

# ICS 321 Fall 2013

# Algebraic and Logical Query Languages

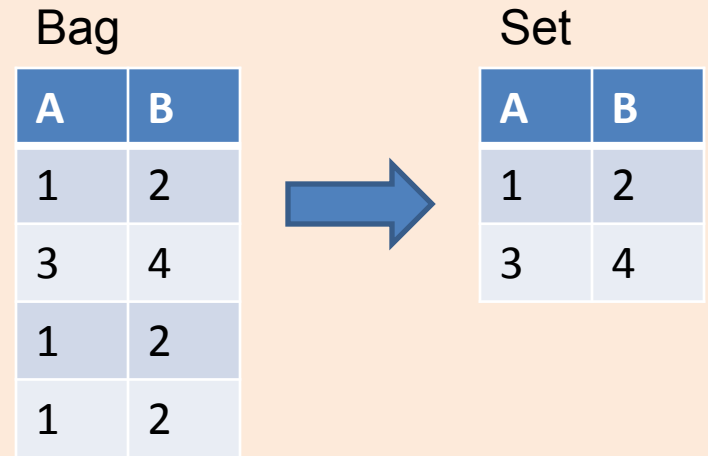
Asst. Prof. Lipyeow Lim  
Information & Computer Science Department  
University of Hawaii at Manoa

# Relational Algebra Review

- Relations are **sets** of tuples – no duplicates allowed
- Basic operations:
  - Selection ( $\sigma$ ) Selects a subset of rows from relation.
  - Projection ( $\pi$ ) Deletes unwanted columns from relation.
  - Cross-product ( $\times$ ) Allows us to combine two relations.
  - Set-difference ( $-$ ) Tuples in reln. 1, but not in reln. 2.
  - Union ( $\cup$ ) Tuples in reln. 1 and in reln. 2.
- Additional operations:
  - Intersection, join, 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



Can relational algebra work with bags ?

# Selection & Projection

- Expected behavior
- No duplicate elimination of results

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

<u>sid</u>	sname	rating	age
23	Yuppy	9	35.0
31	Lubber	8	55.5
44	Guopy	5	35.0
53	Rusty	10	35.0

$\Pi_{\text{age}} (S2)$

<u>sid</u>	sname	rating	age
23	Yuppy	9	35.0
31	Lubber	8	55.5
44	Guopy	5	35.0
53	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

