Relational Algebra Operators

Presentation by

Dr. Jenila Livingston L.M.

VIT Chennai

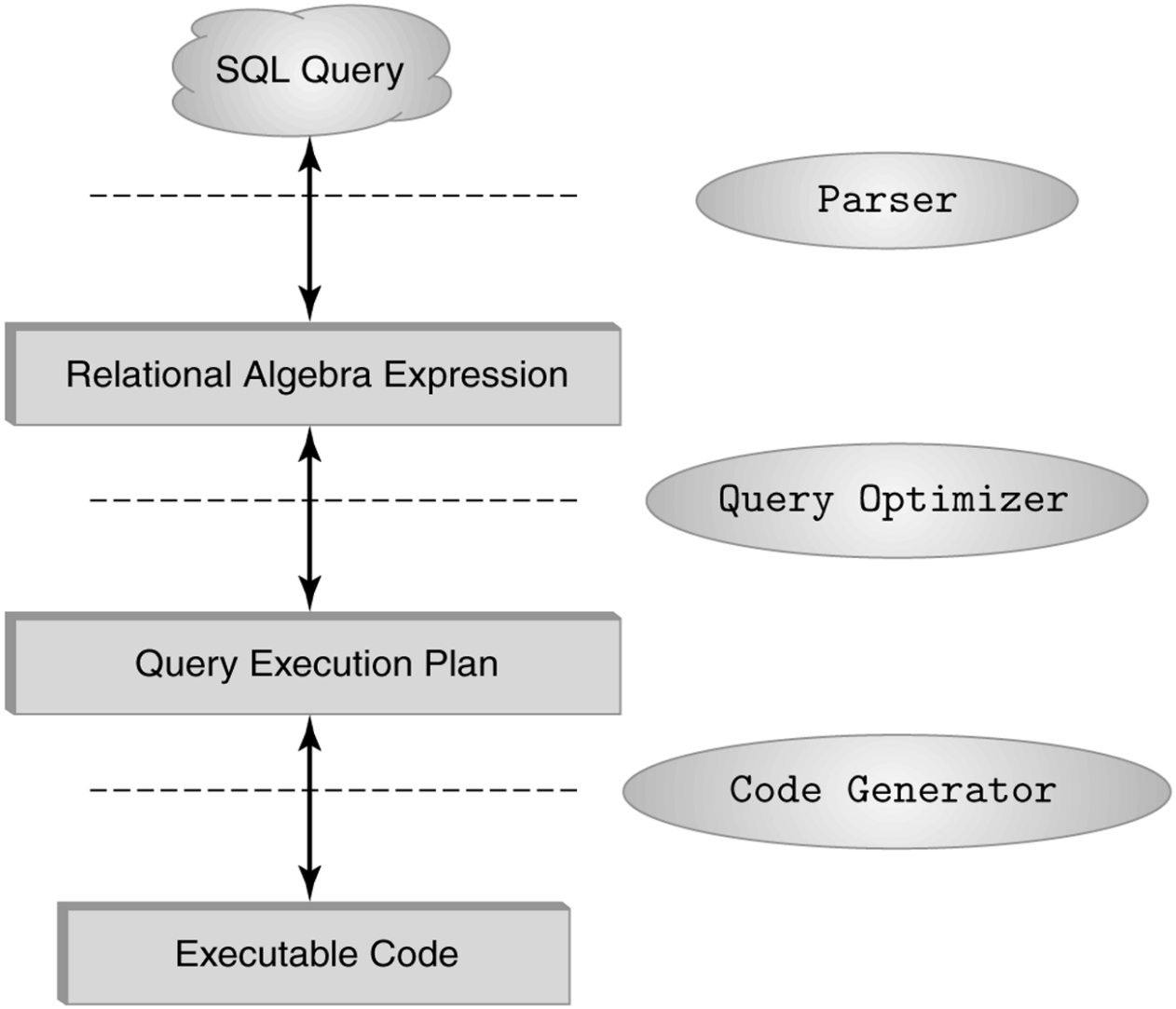
Formal Query Languages

- Specialized languages for asking questions or queries
- Theoretical languages
- Developed before SQL. SQL is based on these languages
  - Relational Algebra
  - Relational Calculus

Relational Algebra

- ***Introduced by E. F. Codd in 1970.***
- **More operational, very useful for representing execution plans.**
- **relational algebra**, a **procedural language** that defines database operations in terms of algebraic expressions.
- Similar to normal algebra (as in 2+3*x-y), except we use relations as values instead of numbers/ operands , and the operations and operators are different.
- We need to know about relational algebra to understand query execution and optimization in a relational DBMS.

