Database System-Relational Algebra

Week12-Course

Database System-Relational Algebra-part1

关系查询语言

(Relational Query Languages)

- Formal: Relational Algebra, Relational Calculus, Datalog
- Practical: SQL, QUEL, QBE(Query-by-Example)
- What is a relational query?
  - Input: a number of relations in your database
  - Output: one relation as the answer

关系代数（Relational Algebra）

- Union, intersection, and difference
- Selection: picking certain rows.
- Projection: picking certain columns.
- Products and joins: compositions of relations.
- Renaming of relations and attributes.

Basic Set Operations

- Union

$$R := R_1 \cup R_2$$

- Intersection

$$R := R_1 \cap R_2$$

- Difference

$$R := R_1 - R_2$$

- Operands must have the same schema

Product

$$R_3 := R_1 \times R_2$$

- Pair each tuple t1 of R1 with each tuple t2 of R2.
- Concatenation t1t2 is a tuple of R3.
- Schema of R3 is the attributes of R1 and then R2, in order.
- But beware attribute *A* of the same name in R1 and R2: use R1.*A* and R2.*A*.

Example

R(A B C)

2	5	7
2	6	9
3	6	7

S(A B C)

2	6	9
2	8	9
3	6	7

R×S(R.A R.B R.C S.A S.B S.C)

2	5	7	2	6	9
2	5	7	2	8	9
2	5	7	3	6	7
2	6	9	2	6	9
2	6	9	2	8	9
2	6	9	3	6	7
3	6	7	2	6	9
3	6	7	2	8	9
3	6	7	3	6	7

R∪S(A B C)

2	5	7
2	6	9
3	6	7
2	8	9

R−S(A B C)

2	5	7

R∩S(A B C)

2	6	9
3	6	7

Selection

$$R_1 := \sigma_C(R_2)$$

- C is a condition (as in "if" statements) that refers to attributes of R2.
- R1 is all those tuples of R2 that satisfy C.

The Example of Selection

Sells

bar	beer	price
Joe’s	Bud	2.50
Joe’s	Miller	2.75
Sue’s	Bud	2.50
Sue’s	Miller	3.00

$$JoeMenu := \sigma_{bar="Joe's"}(Sells)$$

bar	beer	price
Joe’s	Bud	2.50
Joe’s	Miller	2.75

Select Operation – Example

- Relation *r*

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

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

A	B	C	D
α	α	1	7
β	β	23	10

Composition of Operations

- Can build expressions using multiple operations
- Example: $\sigma_{A=C}(r \times s)$

$r \times s$

r

A	B
α	1
β	2

s

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

$\sigma_{A=C}(r \times s)$

A	B	C	D	E
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b

Projection

$$R_1 := \pi_L(R_2)$$

- L is a list of attributes from the schema of R2.
- R1 is constructed by looking at each tuple of R2, extracting the attributes on list L, in the order specified, and creating from those components a tuple for R1.
- Eliminate duplicate tuples, if any.

The Example of Projection

Sells

bar	beer	price
Joe’s	Bud	2.50
Joe’s	Miller	2.75
Sue’s	Bud	2.50
Sue’s	Miller	3.00

$$Prices := \pi_{beer,price}(Sells)$$

beer	price
Bud	2.50
Miller	2.75
Miller	3.00

Theta-Join

$$R_3 := R_1 \bowtie_C R_2$$

- Take the product $R_1 \times R_2$
- Then apply σ_C to the result.
- C can be any boolean-valued condition.
- Historic versions of this operator allowed only A θ B, where θ is =, <, etc.; hence the name "theta-join."

Sells(bar, beer, price) Bars(name, addr)

bar,	beer,	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

name,	addr
Joe's	Maple St.
Sue's	River Rd.