$R \times S$

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

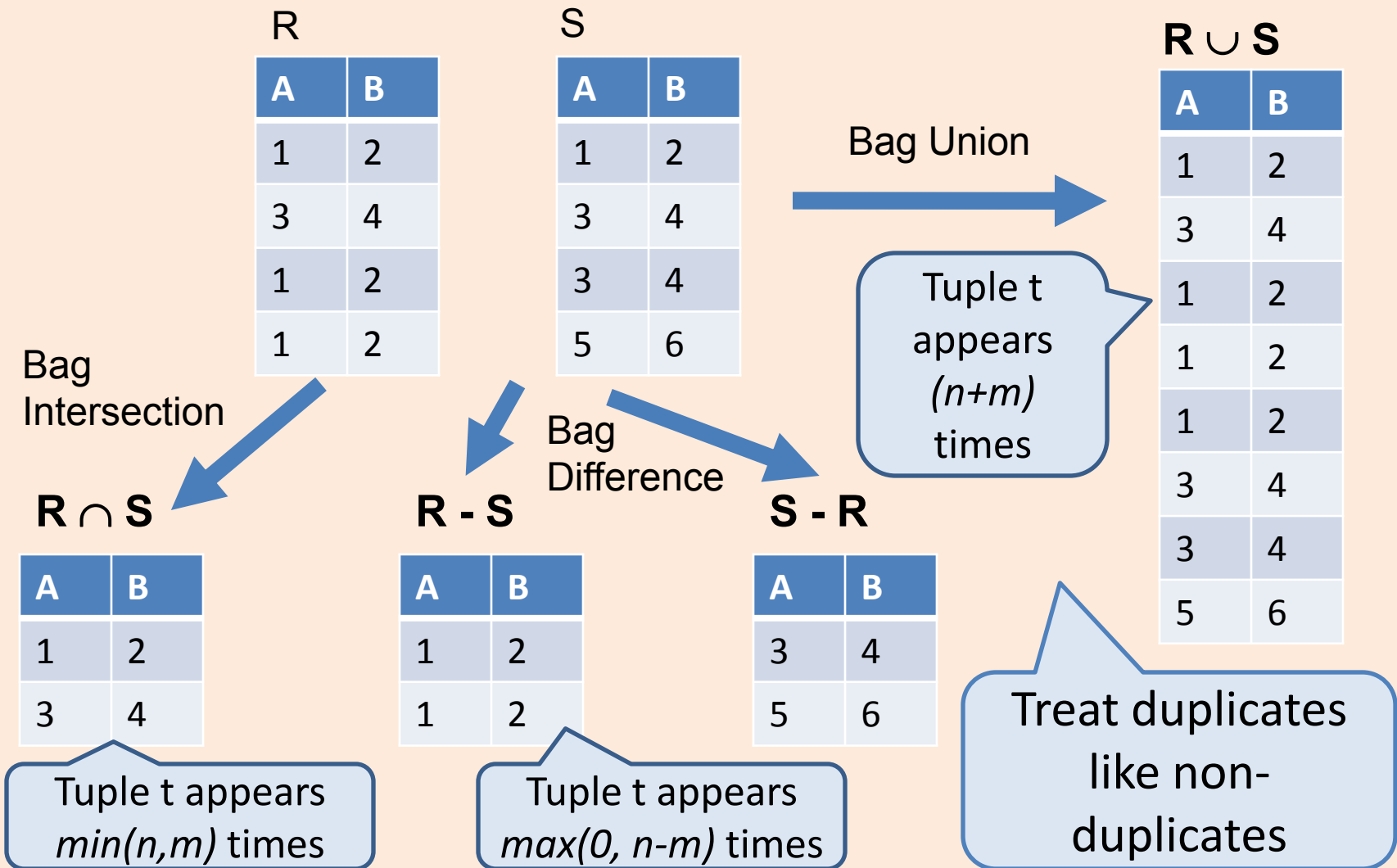
Join

$R \bowtie S$

A	B	C
1	2	3
1	2	3

Treat duplicates like non-duplicates

# Bag Union, Intersection & Difference



# Extended Operators

- Duplicate elimination  $\delta$ 
  - turns a bag into a set
- Aggregation
  - calculates an aggregate (sum, average etc) over the values in a column
- Grouping  $\gamma$ 
  - partitions tuples in a relation into groups based on values in some columns
- Extended projection  $\pi$ 
  - allow computation on column values to produce new values
- Sorting  $\tau$ 
  - 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  $\gamma$ 
  - 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

	<i>studioName</i>	
	Disney	
	Disney	
	Disney	
	MGM	
	MGM	
	○	
	○	
	○	

# Grouping Operator Arguments

Movies

Title	Year	Length	Genre	StudioName	producer
-------	------	--------	-------	------------	----------

$\gamma_{\text{studioName}, \text{SUM}(\text{length}) \rightarrow \text{sumOfLengths}}$

Grouping  
attributes

Aggregation functions on  
aggregated attributes  
with optional renaming

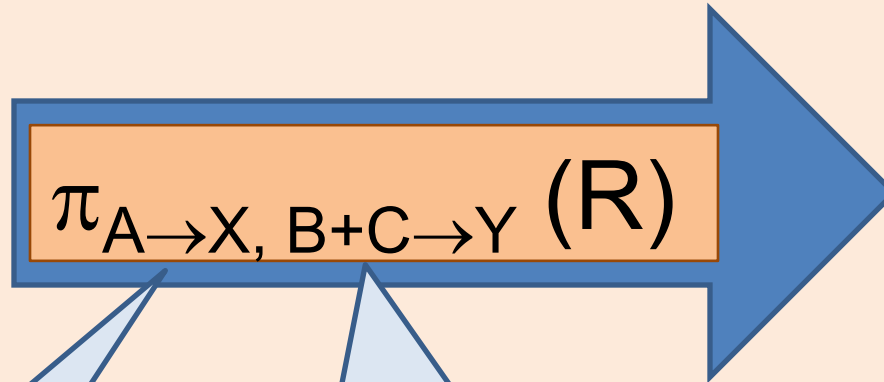
StudioName	SumOfLengths
Disney	12345
MGM	54321
...	...

<i>studioName</i>	
Disney	
Disney	
Disney	
MGM	
MGM	
○	
○	
○	

© CourseSmart

# Extended Projection

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



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

$R \bowtie_L S$

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

Keep left dangling pointers

$R \bowtie_S S$

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

$R \bowtie_R S$

A	B	C	D
1	2	3	10
1	2	3	11
⊥	6	7	12

Keep right dangling pointers