

## 第2章关系模型与关系运算

### 2.1 关系模型与关系运算简述

### 2.2 关系与关系模型

### 2.3 关系代数运算

### 2.4 关系元组演算

- 关系演算概述

- 关系元组演算

- 元组演算公式

- 元组演算符的优先次序

- 元组演算符的等价性

- 用元组演算公式实现关系代数公式

### 2.5 关系域演算

### 2.6 小结

## 2.4 关系元组演算

### —关系演算概述

➤ 我们在前面已经见过关系演算的基本形式：

□ 如，并运算定义中： $R \cup S = \{r \mid r \in R \vee r \in S\}$

□ 再如，差运算定义中： $R - S = \{r \mid r \in R \wedge r \notin S\}$

□ 又如，交运算定义中： $R \cap S = \{r \mid r \in R \wedge r \in S\}$

以上花括号内的演算公式既为关系演算公式

## 2.4 关系元组演算

### —关系演算概述(续)

- 关系演算是以数理逻辑中的谓词演算为基础的
- 关系演算是描述关系运算的另一种思维方式
- SQL语言是继承了关系代数和关系演算各自的优点所形成的
- 按照谓词变量的不同，可分为关系元组演算和关系域演算形式：
  - ❑ 关系元组演算是以元组变量作为谓词变量的基本对象
  - ❑ 关系域演算是以域变量作为谓词变量的基本对象，域演算将在下一小节介绍

## 2.4 关系元组演算

### —关系元组演算

➤ 关系元组演算的基本形式:

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

上式表示：所有使谓词P为真的元组t的集合

- t是元组变量
- $t \in r$ 表示元组t在关系r中
- $t[A]$ 表示元组t的分量，即t在属性A上的值
- P是与谓词逻辑相似的公式， $P(t)$ 表示以元组t为变量的公式

## 2.4 关系元组演算

### --关系元组演算(续)

➤  $P(t)$  可以是如下三种形式之一的原子公式:

□  $t \in R$

- $t$  是关系  $R$  中的一个元组
- 例如:  $\{t \mid t \in \text{Student}\}$

□  $s[A] \theta c$

- 元组分量  $s[A]$  与常量  $c$  之间满足比较关系  $\theta$
- $\theta$ : 比较运算符  $<, \leq, =, <>, >, \geq$
- 例如:  $\{t \mid t \in R \wedge t[\text{Sage}] \leq 19 \wedge t[\text{Sname}] = \text{'张三'}\}$

□  $s[A] \theta u[B]$

- $s[A]$  与  $u[B]$  为元组分量,  $A$  和  $B$  分别是某些关系的属性, 他们之间满足比较关系  $\theta$
- 例如:  $\{t \mid t \in \text{Student} \wedge \exists (u \in \text{Student}) (t[\text{Sage}] > u[\text{Sage}])\}$

## 2.4 关系元组演算

### —关系元组演算（续）

➤  $P(t)$  可以由公式加运算符 $\wedge$  (与)、 $\vee$  (或)、 $\neg$  (非) 递归构成

- 如果  $F$  是一个公式，则  $\neg F$  也是公式
- 如果  $F_1$ 、 $F_2$  是公式，则  $F_1 \wedge F_2$ ， $F_1 \vee F_2$  也是公式

| F1      | F2      | F1 $\wedge$ F2 |
|---------|---------|----------------|
| 真/True  | 真/True  |                |
| 真/True  | 假/False |                |
| 假/False | 真/True  |                |
| 假/False | 假/False |                |

| F1      | F2      | F1 $\vee$ F2 |
|---------|---------|--------------|
| 真/True  | 真/True  |              |
| 真/True  | 假/False |              |
| 假/False | 真/True  |              |
| 假/False | 假/False |              |

| F1      | $\neg F1$ |
|---------|-----------|
| 真/True  |           |
| 假/False |           |