BarInfo := Sells $\bowtie_{Sells.a. = Bars.name}$ Bars

bar,	beer,	price,	name,	addr
Joe's	Bud	2.50	Joe's	Maple St.
~~Joe's~~	~~Bud~~	~~2.50~~	~~Sue's~~	~~River Rd~~
Joe's	Miller	2.75	Joe's	Maple St.
~~Joe's~~	~~Miller~~	~~2.75~~	~~Sue's~~	~~River Rd.~~
~~Sue's~~	~~Bud~~	~~2.50~~	~~Joe's~~	~~Maple St.~~
Sue's	Bud	2.50	Sue's	River Rd.
~~Sue's~~	~~Coors~~	~~3.00~~	~~Joe's~~	~~Maple St.~~
Sue's	Coors	3.00	Sue's	River Rd.

Example

Sells(bar, beer, price)

bar	beer	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

Bars(name, addr)

name	addr
Joe's	Maple St.
Sue's	River Rd.

BarInfo := Sells $\bowtie$ Sells.bar = Bars.name Bars

BarInfo(bar, beer, price, name, addr)

bar	beer	price	name	addr
Joe's	Bud	2.50	Joe's	Maple St.
Joe's	Miller	2.75	Joe's	Maple St.
Sue's	Bud	2.50	Sue's	River Rd.
Sue's	Coors	3.00	Sue's	River Rd.

Natural Join

- A frequent type of join connects two relations by:
  - Equating attributes of the same name, and
  - Projecting out one copy of each pair of equated attributes.
- Called *natural* join.
- Denoted R3 := R1 ⋈ R2.

Example

Sells(bar, beer, price)

bar	beer	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

Bars(bar, addr)

bar	addr
Joe's	Maple St.
Sue's	River Rd.

BarInfo := Sells ⋈ Bars

Note Bars.name has become Bars.bar to make the natural join "work."

BarInfo(bar, beer, price, addr)

bar	beer	price	addr
Joe's	Bud	2.50	Maple St.
Joe's	Millller	2.75	Maple St.
Sue's	Bud	2.50	River Rd.
Sue's	Coors	3.00	River Rd.

Summary

- Basic Relational Algebra
  - ∪ Union
  - ∩ Intersection
  - – Difference
  - X Product
  - σ_C Selection
  - π Projection
  - ⋈$_\theta$ Theta-join
  - ⋈ Natural-join

填空题 3分

互动交流一

对于关系R和S，在经过以下关系代数运算后，结果关系中的元组数目为多少？请填写阿拉伯数字

R(A	B	C)
2	5	7
2	6	9
3	6	7

S(A	B	C)
2	7	9
2	8	9
3	6	7

R∪S包含 [填空1] 个元组

R–S包含 [填空2] 个元组

R∩S 包含 [填空3] 个元组

提交

多选题 1分

互动交流二

State if the following statements are TRUE or FALSE.

(a) Consider relations *R(A,B)* and *S(B,C)* where *T(R) = 5000*, *T(S) = 3000*, and *B* is a primary key on *S*. The expected number of tuples in R ⋈ S is less than or equal to *3000*.

A True B False

(b) Consider two relations R(A, B, C) and S(A, D, E), sharing a common attribute A. It is known that R.A is a foreign key referencing S.A, and that S.A is the primary key of relation S. Then the estimated size of R ⋈ S is T(R).

