数据库系统之一
--基本知识与关系模型

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

第5讲 关系模型之关系演算

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

本讲学习什么?

战德臣 教授

基本内容

1. 关系演算之关系元组演算
2. 关系演算之关系域演算
3. 关系演算之安全性
4. 关于三种关系运算的一些观点

重点与难点

- 关系元组演算公式的递归定义；关系域演算公式的递归定义
- 关系元组演算公式：与 $\wedge$、或 $\vee$、非 $\neg$、**存在量词 $\exists$、全称量词 $\forall$**
- 用关系元组演算公式表达查询的思维训练
- 用QBE语言表达查询的思维训练
- 关系元组演算、域演算和关系代数在表达查询方面的思维差异

关系元组演算

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

关系元组演算
(1)概述

战德臣 教授

- 前面出现过关系演算形式

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

- 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
  - 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

关系元组演算公式的基本形式：

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

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

P(t)可以如下递归地进行定义

关系元组演算
(3)关系元组演算公式的完整定义

战德臣 教授

关系元组演算公式的基本形式：{ t | P(t) }

其中公式P(t)可以递归地进行构造：

- 三种形式的原子公式是公式
  - ✓ s ∈ R
  - ✓ s[A] θ c
  - ✓ s[A] θ u[B]
- 如果*P*是公式，那么¬ P也是公式
- 如果P1 , P2是公式，则P1 ∧ P2 ， P1 ∨ P2 也是公式
- 如果P(t)是公式，R是关系，则∃(t∈R)(P(t))和∀(t∈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；θ；∃；∀；¬；∧；∨；
- 公式只限于以上形式

公式定义很简单，理解运用是关键

关系元组演算公式 之
原子公式及与、或、非之理解与运用

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

关系元组演算公式 之原子公式及与、或、非之理解与运用
(1)元组演算公式之原子公式

战德臣 教授

关系元组演算公式的基本形式：{ t | P(t) }

- P(t)可以是如下三种形式之一的原子公式
  - ❑ t ∈ R

    t 是关系 R 中的一个元组，例如：{ t | t ∈ Student }
  - ❑ s[A] θ c

    元组分量s[A]与常量 c 之间满足比较关系θ.

    θ：比较运算符 <, <=, =, <>, >, >=

    例如：{ t | t∈R ∧ t [Sage] <= 19 ∧ t [Sname] = '张三' }

t →

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

关系元组演算公式 之原子公式及与、或、非之理解与运用

(1)元组演算公式之原子公式

战德臣 教授

- $s[A]\ \theta\ u[B]$

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θ.

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

“检索出年龄不是最小的所有同学”

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

t → 98030102 张四 女 20 03 980301

u → 98040202 王四 男 21 04 980402

关系元组演算公式 之原子公式及与、或、非之理解与运用
(2)元组演算公式之与、或、非运算符

战德臣 教授

- P(t)可以由公式加运算符 ∧(与)、 ∨(或)、 ¬(非)递归地构造
  - 如果F是一个公式，则 ¬F 也是公式
  - 如果F1、F2是公式，则 F1 ∧ F2, F1 ∨ F2也是公式

F1	F2		F1 ∧ 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 ∨ 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			¬ F1
真/True			假/False
假/False			真/True

关系元组演算公式 之原子公式及与、或、非之理解与运用
(2)元组演算公式之与、或、非运算符

战德臣 教授

- 例如：检索出年龄小于20岁并且是男同学的所有学生。

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

- 再例如：检索出年龄小于20岁或者03系的所有男学生。

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

- 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

Student

	S#	Sname	Ssex	Sage	D#	Sclass
	98030101	张三	男	20	03	980301
t →	98030102	张四	女	20	03	980301
	98030103	张五	男	19	03	980301
	98040201	王三	男	20	04	980402
	98040202	王四	男	21	04	980402
	98040203	王五	女	19	04	980402

