Relational Algebra

R & G, Chapter 4

Databases: the continuing saga

- When last we left databases…
  - We learned how to conceptually model them using ER diagrams
  - We learned how to logically model them using relational schemas
  - We learned how to translate ER diagrams to relational schemas

We're almost ready to use SQL to query it, but first…

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*.)**

✉We only cover Relational Algebra in CS442

Preliminaries

- **A query is applied to *relation instances*, and the result of a query is also a relation instance.**
  - *Schemas* of input relations for a query are fixed
  - The schema for the *result* of a given query is also fixed.
    - It is determined by the definitions of the query language constructs.

Relational Algebra: 5 Basic Operations

1. *Selection* (σ) : Selects a subset of ***rows*** from relation (horizontal).
2. *Projection* (π) : Retains only wanted ***columns*** from relation (vertical).
3. *Cross-product* ($\times$): Combine two relations.
4. *Set-difference* (—): Return records in r1, but not in r2.
5. *Union* ($\cup$): Return records in r1 and/or in r2.

Since each operation returns a relation, operations can be *composed!*

Example Instances

R1

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

S1

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

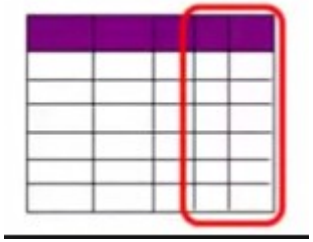
S2

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

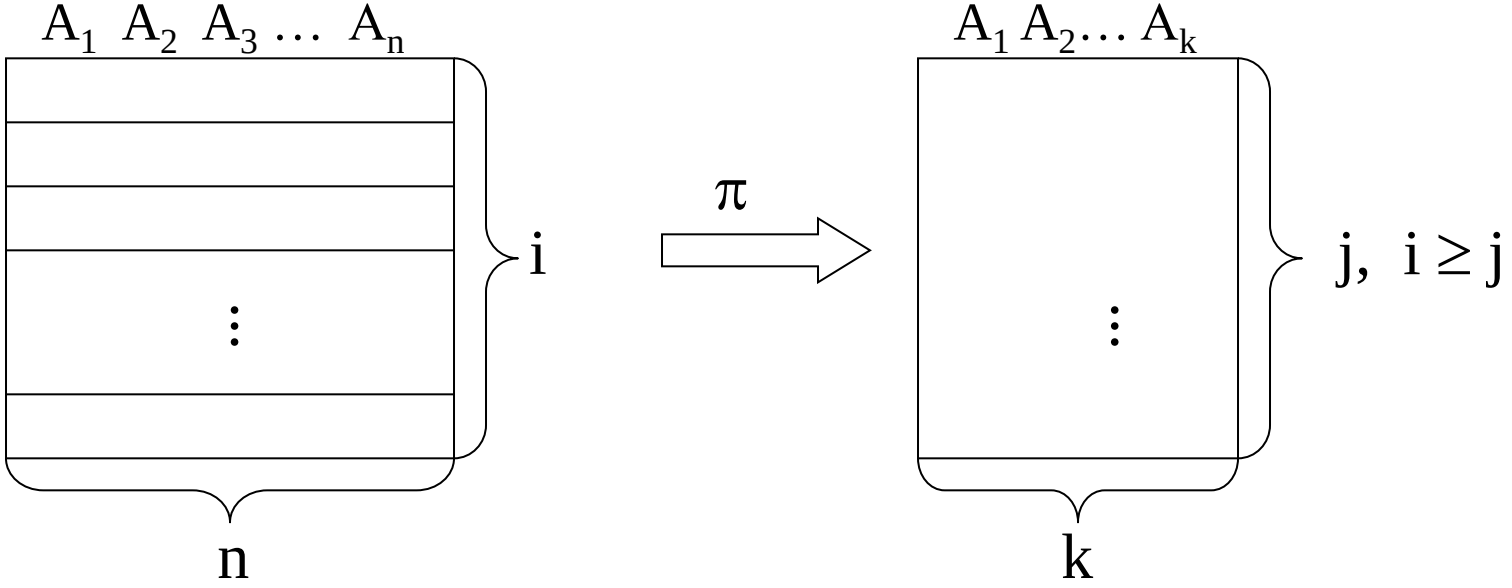
Boats

Bid	Bname	Color
101	Interlake	Blue
102	Interlate	Red
103	Clipper	Green
104	Marine	red

Projection (π)



- Notation: $\pi_{A1,A2\ldots Ak}(r)$
  - where *A1, …, Ak* are attributes (the projection list) and *r* is a relation.
- Retains attributes *A1, …, Ak* that are in the "*projection list*".