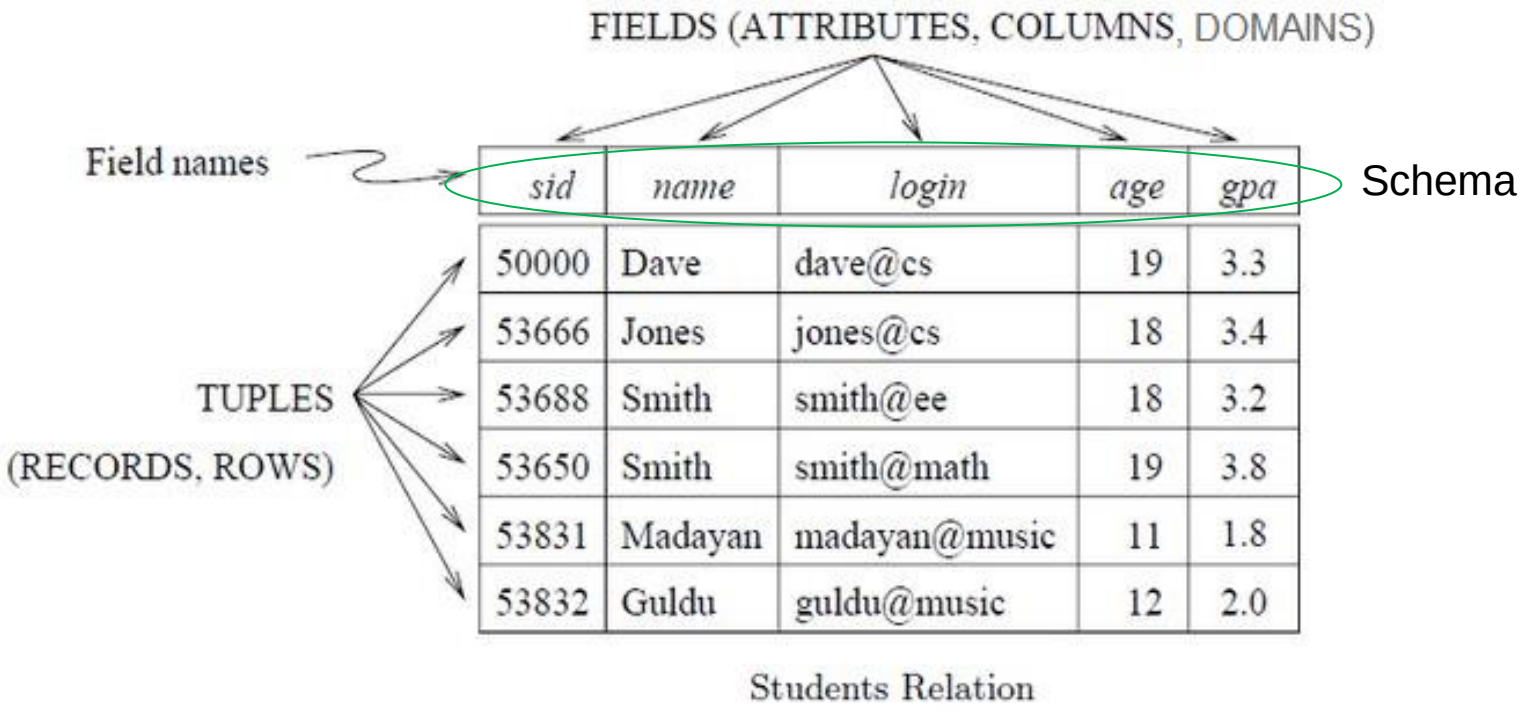
The Role of Relational Algebra in a DBMS



Relational Algebra

1. Selection of Tuples
2. Selection of Attributes
3. Set Operations
4. Join operations
5. Rename
6. Assignment
7. Aggregation

Table or Relation



sid	*name*	*login*	*age*	*gpa*
50000	Dave	dave@cs	19	3.3
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.8
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

