第2章 关系模型

2.1 基本概念

2.2 关系代数

2.3 关系演算

2.1 基本概念

- 关系模型是以集合论中的关系(relation)概念为基础发展起来的数据模型。关系模型理论由IBM公司San Jose研究实验室E.F.Codd率先提出《Communication of the ACM》

"A Rational Model of Data for Large Shared Data Banks"

- 早期代表系统
  - System R：由IBM研制
  - INGRES：由加州Berkeley分校研制
- 目前主流的关系数据库管理系统软件产品包括:
  - IBM DB2 UDB, Oracle，Informix，
  - Sybase，MS SQL Server，
  - MySQL
  - Access，Kingbase

关系模型的直观表示

整数

字符串

1,2,3,4,5

域

主键

属性名

学号	姓名	年龄	性别	系	年级
82021	张平	18	男	计算机	2
82022	王英	17	女	计算机	1
…	…	…	…	…	…
82030	李强	17	男	物理	1

关系

元组（行）

属性列

关系名：学生登记表

关系模式：学生（学号，姓名，年龄…）

关系的定义

- 定义2.1：域(domain)是一组值的集合，同一个域中的所有值均应具有相同的数据类型

- 例2-1：
  - D1={袁玲，吴丹，刘杰} D2={男，女}
  - D3={18,19}
- 其中：D1(姓名)、D2(性别)和D3(年龄)为域名，分别表示“人员”关系中姓名、性别和年龄的可取值范围
- D1的基数m1=3，D2的基数m2=2，D3的基数m3=2
- 域中元素一般无排列次序，如：D2={男，女}={女，男}

关系的定义

- 定义2.2：域D1，D2，…，Dn上的笛卡儿积是一个集合：

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

- 其中允许Di＝Dj且i≠j，将该集合中的每一个元素(d1,d2,…dn)称为一个元组(tuple)，元组中的每一个值di称为一个分量(component)，有n个分量的元组称为n元组

关系的定义

例2-2:已知三个域:
*D*1={袁玲，吴丹，刘杰} *D*2={男，女} *D*3={18,19}

则*D*1、*D*2和*D*3的笛卡尔积可以表示成一张二维表，如表2-1所示。

*D*1	*D*2	*D*3
袁玲	男	18
袁玲	男	19
袁玲	女	18
袁玲	女	19
吴丹	男	18
吴丹	男	19
吴丹	女	18
吴丹	女	19
刘杰	男	18
刘杰	男	19
刘杰	女	18
刘杰	女	19

关系的定义

- 该笛卡尔积的基数为3×2×2=12，即共有12个元组，(袁玲、男、18)和(吴丹、女、19)就是元组，“袁玲”、“女”、“吴丹”、“19”就是分量。
- 有时一个元组也被称为一条记录

关系的定义

- 定义2.3：D1×D2×…×Dn上的任意一个子集均是定义在域D1，D2，…Dn上的一个关系(relation)，记为R
- 由n个域构成的关系通常称为n元关系，关系中的每个元素即是这个关系的元组

姓名	性别	年龄
袁玲	男	18
吴丹	女	19
刘杰	男	18

关系的定义

- 由于域可以相同，为了加以区分，必须为每列起一个名字，称为属性(Attribute)
- 定义2.4：为关系的每个列所起的名字称为关系的属性，一般可表示为A，B，C…
- n目关系必有n个属性，表的任意一列对应一个属性，属性的名称称为“属性名”，具体元组的属性取值称为“属性值”
- 例：学生关系有学号、姓名、性别、系名、年龄等属性

关系的定义

- 定义2.5：如果一个关系中的某个属性或属性集能够唯一的确定一个元组，则称该属性（集）是这个关系上的超键（super key，SK）；如果将超键中的任一属性去掉后剩余的属性集不能唯一标识一个元组，则称该属性集是关系上的候选键（candidate key，CK）；通常从候选键中选择一个使用，这个候选键称为关系的主键（primary key，PK）
- 例 学生（学号,姓名,性别,系名,年龄,……）

关系的定义

- （学号,姓名）　超键

  学号　候选键 主键

- 一般情况下，如不加特别说明，键即指主键。如果关系R1中的某个属性集是另外一个关系R2的候选键，那么该属性集对于关系R1而言是它的外键（foreign key，FK）

- 例　学生表（学号，姓名，年龄,……)

  课程表（课程号，课程名，学分,……)

  选课表（学号，课程号，成绩,……)

关系模式

- 定义2.6：关系模式（relation schema）是对关系的型的描述，可以表示为：

  R(U，D，DOM，I，F)

- 其中，U是R的属性集合{A1，A2，…，An}，D是属性的取值范围，即域的集合{D1，D2，…，Dn}，DOM是U到D的映射集合{A1→D1，A2→D2，…，An→Dn}，I是完整性约束规则集，F是函数依赖集合
- 习惯上将关系简记为R(A1／D1，A2／D2，…，An／Dn)或者R(A1，A2，…，An)

关系模式与关系

- 关系可看作是关系模式在某一时刻的状态或内容，也就是说，关系模式是型，关系是值
- 关系模式是静态的、稳定的，而关系是动态的、随时间不断变化的，因为关系操作在不断地更新着数据库中的数据