C True D False

提交

多选题 1分

互动交流三-不定项选择

对于关系修课（学号，学生姓名，课号，成绩），想要查询修读任意课程有成绩大于90分的学生的学号，需要用到的关系代数运算符为

A 选择运算符σ

B 选择运算符 π

C 投影运算符σ

D 投影运算符 π

提交

填空题 6分

预习效果考察一

对于关系R和S，在经过以下关系代数运算后，结果关系中的属性和元组数目为多少？请填写阿拉伯数字

R(	A	B	C)
	2	5	7
	2	6	9
	3	6	7

S(	B	C	D)
	5	7	9
	5	8	9
	6	7	7

R×S 包含 [填空1] 个属性，包含 [填空2] 个元组

$R \bowtie S$ 包含 [填空3] 个属性，包含 [填空4] 个元组

$R \bowtie_{R.C>7} S$ 包含 [填空5] 个属性， [填空6] 个元组

提交

单选题 1分

预习效果考察二

已知关系模式R(A,B)和S(B,C)，有以下两个关系代数运算：

Q1：$R \bowtie S$　　Q2：$R \underset{R.B=S.B}{\bowtie} S$

请问下列哪个描述是正确的

A　Q1和Q2产生的结果一样

B　Q1的结果总是包含Q2的结果

C　Q2的结果总是包含Q1的结果

D　Q1和Q2产生不同的结果

提交

Database System-Relational Algebra-part2

Renaming

- The ρ operator gives a new schema to a relation.
- R1 := $\rho_{R1(A1,\cdots,An)}$(R2) makes R1 be a relation with attributes A1,⋯,A*n* and the same tuples as R2.
- Simplified notation: R1(A1,⋯,A*n*) := R2.

Example

Bars(name, addr)

Joe’s	Maple St.
Sue’s	River Rd.

R(bar, addr) := Bars

R(bar, addr)

Joe’s	Maple St.
Sue’s	River Rd.

Sequences of Assignments

- Create temporary relation names.
- Renaming can be implied by giving relations a list of attributes.
- Example: $R_3 := R_1 \bowtie_C R_2$ can be written:

$$R_4 := R_1 \times R_2$$
$$R_3 := \sigma_C(R_4)$$

Expressions in a Single Assignment

- Example:

$$R_3 := R_1 \bowtie_C R_2$$

can be rewritten:

$$R_3 := \sigma_C (R_1 \times R_2)$$

Operator Precedence

- Precedence of relational operators:
  1. Unary operators --- select, project, rename --- have highest precedence, bind first.
  2. Then come products and joins.
  3. Then intersection.
  4. Finally, union and set difference bind last.
- But you can always insert parentheses to force the order you desire.

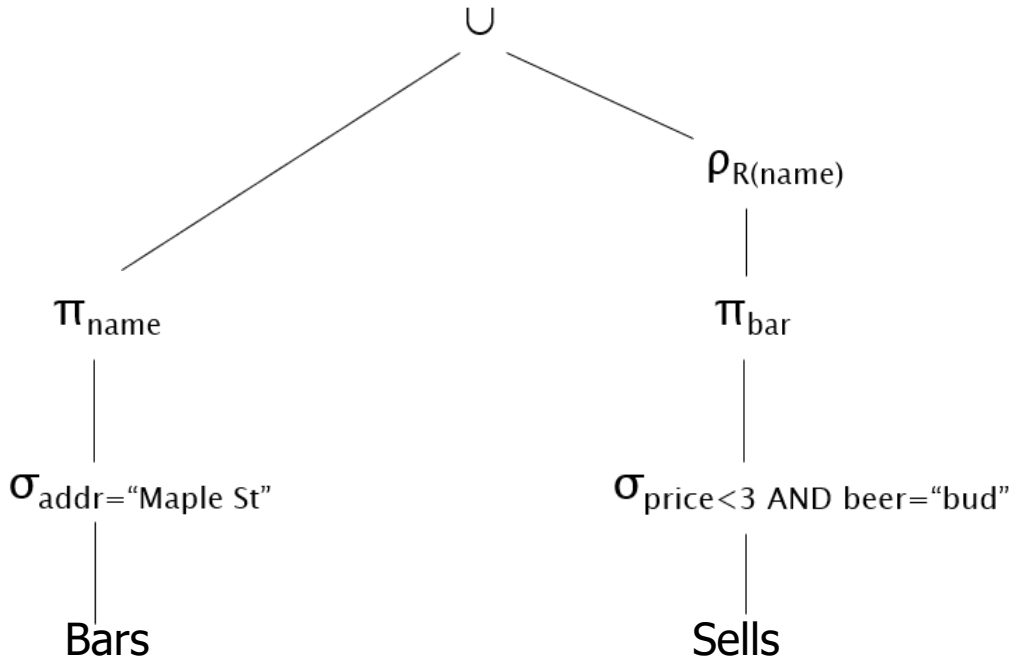
Expression Trees

- Leaves are operands --- either variables standing for relations or particular, constant relations.
- Interior nodes are operators, applied to their child or children.

