CS275 – Intro to Databases

Relational Algebra - *Chap.6.3-6.5*

SPECIAL RELATIONAL OPERATORS

- Join operators

There are several kind of join operators.

(1) Condition Joins

(2) Equijoins

(3) Natural Joins

- Outer Join
- Division

Conditional join

- Conditional joins
- Defined as a cross-product followed by a selection

$$R \triangleright\!\triangleleft\, cS = \sigma_C(R \times S)$$

(⋈ is called the bow-tie)where c is the condition.

Conditional join

Example:

- Given the sample relational instances S1 and R1

sid	*sname*	*rating*	*age*
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0

Figure 4.1 Instance $S1$ of Sailors

sid	*bid*	*day*
22	101	10/10/96
58	103	11/12/96

Figure 4.3 Instance $R1$ of Reserves

The condition join S1 ⋈S1.sid<R1.sid R1 yields

(*sid*)	*sname*	*rating*	*age*	(*sid*)	*bid*	*day*
22	Dustin	7	45.0	58	103	11/12/96
31	Lubber	8	55.5	58	103	11/12/96

Figure 4.12 $S1 \bowtie_{S1.sid<R1.sid} R1$

Equi Join

- Equijoins

$$R \rhd\lhd cS = \sigma_C(R \times S)$$

- Example
- S1 Sailors(<u>sid:integer</u>, sname:string,rating:integer,age:real)
- B1 Boats(<u>bid:integer</u>, bname: string, color: string)
- R1 Reserves(<u>sid:integer, bid: integer, day: date</u>)

$$S_1 \rhd\lhd_{S1.sid=R1.sid} R_1$$

Equi Join

Equi-join: example

- R　　　　S　　　　R equi-join $_{NR = NS}$ S

R

NR	Tel
Joe	1234
Jill	1244
Bill	2244

S

NS	Addr
Joe	NG7
Jill	NG8
Bob	NG1

R equi-join $_{NR = NS}$ S

NR	Tel	NS	Addr
Joe	1234	Joe	NG7
Jill	1244	Jill	NG8

Natural Join

- Natural joins

$$R \rhd\!\lhd cS = \sigma_C(R \times S)$$

- Example

- S1 Sailors(<u>sid:integer</u>, sname:string,rating:integer,age:real)
- B1 Boats(<u>bid:integer</u>, bname: string, color: string)
- R1 Reserves(<u>sid:integer, bid: integer, day: date</u>)

$$S_1 \rhd\!\lhd R_1$$

$$or$$

$$S_1 * R_1$$

Natural Join

Natural join: example

- R S $R \bowtie_{NR = NS} S$

NR	Tel
Joe	1234
Jill	1244
Bill	2244

NS	Addr
Joe	NG7
Jill	NG8
Bob	NG1

NR	Tel	Addr
Joe	1234	NG7
Jill	1244	NG8

Natural join is definable

- Natural join is a very useful operator, but it can be defined using $\times$, π and σ.
- To define equi-join of R and S over attributes X and Y we take a Cartesian product of R and S and then select tuples which satisfy X = Y.
- To obtain natural join, we project the resulting relation on the set of all attributes apart from Y. Let A be the set of all attributes in R and S:

$$R \bowtie_{X=Y} S = \pi_{A-Y}\, \sigma_{X=Y}(R \times S)$$

Joins

- More examples

sid	*sname*	*rating*	*age*
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Figure 4.15 An Instance $S3$ of Sailors

sid	*bid*	*day*
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Figure 4.16 An Instance $R2$ of Reserves

- Find the names of sailors who have reserved boat 103

Joins

- More examples

sid	sname	rating	age
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Figure 4.15 An Instance $S3$ of Sailors

sid	bid	day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Figure 4.16 An Instance $R2$ of Reserves

Find the names of sailors who have reserved boat 103

$$\pi_{sname}((\sigma_{bid=103}\, Reserves) \bowtie Sailors)$$

Ans; {<Dustin>, <Lubber>, <Horatio>}

Joins

Find the names of sailors who have reserved a red boat.