关系的特征

（1）关系中不允许出现相同的元组

（2）关系中元组的顺序(即行序)是无关紧要的，在一个关系中可以任意交换两行的次序

（3）关系中属性的顺序是无关紧要的，即属性的顺序可以任意交换。

sno	sname	sex	age
801	Zhang	F	19
802	Li	M	20
803	Wang	M	20
804	Zhao	F	20
805	Qian	m	19

关系的特征

（4） 同一属性名下的各个属性值必须来自同一个域，是同一类型的数据。

（5） 关系中各个属性必须有不同的名字，不同的属性可来自同一个域，即他们的分量可以取自同一个域

（6） 关系中每个分量必须是不可分的数据项，或者说所有属性值都是原子的，是一个确定的值，而不是值的集合。属性值可以为空值，表示“未知”或“不可使用”，即不可“表中有表”

sno	sname	sex	age
801	Zhang	F	19
802	Li	M	20
803	Wang	M	20
804	Zhao	F	20
805	Qian	m	19

关系的类型

- 在关系数据库中，关系有以下三种类型：

  (1) 基本表：实际存在的表，对应实际存储数据的逻辑表示

  (2) 查询表：对基表查询得到的结果表

  (3) 视图表：从基本表或其他视图中导出的表

返回目录

关系数据库

- 在一个给定的应用领域中，所有关系的集合构成一个关系数据库

关系数据库的型与值

- 关系数据库的型：关系数据库模式

  对关系数据库的描述。

- 关系数据库模式包括
  - 若干域的定义
  - 在这些域上定义的若干关系模式
- 关系数据库的值：关系模式在某一时刻对应的关系的集合，简称为关系数据库

关系的完整性

- 关系模型的完整性（integrity）规则是为了保证数据的正确性和相容性而在关系上施加的约束条件

关系的完整性

（1）域完整性约束

元组分量在某个属性上的取值应在其值域之内；元组是否能在某个属性上取空值null，由该属性的语义决定

sno	sname	sex	age
801	Zhang	F	19
802	Li	M	20
803	Wang	M	20
804	Zhao	F	20
805	Qian	m	19

关系的完整性

（2）实体完整性约束

- 元组在键上的取值不可重复，且不能为null
- 按照键的定义，如果不同的元组在键上取相同值或空值，则将无法区分两个不同的元组

sno	cno	grade
801	01	92
801	02	78
801	03	85
802	01	82
802	02	90
803	01	88

关系的完整性

（3）参照完整性约束

- 属性集FK是关系R上的外键，属性集CK是关系S上的候选键，FK引用CK，R和S可以是不同的关系，也可以是相同的关系。

关系的完整性

（3）参照完整性约束

- 参照完整性约束要求R上的元组t在FK属性上的取值t[FK]必须是如下两种情况之一：
  - 等于关系S中某个元组在候选键CK上的值；
  - 取空值null；

sno	sname	sex	age
801	Zhang	F	19
802	Li	M	20
803	Wang	M	20
804	Zhao	F	20
805	Qian	m	19

sno	cno	grade
801	01	92
801	02	78
801	03	85
802	01	82
802	02	90
803	01	88

关系的完整性

（4）用户定义的完整性约束

- 用户定义的完整性是针对某一具体关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求
- 关系模型应提供定义和检验这类完整性的机制，以便用统一的系统的方法处理它们，而不要由应用程序承担这一功能

sno	cno	grade
801	01	92
801	02	78
801	03	85
802	01	82
802	02	90
803	01	88

关系模型的优点

- 与其它数据模型相比，关系模型突出的优点如下：
  - 关系模型提供单一的数据结构形式，具有高度的简明性和精确性
  - 关系模型的逻辑结构和相应的操作完全独立于数据存储方式，具有高度的数据独立性
  - 关系模型使数据库的研究建立在比较坚实的数学基础上
  - 关系数据库语言与一阶谓词逻辑的固有内在联系，为以关系数据库为基础的推理系统和知识库系统的研究提供了方便

第2章 关系模型

2.1 基本概念

2.2 关系代数

2.3 关系演算

2.2 关系代数

- 关系运算是关系数据库的数学基础，分为关系代数和关系演算两大类
- 关系代数中运算对象是关系，运算结果也是关系，运算符包括四类：

  (1) 集合运算符：∪(并)，－(差)，∩(交)，×(广义笛卡尔积)

  (2) 专门的关系运算符：σ (选择)，π (投影)，⋈(连接)，÷(除)

  (3) 算术比较符：>(大于)，≥(大于等于)，<(小于)，≤(小于等于)，=(等于)，≠(不等于)

  (4) 逻辑运算符：∧(与)，∨(或)，¬ (非)