Example

- Using the relations Bars(name, addr) and Sells(bar, beer, price), find the names of all the bars that are either on Maple St. or sell Bud for less than $3.

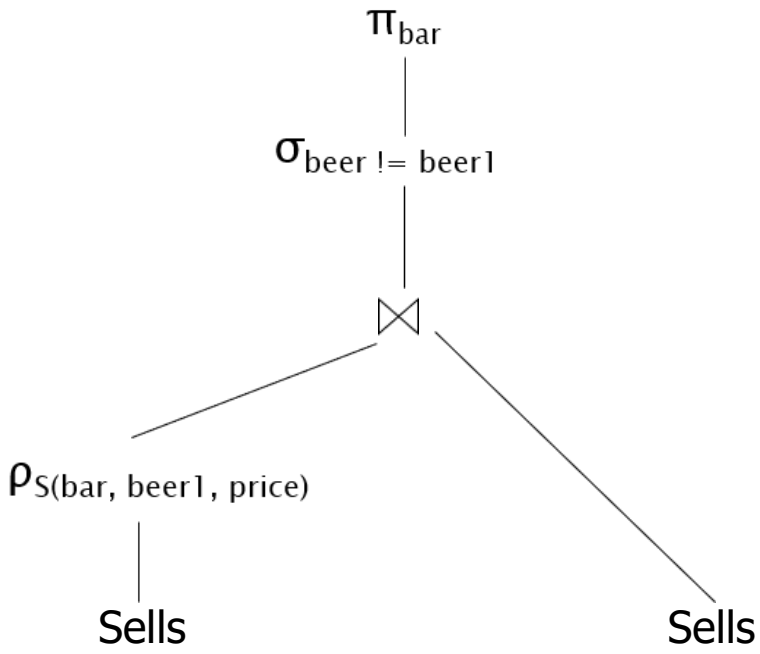
As a Tree:


∪
ρR(name)
πname
πbar
σaddr="Maple St"
σprice<3 AND beer="bud"
Bars
Sells

Example

- Using Sells(bar, beer, price), find the bars that sell two different beers at the same price.
- Strategy: by renaming, define a copy of Sells, called S(bar, beer1, price). The natural join of Sells and S consists of quadruples (bar, beer, beer1, price) such that the bar sells both beers at this price.

The Tree



Schemas for Interior Nodes

- An expression tree defines a schema for the relation associated with each interior node.
- Similarly, a sequence of assignments defines a schema for each relation on the left of the := sign.

Schema-Defining Rules 1

- For union, intersection, and difference, the schemas of the two operands must be the same, so use that schema for the result.
- Selection: schema of the result is the same as the schema of the operand.
- Projection: list of attributes tells us the schema.

Schema-Defining Rules 2

- Product: the schema is the attributes of both relations.
  - Use R.*A*, etc., to distinguish two attributes named *A*.
- Theta-join: same as product.
- Natural join: use attributes of both relations.
  - Shared attribute names are merged.
- Renaming: the operator tells the schema.

