COP 3540: Introduction to Database Structures

Fall 2017

Relational Algebra

M. Rathod

Query Languages

- Query languages are specialized languages for asking data-related questions
- Inputs and outputs of queries are relations
- A query is evaluated using *instances* of each input relation
- We can use either field names or field position (in order)
- Positional notation is used to formally define relational algebra

Relational Algebra

- Formal query language

- Every operator in the algebra accepts either one or two relation instances as arguments and returns a relation instance as the result

- A **relational algebra expression** is recursively defined to be a relation, a unary algebra operator applied to a single expression, or a binary algebra operator applied to two expressions

Sailor Schema

Sailors(*sid*: `integer`, *sname*: `string`, *rating*: `integer`, *age*: `real`)
Boats(*bid*: `integer`, *bname*: `string`, *color*: `string`)
Reserves(*sid*: `integer`, *bid*: `integer`, *day*: `date`)

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

Instance S1 of Sailors

sid	*sname*	*rating*	*age*
28	yuppy	9	35.0
31	Lubber	8	55.5
44	guppy	5	35.0
58	Rusty	10	35.0

Instance S2 of Sailors

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

Instance R1 of Reserves

Relational Algebra

Selection and Projection

Relational algebra consists of the following:

- Operator to select rows (σ) from a relation
- Operator to project columns (π) from relation by extracting columns
- Set Operations: union ($\cup$), intersection ($\cap$), set-difference ($-$), and cross-product ($\times$).

Relational Algebra

Selection Operator (σ)

Consider expression $\underbrace{\sigma_{rating>8}}(S2)$

This subscript specifies the selection criteria to retrieve the tuples

sid	*sname*	*rating*	*age*
28	yuppy	9	35.0
58	Rusty	10	35.0

- $\sigma_{rating>8}(S2)$

sid	*sname*	*rating*	*age*
28	yuppy	9	35.0
31	Lubber	8	55.5
44	guppy	5	35.0
58	Rusty	10	35.0

Instance S2 of Sailors

Relational Algebra

Selection Operator (σ)

Consider expression $\sigma_{\underbrace{rating>8}}(S2)$

This selection condition is boolean combination that are

attribute **op** constant

or

attribute **op** attribute

where **op** is one of the following comparison operators: <, <=, =, ≠, >=, or >

Relational Algebra

Projection Operator (π)

Consider expression $\underbrace{\pi_{sname,rating}}(S2)$

This subscript specifies the Fields to be retained...

sid	*sname*	*rating*	*age*
28	yuppy	9	35.0
31	Lubber	8	55.5
44	guppy	5	35.0
58	Rusty	10	35.0

Instance S2 of Sailors

sname	*rating*
yuppy	9
Lubber	8
guppy	5
Rusty	10

- $\pi_{sname,rating}(S2)$

...the other fields are 'projected out'

Relational Algebra

Projection Operator (π)

If we want to find out the only the ages of sailors we use the expression: $\pi_{age}(S2)$

age
35.0
55.5

$\pi_{age}(S2)$

sid	*sname*	*rating*	*age*
28	yuppy	9	35.0
31	Lubber	8	55.5
44	guppy	5	35.0
58	Rusty	10	35.0

Instance S2 of Sailors

Notice that a single tuple of 35 appears in projection result

Relational Algebra

Selection and Projection

Since the result of a relational algebra expression is always a relation, we can substitute an expression wherever a relation is expected.

Compute the names and ratings of highly rated sailors

$$\pi_{sname,rating}(\sigma_{rating>8}(S2))$$

sname	*rating*
yuppy	9
Rusty	10

$\pi_{sname,rating}(\sigma_{rating>8}(S2))$

sid	*sname*	*rating*	*age*
28	yuppy	9	35.0
31	Lubber	8	55.5
44	guppy	5	35.0
58	Rusty	10	35.0

Instance S2 of Sailors

Apply the selection to S2 and then apply the projection

Relational Algebra

Set Operations

- Union
- Intersection
- Set-difference
- Cross-product or Cartesian product

Relational Algebra

Set Operations

Union:

