Relational Algebra (Part II)

R & G, Chapter 4

Last Lecture

- **Relational Algebra: 5 Basic Operations**
  - *Selection* (σ) Selects a subset of *rows* from relation (horizontal).
  - *Projection* (π) Retains only wanted *columns* from relation (vertical).
  - *Cross-product* ($\times$) Allows us to combine two relations.
  - *Set-difference* (—) Tuples in r1, but not in r2.
  - *Union* ($\cup$) Tuples in r1 and/or in r2.

Today’s Lecture

- Compound operations
  - Intersection ($\cap$)
  - Join ($\bowtie$)
  - Division (/)

Intersection ∩

- Intersection notation: $R \cap S$
- It returns the tuples in both R and S.
- It takes two input relations, which must be ***union-compatible***.
- Q: How to express it using basic operators?

$$R \cap S = R - (R - S)$$

Intersection

Sid	Sname	Rating	Age
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.5

S1

Sid	Sname	Rating	Age
28	Yuppy	9	35.0
31	Lubber	8	55.5
44	Guppy	5	35.0
58	Rusty	10	35.5

S2

Sid	Sname	Rating	Age
31	Lubber	8	55.5
58	Rusty	10	35.5

$$S1 \cap S2$$

Join $\bowtie$

- 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 $\bowtie$ S conceptually is:
  - Step 1: Compute R × S
  - Step 2: Select rows in R × S where attributes appearing in both relations have equal values
  - Step 3: Keep all unique attributes and one copy 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	35.5

S1

$R1 \bowtie S1 =$

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”)*:

$$R \bowtie_{c} S = \sigma_{c}(R \times S)$$

S1.sid	Sname	Rating	Age	R1.sid	Bid	day
22	Dustin	7	45.0	58	103	11/12/96
31	Lubber	8	55.5	58	103	11/12/96

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

- *Result schema* is the same as that of cross-product.
- May have fewer tuples than cross-product.

Equi-Join

- A special case of condition join where condition contains only ***equalities***
- *Result schema:* same as cross-product
- What’s the difference between equi-join and natural-join?

A	B	C
a1	b1	c1
a1	b2	c2

R

A	B	D
a1	b1	d1
a1	b2	d2

S

$\mathbf{R} \bowtie \mathbf{S} = ?$

$\mathbf{R} \bowtie_{\mathbf{R.A=S.A}} \mathbf{S} = ?$

Division

- Notation: ***A*/B** ***or A*÷B**
  - Useful for expressing "for all" queries like: *Find sids of sailors who have reserved all boats*.
  - E.g., let *A* have 2 fields, *x* and *y*; *B* have only field *y*:

A/B **contains all tuples (x) such that for *every y* tuple in *B*, there is an <*x,y*> tuple in *A*.**

Division

- Attributes of B must be a subset of attributes of A.
  - Given A <x1…xn>, B<x1…xk>
  - <x1…xk> must be a SUBSET of <x1…xn>
  - The schema of A/B contains <x1…xn> - <x1…xk>
    - E.g., A (SID, PID, grade), B (PID)
    - The schema of A/B is (SID, grade)

- **Question:**
  - Consider A (SID, Name, Age), B(SID, Address)
  - Is A/B allowed?

Examples of Division A/B

Sno	pno
S1	P1
S1	P2
S1	P3
S1	P4
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4

A

Pno
P2

B1

Pno
P2
P4

B2

Pno
P1
P2
P4

B3

Sno
S1
S2
S3
S4

A/B1

Sno
S1
S4

A/B2

Sno
S1

A/B3

Rename (ρ (rho))

- Rename is NOT a compound operation
- Allows us to name results of relational-algebra expressions.
- Notation: ρ ***(X*, *E*)**
  - Returns the expression *E* under the name *X*
- Rename the results of an expression, e.g.,