## 2.4 关系元组演算

### —关系元组演算（续）

➤  $P(t)$  可以由公式加运算符 $\wedge$  (与)、 $\vee$  (或)、 $\neg$  (非) 递归构成

- 如果  $F$  是一个公式，则  $\neg F$  也是公式
- 如果  $F_1$ 、 $F_2$  是公式，则  $F_1 \wedge F_2$ ， $F_1 \vee F_2$  也是公式

| F1      | F2      | F1 $\wedge$ F2 |
|---------|---------|----------------|
| 真/True  | 真/True  | 真/True         |
| 真/True  | 假/False | 假/False        |
| 假/False | 真/True  | 假/False        |
| 假/False | 假/False | 假/False        |

| F1      | F2      | F1 $\vee$ F2 |
|---------|---------|--------------|
| 真/True  | 真/True  | 真/True       |
| 真/True  | 假/False | 真/True       |
| 假/False | 真/True  | 真/True       |
| 假/False | 假/False | 假/False      |

| F1      | $\neg F_1$ |
|---------|------------|
| 真/True  | 假/False    |
| 假/False | 真/True     |





## 2.4 关系元组演算

### —关系元组演算（续）

➤  $P(t)$  可以由公式加运算符  $\wedge$  (与)、 $\vee$  (或)、 $\neg$  (非) 递归构成

□ 例如：检索出年龄小于20岁并且是男同学的所有学生

$\{ t \mid t \in \text{Student} \wedge t[\text{Sage}] < 20 \wedge t[\text{Ssex}] = \text{'男'} \}$

□ 再例如：检索出年龄小于20岁或者03系的所有男学生。

$\{ t \mid t \in \text{Student} \wedge (t[\text{Sage}] < 20 \vee t[\text{D\#}] = \text{'03'}) \wedge t[\text{Ssex}] = \text{'男'} \}$

在元组演算公式构造过程中，如果需要，可以使用括号，通过括号改变运算的优先次序，即：括号内的运算优先计算。

□ 再例如：检索出不是03系的所有学生

$\{ t \mid t \in \text{Student} \wedge \neg (t[\text{D\#}] = \text{'03'}) \}$

□ 再例如：检索不是(小于20岁的男同学)的所有同学

$\{ t \mid t \in \text{Student} \wedge \neg (t[\text{Sage}] < 20 \wedge t[\text{Ssex}] = \text{'男'}) \}$



## 2.4 关系元组演算

### —关系元组演算（续）

- 构造 $P(t)$ 还有两个运算符： $\exists$ （存在）、 $\forall$ （任意）
  - ▣ 如果 $F$ 是一个公式，则 $\exists(t \in r)(F(t))$ 也是公式
  - ▣ 如果 $F$ 是一个公式，则 $\forall(t \in r)(F(t))$ 也是公式
- 运算符 $\exists$ 和 $\forall$ ，又称为量词，前者称“存在量词”，后者称“全称量词”
- 而被 $\exists$ 或 $\forall$ 限定的元组变量 $t$ ，或者说，元组变量 $t$ 前有存在量词或全称量词，则该变量被称为“约束变量”，否则被称为“自由变量”。

| $t \in r$                     | $F(t)$                    | $\exists(t \in r)(F(t))$ |
|-------------------------------|---------------------------|--------------------------|
| 对 $r$ 中的每一个 $t$ 进行 $F(t)$ 的检验 | 所有 $t$ 都使 $F(t)=\text{假}$ | 假/False                  |
|                               | 有一个 $t$ 使 $F(t)=\text{真}$ | 真/True                   |

| $t \in r$                     | $F(t)$                    | $\forall(t \in r)(F(t))$ |
|-------------------------------|---------------------------|--------------------------|
| 对 $r$ 中的每一个 $t$ 进行 $F(t)$ 的检验 | 所有 $t$ 都使 $F(t)=\text{真}$ | 真/True                   |
|                               | 有一个 $t$ 使 $F(t)=\text{假}$ | 假/False                  |

## 2.4 关系元组演算

### —关系元组演算（续）

➤ 构造 $P(t)$ 还有两个运算符： $\exists$ （存在）、 $\forall$ （任意）

▣ 例如：“检索出年龄不是最小的所有同学

$\{t \mid t \in \text{Student} \wedge \exists(u \in \text{Student}) (t[\text{Sage}] > u[\text{Sage}])\}$