关系元组演算公式 之原子公式及与、或、非之理解与运用
(2)元组演算公式之与、或、非运算符

战德臣 教授

- 再例如：检索出不是03系的所有学生

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

- 再例如：检索不是(小于20岁的男同学)的所有同学

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

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

t →

关系元组演算公式 之原子公式及与、或、非之理解与运用
(3)注意运算符之次序及语义正确性

战德臣 教授

- **P(t)**运算符优先次序(括弧；θ；∃；∀；¬；∧；∨)示例
- 请注意下述语句的结果差异

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

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

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

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

关系元组演算公式 之
存在量词与全称量词之理解与运用

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

关系元组演算公式 之存在量词与全称量词之理解与运用

(1)存在量词与全称量词公式

战德臣 教授

- 构造P(t)还有两个运算符： ∃ (存在)、 ∀ (任意)
  - 如果F是一个公式，则 ∃ (t ∈ r) (F(t)) 也是公式
  - 如果F是一个公式，则 ∀ (t ∈ r) (F(t))也是公式
- 运算符∃ 和∀，又称为量词，前者称**“存在量词”**，后者称**“全称量词”**
- 而被 ∃ 或 ∀ 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 ∈ r	F(t)	∃ (t ∈ 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)=假	假/False
	有一个 t, 使 F(t)=真	真/True

t ∈ r	F(t)	∀ (t ∈ 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)=真	真/True
	有一个 t, 使 F(t)=假	假/False

关系元组演算公式 之存在量词与全称量词之理解与运用
(2)存在量词与全称量词公式之应用

战德臣 教授

- 例如：“检索出年龄不是最小的所有同学”

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

- 请大家写一下，在关系代数中，如何表达上面的查询需求？

$$\pi_{Student.*}(\sigma_{Student.Sage>S1.Sage}(Student \times \rho_{S1}(Student)))$$

Student

	S#	Sname	Ssex	Sage	D#	Sclass
	98030101	张三	男	20	03	980301
t →	98030102	张四	女	20	03	980301
	98030103	张五	男	19	03	980301
	98040201	王三	男	20	04	980402
u →	98040202	王四	男	21	04	980402
	98040203	王五	女	19	04	980402

关系元组演算公式 之存在量词与全称量词之理解与运用
(2)存在量词与全称量词公式之应用

战德臣 教授

- 再例如：检索出课程都及格的所有同学

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

- 用关系代数，如何书写上面例子？
- 请注意上式写成下面的公式会表达什么意思？

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

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

t

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

u

关系元组演算公式 之存在量词与全称量词之理解与运用
(2)存在量词与全称量词公式之应用

战德臣 教授

➢例如：检索计算机系的所有同学

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

➢ 例如：检索出比张三年龄小的所有同学？

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

Student

	S#	Sname	Ssex	Sage	D#	Sclass
w	98030101	张三	男	20	03	980301
	98030102	张四	女	20	03	980301
	98030103	张五	男	19	03	980301
t	98040201	王三	男	20	04	980402
	98040202	王四	男	21	04	980402
	98040203	王五	女	19	04	980402

Dept

D#	Dname	Dean	
01	机电	李三	
02	能源	李四	
03	计算机	李五	u
04	自动控制	李六	

关系元组演算公式 之存在量词与全称量词之理解与运用
(2)存在量词与全称量词公式之应用

战德臣 教授

➢ 例如：检索学过所有课程的同学

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

Course

C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003
002	高等数学	80	12	004

u

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

t

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

s

关系元组演算公式 之存在量词与全称量词之理解与运用
(2)存在量词与全称量词公式之应用

战德臣 教授

➢ 例如：检索所有同学所有课程全都及格的系

$$\{ t \mid t \in Dept \wedge \forall (s \in Student \wedge s[D\#] = t[D\#])$$

$$(\forall (u \in SC \wedge s[S\#] = u[S\#])\ (\ u[Score] >= 60)) \}$$

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

s

Dept

D#	Dname	Dean
01	机电	李三
02	能源	李四
03	计算机	李五
04	自动控制	李六

t

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

u

关系元组演算之应用训练
语义正确性与等价性变换训练

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

关系元组演算之应用训练语义正确性与等价性变换训练
(1)元组演算的等价性变换

战德臣 教授

{ t | P(t) }

- **P(t)**公式，如谓词演算一样，也有一系列演算的等价性
- 例如： θ 与¬ 的等价性(符号⇔表示等价于)

