　　关系名. 属性名| 元组变量. 属性名[，…]**

检索操作 (续)

(1) 简单检索(即不带条件的检索)

(2) 限定的检索(即带条件的检索)

(3) 带排序的检索

(4) 带定额的检索

(5) 用元组变量的检索

(6) 用存在量词的检索

检索操作 (续)

(7) 带有多个关系的表达式的检索

(8) 用全称量词的检索

(9) 用两种量词的检索

(10) 用蕴函（Implication）的检索

(11) 集函数

（1）简单检索

GET　工作空间名 （表达式1）

[例1] 查询所有被选修的课程号码。

GET W (SC.Cno)

[例2] 查询所有学生的数据。

GET W (Student)

（2）限定的检索

格式

GET 工作空间名（表达式1）：操作条件

[例3] 查询信息系(IS)中年龄小于20岁的学生的学号和年龄。

GET W (Student.Sno , Student.Sage):

Student.Sdept='IS'∧ Student.Sage<20

（3）带排序的检索

格式

GET 工作空间名（表达式1）[：操作条件]

DOWN/UP 表达式2

[例4] 查询计算机科学系(CS)学生的学号、年龄，结果按年龄降序排序。

GET W (Student.Sno，Student.Sage):

Student.Sdept='CS' DOWN Student.Sage

（4）带定额的检索

格式：**GET　工作空间名（定额）（表达式1）**
[：操作条件] [DOWN/UP 表达式2]

[例5] 取出一个信息系学生的学号。

GET W (1) (Student.Sno):
Student.Sdept='IS'

[例6] 查询信息系年龄最大的三个学生的学号及其年龄，结果按年龄降序排序。

GET W (3) (Student.Sno，Student.Sage):
Student.Sdept='IS' DOWN Student.Sage

（5）用元组变量的检索

- **元组变量的含义**
  - 表示可以在某一关系范围内变化（也称为范围变量**Range Variable**）
- **元组变量的用途**
  - ① 简化关系名：设一个较短名字的元组变量来代替较长的关系名。
  - ② 操作条件中使用量词时必须用元组变量。
- **定义元组变量**
  - 格式：　**RANGE** 关系名 变量名
  - 一个关系可以设多个元组变量

(6) 用存在量词的检索

[例8] 查询选修2号课程的学生名字。

RANGE SC X

GET W (Student.Sname):

∃X(X.Sno=Student.Sno∧X.Cno='2')

[例9] 查询选修了这样课程的学生学号，其直接先行课是6号课程。

RANGE Course CX

GET W (SC.Sno):

∃CX (CX.Cno=SC.Cno∧CX.Pcno='6')

用存在量词的检索(续)

[例10]查询至少选修一门其先行课为6号课程的学生名字

RANGE Course CX

SC SCX

GET W (Student.Sname): ∃SCX (SCX.Sno=Student.Sno∧

∃CX (CX.Cno=SCX.Cno∧CX.Pcno='6'))

前束范式形式：

GET W (Student.Sname):

∃SCX∃CX (SCX.Sno=Student.Sno∧

CX.Cno=SCX.Cno∧CX.Pcno='6')

（7）带有多个关系的表达式的检索

[例11] 查询成绩为90分以上的学生名字与课程名字。

RANGE SC SCX

GET W(Student.Sname，Course.Cname):

∃SCX (SCX.Grade≥90∧

SCX.Sno=Student.Sno∧

Course.Cno=SCX.Cno)

（8）用全称量词的检索

- [例12] 查询不选1号课程的学生名字。

RANGE SC SCX

GET W (Student.Sname): ∀ SCX

(SCX.Sno≠Student.Sno∨SCX.Cno≠'1')

用存在量词表示：

RANGE SC SCX

GET W (Student.Sname): ¬∃SCX

(SCX.Sno=Student.Sno∧SCX.Cno='1')

（9）用两种量词的检索

- [例13] 查询选修了全部课程的学生姓名。

RANGE Course CX

SC SCX

GET W (Student.Sname): ∀CX∃SCX

(SCX.Sno=Student.Sno∧

SCX.Cno=CX.Cno)