请大家写一下，在关系代数中，如何表达上面的查询需求？

R(学生表)

| S#      | Sname | Ssex | Sage | D# |
|---------|-------|------|------|----|
| 1110101 | 张三    | 女    | 21   | 01 |
| 1110102 | 李四    | 男    | 20   | 02 |
| 1110103 | 王五    | 男    | 19   | 03 |
| 1110104 | 刘六    | 女    | 19   | 03 |

$t \rightarrow$  (points to row 2)

$u \rightarrow$  (points to row 4)

## 2.4 关系元组演算

### —关系元组演算（续）

➤ 构造 $P(t)$ 还有两个运算符： $\exists$ （存在）、 $\forall$ （任意）

▣ 再例如：检索出课程都及格的所有同学

$\{t \mid t \in \text{Student} \wedge \forall (u \in \text{SC} \wedge t[S\#] = u[S\#])(u[\text{Score}] \geq 60)\}$

用关系代数，如何书写上面例子？

Student

$t \rightarrow$

| S#   | Sname | Ssex | Sage | D# | Sclass |
|------|-------|------|------|----|--------|
| 1001 | 张三    | 女    | 21   | 01 | 1101   |
| 1002 | 李四    | 男    | 20   | 02 | 1101   |
| 1003 | 王五    | 男    | 19   | 03 | 1102   |

SC

$u \rightarrow$

| S#   | C#  | Score |
|------|-----|-------|
| 1001 | 001 | 80    |
| 1001 | 002 | 86    |
| 1001 | 003 | 87    |
| 1002 | 001 | 79    |
| 1002 | 002 | 87    |
| 1002 | 003 | 56    |
| 1003 | 001 | 77    |
| 1003 | 003 | 89    |

## 2.4 关系元组演算

### —关系元组演算（续）

#### ➤ 我们再举几个例子

▣ 例如：检索出比张三年龄小的所有同学

$\{ t \mid t \in \text{Student} \wedge \exists (w \in \text{Student})(w[\text{Sname}] = \text{'张三'} \wedge t[\text{Sage}] < w[\text{Sage}]) \}$

|     | S#   | Sname | Ssex | Sage | D# | Sclass |
|-----|------|-------|------|------|----|--------|
| w → | 1001 | 张三    | 女    | 21   | 01 | 1101   |
|     | 1002 | 李四    | 男    | 20   | 02 | 1101   |
| t → | 1003 | 王五    | 男    | 19   | 03 | 1102   |
|     | 1004 | 刘六    | 女    | 21   | 03 | 1103   |

## 2.4 关系元组演算

### —关系元组演算（续）

#### ➤ 我们再举几个例子

▣ 例如：检索计算机系的所有同学

$\{t \mid t \in \text{Student} \wedge \exists(u \in \text{Dept}) (t[D\#] = u[D\#] \wedge u[Dname] = \text{'计算机'})\}$

Student

| S#   | Sname | Ssex | Sage | D# | Sclass |
|------|-------|------|------|----|--------|
| 1001 | 张三    | 女    | 21   | 01 | 1101   |
| 1002 | 李四    | 男    | 20   | 02 | 1101   |
| 1003 | 王五    | 男    | 19   | 03 | 1102   |
| 1004 | 刘六    | 女    | 21   | 03 | 1103   |

t →

Dept

| D# | Dname |
|----|-------|
| 01 | 机电    |
| 02 | 能源    |
| 03 | 计算机   |

← u

## 2.4 关系元组演算

### —关系元组演算（续）

#### ➤ 我们再举几个例子（续）

▣ 例如：检索学过所有课程的同学

$\{ t \mid t \in \text{Student} \wedge \forall (u \in \text{Course})(\exists (s \in \text{SC})(s[S\#]=t[S\#] \wedge u[C\#]=s[C\#])) \}$

