Relational Algebra

Chapter 4

What is it?

- Theoretical Foundations of Queries
- Questions to the Database regarding Data

- Queries in Relational Algebra are formulated using a collection of operators
- Examples
  - π
  - σ
  - $\bowtie$

Introduction and Schemas

- Assumptions

Sailors

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25

Reserves

Sid	Bid	Day
22	101	10/10/96
58	103	11/12/96

Boats

Bid	Bname	color
101	Interlake	Blue
102	Interlake	red
103	Clipper	Green
104	Marine	red

Relational Algebra

- Foundations of Queries
- Fundamental Properties
  - Every operator accepts a relations (one or two) as input
  - Every operator returns a relation as output
- This property allows formation of complex queries.
- Operators
  - π
  - σ
  - ρ
  - Set operators: $\cup \cap - \times$
  - $\bowtie$

Select Operator (σ)

- Formal notation
- $R' = \sigma_{conditions}(R)$
- Works on the horizontal dimension (tuples)

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25
17	Popeye	10	40

Examples on Selection

- $R' = \sigma_{Sailors.age > 40}(Sailors)$

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55

Examples on Selection (2)

- $R' = \sigma_{\substack{Sailors.rating >= 8 \\ and\ Sailors.age > 50}}(Sailors)$

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25

Sid	Sname	Rating	Age
31	Lubber	8	55

- Can apply multiple conditions: AND OR ($\wedge$ $\vee$)
- Can apply multiple selections instead
- $R' = \sigma_{Sailors.rating \geq 8}(Sailors)$
- $Final' = \sigma_{R'.age \geq 50}(R')$

Alias Operator (ρ)

- Alias for relation names and/or column names
- $\rho(newName, oldName)$
- Examples
  - $\rho(S, Sailors)$
    - Returns a relation S with the same data as Sailors. Now we can use S instead of Sailors
    - $R' = \sigma_{\substack{S.rating >= 8 \\ and\ S.age > 50}}(S)$
  - $\rho(S(sid \rightarrow sid1, rating \rightarrow rate), Sailors)$
    - Returns a relation S with the same data as Sailors, with same columns, except sid is now sid1 and rating is now rate.

Project Operator ($\prod R$)

- Formal notation
- $R' = \prod_{columns}(R)$
- Works on the vertical dimension (attributes)
- Will only show unique values.

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25

Examples on Projection

- $\rho(S, Sailors)$
- $R' = \prod_{S.sname,s.age}(S)$

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25

Sname	Age
Dustin	45
Lubber	55
Rusty	35
Andy	25

Examples on Projection(2)

- $\rho(S, Sailors)$
- $R' = \prod_{S.rating}(S)$
- Remember: Unique values only!

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25

Rating
7
8
10

Mixing Operators

- More useful when mixing selections and projections
- Example
- Get sailor names whose age is greater than 40.
  - $\rho(S, Sailors)$
  - $R' = \prod_{S.sname,}(S)$ ------------------→ R' only sname
  - $Final' = \sigma_{R'.age > 40}\left(R'\right)$

  - Can we do this?!!
  - Correct result:
  - $\rho(S, Sailors)$
  - $R' = \sigma_{S.age > 40}(S)$
  - $Final' = \prod_{S.sname,}(R')$

Set Operations: Union

- **Union:** $R \cup S$ 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*, and the schema of the result is defined to be identical to the schema of R.

  Two relation instances are said to be **union-compatible** if the following conditions hold:

  – they have the same number of the fields, and

  – corresponding fields, taken in order from left to right, have the same *domains*.

Set Operations: Intersection and Set Difference

- **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, and the schema of the result is defined to be identical to the schema of R.

- **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, and the schema of the result is defined to be identical to the schema of R.