sid	*sname*	*rating*	*age*
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Figure 4.15 An Instance *S*3 of Sailors

sid	*bid*	*day*
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Figure 4.16 An Instance *R*2 of Reserves

bid	*bname*	*color*
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Figure 4.17 An Instance *B*1 of Boats

$$\pi_{sname}((\sigma_{color='red'}\mathbf{Boats}) \bowtie \mathbf{Reserves} \bowtie \mathbf{Sailors})$$

Joins

Find the names of Sailors who have reserved at least one boat

sid	*sname*	*rating*	*age*
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Figure 4.15 An Instance $S3$ of Sailors

sid	*bid*	*day*
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Figure 4.16 An Instance $R2$ of Reserves

bid	*bname*	*color*
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Figure 4.17 An Instance $B1$ of Boats

$$\pi_{sname}(\textbf{Sailors} \bowtie \textbf{Reserves})$$

Joins

Find the names of sailors who have reserved a red or a green boat.

sid	*sname*	*rating*	*age*
22	Dustin	7	45.0
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Horatio	9	35.0
85	Art	3	25.5
95	Bob	3	63.5

Figure 4.15 An Instance *S*3 of Sailors

sid	*bid*	*day*
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

Figure 4.16 An Instance *R*2 of Reserves

bid	*bname*	*color*
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

Figure 4.17 An Instance *B*1 of Boats

$$\rho(Tempboats, (\sigma_{color='red'} Boats) \cup (\sigma_{color='green'} Boats))$$

$$\pi_{sname}(Tempboats \bowtie Reserves \bowtie Sailors)$$

Joins

Find the names of Sailors who have reserved a red and a green boat.

It seems tempting to use the expression used in Q5, replacing simply $\cup$ by $\cap$. However, this won't work, for such an expression is requesting the names of sailors who have requested a boat that is <u>both red and green</u>! The correct expression is as follows:

$$\rho(\textbf{\textit{Tempred}}, \pi_{sid}((\sigma_{color='red'}\textbf{\textit{Boats}}) \bowtie \textbf{\textit{Reserves}}))$$

$$\rho(\textbf{\textit{Tempgreen}}, \pi_{sid}((\sigma_{color='green'}\textbf{\textit{Boats}}) \bowtie \textbf{\textit{Reserves}}))$$

$$\pi_{sname}((\textbf{\textit{Tempred}} \cap \textbf{\textit{Tempgreen}}) \bowtie \textbf{\textit{Sailors}})$$

Outer Join

- Join selects only tuples satisfying the join condition
- ***Outer Join:***
  - Left outer join (r]∞ s) keeps every tuple in first or left relation
  - Right outer join (r ∞[s) keeps every tuple in second or right relation
  - Full outer join (r]∞[s) keeps every tuple
- Attributes of tuples with no matching tuples are set to NULL

Outer Join

relation **r**

A	B	C
a	b	c
d	a	f
c	b	d

relation **s**

A	B	D
b	g	a
d	a	f

r]∞s

A	B	C	D
a	b	c	Null
d	a	f	f
c	b	d	Null

r ∞[s

A	B	C	D
d	a	f	f
b	g	Null	a

r]∞[s

A	B	C	D
a	b	c	Null
d	a	f	f
c	b	d	Null
b	g	Null	a

Aggregate Functions

- Mathematical and Statistical aggregate functions on collections of values
  - SUM, MAXIMUM, MINIMUM, AVERAGE
  - COUNT number of tuples (cardinality)

$$\mathcal{F}_{\text{<function list>}}(\text{<relation>})$$

  - Function list is a list of pairs
    (< function, attribute>)

- E.g., $\mathcal{F}_{\text{count SSN, AVERAGE QPA}}(\text{STUDENT})$

DIVISION

The division operator is used for queries which involve the 'all' qualifier such as
"Find the names of sailors who have reserved all boats".
The formal definition of division is as follows:

$$A/B = \pi x(A) - \pi x((\pi x(A) \times B) – A)$$

Division: example

- R S R | S

Lecturer	Module
Brown	Compilers
Brown	Databases
Green	Prolog
Green	Databases
Lewis	Prolog
Smith	Databases

Subject
Prolog

Lecturer
Green
Lewis

- R

Lecturer	Module
Brown	Compilers
Brown	Databases
Green	Prolog
Green	Databases
Lewis	Prolog
Smith	Databases