Student

| S#   | Sname | Ssex | Sage | D# | Sclass |
|------|-------|------|------|----|--------|
| 1001 | 张三    | 女    | 21   | 01 | 1101   |
| 1002 | 李四    | 男    | 20   | 02 | 1101   |
| 1003 | 王五    | 男    | 19   | 03 | 1102   |
| 1004 | 刘六    | 女    | 21   | 03 | 1103   |

t →

Course

| C#  | Cname | Chours | Credit |
|-----|-------|--------|--------|
| 001 | 化学    | 40     | 4      |
| 002 | 数学    | 40     | 4      |
| 003 | 物理    | 60     | 6      |

u →

SC

| S#   | C#  | Score |
|------|-----|-------|
| 1001 | 001 | 80    |
| 1001 | 002 | 86    |
| 1001 | 003 | 87    |
| 1002 | 002 | 79    |
| 1002 | 003 | 87    |
| 1003 | 001 | 77    |
| 1003 | 003 | 89    |
| 1004 | 002 | 78    |

s →

## 2.4 关系元组演算

### —关系元组演算（续）

#### ➤ 我们再举几个例子（续）

▣ 例如：检索所有同学所有课程全都及格的系

$\{ t \mid t \in \text{Dept} \wedge \forall (s \in \text{Student} \wedge s[D\#]=t[D\#])(\forall (u \in \text{SC} \wedge s[S\#]=u[S\#])(u[\text{Score}] \geq 60)) \}$

Dept

| D# | Dname |
|----|-------|
| 01 | 化学    |
| 02 | 数学    |
| 03 | 物理    |

t →

Student

| S#   | Sname | Ssex | Sage | D# | Sclass |
|------|-------|------|------|----|--------|
| 1001 | 张三    | 女    | 21   | 01 | 1101   |
| 1002 | 李四    | 男    | 20   | 02 | 1101   |
| 1003 | 王五    | 男    | 19   | 03 | 1102   |

s →

SC

| S#   | C#  | Score |
|------|-----|-------|
| 1001 | 001 | 80    |
| 1001 | 002 | 86    |
| 1001 | 003 | 87    |
| 1002 | 002 | 79    |
| 1002 | 003 | 87    |
| 1003 | 001 | 77    |
| 1003 | 003 | 89    |

u →





## 2.4 关系元组演算

### —关系元组演算（续）

#### ➤ $\{ t \mid P(t) \}$ 中公式 $P(t)$ 的构造小结

- 三种形式的原子公式是公式

(1)  $s \in R$       (2)  $s[A] \theta c$       (3)  $s[A] \theta u[B]$

- 如果  $P$  是公式，那么  $\neg P$  也是公式
- 如果  $P_1, P_2$  是公式，则  $P_1 \wedge P_2, P_1 \vee P_2$  也是公式
- 如果  $P(t)$  是公式， $R$  是关系，则  $\exists (t \in R)(P(t))$  和  $\forall (t \in R)(P(t))$  也是公式
- 需要时可加括弧
- 上述运算符的优先次序自高至低为：括弧； $\theta$ ； $\exists$ ； $\forall$ ； $\neg$ ； $\wedge$ ； $\vee$ ；
- 公式只限于以上形式

## 2.4 关系元组演算

### 一元组演算符的优先次序

➤ P(t)运算符优先次序为:

括弧;  $\theta$ ;  $\exists$ ;  $\forall$ ;  $\neg$ ;  $\wedge$ ;  $\vee$ ;

示例, 请注意下述语句的结果差异

$\{ t \mid t \in \text{Student} \wedge ( t[\text{Sage}] < 20 \vee t[\text{D\#}] = '03' \wedge t[\text{Ssex}] = \text{'男'}) \}$

$\{ t \mid t \in \text{Student} \wedge (( t[\text{Sage}] < 20 \vee t[\text{D\#}] = '03') \wedge t[\text{Ssex}] = \text{'男'}) \}$

$\{ t \mid t \in \text{Student} \wedge \neg ( t[\text{Sage}] < 20 \wedge t[\text{Ssex}] = \text{'男'}) \}$

$\{ t \mid t \in \text{Student} \wedge \forall (u \in \text{SC}) ( t[\text{S\#}] = u[\text{S\#}] \wedge u[\text{Score}] \geq 60 ) \}$