（10）用蕴函（Implication）的检索

[例14] 查询最少选修了**95002**学生所选课程的学生学号。

RANGE Couse CX

SC SCX

SC SCY

GET W (Student.Sno): ∀ CX(∃SCX

(SCX.Sno='95002'∧SCX.Cno=CX.Cno)

⇨ ∃SCY(SCY.Sno=Student.Sno∧

SCY.Cno= CX.Cno))

（11）集函数

常用集函数（Aggregation function）或内部函数（Build-in function）

函 数 名	功　能
COUNT	对元组计数
TOTAL	求 总 和
MAX	求最大值
MIN	求最小值
AVG	求平均值

集函数(续)

[例15] 查询学生所在系的数目。

GET W (COUNT(Student.Sdept))

COUNT函数在计数时会自动排除重复值。

[例16] 查询信息系学生的平均年龄

GET W (AVG(Student.Sage):

Student.Sdept='IS’)

二、更新操作

(1) 修改操作

(2) 插入操作

(3) 删除操作

（1）修改操作步骤

① 用HOLD语句将要修改的元组从数据库中读到工作空间中

HOLD 工作空间名（表达式1）[：操作条件]

HOLD语句是带上并发控制的GET语句

② 用宿主语言修改工作空间中元组的属性

③ 用UPDATE语句将修改后的元组送回数据库中

UPDATE 工作空间名

修改操作(续)

[例17] 把95007学生从计算机科学系转到信息系。

HOLD W (Student.Sno, Student.Sdept):

Student.Sno='95007'

（从**Student**关系中读出**95007**学生的数据）

MOVE 'IS' TO W.Sdept

（用宿主语言进行修改）

UPDATE W

（把修改后的元组送回**Student**关系）

（2）插入操作

步骤

① 用宿主语言在工作空间中建立新元组

② 用**PUT**语句把该元组存入指定关系中

PUT 工作空间名 （关系名）

PUT语句只对一个关系操作

插入操作(续)

[例18]　学校新开设了一门2学分的课程“计算机组织与结构”，其课程号为8，直接先行课为6号课程。插入该课程元组

```
MOVE '8' TO W.Cno
MOVE '计算机组织与结构' TO  W.Cname
MOVE '6' TO W.Cpno
MOVE  '2' TO W.Ccredit
PUT  W  (Course)
```

（3）删除操作

① 用HOLD语句把要删除的元组从数据库中读到工作空间中

② 用DELETE语句删除该元组

DELETE 工作空间名

删除操作(续)

[例19] 95110学生因故退学，删除该学生元组。

HOLD W (Student): Student.Sno='95110'
DELETE W

删除操作(续)

[例20] 将学号95001改为95102。

HOLD W (Student): Student.Sno='95001'

DELETE W

MOVE '95102' TO W.Sno

MOVE '李勇' TO W.Sname

MOVE '男' O W.Ssex

MOVE '20' TO W.Sage

MOVE 'CS' TO W.Sdept

PUT W (Student)

删除操作(续)

[例21] 删除全部学生。

HOLD W (Student)

DELETE W

HOLD W (SC)

DELETE W

在删除操作中保持参照完整性

小结：元组关系演算语言ALPHA

- **检索操作 GET**

  GET 工作空间名 **[**（定额）**]**（表达式**1**）

  [：操作条件**] [DOWN/UP** 表达式**2]**

- **插入操作**
  - 建立新元组**--PUT**
- **修改操作**
  - **HOLD--**修改**--UPDATE**
- **删除操作**
  - **HOLD--DELETE**