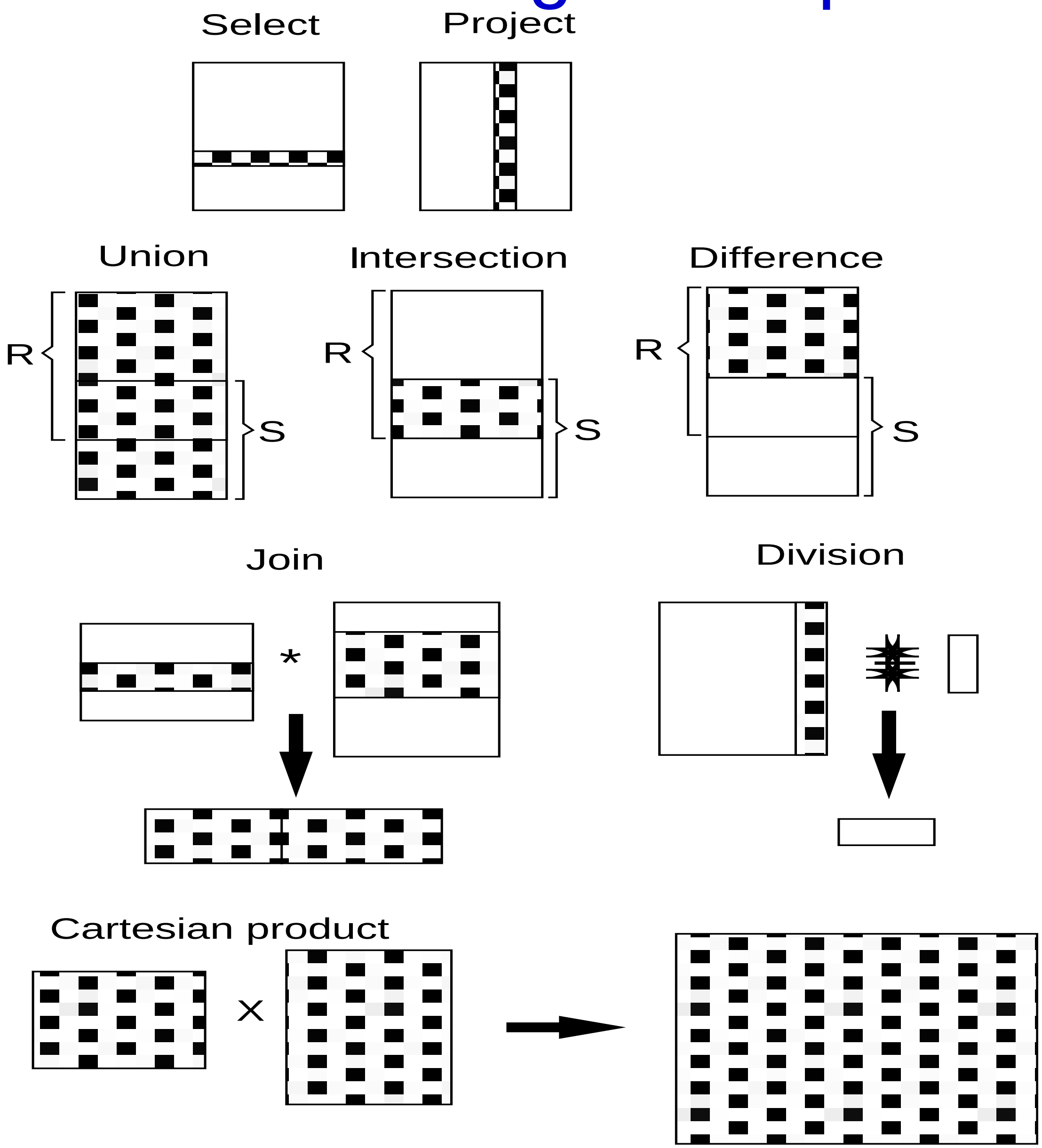
  $\rho(OldSailor, \sigma_{age>60}(Sailor))$
- Use the renamed X as a new relation

  $\rho(OldSailor, \sigma_{age>60}(Sailor)$

  $\pi_{SID}(OldSailor)$

Relational Algebra In-Class Exercise (I)

Summary of Relational Algebra Operations



Set operations (I)

- **Given the schema**
  - – Patients(pnum, pname, age)
  - – Doctors(dnum, dname, rank)
  - – Visits(pnum, dnum, dates, diagnosis)
- **Restate the expression** $\sigma_{age<30 \cup age>50}(Patients)$ **using set operations (i.e., union, set-difference, intersection)**
- ***The way of thinking:***
  - – find the set A of patients with age >50;
  - – find the set B of patients with age <30;
  - – take A union B.
- **Answer (take notes)**

Set operations (II)

- Given the schema
  - Patients(pnum, pname, age)
  - Doctors(dnum, dname, rank)
  - Visits(pnum, dnum, dates, diagnosis)
- Restate the expression

  $$\sigma_{rank \neq 'surgeon' \cap rank \neq 'oculist'}(Doctors)$$

  using set operations
- ***Solution 1***
  - find the set A of non-surgeons;
  - find the set B of non-oculists;
  - Take A intersects B.
- **Answer (take notes)**

Set operations (II)

- Given the schema
  - Patients(pnum, pname, age)
  - Doctors(dnum, dname, rank)
  - Visits(pnum, dnum, dates, diagnosis)
- Restate the expression

$$\sigma_{rank \neq 'surgeon' \cap rank \neq 'oculist'}(doctors)$$

  using set operations

- ***Solution 2***
  - find the set A of oculists;
  - find the set B of surgeon;
  - find the set C of all doctors;
  - take C – (A union B).
- **Answer (take notes)**

Cross-product, Natural-Join and Conditional Join

A	B	C
1	2	3
6	7	8
9	7	8

Relation U

B	C	D
2	3	4
2	3	5
7	8	10

Relation V

Questions:

1.How many tuples and attribute are in $U \times V$?

2.What are the results of $U \bowtie V$ and $U \bowtie_{A<D} V$?

Example

A	B	C	D
1	2	3	4
1	2	3	5
6	7	8	10
9	7	8	10

Result $U \bowtie V$

A	U.B	U.C	V.B	V.C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

Result of $U \bowtie_{A<D} V$