Examples on Set Operators

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35

∪

Sid	Sname	Rating	Age
28	yuppy	9	35
31	Lubber	8	55
44	guppy	5	35
58	Rusty	10	35

=

Sid	Sname	Rating	Age
22	Dustin	7	45
28	yuppy	9	35
31	Lubber	8	55
44	guppy	5	35
58	Rusty	10	35

Examples on Set Operators (2)

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35

$\cap$

Sid	Sname	Rating	Age
28	yuppy	9	35
31	Lubber	8	55
44	guppy	5	35
58	Rusty	10	35

$=$

Sid	Sname	Rating	Age
31	Lubber	8	55
58	Rusty	10	35

Examples on Set Operators (3)

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35

−

Sid	Sname	Rating	Age
28	yuppy	9	35
31	Lubber	8	55
44	guppy	5	35
58	Rusty	10	35

=

Sid	Sname	Rating	Age
22	Dustin	7	45

Introduction and Schemas

- Assumptions

Sailors

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25

Reserves

Sid	Bid	Day
22	101	10/10/96
58	103	11/12/96

Boats

Bid	Bname	color
101	Interlake	Blue
102	Interlake	red
103	Clipper	Green
104	Marine	red

Cross Product:

- **Cross-product:** $R \times S$ returns a relation instance whose schema contains all the fields of R (in the same order as they appear in R) followed by all the fields of S (in the same order as they appear in S). The result of $R \times S$ contains one tuple $\langle r, s \rangle$ (the concatenation of tuples r and s) for each pair of tuples $r \in R$, $s \in S$. The cross-product opertion is sometimes called **Cartesian product**.

  We will use the convention that the fields of $R \times S$ inherit names from the corresponding fields of R and S. It is possible for both R and S to contain one or more fields having the same name; this situation creates a *naming conflict*. The corresponding fields in $R \times S$ are unnamed and are referred to solely by position.

Example:

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35

×

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	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

Figure 4.11 $S1 \times R1$

Condition Join

- The most general version of the join operation accepts a join condition c and a pair of relation instances as arguments, and returns a relation instance

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

Condition Join Example

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

(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

Equijoin

- A common special case of the join operation R S is when the join condition consists solely of equalities of the form R.name1 = S.name2, that is, equalities between two fields in R and S.
- The schema of the result of an equijoin contains the fields of R (with the same names and domains as in R) followed by the fields of S that do not appear in the join conditions.

Natural Join

- A further special case of the join operation R S is an equijoin in which equalities are specified on all fields having the same name in R and S.
- In this case, we can simply omit the join condition.

$$S1 \bowtie_{R.sid=S.sid} R1 \quad \Rightarrow \quad S1 \bowtie R1$$

Examples

Sailors

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25

Reserves

Sid	Bid	Day
22	101	10/10/96
58	103	11/12/96

Boats

Bid	Bname	color
101	Interlake	Blue
102	Interlake	red
103	Clipper	Green
104	Marine	red

- Find the names of Sailors who have reserved boat id = 103

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

- Find the names of Sailors who reserved a red boat

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

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

- Find the color of boats reserved by Rusty

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

Examples (2)

Sailors

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25

Reserves

Sid	Bid	Day
22	101	10/10/96
58	103	11/12/96

Boats

Bid	Bname	color
101	Interlake	Blue
102	Interlake	red
103	Clipper	Green
104	Marine	red

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

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

- Find the names of Sailors who reserved a red boat or a green boat

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

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

- Find the names of Sailors who reserved a red AND a green boat

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

Examples

Sailors

Sid	Sname	Rating	Age
22	Dustin	7	45
31	Lubber	8	55
58	Rusty	10	35
32	Andy	8	25

Reserves

Sid	Bid	Day
22	101	10/10/96
58	103	11/12/96

Boats

Bid	Bname	color
101	Interlake	Blue
102	Interlake	red
103	Clipper	Green
104	Marine	red

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

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

Examples (4)

Sid	Bid	Day
22	101	10/10/96
58	103	11/12/96
22	107	1/2/98

- Find the sids of sailors who reserved at least two different boats

Sid	Bid	Day	Sid’	Bid’	Day’
22	101	10/10/96	22	101	10/10/96
22	101	10/10/96	58	103	11/12/96
22	101	10/10/96	22	107	1/2/98
58	103	11/12/96	22	101	10/10/96
58	103	11/12/96	58	103	11/12/96
58	103	11/12/96	22	107	1/2/98
22	107	1/2/98	22	101	10/10/96
22	107	1/2/98	58	103	11/12/96
22	107	1/2/98	22	107	1/2/98