S

Subject
Databases
Prolog

R | S

Lecturer
Green

- R S R | S

Lecturer	Module
Brown	Compilers
Brown	Databases
Green	Prolog
Green	Databases
Lewis	Prolog
Smith	Databases

Subject
Compilers
Prolog

Lecturer

EXAMPLES OF DIVISION

A

sno	*pno*
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

B1

pno
p2

B2

pno
p2
p4

B3

pno
p1
p2
p4

A/B1

sno
s1
s2
s3
s4

A/B2

sno
s1
s4

A/B3

sno
s1

DIVISION

Interpretation of the division operation A/B:

- Divide the attributes of A into 2 sets: A1 and A2.
- Divide the attributes of B into 2 sets: B2 and B3.
- Where the sets A2 and B2 have the same attributes.
- For each set of values in B2:
    - Search in A2 for the sets of rows (having the same A1 values) whose A2 values (taken together) form a set which is the same as the set of B2's.
    - For all the set of rows in A which satisfy the above search, pick out their A1 values and put them in the answer.

DIVISION

Example: Find the names of sailors who have reserved all boats:

(1) A = πsid,bid(Reserves). A1 = πsid(Reserves)   A2 = πbid(Reserves)

(2) B2 = πbid(Boats)   B3 is the rest of B.

Thus, B2 ={101, 102, 103, 104}

(3)Find the rows of A such that their A.sid is the same and their combined A.bid is the set B2.

Thus we find A1 = {22}

(4) Get the set of A2 corresponding to A1: A2 = {Dustin}

Given a relational database schema as follows:

- SUPPLIER (S#, SNAME, STATUS, CITY),
- PART(P#, PNAME, COLOR, WEIGTH, CITY),
- PROJECT(J#, JNAME, CITY),
- SPJ(S#, P#, J#, QTY).

Please write the following query in Relational Algebra :

1.Find all suppliers who locate in Atlanta.

$$\sigma_{\text{CITY} = \text{'Atlanta'}}(\text{SUPPLIER})$$

2.Find all names of those suppliers who locate in Atlanta.

$$\pi_{NAME}(\sigma_{CITY = \text{'Atlanta'}}(SUPPLIER))$$

3. Find all parts that are red in color and weight more than 1 lb.

$$\pi_{P\#,\ PNAME,\ CITY}(\sigma_{COLOR = \text{'Red'} \text{ and } WEIGHT = \text{'1 lb'}}(PART))$$

4.Find all suppliers that have supplied P2.

$$\pi_{S\#,\ SNAME,\ CITY,}(\sigma_{P\# = \text{'P2'}}(SPJ \bowtie SUPPLIER))$$

5.Find supplier names for suppliers who do not supply part P2.

$$\pi_{S\#,\ SNAME,\ CITY}(\sigma_{P\# <> \text{'P2'}}(SPJ) \bowtie PART)$$

6. list the S# of suppliers that have supplied all the parts listed in SPJ?

$$\pi_{S\#}(SPJ) \div \pi_{P\#}(SPJ)$$

7. list the S# of suppliers who have supplied to all the projects with the project name 'CAD/CAM'

$$\pi_{S\#}(SPJ \div \pi_{J\#}(\sigma_{PNAME='CAD/CAM'}(PROJECT)))$$

8. Find part numbers for parts supplied to any project in London.

$$\pi_{J\#}(\sigma_{CITY='LONDON'}(PROJECT) \bowtie SPJ)$$

9. Find part numbers for parts supplied to all projects in London.

$$\pi_{P\#}(SPJ \div \pi_{J\#}(\sigma_{CITY='LONDON'}(PROJECT)))$$

10. Get a list containing (S#, J#) for all suppliers who live in Atlanta but are not supplied to the project CAD/CAM?

$$(\pi_{S\#}(\sigma_{CITY='ATLANTA'}(SUPPLIER)) \times \pi_{J\#}(\sigma_{PNAME='CAD/CAM'}(PROJECT))) - \pi_{S\#,J\#}(SPJ)$$

11. Find names and city of all suppliers that have supplied more than 100 units of P2.

$$\pi_{SNAME,\ CITY}(\sigma_{P\#='P2' \text{ and } QTY>100}(SPJ) \bowtie_{S\#} SUPPLIER)$$