Summary

- Basic Relational Algebra
  - ρ Renaming
- Sequences of Assignments
- Operator Precedence
- Expression Trees
- Results Schema

填空题 2分

互动交流一

假设有如下关系，并有关系代数

R(A B C)
2 5 6
2 6 9
3 6 7

S(C D)
6 7
7 8
7 9
8 7
8 9
9 9

$R1:= \rho_{R1(bigC\_A,bigC\_D)}(\pi_{A,D}(\sigma_{R.C>7}R \bowtie S))$

则R1中的属性数目为 [填空1] ,元组数目为 [填空2]
(请填写阿拉伯数字)

请弹幕写出R1的模式

提交

填空题 2分

互动交流二

在以下关系中，执行如右图所示的表达式树后，其结果关系中的属性数目为 [填空1]，元组数目为 [填空2]（请填写阿拉伯数字）

Sells(

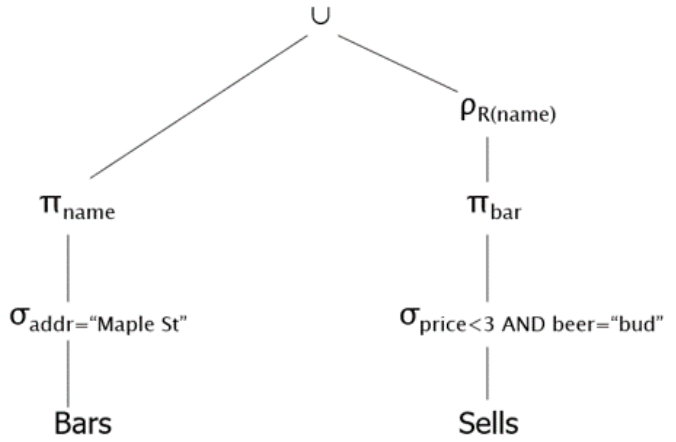
bar,	beer,	price
Joe's	Bud	2.50
Joe's	Miller	2.75
Sue's	Bud	2.50
Sue's	Coors	3.00

)

Bars(

name,	addr
Joe's	Maple St.
Sue's	River Rd.

)



提交

多选题 1分

互动交流三

假设有关系模式R(A,B,C), S(A,B,C)，则在R和S上进行如下关系代数运算后，其结果关系模式有可能不再是(A,B,C)的有

A $R \cup S$

B $R \cap S$

C $R - S$

D $R \times S$

E $\sigma_C(R)$

F $\pi_L(R)$

G $R \bowtie_C S$

H $R \bowtie S$

I $\rho_{R1(A1,\cdots,An)}(R)$

提交

Database System-Relational Algebra-part3

The Extended Algebra

1. DELTA (δ) = eliminate duplicates from bags.
2. TAU (τ) = sort tuples.
3. *Extended projection* : arithmetic, duplication of columns.
4. GAMMA (γ) = grouping and aggregation.
5. *Outerjoin* : avoids "dangling tuples" = tuples that do not join with anything.

Duplicate Elimination

- R1 := δ(R2).
- R1 consists of one copy of each tuple that appears in R2 one or more times.

Example: Duplicate Elimination

R =

A	B
1	2
3	4
1	2

δ(R) =

A	B
1	2
3	4

Sorting

- R1 := τ_L (R2).
  - *L* is a list of some of the attributes of R2.
- R1 is the list of tuples of R2 sorted first on the value of the first attribute on *L*, then on the second attribute of *L*, and so on.
  - Break ties arbitrarily.
- TAU is the only operator whose result is neither a set nor a bag.

Example: Sorting

R =

A	B
1	2
3	4
5	2

$T_B(R)$ = [(5,2), (1,2), (3,4)]

Extended Projection

- Using the same π_L operator, we allow the list L to contain arbitrary expressions involving attributes, for example:
  1. Arithmetic on attributes, e.g., $A+B$.
  2. Duplicate occurrences of the same attribute.

Example: Extended Projection

R =

A	B
1	2
3	4

$\pi_{A+B,A,A}$ (R) =

A+B	A1	A2
3	1	1
7	3	3

Aggregation Operators

- Aggregation operators are not operators of relational algebra.
- Rather, they apply to entire columns of a table and produce a single result.
- The most important examples: SUM, AVG, COUNT, MIN, and MAX.