2.5 关 系 演 算

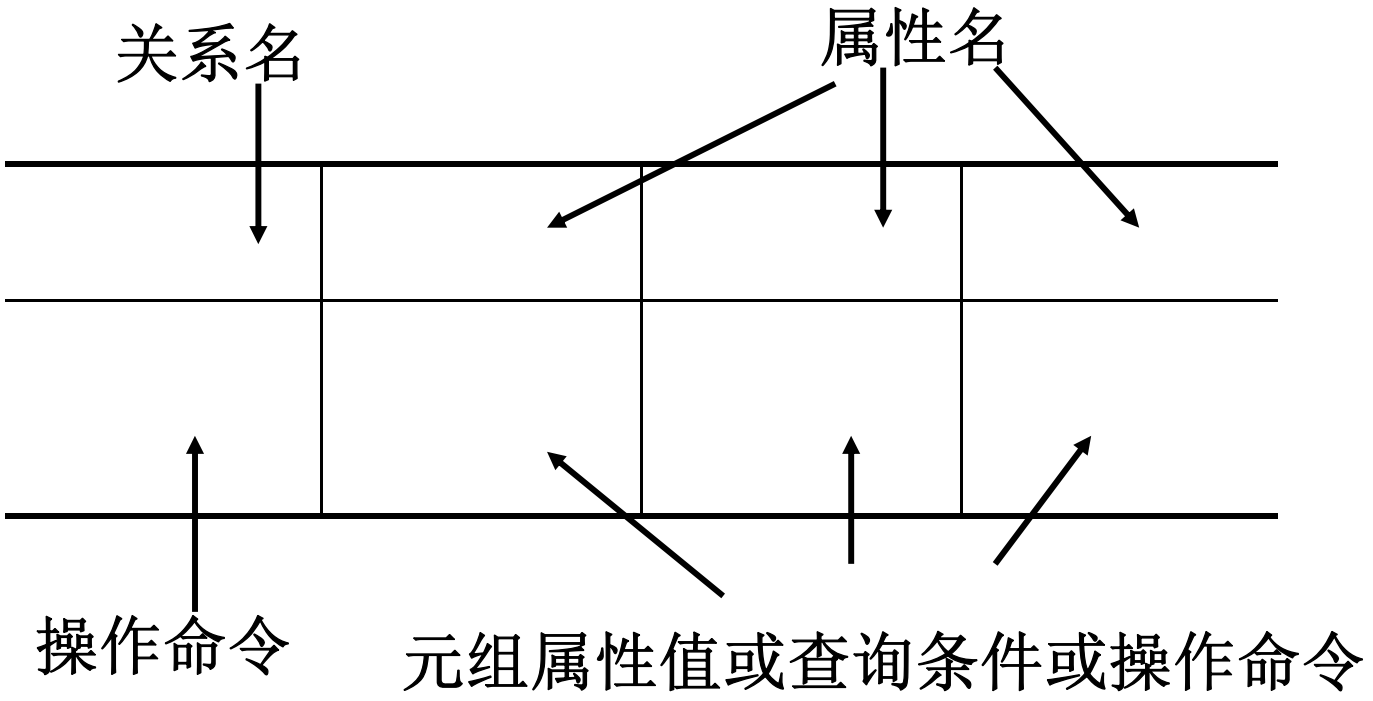
- 2.5.1 元组关系演算语言ALPHA
- 2.5.2 域关系演算语言QBE

2.5.2 域关系演算语言QBE

- 一种典型的域关系演算语言
  - 由**M.M.Zloof**提出
  - **1978**年在**IBM370**上得以实现

  - **QBE：Query By Example**
  - 基于屏幕表格的查询语言
  - 查询要求：以填写表格的方式构造查询
  - 用示例元素(域变量)来表示查询结果可能的情况
  - 查询结果：以表格形式显示

QBE操作框架



一、检索操作

（1）用户提出要求；

（2）屏幕显示空白表格；

（3）用户在最左边一栏输入要查询的关系名，例如Student；

Student					

检索操作（续）

（4）系统显示该关系的属性名

Student	Sno	Sname	Ssex	Sage	Sdept

- （5）用户在上面构造查询要求

Student	Sno	Sname	Ssex	Sage	Sdept
		P. T		AO.	C

检索操作（续）

（6）屏幕显示查询结果

Student	Sno	Sname	Ssex	Sage	Sdept
		李勇 张立			C

构造查询的几个要素

- 示例元素　即域变量　一定要加下划线

  示例元素是这个域中可能的一个值，它不必是查询结果中的元素
- 打印操作符P.　指定查询结果所含属性列
- 查询条件　不用加下划线

  可使用比较运算符＞，≥，＜，≤，＝和≠其中＝可以省略
- 排序要求

1. 简单查询

[例1] 查询全体学生的全部数据。

Student	Sno	Sname	Ssex	Sage	Sdept
	P.95001	P. 李勇	P.男	P.20	P.CS

简单查询（续）

显示全部数据也可以简单地把P.操作符作用在关系名上。

Student	Sno	Sname	Ssex	Sage	Sdept
P.					

