Query Processing and Relational Algebra 2

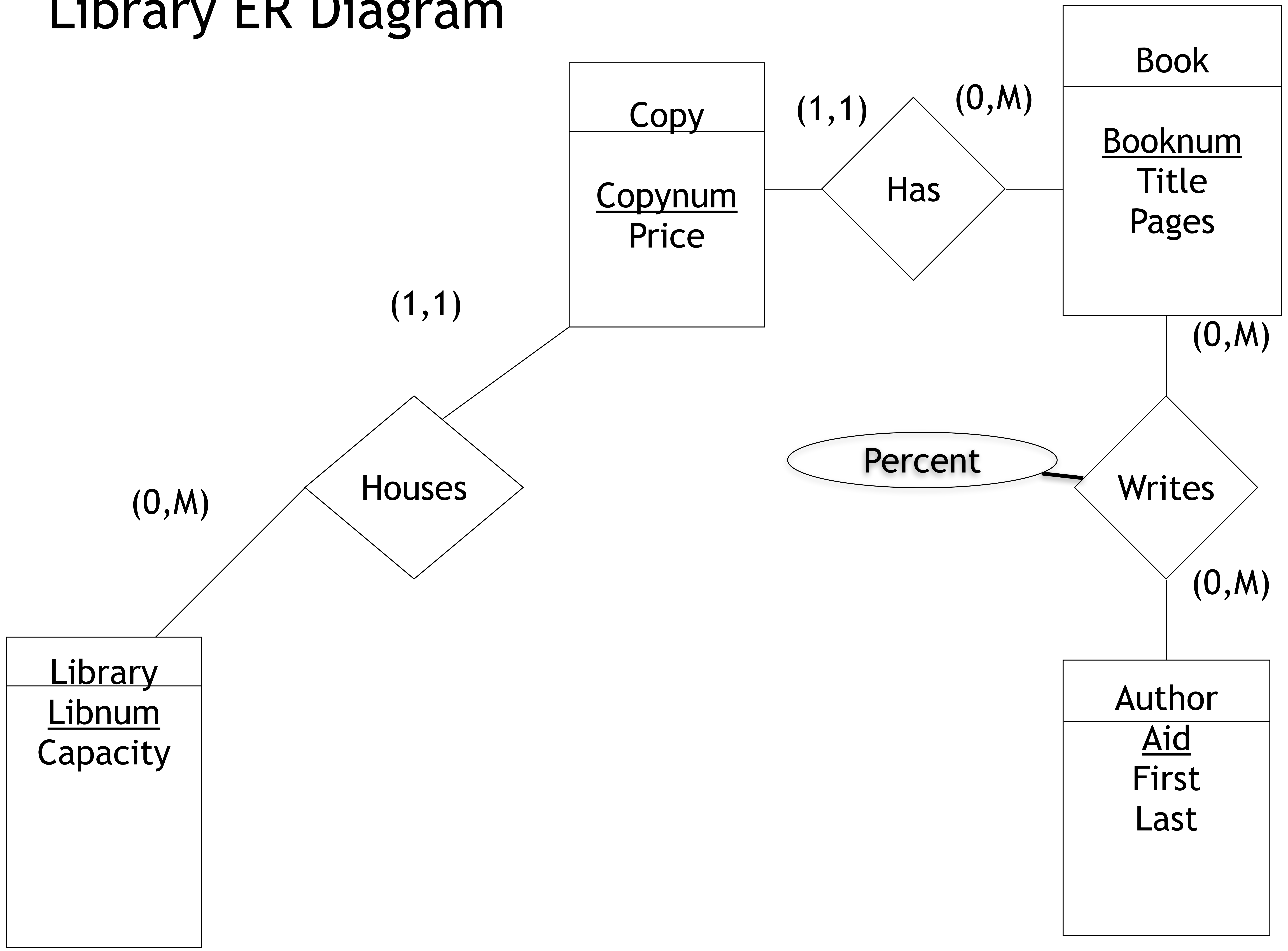
Relational Algebra Operators

- 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
    - $\bowtie_{Condition}$
- Division
  - $\div$ or $/$
- Renaming
  - Expression[A_1, A_2, …A_n]

Library ER Diagram



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.
  - $\pi_{\text{title}} (\sigma_{\text{capacity} > 200} (B \bowtie C \bowtie L))$
  - $\pi_{\text{title}} (B \bowtie C \bowtie (\sigma_{\text{capacity} > 200} L))$

Problems

- Find the names of authors who have written a book housed in a library with a capacity greater than 200
  - $\pi_{\text{first, last}} (\sigma_{\text{capacity} > 200} (A \bowtie W \bowtie B \bowtie C \bowtie L))$

Problems

- Find aid and name of authors who have not written any books
  - $\pi_{\text{first, last}} (A \bowtie ((\pi_{\text{aid}} A) - (\pi_{\text{aid}} W)))$

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
  - $(\pi_{\text{booknum, title, libnum}} (B \bowtie C)) \ / \ (\pi_{\text{libnum}} L)$

More Relational Algebra Problems

Suppose relations R and S contain Size(R) and 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 U S
- R ∩ S
- R – S
- $\pi_A R$ where A is an attribute of R
- R x S
- R ⋈ 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 ∩ S**
- Max: Smaller of Size(R) and Size(S)
- Min: 0

Result Size

- **R - S**
- Max: Size(R)
- Min: 0
- If everything in S is in R the Size(R)-Size(S)

Result Size

- $\pi_A R$
- Max: Size(R)
- Min: 1

Result Size

- **R x S**
- Size(R) * Size(S)

Result Size

- **R** ⋈ **S** where A is the common attribute in R and S
- Max: Size(R) * 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 Size(S) (Note correction mentioned in video)

Result Size

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