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

- 等价性示例：

{ t | t∈Student ∧ ¬ (t[D#] = '03') }

等价于

{ t | t∈Student ∧ t[D#] <> '03' }

关系元组演算之应用训练语义正确性与等价性变换训练
(1)元组演算的等价性变换

战德臣 教授

- 再例如 ∧、∨与¬ 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”

但下式是否正确呢?

$\{ u \mid u \in SC \wedge \neg(u[C\#] <> '001' \wedge u[C\#] <> '002') \}$

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

u

关系元组演算之应用训练语义正确性与等价性变换训练
(1)元组演算的等价性变换

战德臣 教授

- 等价性再示例：

“既年龄小于20岁又是男性的所有学生”

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

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”

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

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

S

关系元组演算之应用训练语义正确性与等价性变换训练
(1)元组演算的等价性变换

战德臣 教授

➢例如∀ 、 ∃与¬ 运算之间的等价性

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

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

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

➢ 等价性示例：

“既学过001课程又学过002课程的学生”

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

或者写成

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

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”

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

关系元组演算之应用训练
四个最复杂的例子

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

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)

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

$\{\ t[Sname]\ |\ t \in Student \wedge \forall(u \in Course \wedge u[Tname]='李明')$

$(\exists(w \in SC)(w[S\#]=t[S\#] \wedge w[C\#]=u[C\#]\ ))\ \}$

战德臣 教授

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)

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

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

战德臣 教授

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)

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

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

战德臣 教授

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)

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

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

关系元组演算之应用训练
将关系代数转换为元组演算

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

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、θ-连接操作。
- 并运算：$R \cup S = \{ t \mid t \in R \vee t \in S \}$
- 差运算：$R - S = \{ t \mid t \in R \wedge \neg t \in S \}$
- 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：$\pi_A(R) = \{ t[A] \mid t \in R \}$

关系元组演算之应用训练将关系代数转换为元组演算
(2)元组演算公式总结

战德臣 教授

关系元组演算公式的基本形式：**{ t | P(t) }**

其中公式**P(t)**的构造：

- 三种形式的原子公式是公式
  - ✓ **s ∈ R；　s[A] θ c；　s[A] θ u[B]**
- 如果***P***是公式，那么**¬ P**也是公式
- 如果**P1 , P2**是公式，则**P1 ∧ P2 ， P1 ∨ P2** 也是公式
- 如果**P(t)**是公式，**R**是关系，则**∃(t∈R)(P(t))**和**∀(t∈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；θ；∃；∀；¬；∧；∨；
- 公式只限于以上形式

•用递归定义公式，组合、递归地构造公式
•一种用逻辑表达查询的思维
•以元组为基本单位进行循环，先找到元组，再找到元组分量，进行谓词判断；
•元组演算与关系代数可以相互转换(有前提，后面介绍)

关系域演算

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

关系域演算
(1)关系域演算公式

战德臣 教授

关系**域演算**公式的基本形式：$\{ < x_1 , x_2 , \dots , x_n > | P (x_1 , x_2 , \dots , x_n) \}$

其中，x_i代表域变量或常量，P为以x_i为变量的公式。

公式P可以递归地进行构造：

- 三种形式的原子公式是公式
  - ✓ $< x_1 , x_2 , \dots , x_n > \in R$。其中$x_i$代表域变量或常量，表示由域变量构成的$< x_1 , x_2 , \dots , x_n >$是属于关系R的
  - ✓ $x \theta c$。其中，域变量x与常量c之间满足比较关系θ。θ：比较运算符 <, <=, =, <>, >, >=
  - ✓ $x \theta y$。其中，域变量x与域变量y之间满足比较关系θ
- 如果*P*是公式，那么¬ P也是公式
- 如果P1 , P2是公式，则P1 ∧ P2 , P1 ∨ P2 也是公式
- 如果P是公式，x是域变量，则∃(x)(P(x))和∀(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；θ；∃；∀；¬；∧；∨；
- 公式只限于以上形式