- R∪S returns a relation instance containing all tuples that occur in either relation instance R or relation instance S (or both)
- R and S must be **union-compatible**
  - have the same number of fields (attributes)
  - fields (taken in order from L to R) have the same domain (type)
- Field names are not used to define union-compatibility

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

Instance S1 of Sailors

sid	*sname*	*rating*	*age*
28	yuppy	9	35.0
31	Lubber	8	55.5
44	guppy	5	35.0
58	Rusty	10	35.0

Instance S2 of Sailors

$S1 \cup S2$

sid	*sname*	*rating*	*age*
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.0
28	yuppy	9	35.0
44	guppy	5	35.0

Relational Algebra

Set Operations

Intersection:

- $R \cap S$ returns a relation instance containing all tuples that occur in both R and S.
- The relations R and S must be **union-compatible**
  - have the same number of fields (attributes)
  - fields (taken in order from L to R) have the same domain (type)
- The schema of the result is defined to be identical to the schema of R

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

Instance S1 of Sailors

sid	*sname*	*rating*	*age*
28	yuppy	9	35.0
31	Lubber	8	55.5
44	guppy	5	35.0
58	Rusty	10	35.0

Instance S2 of Sailors

sid	*sname*	*rating*	*age*
31	Lubber	8	55.5
58	Rusty	10	35.0

$S1 \cap S2$

Relational Algebra

Set Operations

Set-difference:

- R – S returns a relation instance containing all tuples that occur in R but not in S
- The relations R and S must be **union-compatible**
  - have the same number of fields (attributes)
  - fields (taken in order from L to R) have the same domain (type)
- The schema of the result is defined to be identical to the schema of R

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

Instance S1 of Sailors

sid	*sname*	*rating*	*age*
28	yuppy	9	35.0
31	Lubber	8	55.5
44	guppy	5	35.0
58	Rusty	10	35.0

Instance S2 of Sailors

sid	*sname*	*rating*	*age*
22	Dustin	7	45.0

$S1 - S2$

Relational Algebra

Set Operations

Cross-product or Cartesian product:

- R x S returns a relation instance containing all the fields of R in order as they appear in R followed by all the fields of S in order as they appear in S
- Fields of R x S inherit names from the corresponding fields of R and S but a *naming-conflict* can occur if both R and S contain one or more fields of the same name – in this case, only field positions will be used

S1

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

R1

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

not an inherited field name

(*sid*)	*sname*	*rating*	*age*	(*sid*)	*bid*	*day*
22	Dustin	7	45.0	22	101	10/10/96
22	Dustin	7	45.0	58	103	11/12/96
31	Lubber	8	55.5	22	101	10/10/96
31	Lubber	8	55.5	58	103	11/12/96
58	Rusty	10	35.0	22	101	10/10/96
58	Rusty	10	35.0	58	103	11/12/96

$S1 \times R1$

Relational Algebra

Renaming

- Due to *naming-conflicts* that can occur such as in the case of S1 x R1, giving a name for such a field would be very convenient

- Naming operator ρ is used

Example: $\rho(R(\overline{F}), E)$ → Arbitrary relational algebra expression

Renaming list of terms having the form: *oldname* → *newname* or *position* → *newname*

New relation that contains the same tuples and schema as E with some field renamed

Relational Algebra

Renaming

Example: $\rho(R(\overline{F}), E)$ → Arbitrary relational algebra expression

Renaming list of terms having the form:
oldname → *newname* or *position* → *newname*

New relation that contains the same tuples and schema as E with some field renamed

Hence, $\rho(C(1 \rightarrow sid1, 5 \rightarrow sid2),\ S1 \times R1)$

returns

C(*sid1:* `integer`, *sname:* `string`, *rating:* `integer`, *age:* `real`, *sid2:* `integer`, *bid:* `integer`, *day:* `dates`).

