INFO20003 Database Systems

Dr Renata Borovica-Gajic

Lecture 07
Relational Algebra

Coverage

- Relational Query Languages
- Selection & Projection
- Union, Set Difference & Intersection
- Cross product & Joins
- Examples

Readings: Chapter 4, Ramakrishnan & Gehrke, Database Systems

Relational Query Languages

- **Query languages:** Allow manipulation and retrieval of data from a database.
- Relational model supports simple, powerful QLs:
  - Strong formal foundation based on logic
  - Allows for optimization
- Query Languages != programming languages
  - QLs not intended to be used for complex calculations
  - QLs support easy, efficient access to large data sets

Formal Relational Query Languages

Two mathematical Query Languages form the basis for "real" languages (e.g. SQL), and for implementation:

- *Relational Algebra*: More operational, very useful for representing execution plans.
- *Relational Calculus*: Lets users describe what they want, rather than how to compute it. (Non-procedural, *declarative*.)

* *Understanding Algebra & Calculus is key to understanding SQL and query processing*

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.

Since each operation returns a relation, operations can be *composed* (Algebra is "closed".)

Example Instances

R1

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

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

- Relational Query Languages
- Selection & Projection
- Union, Set Difference & Intersection
- Cross product & Joins
- Examples

Readings: Chapter 4, Ramakrishnan & Gehrke, Database Systems

Projection

- Retains only attributes that are in the *"projection list"*.
- Examples: $\pi_{age}(S2)$; $\pi_{sname,rating}(S2)$

- *Schema* of result:
  - exactly the fields 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.

Projection (π)

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

$$\pi_{age}(S2)$$

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

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

$\Rightarrow$

sid	sname	rating	age
28	yuppy	9	35.0
58	rusty	10	35.0

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

Selection & Projection

Selects rows that satisfy *selection condition*
& retain only *certain attributes (columns)*

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

sname	rating
yuppy	9
rusty	10

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

Relational Algebra

- Relational Query Languages
- Selection & Projection
- Union, Set Difference & Intersection
- Cross product & Joins
- Examples

Readings: Chapter 4, Ramakrishnan & Gehrke, Database Systems

Union and Set-Difference

- All of these operations take two input relations, which must be *union-compatible*:
  - Same number of fields.
  - **Corresponding** fields have the same type.

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

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

S1

<u>sid</u>	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

$S1 - S2$

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
22	dustin	7	45.0

$S1 - S2$

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

S2

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

$S2 - S1$

Compound Operator: Intersection

- In addition to the 5 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?

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

Intersection

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

S1

<u>sid</u>	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

$$S1 \cap S2$$

- Relational Query Languages
- Selection & Projection
- Union, Set Difference & Intersection
- Cross product & Joins
- Examples

Readings: Chapter 4, Ramakrishnan & Gehrke, Database Systems

Cross-Product

- S1 x R1: Each row of S1 paired with each row 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*:

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

Cross 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

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:
  1. Compute R X S
  2. Select rows where attributes that appear in both relations have equal values
  3. Project all unique attributes and one copy of each of the common ones.

Natural Join 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 $\bowtie$ 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

Natural Join Example

1

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

Natural Join Example

1

S1 X R1 =

2

σ

(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

Natural Join Example

1

S1 X R1 =

2

σ

(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

π 3

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)**:

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

(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

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

  – **Result schema** same as that of cross-product.

- **Equi-Join:** Special case, condition *c* contains only conjunction of *equalities*.

Relational Algebra

- Relational Query Languages
- Selection & Projection
- Union, Set Difference & Intersection
- Cross product & Joins
- Examples

Readings: Chapter 4, Ramakrishnan & Gehrke, Database Systems

Let's try it...

Boats

bid	bname	color
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	Red

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

Question 1

Find names of sailors who have reserved boat #12

Solution 1: $\pi_{sname}((\sigma_{bid=12}\text{Reserves}) \bowtie Sailors)$

Solution 2: $\pi_{sname}(\sigma_{bid=12}(\text{Reserves} \bowtie Sailors))$

Question 2

Find names of sailors who have reserved a blue boat

- Information about boat color only available in Boats; so need an extra join:

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

- A more efficient solution:

$$\pi_{sname}(\pi_{sid}((\pi_{bid}\sigma_{color='blue'}Boats) \bowtie \text{Res.}) \bowtie Sailors)$$

* *A query optimizer can find this given the first solution!*

Your turn …

Find all pairs of sailors in which the older sailor has a lower rating

Practice at home

1. Find (the name of) all sailors whose rating is above 9

2. Find all sailors who reserved a boat prior to November 1, 1996

3. Find (the names of) all boats that have been reserved at least once

4. Find all pairs of sailors with the same rating

Summary

- You have learned about relational algebra and calculus
- This is important for writing SQL statements and to understand query processing!

What’s Examinable

- Relational Algebra Operations: Selection, Projection, Union, Set, Difference, Intersection, **JOINS**…
- Draw different queries with Relational Algebra operations

Next Lecture

- Introducing SQL queries