## 2.4 关系元组演算

### 一元组演算公式的等价性

➤  $P(t)$  公式, 如谓词演算一样, 也有一系列演算的等价性

▣ 例如:  $\theta$  与  $\neg$  的等价性 (符号  $\Leftrightarrow$  表示等价于)

$$\neg(\alpha = \beta) \Leftrightarrow \alpha \neq \beta$$

$$\neg(\alpha > \beta) \Leftrightarrow \alpha \leq \beta \Leftrightarrow \alpha < \beta \vee \alpha = \beta$$

$$\neg(\alpha < \beta) \Leftrightarrow \alpha \geq \beta \Leftrightarrow \alpha > \beta \vee \alpha = \beta$$

$$\neg(\alpha \geq \beta) \Leftrightarrow \alpha < \beta$$

$$\neg(\alpha \leq \beta) \Leftrightarrow \alpha > \beta$$

$$\neg(\alpha \neq \beta) \Leftrightarrow \alpha = \beta$$

▣ 等价性示例:

$\{ t \mid t \in \text{Student} \wedge \neg (t[D\#] = '03') \}$  等价于

$\{ t \mid t \in \text{Student} \wedge t[D\#] \neq '03' \}$

## 2.4 关系元组演算

### 一元组演算公式的等价性(续)

➤  $P(t)$ 公式,如谓词演算一样,也有一系列演算的等价性

▣ 再例如 $\wedge$ 、 $\vee$ 与 $\neg$ 运算之间的等价性)

$$P1 \wedge P2 \Leftrightarrow \neg(\neg P1 \vee \neg P2)$$

$n$ 个否定的或操作的再否定,便是 $n$ 个肯定的与操作

$$P1 \vee P2 \Leftrightarrow \neg(\neg P1 \wedge \neg P2)$$

$n$ 个否定的与操作的再否定,便是 $n$ 个肯定的或操作

▣ 等价性示例:

“或者学过001课程或者学过002课程”

$$\{u \mid u \in SC \wedge (u[C\#] = '001' \vee u[C\#] = '002')\}$$

等价于“去掉既未学过001课程又未学过002课程的所有人”

## 2.4 关系元组演算

### 一元组演算公式的等价性(续)

➤  $P(t)$ 公式,如谓词演算一样,也有一系列演算的等价

#### ▣ 等价性再示例:

“既年龄小于20岁又是男性的所有学生”

$\{ s \mid s \in \text{Student} \wedge (s[\text{Sage}] < 20 \wedge s[\text{Ssex}] = \text{'男'}) \}$

等价于“去掉: 或者年龄大于等于20岁, 或者不是男性的所有人”

$\{ s \mid s \in \text{Student} \wedge \neg (s[\text{Sage}] \geq 20 \vee s[\text{Ssex}] \neq \text{'男'}) \}$

## 2.4 关系元组演算

### 一元组演算公式的等价性(续)

➤  $P(t)$  公式, 如谓词演算一样, 也有一系列演算的等价

▣ 还例如  $\exists$ 、 $\forall$  与  $\neg$  运算之间的等价性

$$\forall(t \in R)(P(t)) \Leftrightarrow \neg(\exists(t \in R)(\neg P(t)))$$

$$\exists(t \in R)(P(t)) \Leftrightarrow \neg(\forall(t \in R)(\neg P(t)))$$

▣ 等价性示例

“既学过001课程又学过002课程的学生”

$$\{ t \mid t \in \text{Student} \wedge \exists(s \in \text{SC} \wedge u \in \text{SC} \wedge s[S\#] = t[S\#] \wedge u[S\#] = t[S\#]) (s[C\#] = '001' \wedge u[C\#] = '002') \}$$

等价于“去掉：或者未学过001课程，或者未学过002课程的所有人”

$$\{ t \mid t \in \text{Student} \wedge \neg(\forall(s \in \text{SC} \wedge s[S\#] = t[S\#])(s[C\#] \neq '001') \vee \forall(s \in \text{SC} \wedge s[S\#] = t[S\#])(s[C\#] \neq '002')) \}$$