- Examples: $\pi_{age}(S2)$; $\pi_{sname,rating}(S2)$

Projection (π)

- *Schema* of result of $\pi_{A1,A2\ldots Ak}(r)$
  - Contains (A1, ...Ak)

- 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

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

Sname	Rating
Yuppy	9
Lubber	8
Guppy	5
Rusty	10

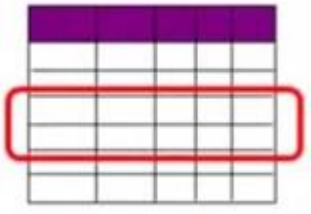
$$\pi_{sname,rating}(S2)$$

Age
35.0
55.5

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

(after duplicates elimination)

Selection (σ)



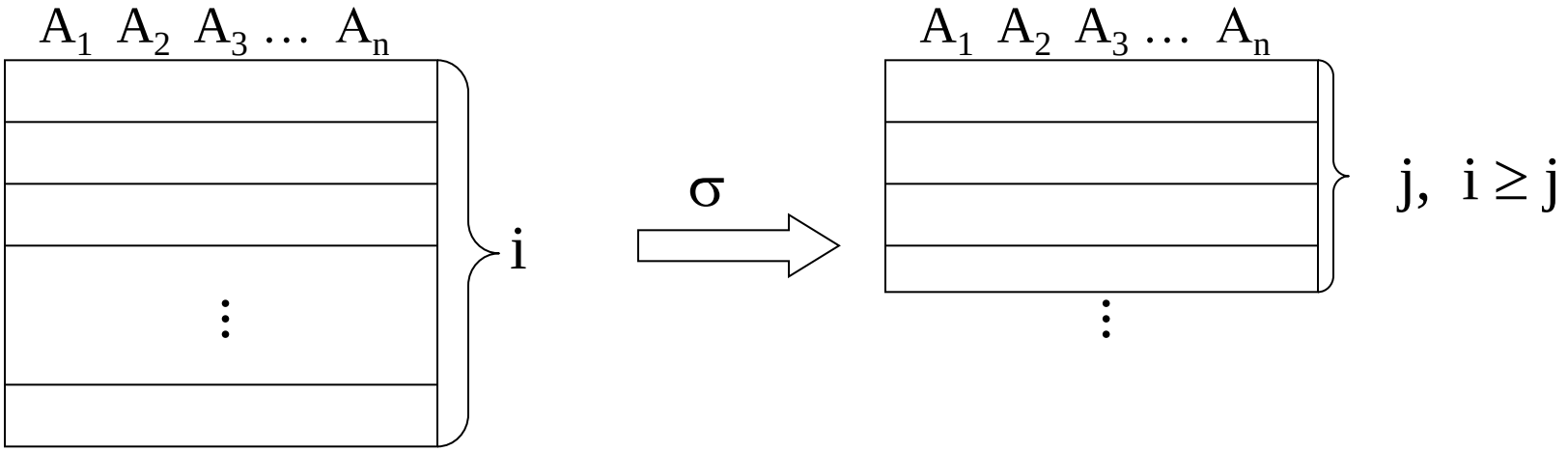
- Notation: $\sigma_{p}(r)$
- Selects rows that satisfy *selection condition p*.
- Selection condition ***p***
  - consisting of: **connectives** : ∧ (and), ∨ (or), ¬ (not)