Relational Algebra

Joins

- one of the most useful operations
- combine information from two or more relations
- can be defined as a cross-product followed by selections and projections
- there are different types of joins

Condition Joins
Equijoins
Natural Joins
Outer Joins

Relational Algebra

Condition Joins

- $R \bowtie_c S$ is defined as being a cross-product followed by a selection
- The condition c typically refers to attributes of both *R* and *S*
- The reference to an attribute of a relation can be by position or name (*R.i* or *R.name* respectively)
- The condition in the form attribute **op** attribute where **op** is one of the following comparison operators: <, <=, =, ≠, >=, or >
- We can rewrite the join using cross-product and selection operations:

$$R \bowtie_c S = \sigma_c(R \times S)$$

Relational Algebra

Condition Joins

- *Example:* $$S1 \bowtie_{S1.sid<R1.sid} R1$$

sid appears in both *S1* and *R1*, the resulting fields are unnamed but the domains are inherited from corresponding fields

S1

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

R1

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

(*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

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

Relational Algebra

Equijoin

- A special case of join operation $R \bowtie S$ is when the the join condition based on **equalities** in the form of *R.name1 = S.name2*
- Since there is some redundancy in retaining both attributes, an additional projection is done and S.name2 is dropped
- Resulting schema of an equijoin contains fields from *R* followed by fields from *S* which are not in the join condition
- If the result relation contains two fields that inherit the same name from *R* and *S*, they will be unnamed

Example: $S1 \bowtie_{R.sid=S.sid} R1$

S1

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

R1

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

sid	*sname*	*rating*	*age*	*bid*	*day*
22	Dustin	7	45.0	101	10/10/96
58	Rusty	10	35.0	103	11/12/96

Relational Algebra

Natural Join

- A special case of equijoin join where the equalities are specified on all fields having the same name in *R* and *S*
- The join condition is omitted
- Default is a collection of equalities on all common fields
- The result never has two fields with the same name
- $S1 \bowtie_{R.sid=S.sid} R1$ is a natural join and can be denoted by $S1 \bowtie R1$, as the only common field is *sid*
- If the two relations have nothing in common, $S1 \bowtie R1$ is simply the cross product

Relational Algebra

Outer Join

- When you join two tables $\text{Sailors} \bowtie_c \text{Reserves}$. based on the join condition c, tuples from *Sailors* that do not match the rows from *Reserves* will not be included in the result
- Outer join allows *Sailor* tuples without a matching *Reserves* row to appear once in the result
  - **Left outer join**: *Sailor* rows without a matching *Reserves* row appear in the result
  - **Right outer join**: *Reserves* rows without a matching *Sailor* row appear in the result
  - **Full outer join**: Both *Sailor* and *Reserves* rows without a match appear in the result

Relational Algebra

Left Outer Join

- *Sailor* rows without a matching *Reserves* row appear in the result

Example:

```
SELECT Sailors.sid, Reserves.bid
FROM   Sailors NATURAL LEFT OUTER JOIN Reserves R
```

Sailors

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

Reserves

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

sid	*bid*
22	101
31	*null*
58	103

Left outer join

Relational Algebra

Division

- Used to express queries where rows from *R* match the combination of **<u>all</u>** rows in *S*

Example1: Find the names of sailors who have reserved all boats

Example2: Instance A has fields x and y
Instance B has field y
Both A.y and B.y have the same domain

A/B = set of all x values such that for every y value in B, there is a tuple ⟨x,y⟩ in A

Relational Algebra

Division

- Used to express queries where rows from *B* match the combination of **all** rows in *A*

A

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

B1

pno
p2

A/B1

sno
s1
s2
s3
s4

Relational Algebra

Division

- Used to express queries where rows from *B* match the combination of **all** rows in *A*

A

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

B2

pno
p2
p4

A/B2

sno
s1
s4

Relational Algebra

Division

- Used to express queries where rows from *B* match the combination of **<u>all</u>** rows in *A*