关系域演算
(2)关系域演算公式构造示例

战德臣 教授

- 例如：检索出不是03系的所有学生

  { <a,b,c,d,e,f> | <a,b,c,d,e,f>∈Student ∧ e<>'03' }

- 例如：检索不是(小于20岁的男同学)的所有同学的姓名

  { <b> | ∃a,c,d,e,f (<a,b,c,d,e,f>∈Student ∧ ¬ (d < 20 ∧ c = '男')) }

- 例如：检索成绩不及格的同学姓名、课程及其成绩

  { <b,h,m> |∃a,c,d,e,f,g,l,j,k (<a,b,c,d,e,f>∈Student ∧ (g,h,l,j,k) ∈Course ∧ (a,g,m) ∈SC ∧ m<60) }

a b c d e f

Student

S#	Sname	Ssex	Sage	D#	Sclass
98030101	张三	男	20	03	980301
98030102	张四	女	20	03	980301
98030103	张五	男	19	03	980301
98040201	王三	男	20	04	980402
98040202	王四	男	21	04	980402
98040203	王五	女	19	04	980402

g h i j k

Course

C#	Cname	Chours	Credit	T#
001	数据库	40	6	001
003	数据结构	40	6	003
004	编译原理	40	6	001
005	C 语言	30	4.5	003
002	高等数学	80	12	004

a g m

SC

S#	C#	Score
98030101	001	92
98030101	002	85
98030101	003	88
98040202	002	90
98040202	003	80
98040202	001	55
98040203	003	56
98030102	001	54
98030102	002	85
98030102	003	48

关系域演算
(3)关系域演算与关系元组演算的比较

战德臣 教授

➢ **元组演算**的基本形式：　{ t | P(t) }

域演算的基本形式：　{ < x_1 , x_2 , … , x_n > | P (x_1 , x_2 , … , 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。

➢ 公式的运算符(∧ (与)、 ∨ (或)、 ¬(非)、 ∀ (全称量词)和∃(存在量词))是相同的，只是其中的变量不同

➢ 元组演算和域演算可以等价互换。

基于关系域演算的QBE语言

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

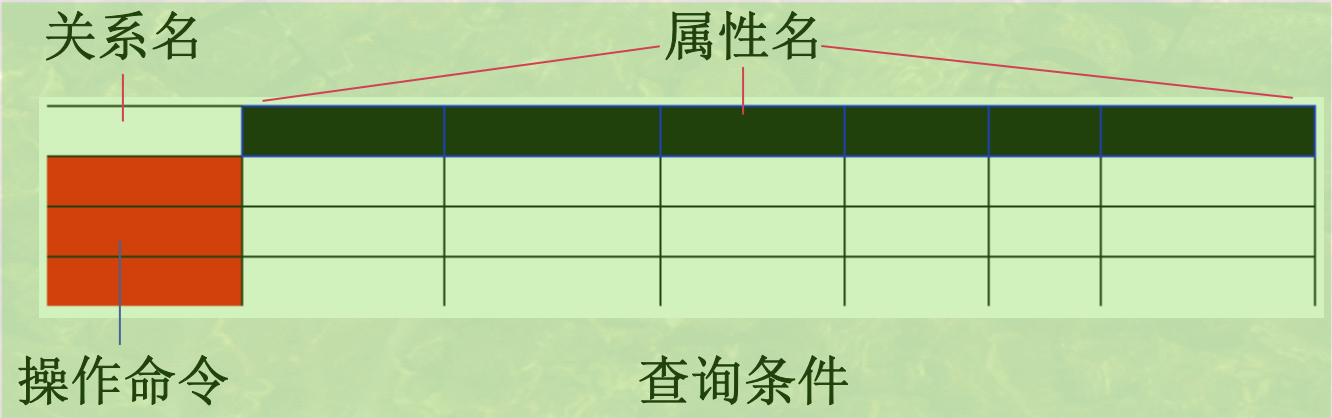
域演算语言QBE

- QBE: Query By Example
- 1975年由M. M. Zloof提出，1978年在IBM370上实现
- 特点：操作独特，基于屏幕表格的查询语言，不用书写复杂的公式，只需将条件填在表格中即可
- 是一种高度非过程化的查询语言
- 特别适合于终端用户的使用