Example: Aggregation

R =

A	B
1	3
3	4
3	2

SUM(A) = 7
COUNT(A) = 3
MAX(B) = 4
AVG(B) = 3

Grouping Operator

- R1 := γ_L (R2). L is a list of elements that are either:

1. Individual (*grouping*) attributes.
2. AGG(A), where AGG is one of the aggregation operators and A is an attribute.

Applying GAMMA$_L$(R)

- Group *R* according to all the grouping attributes on list *L*.
  - That is, form one group for each distinct list of values for those attributes in *R*.
- Within each group, compute AGG(*A*) for each aggregation on list *L*.
- Result has one tuple for each group:

1. The grouping attributes and
2. Their group’s aggregations.

Example: Grouping/Aggregation

R =

A	B	C	
1	2	3	
4	5	6	
1	2	5	

$\gamma_{A,B,AVG(C)}$ (R) = ??

First, group R :

A	B	C
1	2	3
1	2	5
4	5	6

Then, average C within groups:

A	B	AVG(C)
1	2	4
4	5	6

Outerjoin

- Suppose we join $R \bowtie S$.
- A tuple of R that has no tuple of S with which it joins is said to be *dangling*.
  - Similarly for a tuple of S.
- Outerjoin ($\overset{\circ}{\bowtie}$) preserves dangling tuples by padding them with a special NULL symbol in the result.

Example: Outerjoin

R =

A	B
1	2
4	5

S =

B	C
2	3
6	7

(1,2) joins with (2,3), but the other two tuples are dangling.

R ⟕ S =

A	B	C
1	2	3
4	5	NULL
NULL	6	7

Relational algebra operators

SELECT	σ	INTERSECT	$\cap$
PROJ	π	MINUS	-
*	$\times$	TAU	τ
JOIN	$\bowtie$	DELTA	δ
RENAME	ρ	GAMMA	γ
UNION	$\cup$	OUTERJOIN	$\overset{\circ}{\bowtie}$

Summary-Extended Algebra

- DELTA (δ)
- TAU (τ)
- Extended projection (π)
- GAMMA (γ)
- Outerjoin ($\overset{\circ}{\bowtie}$)

单选题 1分

互动交流一

运算符δ和SQL中哪个关键字的功能类似？(请弹幕写出SQL中的其他关键字的功能在关系代数中用什么运算符实现？)

A SELECT

B FROM

C WHERE

D DISTINCT

提交

多选题 1分

互动交流二-不定项选择

对于关系修课（学号，姓名，课号，成绩），若想将学生修课信息按照成绩升序排列，则会用到的关系代数运算符为

A δ

B τ

C π

D γ

提交

多选题 1分

互动交流三-不定项选择

若想从关系 考试成绩（考号，姓名，语文成绩，数学成绩，英语成绩）中获得关系 总成绩（考号，姓名，总成绩），使用关系代数会用到的运算符为

A δ

B τ

C π

D γ

E 无法用关系代数完成此运算

提交

主观题 10分

互动交流四

若想从以下关系中获得每位员工参与的项目数信息，使结果关系模式为 NumberOfProjects(empno, projects_count)，应使用的关系代数表达式为

Workson

empno	projectno	job	enterdate
2581	p3	分析员	98-10-15
9031	p1	管理员	98-4-15
9031	p3	职员	97-11-15
10102	p1	分析员	97-1-10
10102	p3	管理员	99-1-1
18316	p2	职员	98-2-15
25348	p2	<NULL>	98-6-1
28559	p1	<NULL>	98-8-1
28559	p2	职员	99-2-1
29346	p1	职员	98-1-4
29346	p2	<NULL>	97-12-15