A

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

B3

pno
p1
p2
p4

A/B3

sno
s1

Relational Algebra

Division

- Compute all x values in *A* that are not disqualified
- x value is disqualified if by attaching a *y* value from *B*, a tuple <x,*y*> is created that is not in *A*

$$\pi_x(\underbrace{\overbrace{(\pi_x(A) \times B)}^{\text{all <x,y> combinations}} - \overbrace{A}^{\text{<x,y> combinations in A}}}_{\text{possible <x,y> combinations not in A}})$$

$$\underbrace{\pi_x(A)}_{\text{disqualified tuples in A}}$$

$$\underbrace{\pi_x(A) - \pi_x((\pi_x(A) \times B) - A)}_{\text{A/B}}$$

Relational Algebra Queries

Sailors, Reserves, Boats Schema

We will use the instances below to illustrate examples of relational algebra queries

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

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

Instance R2 of Reserves

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

Instance B1 of Boats

Relational Algebra Queries

1. Find the names of Sailors who have reserved boat 103

S3

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

R2

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

B1

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

$$\underbrace{\pi_{sname}}_{3}((\underbrace{\sigma_{bid=103}Reserves}_{1}) \bowtie \underbrace{Sailors}_{2})$$

Relational Algebra Queries

1. Find the names of Sailors who have reserved boat 103

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

Using the renaming ρ, we can break this query down further:

$$\rho(Temp1, \sigma_{bid=103} Reserves)$$

sid	bid	day
22	103	10/8/98
31	103	11/6/98
74	103	9/8/98

Instance of Temp1

$$\rho(Temp2, Temp1 \bowtie Sailors)$$

sid	sname	rating	age	bid	day
22	Dustin	7	45.0	103	10/8/98
31	Lubber	8	55.5	103	11/6/98
74	Horatio	9	35.0	103	9/8/98

$$\pi_{sname}(Temp2)$$

Instance of Temp2

Relational Algebra Queries

1. Find the names of Sailors who have reserved boat 103

- there are several distinct ways to write a query in relational algebra
- another way to write this is:

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

- we first compute the natural join of *Reserves* and *Sailors* and then apply the selection σ and the projection π

Relational Algebra Queries

1. Find the names of Sailors who have reserved boat 103

- DBMS translates the SQL query into relational algebra
- then looks for other algebra expressions that will produce the same answers but are cheaper to evaluate
- Query Optimizer will pick the cheapest one
- Which one is cheaper?

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

OR

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

Relational Algebra Queries

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

S3

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

R2

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

B1

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

$$\underbrace{\pi_{sname}}_{4}((\underbrace{\sigma_{color='red'}}_{1}Boats) \underbrace{\bowtie}_{2} Reserves \underbrace{\bowtie}_{3} Sailors)$$

Relational Algebra Queries

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

- Alternative expression is :

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

- Which one is cheaper?

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

OR

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

Relational Algebra Queries

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

- Alternative expression is :

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

- Which one is cheaper?

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

OR

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

Relational Algebra Queries

3. Find the colors of boats reserved by Lubber

S3

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

R2

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

B1

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

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

4 1 2 3

Relational Algebra Queries

4. Find the names of sailors who have reserved at least one boat

S3

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

R2

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

B1

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

$$\underbrace{\pi_{sname}}_{2}(\underbrace{Sailors \bowtie Reserves}_{1})$$

Relational Algebra Queries

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

S3

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

R2

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

B1

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

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

1

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

4 2 3

Relational Algebra Queries

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

- Can we use the previous expression and flip the union to intersection?

$$\rho(Tempboats, (\sigma_{color='red'} Boats) \cup (\sigma_{color='green'} Boats))$$
$$\pi_{sname}(Tempboats \bowtie Reserves \bowtie Sailors)$$

TO

$$\rho(Tempboats2, (\sigma_{color='red'} Boats) \cap (\sigma_{color='green'} Boats))$$
$$\pi_{sname}(Tempboats \bowtie Reserves \bowtie Sailors)$$

Relational Algebra Queries

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

S3

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

R2

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

B1

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

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

1

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

2

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

4 3

M. Rathod

Relational Algebra Queries

Changing 6 (and) to 5 (or)