2. 条件查询

(1) 简单条件

[例2] 求信息系全体学生的姓名。

Student	Sno	Sname	Ssex	Sage	Sdept
		P. 李勇			IS

条件查询（续）

[例3] 求年龄大于19岁的学生的学号。

Student	Sno	Sname	Ssex	Sage	Sdept
	P.95001			>19	

条件查询（与条件）

[例4]　求计算机科学系年龄大于19岁的学生的学号。

方法（1）：把两个条件写在同一行上

Student	Sno	Sname	Ssex	Sage	Sdept
	P.95001			>19	CS

条件查询（续）

方法（2）：把两个条件写在不同行上，但使用相同的示例元素值

Student	Sno	Sname	Ssex	Sage	Sdept
	P.95001 P.95001			>19	CS

条件查询（续）

[例5] 查询既选修了1号课程又选修了2号课程的学生的学号。

SC	Sno	Cno	Grade
	P.95001 P.95001	1 2	

条件查询（续）

[例6] 查询计算机科学系或者年龄大于19岁的学生的学号。

Student	Sno	Sname	Ssex	Sage	Sdept
	P.95001 P.95002			>19	CS

多表连接

[例7] 查询选修1号课程的学生姓名。

Student	Sno	Sname	Ssex	Sage	Sdept
	95001	P.李勇			

SC	Sno	Cno	Grade
	95001	1	

注意：示例元素Sno是连接属性，其值在两个表中要相同。

条件查询（非条件）

[例8] 查询未选修1号课程的学生姓名

Student	Sno	Sname	Ssex	Sage	Sdept
	95001	P.李勇			

SC	Sno	Cno	Grade
¬	95001	1	

思路：显示学号为95001的学生名字，而该学生选修1号课程的情况为假

条件查询（续）

[例9] 查询有两个人以上选修的课程号

SC	Sno	Cno	Grade
	95001 ¬95001	P.1 1	

思路：查询这样的课程1，它不仅被95001选修而且也被另一个学生（¬95001）选修了

3. 集函数

常用集函数：

函 数 名	功　能
CNT	对元组计数
SUM	求 总 和
AVG	求平均值
MAX	求最大值
MIN	求最小值

集函数（续）

[例10] 查询信息系学生的平均年龄。

Student	Sno	Sname	Ssex	Sage	Sdept
				P.AVG.ALL.	IS

4.对查询结果排序（续）

[例11]　查全体男生的姓名，要求查询结果按所在系升序排序，对相同系的学生按年龄降序排序。

Student	Sno	Sname	Ssex	Sage	Sdept
		P.李勇	男	DO（2）.	AO（1）.

二、修改操作

[例12] 把95001学生的年龄改为18岁。

方法(1)：将操作符“U.”放在值上

Student	Sno	Sname	Ssex	Sage	Sdept
	95001			U. 18	

修改操作(续)

方法(2)： 将操作符“U.”放在关系上

Student	Sno	Sname	Ssex	Sage	Sdept
U.	95001			18	

码95001标明要修改的元组。“U.”标明所在的行是修改后的新值。

由于主码是不能修改的，**所以系统不会混淆要修改的属性。**

修改操作(续)

[例13]将计算机系所有学生的年龄都改为18岁

Student	Sno	Sname	Ssex	Sage	Sdept
	95008			U.18	CS

修改操作(续)

[例14] 把95001学生的年龄增加1岁

Student	Sno	Sname	Ssex	Sage	Sdept
	95001			17	
U.	95001			17+1	

分两行分别表示改前和改后的示例元素

必须将操作符“U.”放在关系上

修改操作(续)

[例15] 将计算机系所有学生的年龄都增加1岁

Student	Sno	Sname	Ssex	Sage	Sdept
U.	95008 95008			18 18+1	CS

2.插入操作

[例16] 把信息系女生95701，姓名张三，年龄17岁存入数据库中。

Student	Sno	Sname	Ssex	Sage	Sdept
I.	95701	张三	女	17	IS

3. 删除操作

[例17] 删除学生95089

Student	Sno	Sname	Ssex	Sage	Sdept
D.	95089				

为保证参照完整性，删除95089学生前，先删除95089学生选修的全部课程

SC	Sno	Cno	Grade
D.	95089		