Students Relation

Degree: No. of attributes = 5
Cardinality: No. of tuples = 6

1. Selection of tuples

Returns all tuples which satisfy a condition

- Relation r
- **Selection (σ)**

Notation: $\sigma_c(r)$

The condition c can be =, <, ≤, >, ≥, <>

- Select all attributes/ tuples from r

  $\sigma(r)$

- Select tuples with A=B and D > 5

  $\sigma_{A=B \text{ and } D > 5}(r)$

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

A	B	C	D
α	α	1	7
β	β	23	10

Select Operator

- Produce table containing subset of rows of argument table satisfying condition

$$\sigma_{condition}(relation)$$

- Example: List the persons where hobby is stamp collection

$$\sigma_{Hobby='stamps'}(Person)$$

Id	*Name*	*Address*	*Hobby*
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Id	*Name*	*Address*	*Hobby*
1123	John	123 Main	stamps
9876	Bart	5 Pine St	stamps

Selection Condition - Examples

- σ *Id*>3000 OR *Hobby*='hiking' (Person)
- σ *Id*>3000 AND *Id* <3999 (Person)
- σ NOT(*Hobby*='hiking') (Person)
- σ *Hobby*$\neq$'hiking' (Person)

2. Selection of Columns (Attributes)

Retains only wanted ***columns*** from relation (vertical).

- Relation *r*:
- **Projection (Π)**

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

Notation: $\Pi_{A1,\ldots,An}(r)$

- Select A and C
  - $\Pi_{A, C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Project Operator

- Produces table containing subset of columns of argument table

$$\pi_{attribute\ list}(relation)$$

- Example: Display name and hobby of persons

$$\pi_{Name,Hobby}(\text{Person})$$

Id	*Name*	*Address*	*Hobby*
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Name	*Hobby*
John	stamps
John	coins
Mary	hiking
Bart	stamps

Project Operator

- Example:

$\pi_{Name,Address}(Person)$

Id	*Name*	*Address*	*Hobby*
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Name	*Address*
John	123 Main
Mary	7 Lake Dr
Bart	5 Pine St

Result is a table (no duplicates); can have fewer tuples than the original

Expressions

$\pi_{Id, Name} (\sigma_{Hobby='stamps' \text{ OR } Hobby='coins'} (Person))$

Id	*Name*	*Address*	*Hobby*
1123	John	123 Main	stamps
1123	John	123 Main	coins
5556	Mary	7 Lake Dr	hiking
9876	Bart	5 Pine St	stamps

Person

Id	*Name*
1123	John
9876	Bart

Result

3. Set Operations

- Union (∪)
- Intersection (∩)
- Set Difference (-)
- Division (/)
- Cartesian Product / Cross Product

Union of two relations

- Relations *r, s:*

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- r ∪ s:

A	B
α	1
α	2
β	1
β	3

Union

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

$S1 \cup S2$

Set Intersection of two relations

- Relation *r, s*:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- $r \cap s$

A	B
α	2

Intersection

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

sid	sname	rating	age
31	lubber	8	55.5
58	rusty	10	35.0

$S1 \cap S2$

Set difference of two relations

- Relations *r*, *s*:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- *r – s:*

A	B
α	1
β	1

r-S means Tuples in r1, but not in r2.

Set Difference

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S1

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

S2

sid	sname	rating	age
22	dustin	7	45.0

$S1 - S2$

sid	sname	rating	age
28	yuppy	9	35.0
44	guppy	5	35.0

$S2 - S1$

4. Division

- Denoted by ÷
- Binary Operation
- Used in queries that include the phrase “for all”.

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

Exercise

- Selects tuples from books where subject is 'database'.
- Selects tuples from books where subject is 'database' and 'price' is 450.
- Selects tuples from books where subject is 'database' and 'price' is 450 or those books published after 2010
- Selects subject and author from the relation Books.

- Projects the names of the authors who have either written a book or an article or both
- Provides the name of authors who have written books but not articles.
- Yields a relation, which shows all the books and articles written by bala.

- $\sigma_{subject = "database"}(Books)$
- $\sigma_{subject = "database" \text{ and } price = "450"}(Books)$
- $\sigma_{subject = "database" \text{ and } price = "450" \text{ or } year > "2010"}(Books)$
- $\prod_{subject, author}(Books)$

- $\prod_{author}$ (Books) ∪ $\prod_{author}$ (Articles)
- $\prod_{author}$ (Books) − $\prod_{author}$ (Articles)
- $\sigma_{author = \text{'bala'}}$(Books X Articles)

Group by and Order by

- Grouping γ
- Sorting τ

4. Joining two relations
Cartesian Product

- Relations *r, s*:

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

- *r* x *s*:

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

Cartesian Product - Example

r

A	B
a1	2
a2	4

s

C	D	E
4	d1	e1
3	d2	e1
5	d3	e2

rx s

A	B	C	D	E
a1	2	4	d1	e1
a1	2	3	d2	e1
a1	2	5	d3	e2
a2	4	4	d1	e1
a2	4	3	d2	e1
a2	4	5	d3	e2

Cartesian Product: $R \times S$

R

A	B	C
a1	b1	c3
a2	b1	c5
a3	b4	c7

S

E	F
e1	f1
e2	f5

$R \times S$

A	B	C	E	F
a1	b1	c3	e1	f1
a1	b1	c3	e2	f5
a2	b1	c5	e1	f1
a2	b1	c5	e2	f5
a3	b4	c7	e1	f1
a3	b4	c7	e2	f5

Inner Join

- 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	ε

s

- Natural Join
  - 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

Outer Join

- Left Outer Join ()
- Right Outer Join ()
- Full Outer Join ()

Join – Example

- Relation *loan*

loan_number	*branch_name*	*amount*
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

- Relation *borrower*

customer_name	*loan_number*
Jones	L-170
Smith	L-230
Hayes	L-155

Outer Join – Example

- Inner Join

$$loan \bowtie Borrower$$

loan_number	*branch_name*	*amount*	*customer_name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

- Left Outer Join

loan ⟕ *Borrower*

loan_number	*branch_name*	*amount*	*customer_name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	*null*

Outer Join – Example

- Right Outer Join

  loan ⟖ *borrower*

loan_number	*branch_name*	*amount*	*customer_name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	*null*	*null*	Hayes

- Full Outer Join

  loan ⟗ *borrower*

loan_number	*branch_name*	*amount*	*customer_name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	*null*
L-155	*null*	*null*	Hayes

5. Rename operation - ρ

The operation $\rho_x(r)$

- means relation r renamed to x.

6. Assignment

- The assignment operation (←) provides a convenient way to express complex queries.
  - Write query as a sequential program consisting of
    - a series of assignments
    - followed by an expression whose value is displayed as a result of the query.
  - Assignment must always be made to a temporary relation variable.
- Example: Write $r \div s$ as

$$temp1 \leftarrow \Pi_{R-S}(r)$$
$$temp2 \leftarrow \Pi_{R-S}((temp1 \times s) - \Pi_{R-S,S}(r))$$
$$result = temp1 - temp2$$

  - The result to the right of the ← is assigned to the relation variable on the left of the ←.
  - May use variable in subsequent expressions

7. Aggregate

- Functions on more than one tuple
- Samples:
  - Sum
  - Count-distinct
  - Max
  - Min
  - Count
  - Avg
- Use “as” to rename

$$\Pi_{branchname,\ \mathcal{G}\ \textbf{sum}(balance)\ \textbf{as}\ totalbalance}(account)$$

Aggregate Operation – Example

- Relation *account* grouped by *branch-name*:

branch_name	*account_number*	*balance*
Perryridge	A-102	400
Perryridge	A-201	900
Brighton	A-217	750
Brighton	A-215	750
Redwood	A-222	700

$$\Pi_{branch\_name}\, g_{\mathbf{sum}(balance)}\,(account)$$

branch_name	**sum**(*balance*)
Perryridge	1300
Brighton	1500
Redwood	700

Summary

Symbol (Name)	Example of Use
σ (Selection)	$\sigma_{\text{salary}>=85000}(instructor)$
	Return rows of the input relation that satisfy the predicate.
Π (Projection)	$\Pi_{ID,\ salary}(instructor)$
	Output specified attributes from all rows of the input relation. Remove duplicate tuples from the output.
⋈ (Natural Join)	$instructor \bowtie department$
	Output pairs of rows from the two input relations that have the same value on all attributes that have the same name.
× (Cartesian Product)	$instructor \times department$
	Output all pairs of rows from the two input relations (regardless of whether or not they have the same values on common attributes)
∪ (Union)	$\Pi_{name}(instructor) \cup \Pi_{name}(student)$
	Output the union of tuples from the two input relations.

EXAMPLES OF ALGEBRA QUERIES

In the rest of this section we shall illustrate queries using the following relations

Sailors, Reserves and 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

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

QUERY Q1

(Q1) Find the names of sailors who have reserved boat 103

Two relations involved here are

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

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

The answer is thus the following relational instance

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

QUERY Q1 (cont'd)

There are of course several ways to express Q1 in relational algebra.

Here is another:

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

Which of these expressions should we use?