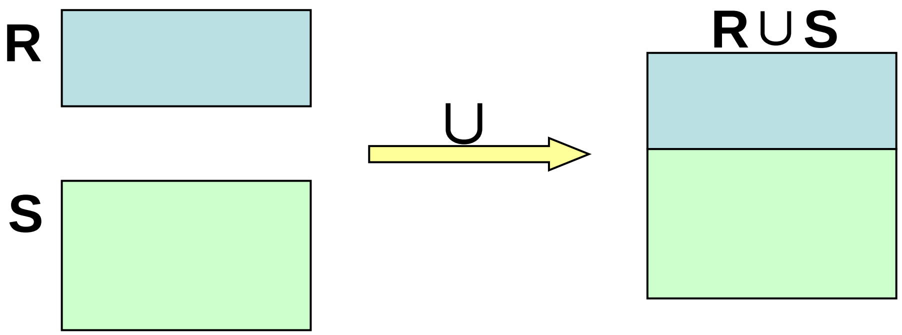
传统的集合运算

（1）并（union）

- R、S都是n元关系，且各自的对应属性具有相同的域，则关系R和S并的结果R∪S仍为n元关系，表示为：

$$R \cup S = \{t \mid t \in R \lor t \in S\}$$

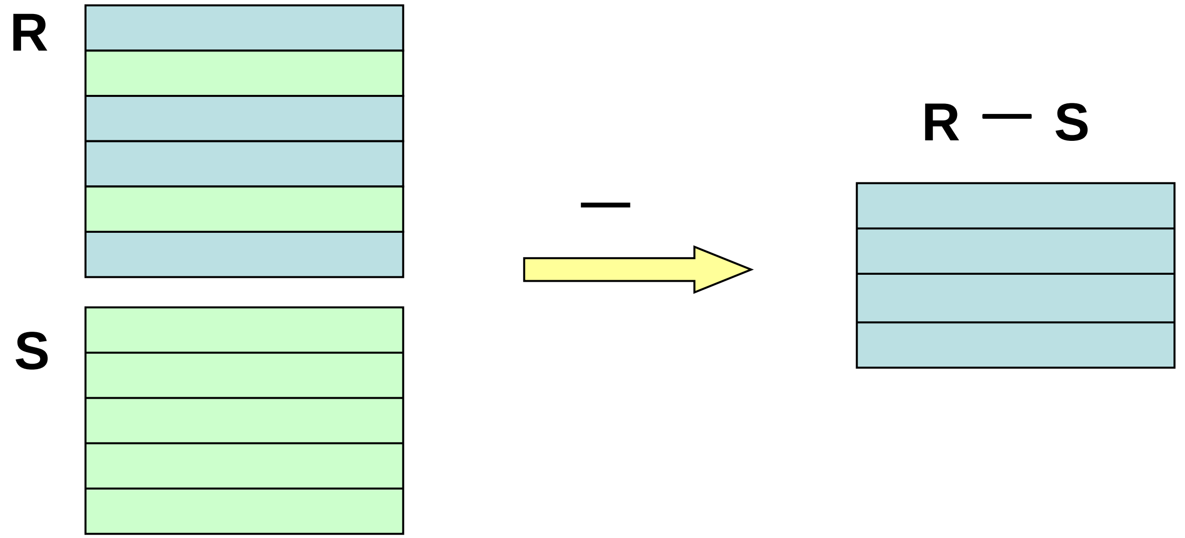
- 满足并运算要求的两个关系是“并兼容”的



传统的集合运算

（2）差（difference）

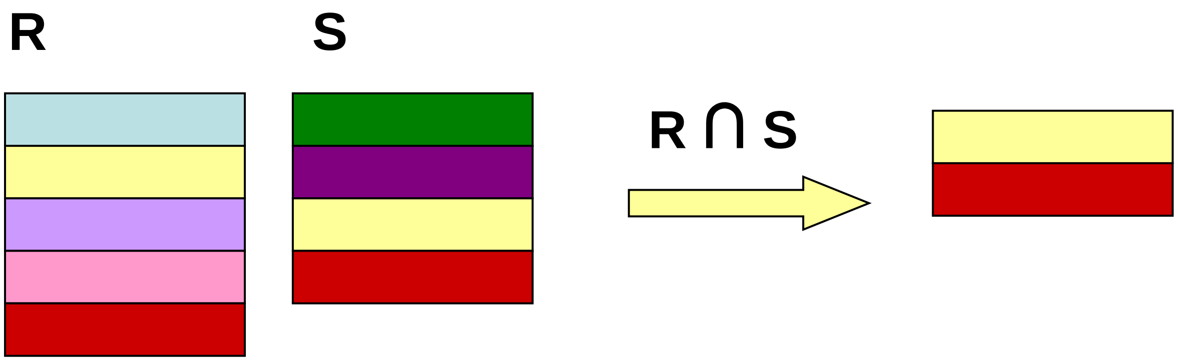
- R和S是并兼容的n元关系，R与S的差R－S仍为n元关系，表示为：$R-S=\{t \mid t \in R \wedge t \notin S\}$ 即R－S由那些仅出现在R中且不出现在S中的元组构成