    And **predicates:**

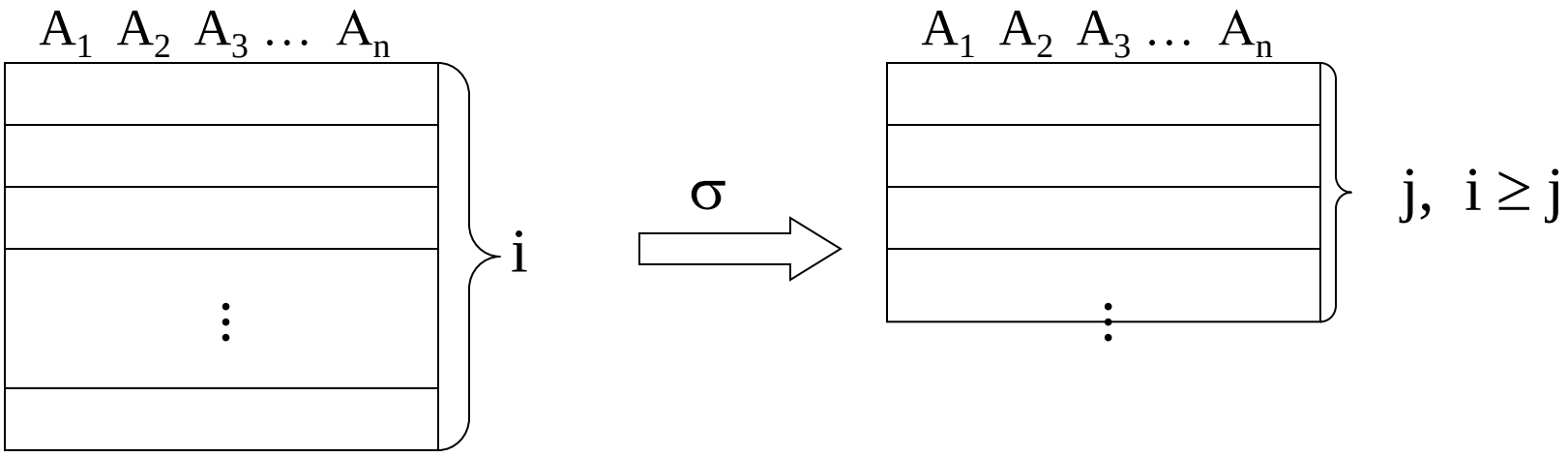
    <attribute> *op* <attribute> or

    <attribute> *op* <constant>

    •op: *<, <=, =, ≠, >=, >*



Selection (σ)



- Selection result is a relation.

  Schema of result is same as that of the input relation.

- Do we need to do duplicate elimination?

Selection (σ)

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

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

Selection (σ) and Projection (π)

- Order rule:
  - Projection is always applied AFTER selection.
  - Format:

$$\pi_{A1,A2\ldots Ak}\,\sigma_{p}(r)$$

Example of Selection (σ) and Projection (π)

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

Sname	Rating
Yuppy	9
Rusty	10

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

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

Example (I)

Bid	Bname	Color
101	Interlake	Blue
102	Interlate	Red
103	Clipper	Green
104	Marine	red

$Boats$

- Find the color of the boat named "Marine"

$$\pi_{color}(\sigma_{bname='Marine'}(Boats))$$

Example (II)

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

S1

- Find the name of the sailor whose rating is at least 8

$$\pi_{name}(\sigma_{rating>=8}(S1))$$

Union and Set-Difference

- All of these operations take two input relations, which must be ***union-compatible***:
  - Same schema:
    - The same number of attributes;
    - `Corresponding' attributes have the same type.

Union

- **Notation:** $R \cup S$
- Returns a relation instance containing all tuples that in either R or S (or both)

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
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.5
28	Yuppy	9	35.0
44	Guppy	5	35.0

$S1 \cup S2$

Set Difference

- **Notation: R - S**
- returns a relation instance containing all tuples that in R but not S.

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
22	Dustin	7	45.0

$S1 - S2$

Sid	Sname	Rating	Age
28	Yuppy	9	35.0
44	Guppy	5	35.0

$S2 - S1$

Q: Do we need to eliminate duplicate tuples from the set difference result?

Cross-Product

- Notation: **S** × **R**
- Each row of S paired with each row of R
- Q: How many rows in the result?
- *Result schema* has one field per field of S and R, with field names `inherited' if possible.
  - *May have a naming conflict*: Both S and R have a field with the same name.
  - In this case, can use the *renaming operator*:

$$\rho(C(S.sid \to sid1, R.sid \to sid2), S \times R)$$

  - Q: How many attributes in the result?

Cross Product Example

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

R

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

S

R X S =

(S1.sid)	sname	rating	age	(R1.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