Student	S#	Sname	Ssex	Sage	D#	Sclass
		P.X	Female	<17		

QBE操作框架由四个部分构成

- 关系名区：用于书写欲待查询的关系名
- 属性名区：用于显示对应关系名区关系的所有属性名
- 操作命令区：用于书写查询操作的命令
- 查询条件区：用于书写查询条件



QBE的操作命令

- Print 或 P. ---- 显示输出操作
- Delete或D. ---- 删除操作
- Insert或I. ---- 插入操作
- Update或U. ---- 更新操作

➢ 示例，如下图表示, 向Student表中插入两个元组

<98030105, 刘二, Male, 35, 03, 980301>

<98030106, 刘三, Male, 32, 03, 980301>

Student	S#	Sname	Ssex	Sage	D#	Sclass
I.	98030105	刘二	Male	35	03	980301
I.	98030106	刘三	Male	32	03	980301

基于关系域演算的QBE语言
(3)QBE的操作命令

战德臣 教授

- 再示例，如下图表示, 将**Student**表中如下的两个元组删除掉

  <98030105, 刘二, Male, 35, 03, 980301>

  <98030106, 刘三, Male, 32, 03, 980301>

Student	S#	Sname	Ssex	Sage	D#	Sclass
D.	98030105	刘二	Male	35	03	980301
D.	98030106	刘三	Male	32	03	980301

QBE的查询条件----简单条件

- QBE查询条件可以直接在查询条件区中书写，形式为 θ 参量
- θ 可以是<, >, >=, <=, =, <>；如省略θ，则默认为 =
- θ 参量中的参量可以是常量，直接书写；

➢ 例如：找出年龄小于17岁的所有同学

Student	S#	Sname	Ssex	Sage	D#	Sclass
P.				<17		

QBE的查询条件----不同属性上的与条件

- QBE不同属性上的与条件可以写在同一行中

➢ 例如：找出年龄小于17岁的所有女同学

Student	S#	Sname	Ssex	Sage	D#	Sclass
P.			Female	<17		

- 同一行中各个条件之间是“∧”关系(与), 如上图，可写为域演算公式：

{ <a,b,c,d,e,f> | <a,b,c,d,e,f> ∈Student ∧ c =‘Female’ ∧ d <17 }

战德臣 教授

QBE的查询条件----示例元素与投影

- 条件 θ 参量中的参量也可以是域变量，用任何一个值(不必是结果中的值)带有下划线表示，被称为示例元素. 示例元素下划线上面的值不起作用，被当作域变量名称来对待，只用于占位或是连接条件。不带下划线的则是构成实际条件一部分的值。

➢ 例如：找出年龄小于17岁的所有女同学的姓名

Student	S#	Sname	Ssex	Sage	D#	Sclass
		P.X	Female	<17		

- 当不是显示所有内容时，可在条件区对应要显示的列下面书写显示输出命令(即投影运算), 如上例输出姓名 P.x, 你可以任意写一个名字，如P.张三都是一样的, 他们是示例元素不是条件

战德臣 教授

QBE的查询条件----用示例元素实现‘与’运算和‘或’运算

- 当书写 ∨ 条件(或运算)时，可以采用在多行书写，然后在打印命令后使用不同的示例元素来表征，如下图, 一行写为P.X, 一行使用P.Y。
- 例如：找出年龄小于17岁或者年龄大于20岁的所有同学的姓名