传统的集合运算

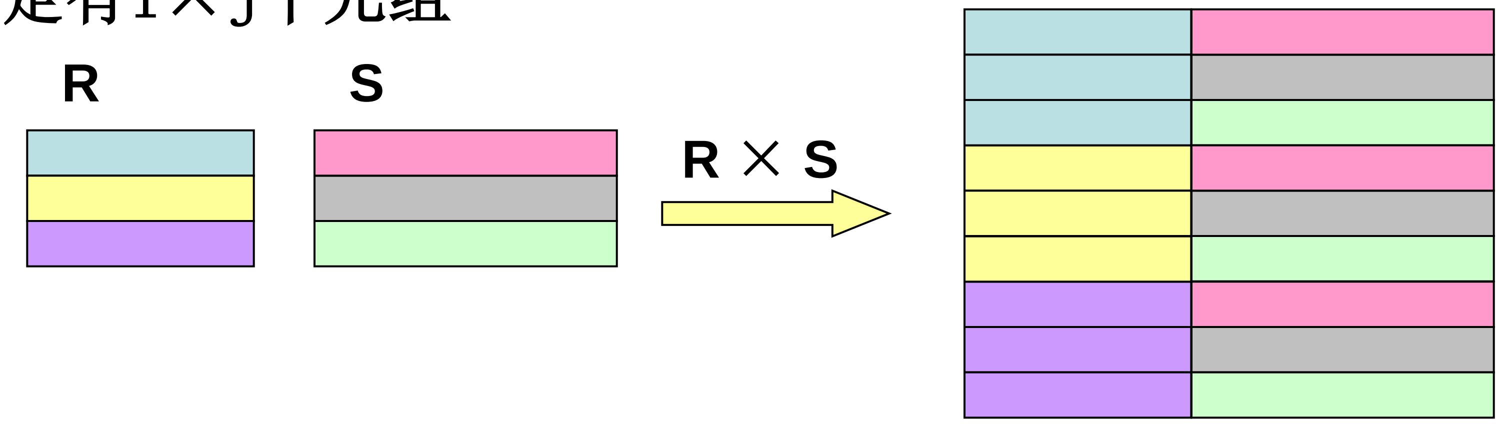
（3）交（intersection）

- R和S是并兼容的n元关系，R与S的交R∩S仍为n元关系，表示为：$R \cap S = \{t \mid t \in R \land t \in S\}$ 即R∩S由那些既在R中出现又在S中出现的元组构成



传统的集合运算

- （4）笛卡儿积（Cartesian product）
- m元关系R与n元关系S的笛卡儿积是（m＋n）元关系，且每个元组的前m个分量是R的一个元组，后n个分量是S的一个元组，可以表示为：
- $R \times S = \{t(m+n) \mid t(m) \in R \wedge t(n) \in S\}$
- 如果R中共有i个元组，S中共有j个元组，则R×S一定有i×j个元组



传统的集合运算

A	B	C
a	b	c
d	e	f
g	h	I

(a) R(A,B,C)

A	B	C
a	b	c
j	k	l
m	n	p

(b) S(A,B,C)

A	B	C
a	b	c
d	e	f
g	h	i
j	k	l
m	n	p

(a)R∪S

A	B	C
d	e	f
g	h	i

(b)R－S

A	B	C
a	b	c

(c)R∩S

传统的集合运算

A	B	C
a	b	c
d	e	f
g	h	I

(a) R(A,B,C)

A	B	C
a	b	c
j	k	l
m	n	p

(b) S(A,B,C)

A	B	C	A	B	C
a	b	c	a	b	c
a	b	c	j	k	l
a	b	c	m	n	p
d	e	f	a	b	c
d	e	f	j	k	l
d	e	f	m	n	p
g	h	i	a	b	c
g	h	i	j	k	l
g	h	I	m	n	p

(d)R×S

专门的关系运算

（1）选择（selection）

$$\sigma_F(R) = \{t \mid t \in R \land F(t)=\text{true}\}$$

其中σ 是选择算符，表示进行选择运算，F是选择条件，最简单的选择条件形如Xθ Y或者Xθ c，X，Y是属性名，c是常量，θ 是算术比较运算符，包括>，≥，<，≤，=，≠等。可以由简单的选择条件通过逻辑运算符构成更为复杂的条件，逻辑运算符包括∧，∨和￢。

专门的关系运算

- Relation r

A	B	C	D	
α =	α	1	7	>5
α ≠	β	5	7	>5
β =	β	12	3	≤5
β =	β	23	10	>5

- $\sigma_{A=B \wedge D > 5}(r)$

A	B	C	D
α	α	1	7
β	β	23	10

专门的关系运算

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c3
a3	b3	c3
a4	b4	c5

(a)关系R

A	B	C
a1	b1	c1
a1	b2	c2
a3	b3	c3

(b)$\sigma_{A='a1' \vee B='b3'}(R)$

A	B	C
a2	b2	c3
a3	b3	c3

(c)$\sigma_{A>'a1' \wedge C='c3'}(R)$

专门的关系运算

（2）投影（projection）

$$\Pi_X(R) = \{t[X] \mid t \in R\}$$

- 其中，X是关系R的一个属性子集，t[X]表示关系R上的元组t在属性子集X上的分量
- 由于关系的属性间是无序的，因此X中的属性顺序可以和它们在R中的出现顺序不一致。此外，需要注意的是因为关系中不允许有重复元组，因此在投影后需要将可能出现的重复元组进行合并