## 2.4 关系元组演算

### —用元组演算公式实现关系代数

➤ 关系代数有五种基本操作：并、差、广义积、选择、投影操作，还有：交、 $\theta$ -连接操作。

□ 并运算：  $R \cup S = \{ t \mid t \in R \vee t \in S \}$

□ 差运算：  $R - S = \{ t \mid t \in R \wedge t \notin S \}$

□ 交运算：  $R \cap S = \{ t \mid t \in R \wedge t \in S \}$

□ 广义笛卡尔积

$R(A) \times S(B) = \{ t \mid \exists (u \in R) \exists (s \in S) (t[A] = u[A] \wedge t[B] = s[B]) \}$

□ 选择运算：  $\sigma_{con}(R) = \{ t \mid t \in R \wedge F(con) \}$

□ 投影运算：  $\pi_A(R) = \{ t[A] \mid t \in R \}$



## 2.4 关系元组演算

### 一元组演算公式与关系代数对比应用的例子

➤ 已知:

- ▣ 学生关系: Student(S#, Sname, Sage, Ssex, Sclass)
- ▣ 课程关系: Course(C#, Cname, Chours, Credit, Tname)
- ▣ 选课关系: SC(S#, C#, Score)

➤ 求学过李明老师讲授所有课程的学生姓名(全都学过)

✓  $\pi_{Sname}((Student \bowtie Course \bowtie SC) \div \sigma_{Tname='李明'}(C))$

✓  $\{ t[Sname] \mid t \in Student \wedge \forall (u \in Course \wedge u[Tname]='李明') \\ (\exists (w \in SC)(w[S\#]=t[S\#] \wedge w[C\#]=u[C\#])) \}$

## 2.4 关系元组演算

### 一元组演算公式与关系代数对比应用的例子(续)

➤ 已知:

- ▣ 学生关系: Student(S#, Sname, Sage, Ssex, Sclass)
- ▣ 课程关系: Course(C#, Cname, Chours, Credit, Tname)
- ▣ 选课关系: SC(S#, C#, Score)

➤ 求没学过李明老师讲授任一门课程的学生姓名(全没学过)

✓  $\pi_{Sname}(\text{Student}) - \pi_{Sname}(\sigma_{Tname='李明'}(\text{Student} \bowtie \text{Course} \bowtie \text{SC}))$

✓  $\{ t[Sname] \mid t \in \text{Student} \wedge \forall (u \in \text{Course} \wedge u[Tname]='李明') (\neg \exists (w \in \text{SC})(w[S\#]=t[S\#] \wedge w[C\#]=u[C\#])) \}$

## 2.4 关系元组演算

### 一元组演算公式与关系代数对比应用的例子(续)

➤ 已知:

- ▣ 学生关系:  $Student(S\#, Sname, Sage, Ssex, Sclass)$
- ▣ 课程关系:  $Course(C\#, Cname, Chours, Credit, Tname)$
- ▣ 选课关系:  $SC(S\#, C\#, Score)$

➤ 求至少学过一门李明老师讲授课程的学生姓名(至少学过一门)

- ✓  $\pi_{Sname}(\sigma_{Tname='李明'}(Student \bowtie Course \bowtie SC))$
- ✓  $\{ t[Sname] \mid t \in Student \wedge \exists(u \in Course) \exists(w \in SC) (u[Tname]='李明' \wedge w[S\#]=t[S\#] \wedge w[C\#]=u[C\#]) \}$

## 2.4 关系元组演算

### 一元组演算公式与关系代数对比应用的例子(续)

#### ➤ 已知:

- ▣ 学生关系:  $Student(S\#, Sname, Sage, Ssex, Sclass)$
- ▣ 课程关系:  $Course(C\#, Cname, Chours, Credit, Tname)$
- ▣ 选课关系:  $SC(S\#, C\#, Score)$

#### ➤ 求至少有一门李明老师讲授课程没有学过的学生姓名(至少有一门没学过)

- ✓  $\pi_{Sname}(Student) - \pi_{Sname}((Student \bowtie Course \bowtie SC) \div \sigma_{Tname='李明'}(C))$
- ✓  $\{ t[Sname] \mid t \in Student \wedge \exists (u \in Course \wedge u[Tname]='李明') \wedge (\neg \exists (w \in SC) \wedge w[S\#]=t[S\#] \wedge w[C\#]=u[C\#]) \}$