提交

填空题 2分

互动交流五

对于以下关系R和S，R ⋈S结果关系中的属性数目为 [填空1]，元组数目为 [填空2]（请使用阿拉伯数字填写）

R(A	B	C)
2	5	7
2	6	9
3	6	7

S(B	C	D)
5	7	9
5	8	9
6	7	7

提交

Database System-Relational Algebra-part4

Division Operation

$$R \div S$$

- Suited to queries that include the phrase "for all".
- Let r and s be relations on schemas R and S respectively where
  - $R = (A_1, \ldots, A_m, B_1, \ldots, B_n)$
  - $S = (B_1, \ldots, B_n)$

  The result of r ÷ s is a relation on schema $R - S = (A_1, \ldots, A_m)$

Division Operation

R(A,B,C)

r

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

S(B,C,D)

s

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

$$r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{(R-S}(r) \times s) - \Pi_{R-S,S}(r))$$

除运算

S(S#,C#)
1 C1
1 C2
1 C3
1 C4
2 C1
2 C2
3 C2
4 C2
4 C3
4 C4
5 C4

CA(C#)
C2

⇓

S÷CA(S#)
1
2
3
4

CB(C#)
C1
C2

⇓

S÷CB(S#)
1
2

CC(C#)
C1
C2
C3
C4

⇓

S÷CC(S#)
1

综合练习

S(S#, SN, SD, SA)

S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

C(C#, CN, PC#)

C1	G	—
C2	H	C1
C3	I	C1
C4	J	C2
C5	K	C4

SC(S#, C#, G)

S1	C1	A
S1	C2	A
S1	C3	A
S1	C5	B
S2	C1	B
S2	C2	C
S2	C4	C
S3	C2	B
S3	C3	C
S3	C4	B
S4	C3	D
S4	C5	A
S5	C2	C
S5	C3	B

1. 查询数学系(MA)全体学生

综合练习

S(S#, SN, SD, SA)

S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

C(C#, CN, PC#)

C1	G	—
C2	H	C1
C3	I	C1
C4	J	C2
C5	K	C4

SC(S#, C#, G)

S1	C1	A
S1	C2	A
S1	C3	A
S1	C5	B
S2	C1	B
S2	C2	C
S2	C4	C
S3	C2	B
S3	C3	C
S3	C4	B
S4	C3	D
S4	C5	A
S5	C2	C
S5	C3	B

2. 查询学生的姓名和所在的系

综合练习

S(S#, SN, SD, SA)

S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

C(C#, CN, PC#)

C1	G	—
C2	H	C1
C3	I	C1
C4	J	C2
C5	K	C4

SC(S#, C#, G)

S1	C1	A
S1	C2	A
S1	C3	A
S1	C5	B
S2	C1	B
S2	C2	C
S2	C4	C
S3	C2	B
S3	C3	C
S3	C4	B
S4	C3	D
S4	C5	A
S5	C2	C
S5	C3	B

3. 查询年龄小于20岁的学生的学号和姓名

综合练习

S(S#, SN, SD, SA)

S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

C(C#, CN, PC#)

C1	G	—
C2	H	C1
C3	I	C1
C4	J	C2
C5	K	C4

SC(S#, C#, G)

S1	C1	A
S1	C2	A
S1	C3	A
S1	C5	B
S2	C1	B
S2	C2	C
S2	C4	C
S3	C2	B
S3	C3	C
S3	C4	B
S4	C3	D
S4	C5	A
S5	C2	C
S5	C3	B

4. 查询选修了‘C1’课的学生学号和姓名

综合练习

S(S#, SN, SD, SA)

S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

C(C#, CN, PC#)

C1	G	—
C2	H	C1
C3	I	C1
C4	J	C2
C5	K	C4

SC(S#, C#, G)

S1	C1	A
S1	C2	A
S1	C3	A
S1	C5	B
S2	C1	B
S2	C2	C
S2	C4	C
S3	C2	B
S3	C3	C
S3	C4	B
S4	C3	D
S4	C5	A
S5	C2	C
S5	C3	B

5. 查询至少选修了一门其直接先行课为`C1’ 课程的学生姓名

综合练习

S(S#, SN, SD, SA)