专门的关系运算

- Relation *r*:

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

$\Pi_{A,C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

专门的关系运算

A	B	C
a1	b1	c1
a1	b2	c2
a2	b2	c3
a3	b3	c3
a4	b4	c5

R

A	B
a1	b1
a1	b2
a2	b2
a3	b3
a4	b4

(a) $\Pi_{A,B}(R)$

A
a1
a2
a3
a4

(b) $\Pi_{A}(R)$

B	C
b1	c1
b2	c2
b2	c3
b3	c3
b4	c5

(c) $\Pi_{B,C}(R)$

专门的关系运算

（3）连接（join）

■θ 连接

连接是笛卡儿积和选择的复合运算，可以表示为：

$$R \bowtie_{A\theta B} S = \{rs \mid r \in R \wedge s \in S \wedge r[A]\,\theta\, s[B]\} = \sigma_{A\theta B}(R \times S)$$

其中，⋈是连接运算符，Aθ B是连接条件，θ 是算术比较运算符。$R \bowtie_{A\theta B} S$表示将关系R和S在各自的属性A，B上满足θ 条件的元组连接起来。θ 条件可以有多个，它们之间用逻辑算符连接。这样的连接一般被称为“θ 连接”。特别的，如果θ 是“等于”比较，称这样的连接运算是“等连接”或“等值连接”

专门的关系运算

（3）连接（join）

- 自然连接

消除了重复属性的等值连接，称为“自然连接”。表示如下：

$$R \bowtie S = \pi_{Attr(R) \cup (Attr(S)-A)}(\sigma_{R.A=S.A}(R \times S))$$

其中Attr(R)是关系R的属性全集

$R \bowtie S$

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2

$R \underset{R.B=S.B}{\bowtie} S$

A	R.B	C	S.B	E
a1	b1	5	b1	3
a1	b2	6	b2	7
a2	b3	8	b3	10
a2	b3	8	b3	2

专门的关系运算

（3）连接（join）

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

R

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

$R \bowtie_{B<D} S$

D	E
3	4
7	6

S

A	B	C	D	E
1	2	3	3	4
1	2	3	7	6

$R \bowtie_{B<D \wedge C<E} S$

专门的关系运算

- Relations r, s:

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

s

- r ⋈ s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

专门的关系运算

- Relations r, s:

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

s

- r ⋈ s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

专门的关系运算

- Relations r, s:

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

s

- r ⋈ s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

专门的关系运算

- Relations r, s:

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

s

- r ⋈ s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

专门的关系运算

- Relations r, s:

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	$\in$

s

- r $\bowtie$ s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

专门的关系运算

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

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

A	R.B	C	S.B	E
a1	b1	5	b2	7
a1	b1	5	b3	10
a1	b2	6	b2	7
a1	b2	6	b3	10
a2	b3	8	b3	10

$R \bowtie S$

A	B	C	E
a1	b1	5	3
a1	b2	6	7
a2	b3	8	10
a2	b3	8	2

$R \underset{R.B=S.B}{\bowtie} S$

A	R.B	C	S.B	E
a1	b1	5	b1	3
a1	b2	6	b2	7
a2	b3	8	b3	10
a2	b3	8	b3	2

专门的关系运算

Student

sno	sname	sex	age
801	Zhang	F	19
802	Li	M	20
803	Wang	M	20
804	Zhao	F	20
805	Qian	m	19

SC

sno	cno	grade
801	01	92
801	02	78
801	03	85
802	01	82
802	02	90
803	01	88

Course

cno	cname	credit
01	DB	4
02	DS	4
03	OS	4

- Relation schema
  - *Student* (**sno**, name, sex, age)
  - *Course* (**cno**, cname, credit)
  - *SC* (**sno, cno**, grade)

专门的关系运算

（3）连接（join）

■外连接

外连接将不满足连接条件的元组仍然保留在结果关系中，并在缺失属性上取null。根据保留元组的不同，外连接又分为左外连接，右外连接，全外连接

sno	sname	sex	age
801	Zhang	F	19
802	Li	M	20
803	Wang	M	20
804	Zhao	F	20
805	Qian	m	19

sno	sname	sex	age	cno	cname	credit	grade
801	Zhang	F	19	01	DB	4	92
801	Zhang	F	19	02	DS	4	78
801	Zhang	F	19	03	OS	4	85
802	Li	M	20	01	DB	4	82
802	Li	M	20	02	DS	4	90
803	Wang	M	20	01	DB	4	88