## 第2章 关系模型与关系运算

### 2.1 关系模型与关系运算简述

### 2.2 关系与关系模型

### 2.3 关系代数运算

### 2.4 关系元组演算

### 2.5 关系域演算

--关系域演算简要介绍

--关系域演算与关系元组演算比较

### 2.6 小结

## 2.5 关系域演算

### —关系域演算简要介绍

#### ➤ 关系域演算的基本形式

$\{ \langle x_1, x_2, \dots, x_n \rangle \mid P(x_1, x_2, \dots, x_n) \}$

其中 $x_i$  代表域变量或常量,  $P$ 为以 $x_i$ 为变量的公式。

#### □ $P$ 的构造如下:

✓  $\langle x_1, x_2, \dots, x_n \rangle \in R$

$x_i$  代表域变量或常量, 表示由域变量构成的 $\langle x_1, x_2, \dots, x_n \rangle$  属于关系 $R$

✓  $x \theta c$

域变量 $x$ 与常量 $c$ 之间满足比较关系

$\theta$ : 比较运算符 $\{<, <=, =, >=, >\}$

✓  $x \theta y$

域变量 $x$ 与域变量 $y$ 之间满足比较关系 $\theta$

## 2.5 关系域演算

### —关系域演算简要介绍(续)

#### ➤ 关系域演算的基本形式

$\{ \langle x_1, x_2, \dots, x_n \rangle \mid P(x_1, x_2, \dots, x_n) \}$

其中 $x_i$ 代表域变量或常量， $P$ 为以 $x_i$ 为变量的公式。

- 如果 $P$ 是公式，那么 $\neg P$ 也是公式
- 如果 $P_1, P_2$ 是公式，则 $P_1 \wedge P_2, P_1 \vee P_2$ 也是公式
- 如果 $P$ 是公式， $x$ 是域变量，则 $\exists(x)(P(x))$ 和 $\forall(x)(P(x))$ 也是公式
- 需要时可加括弧
- 上述运算符的优先次序自高至低为：括弧； $\theta$ ； $\exists$ ； $\forall$ ； $\wedge$ ； $\vee$ ；
- 公式只限于以上形式



## 2.5 关系域演算

### —关系域演算简要介绍(续)

□ 例如：检索出不是03系的所有学生

$\{ \langle a,b,c,d,e,f \rangle \mid \langle a,b,c,d,e,f \rangle \in \text{Student} \wedge e \neq '03' \}$

Student

| S#   | Sname | Ssex | Sage | D# | Sclass |
|------|-------|------|------|----|--------|
| 1001 | 张三    | 女    | 21   | 01 | 1101   |
| 1002 | 李四    | 男    | 20   | 02 | 1101   |
| 1003 | 王五    | 男    | 19   | 03 | 1102   |
| 1004 | 刘六    | 女    | 21   | 03 | 1103   |

a

b

c

d

e

f



## 2.5 关系域演算

### —关系域演算简要介绍(续)

- 再例如：检索不是(小于20岁的男同学)的所有同学的姓名  
 $\{ \langle b \rangle \mid \exists a, c, d, e, f (\langle a, b, c, d, e, f \rangle \in \text{Student} \wedge \neg (d < 20 \wedge c = \text{'男'})) \}$

Student

| S#   | Sname | Ssex | Sage | D# | Sclass |
|------|-------|------|------|----|--------|
| 1001 | 张三    | 女    | 21   | 01 | 1101   |
| 1002 | 李四    | 男    | 20   | 02 | 1101   |
| 1003 | 王五    | 男    | 19   | 03 | 1102   |
| 1004 | 刘六    | 女    | 21   | 03 | 1103   |

a            b            c            d            e            f



## 2.5 关系域演算

### —关系域演算简要介绍(续)

□ 再例如：检索成绩不及格的同学姓名、课程及其成绩

$\{ \langle b, h, m \rangle \mid \exists a, c, d, e, f, g, I, j, k (\langle a, b, c, d, e, f \rangle \in \text{Student} \wedge (g, h, I, j, k) \in \text{Course} \wedge (a, g, m) \in \text{SC} \wedge m < 60) \}$