S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

C(C#, CN, PC#)

C1	G	—
C2	H	C1
C3	I	C1
C4	J	C2
C5	K	C4

SC(S#, C#, G)

S1	C1	A
S1	C2	A
S1	C3	A
S1	C5	B
S2	C1	B
S2	C2	C
S2	C4	C
S3	C2	B
S3	C3	C
S3	C4	B
S4	C3	D
S4	C5	A
S5	C2	C
S5	C3	B

6. 查询选修了全部课程的学生号码和姓名

综合练习

S(S#, SN, SD, SA)

S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

C(C#, CN, PC#)

C1	G	—
C2	H	C1
C3	I	C1
C4	J	C2
C5	K	C4

SC(S#, C#, G)

S1	C1	A
S1	C2	A
S1	C3	A
S1	C5	B
S2	C1	B
S2	C2	C
S2	C4	C
S3	C2	B
S3	C3	C
S3	C4	B
S4	C3	D
S4	C5	A
S5	C2	C
S5	C3	B

7. 将新开课<‘C6’,’ Pascal’,’ -’ >插入到关系C中

综合练习

S(S#, SN, SD, SA)

S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

C(C#, CN, PC#)

C1	G	—
C2	H	C1
C3	I	C1
C4	J	C2
C5	K	C4

SC(S#, C#, G)

S1	C1	A
S1	C2	A
S1	C3	A
S1	C5	B
S2	C1	B
S2	C2	C
S2	C4	C
S3	C2	B
S3	C3	C
S3	C4	B
S4	C3	D
S4	C5	A
S5	C2	C
S5	C3	B

8. 将S1学生的‘C1’课成绩改为‘B’

关系代数基本运算

- 人们证明了∪(并)、－(差)、×(笛卡儿积)、σ(选择)、π(投影)五种运算构成了关系运算的最小完备集，经过它们有限次复合而构成的关系代数表达式，可以实现人们所需要的对数据库的所有查询和更新操作。
- E.F.Codd把关系代数的这种处理能力称为关系完备性(Relational Completeness)。这一概念后来进一步发展为关系DBMS的一个重要评价标准

Summary

- Division
- 关系运算的最小完备集
  – ∪(并)、– (差)、× (笛卡儿积)、 σ(选择)、π(投影)
- 所有SQL操作都可以转化为关系代数运算

主观题 10分

互动交流一

对以下关系，查询修了所有课的学生姓名

S(S#, SN, SD, SA)

S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

C(C#, CN, PC#)

C1	G	—
C2	H	C1
C3	I	C1
C4	J	C2
C5	K	C4

SC(S#, C#, G)

S1	C1	A
S1	C2	A
S1	C3	A
S1	C5	B
S2	C1	B
S2	C2	C
S2	C4	C
S3	C2	B
S3	C3	C
S3	C4	B
S4	C3	D
S4	C5	A
S5	C2	C
S5	C3	B

提交

多选题 1分

互动交流二-不定项选择

关系运算的最小完备集为

A 并、交、差、选择、投影

B 并、差、笛卡尔积、选择、投影

C ∪、－、×、σ、π

D ∪、∩、－、σ、π

提交

主观题 10分

互动交流三

对以下关系，查询不学‘C2’课的学生姓名和年龄

S(S#, SN, SD, SA)

S1	A	CS	20
S2	B	CS	21
S3	C	MA	19
S4	D	CI	19
S5	E	MA	20
S6	F	CS	22

C(C#, CN, PC#)

C1	G	—
C2	H	C1
C3	I	C1
C4	J	C2
C5	K	C4

SC(S#, C#, G)

S1	C1	A
S1	C2	A
S1	C3	A
S1	C5	B
S2	C1	B
S2	C2	C
S2	C4	C
S3	C2	B
S3	C3	C
S3	C4	B
S4	C3	D
S4	C5	A
S5	C2	C
S5	C3	B

提交