专门的关系运算

（3）连接（join）

A	B	C
a1	b1	c1
a2	b2	c2
a3	b3	c3

R

B	C	D
b1	c1	d1
b3	c3	d3
b4	c4	d4

S

A	B	C	D
a1	b1	c1	d1
a2	b2	c2	null
a3	b3	c3	d3

R*⋈S

A	B	C	D
a1	b1	c1	d1
a3	b3	c3	d3
null	b4	c4	d4

R⋈*S

A	B	C	D
a1	b1	c1	d1
a2	b2	c2	null
a3	b3	c3	d3
null	b4	c4	d4

R*⋈*S

2. 专门的关系运算

（3）连接（join）

- 半连接

 R和S的半连接是将其自然连接的结果在R的属性集上投影，可以表示为：

$$R \ltimes S = \pi_R (R \bowtie S), \quad S \ltimes R = \pi_S (R \bowtie S)$$

- 半连接不具有交换率

专门的关系运算

（3）连接（join）

A	B	C
a1	b1	c1
a2	b1	c1
a3	b2	c3
a4	b4	c4

R

B	C	D
b1	c1	d1
b1	c1	d2
b2	c2	d2
b4	c4	d3

S

A	B	C
a1	b1	c1
a2	b1	c1
a4	b4	c4

$R \ltimes S$

B	C	D
b1	c1	d1
b1	c1	d2
b4	c4	d3

$S \ltimes R$

专门的关系运算

（4）除（Divison）

给定两个关系R(X，Y)，S(Y)，其中X，Y均是属性的集合。设T(X)＝R(X，Y)÷S(Y)表示R与S除运算的结果，÷是除运算的算符，结果关系T的属性集由X构成，则T与S的笛卡儿积必定是R的一个子集。R÷S可表示如下：

$$T=R\div S=\{t[X]\mid t\in R\wedge \forall s\in S,\ t[X]s\in R\}=\pi_X(R)-\pi_X((\pi_X(R)\times S)-R)$$

式中t[X]s表示由元组t在属性子集X上的分量同关系S的元组s构成的新元组

专门的关系运算

（4）除（Divison）

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a1	b1	c3	d3
a2	b2	c2	d2
a3	b3	c1	d1
a3	b3	c3	d3

R

C	D
c1	d1
c3	d3

S

A	B
a1	b1
a3	b3

R÷S

专门的关系运算

结果属性　象集属性

A	B	C
a1	b1	c2
a2	b3	c7
a3	b4	c6
a1	b2	c3
a4	b6	c6
a2	b2	c3
a1	b2	c1

关系R

B	C	D
b1	c2	d1
b2	c1	d1
b2	c3	d2

关系S

A
a1

R÷S

- 在关系R中，A属性可以取{a1, a2, a3, a4}。
  - a1的象集为{(b1, c2), (b2, c3), (b2, c1)}
  - a2的象集为{(b3, c7), (b2, c3)}
  - a3的象集为{(b4, c6)}
  - a4的象集为{(b6, c6)}
- S在(B, C)上的投影为：
- {(b1, c2), (b2, c3), (b2, c1)}
- 显然，只有a1的象集$(B, C)_{a1}$包含了S在(B, C)属性组上的投影，所以R÷S={a1}

关系运算

- 关系代数运算
  - 并、差、交、笛卡尔积、选择、投影、连接、除
- 基本运算
  - 并、差、笛卡尔积、选择、投影
- 交、连接、除
  - 可以用5种基本运算来表达
  - 引进它们不能增强关系代数的表达能力，但可以简化表达
- 最小完备集，如 {∪, －, ⋈, σ, Π}
- 最小完备集不是唯一的，如{∪, －, ×, σ, Π}

关系代数查询实例

Student(S#, Class, Sname, Sex, Age)

Course（C#, Cname, Teacher）

SC（S#, C#, Grade）

- 例2.1.王老师开设课程的编号及课程名：

  $\pi_{C\#,\ Cname}(\sigma_{Teacher='王'}(Course))$

- 例2.2. 至少选修了一门王老师所开设课程的学生学号：

  $\pi_{S\#}(SC \bowtie \pi_{C\#}(\sigma_{Teacher='王'}(Course)))$

关系代数查询实例

例：检索选修课程名为Maths的学生学号与姓名

$\Pi_{S\#,\ Sname}(\sigma_{Cname=\text{'Maths'}}(S \bowtie SC \bowtie C))$

例：检索选修课程号为CS-02或CS-04的学生学号

$\Pi_{S\#}(\sigma_{C\#=\text{'CS-02'} \vee C\#=\text{'CS-04'}}(SC))$

例:检索至少选修课程号为CS-02和CS-04学生学号

$\Pi_{S\#}(\sigma_{1=4 \wedge 2=\text{'CS-02'} \wedge 5=\text{'CS-04'}}(SC \times SC))$

这里(SC×SC)表示关系SC自身相乘的笛卡儿积操作。

Student(S#, Class, Sname, Sex, Age)

Course (C#, Cname, Teacher)

SC (S#, C#, Grade)

关系代数查询实例

- ✓ 例2.3.李同学未选课程的编号：

$$\pi_{C\#}(Course) - \pi_{C\#}(\sigma_{Sname='李'}(Student \bowtie SC))$$

$$\pi_{C\#}(Course) - \pi_{C\#}(\sigma_{Sname='李'}(Student) \bowtie SC)$$

- ✓ 例2.4.至少选修了两门课程的学生学号：

$$\pi_{1}(\sigma_{1=4 \wedge 2 \neq 5}(SC \times SC))$$

- ✓ 例2.5.全部学生均选修了的课程编号及名称：

$$\pi_{C\#,\ Cname}(Course \bowtie (\pi_{S\#,\ C\#}(SC) \div \pi_{S\#}(Student)))$$

