Query Processing and Relational Algebra 2
Relational Algebra Operators

- Select: \( \sigma_{\text{condition}} \)
- Project: \( \pi_{\text{attr list}} \)
- Union: \( \cup \)
- Set Difference: \( - \)
- Intersection: \( \cap \)
- Cartesian product: \( \times \)
Relational Algebra Operators

• Joins
  – Natural join
    • \( \bowtie \)
  – Equi join and Theta join
    • \( \bowtie_{\text{Condition}} \)

• Division
  – \( \div \) or /

• Renaming
  – Expression\([A_1, A_2, ...A_n]\)
Problems

- Library L
- Copy C
- Book B
- Writes W
- Author A
Problems

• Find libnums of libraries with a capacity greater than 200.
  \[ \pi \text{ libnum} (\sigma_{\text{capacity} > 200 \ L}) \]
Problems

• Find the titles of books with copies housed in a library with a capacity greater than 200.
  \[ \pi \text{title} \left( \sigma_{\text{capacity} > 200} \left( B \bowtie C \bowtie L \right) \right) \]
  \[ \pi \text{title} \left( B \bowtie C \bowtie \left( \sigma_{\text{capacity} > 200} L \right) \right) \]
Problems

• Find the names of authors who have written a book housed in a library with a capacity greater than 200
  \[ \pi \text{ first, last} \left( \sigma_{\text{capacity} > 200} \left( A \bowtie W \bowtie B \bowtie C \bowtie L \right) \right) \]
Problems

• Find aid and name of authors who have not written any books
  \[ \pi \text{ first, last } (A \Join (((\pi \text{ aid } A) - (\pi \text{ aid } W)))) \]
Problems

• Find booknum and title of books with no copies.
  \[ \pi \text{ booknum, title} (B \bowtie ((\pi \text{ booknum } B) - (\pi \text{ booknum } C))) \]
Problems

• Find the booknum and title of books with a copy in every library
  \[ (\pi \text{booknum, title, libnum} (B \bowtie C)) \div (\pi \text{libnum} L) \]
More Relational Algebra Problems

Suppose relations $R$ and $S$ contain $\text{Size}(R)$ and $\text{Size}(S)$ tuples. What are the minimum and maximum number of tuples in the results of relational algebra expression shown to the right (assume union compatibility where needed)?

- $R \cup S$
- $R \cap S$
- $R - S$
- $\pi_A R$ where $A$ is an attribute of $R$
- $R \times S$
- $R \bowtie S$ where $A$ is the common attribute in $R$ and $S$
- $R / S$ assume all attributes of $S$ are also attributes of $R$
Result Size

• R U S

• Max: Size(R) + Size(S)

• Min: greater of Size(R) and Size(S)
Result Size

- $R \cap S$

- Max: Smaller of Size($R$) and Size($S$)

- Min: 0
Result Size

- $R - S$
- Max: $\text{Size}(R)$
- Min: 0
- If everything in $S$ is in $R$ the $\text{Size}(R) - \text{Size}(S)$
Result Size

• $\pi_A R$

• Max: Size(R)

• Min: 1
Result Size

- $R \times S$
- $\text{Size}(R) \times \text{Size}(S)$
Result Size

- \( R \bowtie S \) where \( A \) is the common attribute in \( R \) and \( S \)

- Max: \( \text{Size}(R) \times \text{Size}(S) \)

- Min: 0

- If \( A \) is the primary key in \( R \) and a foreign key in \( S \) and no \( A \)'s in \( S \) are NULL then Max is \( \text{Size}(S) \) (Note correction mentioned in video)
Result Size

- **R / S** assume all attributes of S are also attributes of R
- Max: \( \text{Size}(R) / \text{Size}(S) \)
- Min: 0