Student	S#	Sname	Ssex	Sage	D#	Sclass
		P.X		<17		
		P.Y		>20		

- 如果一批 ∧ 条件分多行书写，则相互存在∧关系的行要采用相同的示例元素
- 例如：表示找出年龄大于17并且年龄小于等于20的同学姓名

Student	S#	Sname	Ssex	Sage	D#	Sclass
		P.X		>17		
		P.X		<=20		

战德臣 教授

QBE的查询条件----相当于括号的条件表示

- 也可以将∧、 ∨和¬ 条件写在操作命令区，如下所示。

Student	S#	Sname	Ssex	Sage	D#	Sclass
		P.X		<17		
∨		P.Y		>20		

- 当∧ 、∨和 ¬运算符写在操作区时，是对整行条件而言，相当于将该行条件放在括号中一样

Student	S#	Sname	Ssex	Sage	D#	Sclass
		P.X	Male	<17		
∨		P.Y	Female	>20		

- 如上例，表达的是：(年龄<17 并且 是男的) 或者 (年龄>20 并且是女的)

战德臣 教授

QBE的查询条件----用示例元素实现多个表的连接

- 当检索涉及多个表时，可利用同一连接条件使用相同的示例元素，来实现多个表的连接.
- 例如：李明老师教过的所有学生

Student	S#	Sname	Ssex	Sage	D#	Sclass
P.	X					

SC	S#	C#	Score
	X	Y	

Course	C#	Cname	Chours	Credit	Tname
	Y				李明

基于关系域演算的QBE语言
之应用训练

战德臣

哈尔滨工业大学　教授.博士生导师
黑龙江省教学名师
教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

基于关系域演算的QBE语言之应用训练
(1)用QBE进行“查询”的构造

战德臣 教授

➢例如：查询计算机系年龄大于19岁的男同学的姓名

Student	S#	Sname	Ssex	Sage	D#	Sclass
		P.张三		>19	X	
		P.张三	男		X	

Dept	D#	Dname	Dean
	X	计算机	

基于关系域演算的QBE语言之应用训练
(1)用QBE进行“查询”的构造

战德臣 教授

➢例如：查询计算机系或者年龄大于19岁或者是男同学的姓名

Student	S#	Sname	Ssex	Sage	D#	Sclass
		P.张三		>19	X	
		P.李四	男		Y	

Dept	D#	Dname	Dean
	X	计算机	
	Y	计算机	

基于关系域演算的QBE语言之应用训练
(1)用QBE进行“查询”的构造

战德臣 教授

➢例如：查询既选修了001号课程又选修了002号课程的学生的学号

SC	S#	C#	Score
	P.X	001	
	P.X	002	

➢ 再例如：找出比男同学张三年龄大的所有男同学的姓名

Student	S#	Sname	Ssex	Sage	D#	Sclass
		张三	男	X		
		P. 李四	男	> X		

基于关系域演算的QBE语言之应用训练
(1)用QBE进行“查询”的构造

战德臣 教授

➢例如：将张三同学的年龄更新为19岁

Student	S#	Sname	Ssex	Sage	D#	Sclass
		张三		U.19		

➢ 再例如：将每位同学的年龄增加1岁

Student	S#	Sname	Ssex	Sage	D#	Sclass
	0001			19		
U.	0001			19 + 1		

基于关系域演算的QBE语言之应用训练
(1)用QBE进行“查询”的构造

战德臣 教授

➢例如：找出成绩不及格同学的姓名及不及格的课程和分数

Student	S#	Sname	Ssex	Sage	D#	Sclass
	X	P.张三				

SC	S#	C#	Score
	X	Y	< 60
	X	Y	P. 50

Course	C#	Cname	Chours	Credit	Tname
	Y	P.数据库			

基于关系域演算的QBE语言之应用训练
(2)用QBE实现关系代数

战德臣 教授

- 关系代数的并、差、乘积、选择和投影运算可以用QBE来实现
  - T = R U S

R	A1	A2	…	Ak
	RA1	RA2	…	RAk

S	A1	A2	…	Ak
