ICS 321 Spring 2012
Algebraic and Logical Query Languages

Asst. Prof. Lipyeow Lim

Information & Computer Science Department

University of Hawaii at Manoa

Relational Algebra Review

- Relations are **<u>sets</u>** of tuples – no duplicates allowed
- Basic operations:
    - *<u>Selection</u>* (σ)   Selects a subset of rows from relation.
    - *<u>Projection</u>* (π)   Deletes unwanted columns from relation.
    - *<u>Cross-product</u>* (✕) Allows us to combine two relations.
    - *<u>Set-difference</u>* (−) Tuples in reln. 1, but not in reln. 2.
    - *<u>Union</u>* (∪) Tuples in reln. 1 and in reln. 2.
- Additional operations:
    - Intersection, *<u>join</u>*, division, renaming: Not essential, but (very!) useful.
- Each operation returns a relation, operations can be *composed*! (Algebra is "closed".)

Bag Semantics

- Commercial DBMS implements relations as bags

- Avoid duplicate elimination

- Support aggregations

Bag

A	B
1	2
3	4
1	2
1	2

Set

A	B
1	2
3	4
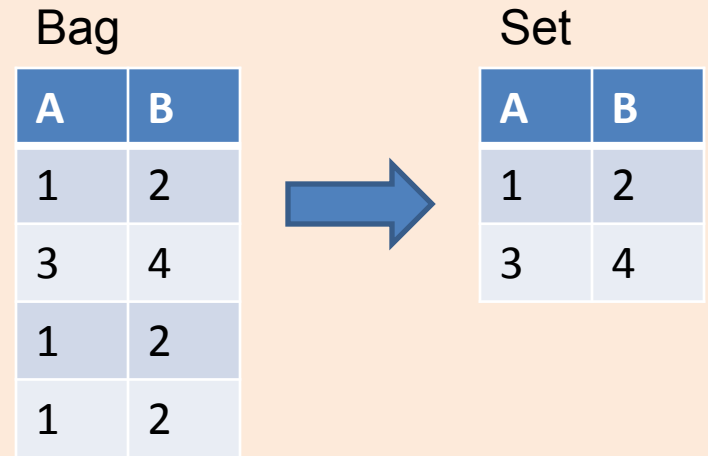
Can relational algebra work with bags ?

Selection & Projection

- Expected behavior
- No duplicate elimination of results

$$\pi_{age} (\sigma_{rating>6} (S2))$$

sid	sname	rating	age
28	Yuppy	9	35.0
31	Lubber	8	55.5
44	Guppy	5	35.0
58	Rusty	10	35.0

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

sid	sname	rating	age
28	Yuppy	9	35.0
31	Lubber	8	55.5
44	Guppy	5	35.0
58	Rusty	10	35.0