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

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

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

TO

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

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

$$\pi_{sname}((Tempred \cup Tempgreen) \bowtie Sailors)$$

Relational Algebra Queries

Consider for 6 (and):

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

$$\rho(Tempgreen, \pi_{sname}((\sigma_{color='green'}Boats) \bowtie Reserves \bowtie Sailors))$$

$$Tempred \cap Tempgreen$$

S3

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

R2

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

B1

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

Relational Algebra Queries

7. Find the names of sailors who have reserved at least two boats

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

S3

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

R2

sid1	*sname1*	*bid1*
22	Dustin	101
22	Dustin	102
22	Dustin	103
22	Dustin	104
31	Lubber	102
31	Lubber	103
31	Lubber	104
64	Horatio	101
64	Horatio	102
74	Horatio	103

Reservations

$$\rho(Reservations, \pi_{sid,sname,bid}(Sailors \bowtie Reserves))$$ 1

$$\rho(Reservationpairs(1 \rightarrow sid1, 2 \rightarrow sname1, 3 \rightarrow bid1, 4 \rightarrow sid2, 5 \rightarrow sname2, 6 \rightarrow bid2), Reservations \times Reservations)$$ 2

$$\pi_{sname1}\sigma_{(sid1=sid2)\wedge(bid1\neq bid2)} Reservationpairs$$ 3

Relational Algebra Queries

7. Find the names of sailors who have reserved at least two boats

sid1	*sname1*	*bid1*
22	Dustin	101
22	Dustin	102
22	Dustin	103
22	Dustin	104
31	Lubber	102
31	Lubber	103
31	Lubber	104
64	Horatio	101
64	Horatio	102
74	Horatio	103

Reservations

sid1	*sname1*	*bid1*	*Sid2*	*Sname2*	*bid2*
22	Dustin	101			
22	Dustin	102			
22	Dustin	103			
22	Dustin	104			
31	Lubber				

$$\rho(Reservationpairs(1 \rightarrow sid1, 2 \rightarrow sname1, 3 \rightarrow bid1, 4 \rightarrow sid2,$$

$$5 \rightarrow sname2, 6 \rightarrow bid2), Reservations \times Reservations)$$

Relational Algebra Queries

8. Find the sids of sailors with age over 20 who have not reserved a red boat

S3

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

R2

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

B1

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

102, 104

22, 29, 31, 32, 58, 64, 74, 85, 95

22, 31, 64

1 $\pi_{sid}(\sigma_{age>20} Sailors) -$

2 $\pi_{sid}((\sigma_{color='red'} Boats) \bowtie Reserves \bowtie Sailors)$

{22, 29, 31, 32, 58, 64, 74, 85, 95} - {22, 31, 64} = {29, 32, 58, 74, 85, 95}

- To compute the names of such sailors, we must first compute their *sids* (as shown above), and then join with *Sailors* and project the *sname* values.

Relational Algebra Queries

9. Find the names of sailors who have reserved all boats

The use of the word all (or every) indicates that the division operation might be applicable

S3

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

R2

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

B1

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

$$\rho(Tempsids, (\pi_{sid,bid}Reserves)/(\pi_{bid}Boats))$$

2 1

$$\pi_{sname}(Tempsids \bowtie Sailors)$$

4 3

Relational Algebra Queries

10. Find the names of sailors who have reserved all boats called Interlake

S3

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

R2

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

B1

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

$$\rho(Tempsids, (\pi_{sid,bid} Reserves)/(\pi_{bid}(\sigma_{bname='Interlake'} Boats)))$$

2 1

$$\pi_{sname}(Tempsids \bowtie Sailors)$$

4 3

Summary

Database systems use some variant of relational algebra to represent query evaluation plans

Relational algebra consists of the following:

- operator to select rows (σ) from a relation
- operator to project columns (π) from relation by extracting columns
- set operations: union ($\cup$), intersection ($\cap$), set-difference ($-$), and cross-product ($\times$)
- naming operator ρ
- join $\bowtie$.
- division (/)