Student

| S#   | Sname | Ssex | Sage | D# | Sclass |
|------|-------|------|------|----|--------|
| 1001 | 张三    | 女    | 21   | 01 | 1101   |
| 1002 | 李四    | 男    | 20   | 02 | 1101   |
| 1003 | 王五    | 男    | 19   | 03 | 1102   |
| 1004 | 刘六    | 女    | 21   | 03 | 1103   |

a

b

c

d

e

f

Course

| C#  | Cname | Chours | Credit |
|-----|-------|--------|--------|
| 001 | 化学    | 40     | 4      |
| 002 | 数学    | 40     | 4      |
| 003 | 物理    | 60     | 6      |

g

h

i

j

SC

| S#   | C#  | Score |
|------|-----|-------|
| 1001 | 001 | 80    |
| 1001 | 002 | 86    |
| 1001 | 003 | 87    |
| 1002 | 002 | 79    |
| 1002 | 003 | 87    |
| 1003 | 001 | 77    |
| 1003 | 003 | 89    |
| 1004 | 002 | 78    |

a

g

m

## 2.5 关系域演算

### —关系域演算与元组演算的比较

- 元组演算的基本形式:  $\{ t \mid P(t) \}$
- 域演算的基本形式:  $\{ \langle x_1, x_2, \dots, x_n \rangle \mid P(x_1, x_2, \dots, x_n) \}$
- 元组演算是以元组为变量, 以元组为基本处理单位, 先找到元组, 然后再找到元组分量, 进行谓词判断;
- 域演算是以域变量为基本处理单位, 先有域变量, 然后再判断由这些域变量组成的元组是否存在或是否满足谓词判断。
- 公式的构造过程是相似的
- 元组演算和域演算可以等价互换。

## 2.6 小结

——本章我们学习了以下一些概念

### ➤ 关系

- 域、笛卡尔积、关系；基数与目(度)数
- 属性/字段/列/数据项、元组/记录/行、关系模式、关系/Table
- 候选码/候选键、主码/主键、主属性和非主属性、外码/外键
- 关系的特性

### ➤ 关系模型

- 基本数据结构：关系
- 基本操作：并、差、广义积、选择、投影
- 完整性约束：实体完整性, 参照完整性, 用户自定义完整性

## 2.6 小结

—本章我们学习了以下一些概念

### ➤ 关系运算

#### ▣ 关系代数

- 并、差、广义积、选择、投影;
- 交、除、 $\theta$ -连接、等值连接、自然连接、外连接
- 用关系代数表达日常检索需求是按集合思维方式进行

#### ▣ 关系演算

- 关系元组演算  $\{ t \mid P(t) \}$ 
  - ✓以元组为变量, 由 $\in, \theta, \wedge, \vee, \neg, \exists, \forall$ , 括号等算式构成 $P(t)$
- 关系域演算  $\{ xyz \mid P(xyz) \}$ 
  - ✓以域为变量, 由 $\in, \theta, \wedge, \vee, \neg, \exists, \forall$ , 括号等算式构成 $P(x)$
  - ✓QBE: Query By Example
- 用关系演算表达日常检索需求是按逻辑思维方式进行

## 2.6 小结

——本章我们学习了以下一些概念

- 三种关系运算之间是具有等价性的
- 三种关系运算都可说是非过程性的，但相比之下：域演算的非过程性最好，元组演算次之，关系代数最差
- 三种关系运算虽是抽象的，但他们是衡量数据库语言完备性的基础
  - ▣ 一个数据库语言能够等价地实现这三种关系运算的操作，则说该语言是完备的



## 2.6 小结

——本章我们学习了以下一些概念

- 目前多数数据库语言都能够实现这三种运算的操作，在此基础上还增加了许多其他的操作，如赋值操作、聚集操作等。

## 下一章的学习内容

基于关系代数和关系演算而提出的SQL语言目前已成为数据库语言的标准，我们将在下一章进行介绍。

### 第3章SQL语言：标准数据库语言的语法 及其交互式应用训练

- SQL的各种操作语句；
- SQL的视图应用语句；