Cross Product & Joins

R

A	B
1	2
1	2

S

B	C
2	3
4	5
4	5

Cross Product

Join

$R \times S$

A	R.B	R.B	C
1	2	2	3
1	2	2	3
1	2	4	5
1	2	4	5
1	2	4	5
1	2	4	5

$R \bowtie S$

A	B	C
1	2	3
1	2	3

Treat duplicates like non-duplicates

Bag Union, Intersection & Difference

R

A	B
1	2
3	4
1	2
1	2

S

A	B
1	2
3	4
3	4
5	6

Bag Union

R ∪ S

A	B
1	2
3	4
1	2
1	2
1	2
3	4
3	4
5	6

Tuple t appears $(n+m)$ times

Bag Intersection

Bag Difference

R ∩ S

A	B
1	2
3	4

R - S

A	B
1	2
1	2

S - R

A	B
3	4
5	6

Tuple t appears $min(n,m)$ times

Tuple t appears $max(0, n-m)$ times

Treat duplicates like non-duplicates
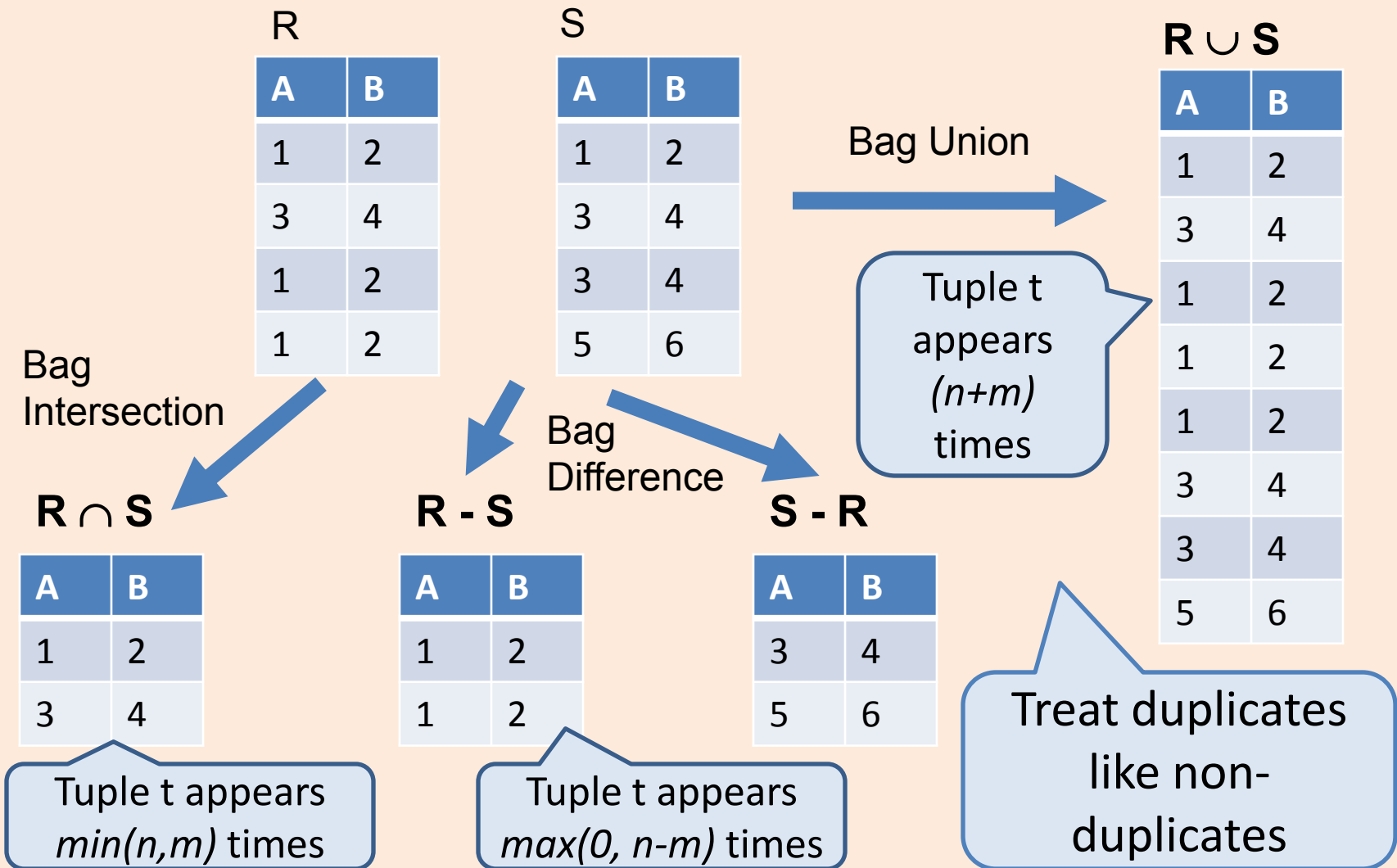
Extended Operators

- Duplicate elimination δ
  - turns a bag into a set
- Aggregation
  - calculates an aggregate (sum, average etc) over the values in a column
- Grouping γ
  - partitions tuples in a relation into groups based on values in some columns
- Extended projection π
  - allow computation on column values to produce new values
- Sorting τ
  - sorts a relation according to the values in some column(s)
- Outer join
  - preserves dangling pointers in the results of joins

Aggregation

- Standard: SUM, AVG, MIN, MAX, COUNT

- DBMS supports more sophisticated functions like Variance, standard deviation etc.

- SUM(B) = 2+4+2+2 = 10

- AVG(A) = (1+3+1+1)/4 = 1.5

- MIN(A) = 1

- MAX(B) = 4

- COUNT(A) = 4

A	B
1	2
3	4
1	2
1	2

Grouping

Movies	Title	Year	Length	Genre	StudioName	producer

- Grouping operator γ
  - Groups tuples by some columns
  - Apply aggregation function to each group
  - Generate a result tuple per group

For each studio, find the total lengths of movies produced

studioName
Disney
Disney
Disney
MGM
MGM
o
o
o

Grouping Operator Arguments

Movies	Title	Year	Length	Genre	StudioName	producer

γstudioName, SUM(length) $\rightarrow$sumOfLengths

Grouping attributes

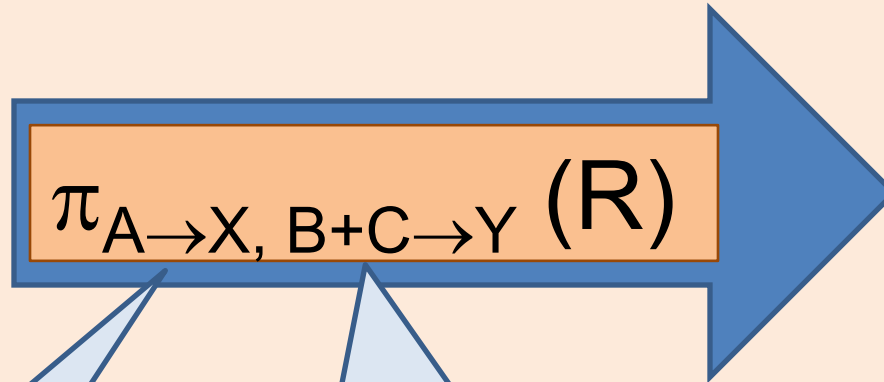Aggregation functions on aggregated attributes with optional renaming

StudioName	SumOfLengths
Disney	12345
MGM	54321
…	…

studioName

Disney
Disney
Disney

MGM
MGM

o
o
o

Extended Projection

A	B	C
0	1	2
0	1	2
3	4	5

$$\pi_{A \rightarrow X, \ B+C \rightarrow Y} \ (R)$$

X	Y
0	3
0	3
3	9

Renaming of attributes

Expressions that compute new values from attributes and naming the new values

Outer Join

R

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

S

B	C	D
2	3	10
2	3	11
6	7	12

$R \bowtie S$

A	B	C	D
1	2	3	10
1	2	3	11

Discard right & left dangling pointers

R ⋈$_L$ S

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	⊥
7	8	9	⊥

Keep left dangling pointers

R ⋈ S

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	⊥
7	8	9	⊥
⊥	6	7	12

R ⋈$_R$ S

A	B	C	D
1	2	3	10
1	2	3	11
⊥	6	7	12

Keep right dangling pointers