Faculty of Engineering and Tecnology

Computer Science Department

Relational Algebra

Chapter 4

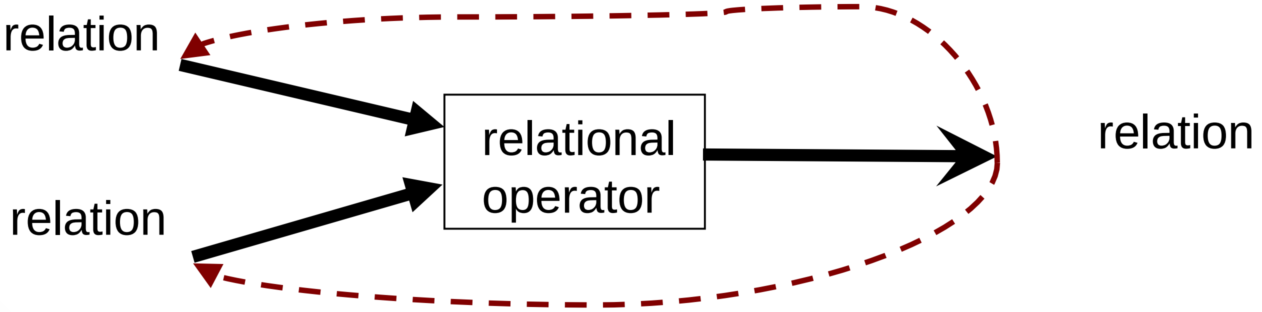
Relational Query Languages

- Query = “retrieval program”
- Language examples:
  - Theoretical:
    1. Relational Algebra
    2. Relational Calculus
  - Practical
    1. SQL (SEQUEL from System R)
    2. QUEL (Ingres)
    3. Datalog (Prolog-like)

- Theoretical QL’s:
  - give semantics to practical QL’s
  - key to understand query optimization in relational DBMSs

Relational Algebra

- Basic operators
  - select (σ)
  - project (π)
  - union ($\cup$)
  - set difference (-)
  - cartesian product (x)
  - rename (ρ)
- The operators take one or two relations as inputs and give a new relation as a result.



Example Instances

R1

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

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

Boats

S1

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

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

Schema:
Boats(bid, bname, color)
Sailors(sid, sname, rating, age)
Reserves(sid, bid, day)

Projection

- **Examples:** $\rho_{age}(S2)$ $\pi_{sname,rating}(S2)$
- Retains only attributes that are in the "*projection list*".
- *Schema* of result:
  - exactly the columns in the projection list, with the same names that they had in the input relation.
- Projection operator has to *eliminate duplicates* (How do they arise? Why remove them?)
  - Note: real systems typically don't do duplicate elimination unless the user explicitly asks for it. (Why not?)

Projection

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

sname	rating
yuppy	9
lubber	8
guppy	5
rusty	10

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

age
35.0
55.5

$$p_{age}(S2)$$

Selection (σ)

- **Selects rows that satisfy *selection condition*.**
- Result is a relation.

  Schema of result is same as that of the input relation.
- Do we need to do duplicate elimination?

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

$$s_{rating>8}(S2)$$

Selection

- Notation: $\sigma_p(r)$
- *p* is called the selection predicate , **r** can be the name of a table, or another query
- Predicate:
  1. Simple
     - □ attr1 = attr2
     - □ Attr = constant value
       - (also, <, > , etc)
  2. Complex
     - □ predicate1 AND predicate2
     - □ predicate1 OR predicate2
     - □ NOT (predicate)

Union and Set-Difference

- **All of these operations take two input relations, which must be *union-compatible*:**
  - Same number of columns (attributes).
  - `Corresponding’ collumns have the same domain (type).
- **For which, if any, is duplicate elimination required?**

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 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

Cartesian-Product

- **S1 × R1: Each row of S1 paired with each row of R1.**

  Like the c.p for mathematical relations: every tuple of S1 “appended” to every tuple of R1
- Q: How many rows in the result?
- *Result schema* has one field per field of S1 and R1, with field names `inherited’ if possible.
  - *May have a naming conflict*: Both S1 and R1 have a field with the same name.
  - In this case, can use the *renaming operator*…

BIRZEIT UNIVERSITY

Cartesian Product Example

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

S1

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

R1

S1 X R1 =

(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

Rename (ρ)

- Allows us to refer to a relation by more than one name and to rename conflicting names

Example:

$$\rho\,(X, E)$$

returns the expression *E* under the name *X*

- If a relational-algebra expression *E* has arity *n*, then

$$\rho(X(1\text{->}A1,\ 2\text{->}A2,\ \ldots,\ n\text{->}An),\ E)$$

returns the result of expression *E* under the name *X*, and with the attributes renamed to *A1, A2, …., An*.

Ex. $\rho(C(1\text{->}sid1,\ 5\text{->}sid2),\ S1xR1)$

Compound Operator: Intersection

- In addition to the 6 basic operators, there are several additional "Compound Operators"
  - These add no computational power to the language, but are useful shorthands.
  - Can be expressed solely with the basic ops.

- Intersection takes two input relations, which must be *union-compatible*.
- Q: How to express it using basic operators?

$$\mathbf{R \cap S = R - (R - S)}$$

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

$$S1 \cap S2$$

Compound Operator: Join

- Joins are compound operators involving cross product, selection, and (sometimes) projection.
- Most common type of join is a "*natural join*" (often just called "join").
- R⋈S conceptually is:
  - Compute R × S
  - Select rows where attributes that appear in both relations have equal values
  - Project all unique attributes and one copy of each of the common ones.
- Note: Usually done much more efficiently than this.

Natural Join Example

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

R1

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

S1

S1 $\bowtie$ R1 =

sid	sname	rating	age	bid	day
22	dustin	7	45.0	101	10/10/96
58	rusty	10	35.0	103	11/12/96

Other Types of Joins

- *Condition Join (or "theta-join")*: $R \bowtie_{c} S = s_{c}(R ´ \ S)$

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

(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

- *Result schema* same as that of cross-product.
- May have fewer tuples than cross-product.
- Equi-join: special case: condition *c* contains only conjunction of ***equalities***.

Example Instances

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

Reserves

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

Boats

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

Sailors

Schema:
Boats(bid, bname, color)
Sailors(sid, sname, rating, age)
Reserves(sid, bid, day)

Examples of RA Queries

1. Find the names of sailors who have reserved boat 103
2. Find the name of sailors who reserved a red boat
3. Find the color of boats reserved by Dustin
4. Find names of sailors who have reserved a red or a green boat
5. Find names of sailors who have reserved a red and a green boat

Examples of RA Queries

- Find the names of sailors who reserved at least two boats
- Find the sids of sailors with age over 20 who have not reserved a red boat
- Find the sids of sailors who reserved all boats