Student(S#, Class, Sname, Sex, Age)

Course (C#, Cname, Teacher)

SC (S#, C#, Grade)

第2章 关系模型

2.1 基本概念

2.2 关系代数

2.3 关系演算

2.3 关系演算

- 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

  2. 域关系演算：

     以域变量作为谓词变元的基本对象

元组关系演算

- 元组关系演算表达式的一般形式是{t| P(t)}，其中t是元组变量，P(t)是元组关系演算公式，一般由原子公式构成

元组关系演算

- 原子公式有以下的三种形式：

1)R(t)，t是元组变量，R是关系，R(t)表示t∈R，即t是R的一个元组；

2) s[i]θ t[j]，s和t都是元组变量，s[i]表示元组s的第i个分量，t[j]表示t的第j个分量，θ 是算术比较运算符。s[i]θ t[j]表示s的第i个分量和t的第j个分量满足θ 比较关系。

3) s[i]θ c或cθ s[i]，c是常量，s[i]θ c表示元组s的第i个分量和常量c满足θ 这样的比较关系。

元组关系演算

- 在原子公式的基础上可以通过以下递归定义构成元组关系演算公式：

1) 原子公式也是公式；

2) 如果P1，P2是公式，则P1∧P2，P1∨P2，¬P1也是公式，分别表示：

如果P1，P2同时为真，则P1∧P2为真，否则P1∧P2为假；

如果P1，P2中至少有一个为真，则P1∨P2为真，否则P1∨P2为假；

如果P1为真，则¬P1为假，如果P1为假，则¬P1为真；

元组关系演算

3) 如果P是公式，则 $\exists t(P)$ 也是公式。$\exists$是存在量词符号，$\exists t(P)$ 表示当至少存在一个t使得P为真，则 $\exists t(P)$ 为真，否则为假；

4) 如果P是公式，则 $\forall t(P)$ 也是公式。$\forall$是全称量词符号，$\forall t(P)$ 表示如果所有的t都使P为真，则 $\forall t(P)$ 为真，否则为假；

元组关系演算

5) 在元组关系演算公式中，各类算符优先级为：

①算术比较运算符优先级最高；

②量词 ∃和∀ 优先级其次，且 ∃的优先级高于∀ ；

③逻辑运算符优先级最低，且¬ 的优先级高于∧，∧的优先级高于∨；

④括号可以改变算符的优先级，加括号时，括号内算符优先，同一括号内算符优先级遵循上述各项。

6) 有限次使用上述规则得到的即是元组关系演算公式，除此之外的构成形式不是元组关系演算公式

元组关系演算

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

R

D	E
3	4
7	6

S

A	B	C
4	5	6
7	8	9

$R1 = \{t \mid \exists(u)(R(t) \land S(u) \land t[3] > u[1])\}$

A	B	E
1	2	4
1	2	6
4	5	6

$R2 = \{t \mid (\exists u)(\exists v)(R(u) \land S(v) \land u[2] < v[1] \land t[1] = u[1] \land t[2] = u[2] \land t[3] = v[2])\}$

元组关系演算

- 基本关系代数运算的元组关系演算等价表示：

1) $\cup$

$R \cup S = \{t \mid R(t) \vee S(t)\}$

2) $-$

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

3) $\times$

$R \times S = \{t(m+n) \mid (\exists u(m))(\exists v(n))(R(u) \wedge S(v) \wedge t[1] = u[1] \wedge \cdots \wedge t[m] = u[m] \wedge t[m+1] = v[1] \wedge \cdots \wedge t[m+n] = v[n])\}$，其中$t(m+n)$表示t是m+n元关系的元组。

元组关系演算

- 基本关系代数运算的元组关系演算等价表示：

4) σ

$\sigma_F(R)=\{t \mid R(t) \wedge F'\}$，其中F’是选择条件F在元组关系演算中的等价表示。

5) π

$\pi_{i1,i2,\cdots,ik}(R)=\{t^{(k)} \mid (\exists u)(R(u) \wedge t[1]=u[i1] \wedge \cdots \wedge t[k]=u[ik])\}$

元组关系演算

- 不产生无限关系和无穷验证的表达式称为安全表达式，为了保证运算的安全性而采取的限制措施称为安全约束
  - 例如表达式$\{t \mid \neg R(t)\}$
- 如果$\{t \mid P(t)\}$满足下面的三个条件，则$\{t \mid P(t)\}$是安全的：

1) 如果P(t)为真，则t的每个分量在Dom(P)中；
2) 对于P中每个形如$(\exists u)(Q(u))$的子表达式，如果u使Q(u)为真，则u的每个分量在Dom(P)中；
3) 对于P中每个形如$(\forall u)(Q(u))$的子表达式，如果u使Q(u)为假，则u的每个分量在Dom(P)中。

元组关系演算

例2.6. 查询“离散数学”课程的编号及任课教师

$R=\{t \mid (\exists u)(Course(u) \wedge u[2]='\text{离散数学}' \wedge t[1]=u[1] \wedge t[2]=u[3])\}$