	SA1	SA2	…	SAk

T	A1	A2	…	Ak
I.	RA1	RA2	…	RAk
I.	SA1	SA2	…	SAk

基于关系域演算的QBE语言之应用训练
(2)用QBE实现关系代数

战德臣 教授

- 关系代数的并、差、乘积、选择和投影运算可以用QBE来实现
  - T = R – S

R	A1	A2	…	Ak
	RA1	RA2	…	RAk

S	A1	A2	…	Ak
	SA1	SA2	…	SAk

T	A1	A2	…	Ak
I.	RA1	RA2	…	RAk
D.	SA1	SA2	…	SAk

基于关系域演算的QBE语言之应用训练
(2)用QBE实现关系代数

战德臣 教授

- 关系代数的并、差、乘积、选择和投影运算可以用QBE来实现
  - $T = R \times S$

R	A1	A2	…	An
	RA1	RA2	…	RAn

S	B1	B2	…	Bm
	SB1	SB2	…	SBm

T	A1	A2	…	An	B1	B2	…	Bm
I.	RA1	RA2	…	RAn	SB1	SB2	…	SBm

基于关系域演算的QBE语言之应用训练
(2)用QBE实现关系代数

战德臣 教授

- 关系代数的并、差、乘积、选择和投影运算可以用QBE来实现
  - $T = \pi_{i1,i2,\ldots,im}(R)$
  - 其中RAij 是某一个RAk

R	A1	A2	…	An
	RA1	RA2	…	RAn

T	T1	T2	…	Tm
I.	RAi1	RAi2	…	RAim

基于关系域演算的QBE语言之应用训练
(2)用QBE实现关系代数

战德臣 教授

- 关系代数的并、差、乘积、选择和投影运算可以用QBE来实现
  - $T = \sigma_F(R)$
  - 其中Condition Box中表示选择运算中的F公式(详细转换略)

R	A1	A2	…	An
	RA1	RA2	…	RAn

T	A1	A2	…	An
I.	RA1	RA2	…	RAn

Condition Box

关系演算的安全性

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

战德臣 教授

关系运算的安全性

“不产生无限关系和无穷验证的运算被称为是安全的”

- 关系代数是一种集合运算，是安全的
  - ✓集合本身是有限的，有限元素集合的有限次运算仍旧是有限的。
- 关系演算不一定是安全的
  - ❑例如：$\{ t \mid \neg(R(t)) \}$, $\{ t \mid R(t) \vee t[2]>3 \}$可能表示无限关系
  - ✓R(t)是有限的，但不在R(t)中的元素就可能是无限的，后例中的t[2]>3是无限的。
- 再例如：$(\exists u)(\omega(u))$, $(\forall u)(\omega(u))$可能导致无穷验证
- 前者被称为“假验证”，即验证所有元素是否都使得ω(u)为false；后者被称为“真验证”，即验证所有元素是否都使得ω(u)为true。检验所有元素就可能造成无穷。

关系演算的安全性
(2)关系演算的约束?

战德臣 教授

➢需要对关系演算施加约束条件，即任何公式都在一个集合范围内操作，而不是无限范围内操作，才能保证其安全性。

安全约束有限集合DOM

- DOM(ψ)是一个有限集合，其中的每个符号要么是ψ中明显出现的符号，要么是出现在ψ中的某个关系R的某元组的分量。
- DOM主要用于约束ψ中一些谓词的计算范围，它不必是最小集合。

安全元组演算表达式

满足下面三个条件的元组演算表达式{ t|ψ(t) }称为安全表达式。

- 只要t满足ψ，t的每个分量就是DOM(ψ)的一个成员。
  - ✓{ t | ψ(t) }中t的取值只能是DOM中的值，有限的。
- 对于ψ中形如(∃u)(ω(u))的子表达式，若u满足ω，则u的每个分量都是DOM(ω)中的成员。
  - ✓{ t | ψ(t) }中的每个(∃u)(ω(u))子表达式，只需要验证DOM中的元素是否有使ω(u)为真的元素。而对于DOM以外的元素，已经明确其都不满足ω(u)，无需验证。
- 对于ψ中形如(∀u)(ω(u))的子表达式，若u不满足ω，则u的每个分量都是DOM(ω)中的成员。
  - ✓{ t | ψ(t) }中的每个(∀u)(ω(u))子表达式，只需要验证DOM中的元素是否有使ω(u)为假的元素。而对于DOM以外的元素，已经明确其都满足ω(u)，无需验证。