That is a question of optimization. Indeed, when we describe how to state queries in SQL, we can leave it to the optimizer in the DBMS to select the nest approach.

QUERY Q2

(Q2) 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 $S3$ 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

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

QUERY Q3

(Q3) Find the colors of boats reserved by Lubber.

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

QUERY Q3

(Q3) Find the colors of boats reserved by Lubber.

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

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

QUERY Q4

(Q4) 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

QUERY Q4

(Q4) 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

π_{sname}(Sailors $\bowtie$ Reserves)

QUERY Q5

(Q5) 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 $S3$ 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

QUERY Q5

(Q5) 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 $S3$ 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

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

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

QUERY Q6

(Q6) 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 ∪ by ∩. However, this won't work, for such an expression is requesting the names of sailors who have requested a boat that is both red and green! The correct expression is as follows:

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

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

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

QUERY 7

(Q7) Find the sids of sailors with age over 20 who have not 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 $S3$ 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

$\pi_{sid}(\sigma_{age>20}Sailors) - \pi_{sid}((\sigma_{color='red'}Boats) \bowtie Reserves \bowtie Sailors)$

QUERY 8

(Q) Find the names of sailors who have reserved all 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

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

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

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

QUERY Q9

(Q9) Find the names of sailors who have reserved all boats called Interlake.

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

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

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

Examples

- *Person* (name, age, gender)
  Frequents (name, pizzeria)
  Eats (name, pizza)
  Serves (pizzeria, pizza, price)

Examples

- a. Find all pizzerias frequented by at least one person under the age of 18.

  b. Find the names of all females who eat either mushroom or pepperoni pizza (or both).

  c. Find the names of all females who eat both mushroom and pepperoni pizza.

a. $\pi_{pizzeria}(\sigma_{age<18}(Person) \bowtie Frequents)$

b. $\pi_{name}\left(\sigma_{gender='female' \wedge (pizza='mushroom' \vee pizza='pepperoni')}(Person \bowtie Eats)\right)$

c. $\pi_{name}(\sigma_{gender='female' \wedge pizza='mushroom'}(Person \bowtie Eats)) \cap$
$\pi_{name}(\sigma_{gender='female' \wedge pizza='pepperoni'}(Person \bowtie Eats))$

- d. Find all pizzerias that serve at least one pizza that Amy eats for less than $10.00.

  e. Find all pizzerias that are frequented by only females or only males.

  f. For each person, find all pizzas the person eats that are not served by any pizzeria the person frequents. Return all such person (name) / pizza pairs.

d. $\pi_{pizzeria}\left(\sigma_{name='\text{Amy}'}(Eats) \bowtie \sigma_{price<10}(Serves)\right)$

e. $$\left(\begin{array}{l}\pi_{pizzeria}(\sigma_{gender='\text{female}'}(Person) \bowtie Frequents) \; - \\ \pi_{pizzeria}(\sigma_{gender='\text{male}'}(Person) \bowtie Frequents)\end{array}\right) \cup \left(\begin{array}{l}\pi_{pizzeria}(\sigma_{gender='\text{male}'}(Person) \bowtie Frequents) \; - \\ \pi_{pizzeria}(\sigma_{gender='\text{female}'}(Person) \bowtie Frequents)\end{array}\right)$$

f. $Eats - \pi_{name,pizza}(Frequents \bowtie Serves)$

- g. Find the names of all people who frequent only pizzerias serving at least one pizza they eat.

  h. Find the names of all people who frequent every pizzeria serving at least one pizza they eat.

  i. Find the pizzeria serving the cheapest pepperoni pizza. In the case of ties, return all of the cheapest-pepperoni pizzerias.

g. $\pi_{name}(Person) - \pi_{name}\left(Frequents - \pi_{name,pizzeria}(Eats \bowtie Serves)\right)$

h. $\pi_{name}(Person) - \pi_{name}\left(\pi_{name,pizzeria}(Eats \bowtie Serves) - Frequents\right)$

i. $$\pi_{pizzeria}(\sigma_{pizza='\text{pepperoni}'} Serves) \; - \; \pi_{pizzeria}\left(\sigma_{price>price2}\left(\begin{array}{l}\pi_{pizzeria,price}(\sigma_{pizza='\text{pepperoni}'} Serves) \\ \quad \times \\ \rho_{pizzeria2,price2}\left(\pi_{pizzeria,price}(\sigma_{pizza='\text{pepperoni}'} Serves)\right)\end{array}\right)\right)$$

Thank You!