例2.7. 选修了’数据结构’的学生学号及姓名

$R=\{t \mid (\exists u)(\exists v)(\exists w)(Student(u) \wedge SC(v) \wedge Course(w) \wedge u[1]=v[1] \wedge v[2]=w[1] \wedge w[2]='\text{数据结构}' \wedge t[1]=u[1] \wedge t[2]=u[3])\}$

Student(S#, Class, Sname, Sex, Age)

Course (C#, Cname, Teacher)

SC (S#, C#, Grade)

元组关系演算

- 例：检索选修课程号为C2或C4的学生学号

  $\Pi_{S\#}(\sigma_{C\#='C2' \vee C\#='C4'}(SC))$

  $\{t \mid (\exists u)(SC(u) \wedge (u[2]='C2' \vee u[2] = 'C4') \wedge t[1]=u[1])\}$

- 例：检索至少选修课程号为C2和C4的学生学号

  $\Pi_{S\#}(\sigma_{1=4 \wedge 2='C2' \wedge 5='C4'}(SC \times SC))$

这里(SC×SC)表示关系SC自身相乘的笛卡儿积操作。

$\{t \mid (\exists u)(\exists v)(SC(u) \wedge SC(v) \wedge u[2] = 'C2' \wedge v[2] = 'C4' \wedge u[1] = v[1] \wedge t[1] = u[1])\}$

Student(S#, Class, Sname, Sex, Age)

Course (C#, Cname, Teacher)

SC (S#, C#, Grade)

域关系演算

- 域关系演算表达式的一般形式是{t1…tk ｜P(t1，…，tk)}，其中t1，…，tk都是域变量，P(t1，…，tk)是域关系演算公式，一般由原子公式构成

域关系演算

- 类似元组演算，域演算原子公式定义如下：

1)R(x1，…，xk)，其中x1，…，xk都是域变量或常量，R是k元关系，R(x1，…，xk)表示由x1，…，xk构成的元组满足关系R；

2)xθy，其中x和y都是域变量，θ是算术比较运算符。xθy表示x和y满足θ比较关系；

3) xθc或cθx，其中x是域变量，c是常量，xθc表示变量x与常量c满足θ比较关系。xθc和cθx是等价的。

域关系演算

- 在原子公式的基础上域演算公式递归定义如下：

1) 原子公式也是公式；

2) 如果P1，P2是域演算公式，则$P1 \wedge P2$，$P1 \vee P2$，$\neg P1$也是域演算公式；

3) 如果P是域演算公式，则 $\exists x(P)$ 也是域演算公式；

4) 如果P是域演算公式，则 $\forall x(P)$ 也是域演算公式；

5) 域演算公式中各类算符优先级与元组演算相同；

6) 有限次使用上述规则得到的即是域演算公式，除此之外的构成形式不是域演算公式。

域关系演算

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

R

D	E
3	4
7	6

S

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

R1＝{xyz| (R(xyz)∧y＜10)}

B	C	D
5	6	3
8	9	3
8	9	7

R2＝{xyz| (∃u)(∃v)(R(uxy)∧S(zv)∧u≥v)}

域关系演算

- 例：检索学习课程号为C2的学生学号与成绩

  $\{t_1t_2 \mid (\exists u_1u_2u_3)(SC(u_1u_2u_3) \land u_2=\text{'C2'} \land t_1 = u_1 \land t_2 = u_3)\}$

  可简化为$\{t_1t_2 \mid SC(t_1\ \text{'C2'}\ t_2)\}$

- 例：检索学习课程号为C2的学生学号与姓名

  $\{t_1t_2 \mid (\exists u_1u_2u_3u_4)(\exists v_1v_2v_3)(S(u_1u_2u_3u_4) \land SC(v_1v_2v_3) \land v_2=\text{'C2'} \land u_1= v_1 \land t_1= u_1 \land t_2 = u_2)\}$

  可简化为$\{t_1t_2 \mid (\exists u_3u_4)(\exists v_3)(S(t_1t_2u_3u_4) \land SC(t_1\ \text{'C2'}\ v_3))\}$

Student(S#, Class, Sname, Sex, Age)

Course (C#, Cname, Teacher)

SC (S#, C#, Grade)

第2章 小结

- 关系数据结构
  - 关系
    - 域
    - 笛卡尔积
    - 关系（关系，属性，元组，超键，候选键，主键，外键，关系的特征）
  - 关系模式
  - 关系数据库

第2章 小结（续）

- 关系的完整性约束
  - 域完整性约束
  - 实体完整性约束
    - 主键
  - 参照完整性约束
    - 外键
  - 用户定义的完整性约束

第2章 小结（续）

- 关系代数
  - 关系代数运算
    - 并、差、交、笛卡尔积、选择、投影、连接、除
  - 基本运算
    - 并、差、笛卡尔积、选择、投影
- 关系演算
  - 元组关系演算
    - 以元组变量作为谓词变元的基本对象
    - 一般形式{t| P(t)}
  - 域关系演算
    - 以域变量作为谓词变元的基本对象
    - 一般形式{t1…tk | P(t1，…，tk)}