安全域演算表达式

略。同学可仿照安全元组演算表达式自己去定义。

关于关系运算的一些观点

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

关于关系运算的一些观点

战德臣 教授

- 关系运算有三种：关系代数、关系元组演算和关系域演算
- 三种关系运算都是抽象的数学运算，体现了三种不同的思维
  - ✓关系代数---以集合为对象的操作思维，由集合到集合的变换
  - ✓元组演算---以元组为对象的操作思维，取出关系的每一个元组进行验证，有一个元组变量则可能需要一个循环，多个元组变量则需要多个循环
  - ✓域演算---以域变量为对象的操作思维，取出域的每一个变量进行验证看其是否满足条件

关于关系运算的一些观点

战德臣 教授

- 三种运算之间是等价的
  - ✓关系代数 与 安全的元组演算表达式 与 安全的域演算表达式 是等价的。即一种形式的表达式可以被等价地转换为另一种形式
- 三种关系运算都可说是非过程性的
  - ✓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，但却是衡量数据库语言完备性的基础
  - ✓一个数据库语言如果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
  - ✓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

关于关系运算的一些观点

战德臣 教授

- 数据库语言可以基于这三种抽象运算来设计
  - ✓用“键盘符号”来替换抽象的数学符号
  - ✓用易于理解的符号组合来表达抽象的数学符号
  - ✓例如：ISBL语言---基于关系代数的数据库语言

ISBL 运算	对应的关系代数运算	说明/差别说明
R + S	$R \cup S$	
R – S	$R - S$	当 R 与 S 有相异属性名时，关系代数不能操作，而 ISBL 能按如下操作：设 $R(A_1,\ldots,A_n,A_{n+1},\ldots,A_{n+m})$, $S(A_1,\ldots,A_n,B_1,\ldots,B_k)$ 则 $R-S = R-(\pi_{A1,\ldots,An}(S) \times \pi_{An+1,\ldots,An+m}(R))$
R.S	$R \cap S$	
R:F	$\sigma_F(R)$	
R%A₁,A₂,…,Aₙ	$\pi_{A1,A2,\ldots,An}(R)$	可以重新命名关系的属性名。R%A→B
R*S	R (Join) S	R*S 隐含了 R×S，当 R, S 有相同列名时，完成的是连接操作，否则完成的是乘积操作
LIST		打印结果语句
=		赋值语句
N!		赋值延迟符号
		ISBL 没有集聚函数，但其可与 PL/1 程序结合使用，可利用 PL/1 的函数实现集聚函数

关于关系运算的一些观点

战德臣 教授

✓ 再例如：**Ingres**系统的**QUEL**语言

range of t_1 is R_1
range of t_2 is R_2
… …
range of t_r is R_r
retrieve($t_{i1}.A_1$,$t_{i2}.A_2$,…,$t_{ik}.A_k$)
where $\varphi(t_1,\ldots,t_r)$

$$\{ u^k \mid (\exists t_1)(\exists t_2)\ldots(\exists t_r)(R_1(t_1) \wedge R_2(t_2) \wedge \ldots \wedge R_r(t_r) \wedge u[1]=t_{i1}[j1] \wedge u[2]=t_{i2}[j2] \ldots \wedge u[k]=t_{ik}[jk] \wedge \varphi \}$$

回顾本讲学了什么?

战德臣

哈尔滨工业大学　教授.博士生导师

黑龙江省教学名师

教育部大学计算机课程教学指导委员会委员

Research Center on Intelligent Computing for Enterprises & Services, Harbin Institute of Technology

回顾本讲学习了什么?

战德臣 教授