











Course Name: Database Management Systems

Lecture 9 Topics to be covered













Relational Algebra





- Basic operations:
 - \bullet <u>Selection</u> (\mathcal{O}) Selects a subset of rows from relation.
 - \bullet <u>Projection</u> (\mathcal{T}) Deletes unwanted columns from relation.
 - \bullet <u>Cross-product</u> (\times) Allows us to combine two relations.
 - <u>Set-difference</u> (—) Tuples in reln. 1, but not in reln. 2.
 - **o** <u>Union</u> (\bigcup) Tuples in reln. 1 and in reln. 2.
- Additional operations:
 - Intersection, join, division, renaming: Not essential, but (very!) useful.
- Since each operation returns a relation, operations can be composed! (Algebra is "closed".)









Projection

- Deletes attributes that are not in *projection list*.
- Schema of result contains exactly the fields in the projection list, with the same names that they had in the (only) input relation.
- Projection operator has to eliminate duplicates! (Why??)
 - Note: real systems typically don't do duplicate elimination unless the user explicitly asks for it. (Why not?)

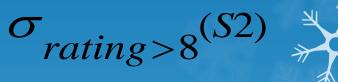
		9	/ 1/2	
snam	e	rating	g *	
yupp	y	9	F	****
lubbe	r	8		
gupp	y	5	<u> </u>	
rusty		10	*	*
T Snai	ne,	ratin	g(Ŝ2	
	ag	e	1	(N
	35.	.0		
	55.	.5	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	*
T	τ_{ag}	$e^{(S2)}$	》	K K



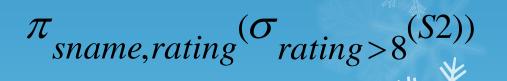
Selection

- Selects rows that satisfy selection condition.
- No duplicates in result! (Why?)
- Schema of result identical to schema of (only) input relation.
- Result relation can be the input for another relational algebra operation! (Operator composition.)

			**
sid	sname	rating	age
28	yuppy	9	35.0
58	rusty	10	35.0



sname	rating
yuppy	9
rusty	10









Union, Intersection, Set-Difference

- All of these operations take two input relations, which must be <u>union-compatible</u>:
 - Same number of fields.
 - Corresponding' fields have the same type.

sid	sname	rating	age
22	dustin	7	45.0

~4	~~
\mathbf{C}	C
,) -	— 、) / .

			<u>-71 k</u>
sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0



sid	sname	rating	age
31	lubber	8	55.5
58	rusty	10	35.0





Cross-Product

Each row of S1 is paired with each row of R1.

o Conflict: Both S1 and R1 have a field called sid.

(sid)	sname	rating	age	(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

■ Renaming operator: $\rho(C(1 \rightarrow sid1, 5 \rightarrow sid2), S1 \times R1)$



Joins





$$R \bowtie_{c} S = \sigma_{c} (R \times S)$$

Condition Join:

(sid)	sname	rating	age	` ′		2	
22	dustin	7	45.0	58	103	11/12/96*	IN
31	lubber	8	55.5	58	103	11/12/96	

$$S1 \bowtie_{S1.sid} < R1.sid$$



• Fewer tuples than cross-product, might be able to compute more efficiently

Sometimes called a theta-join.



Joins



• <u>Equi-Join</u>: A special case of condition join where the condition *c* contains only **equalities**.

sid	sname	rating	age	bid	day
22	dustin	7	45.0	101	10/10/96
58	rusty	10	35.0	103	11/12/96

$$S1 \bowtie_{sid} R1$$

- O Result schema similar to cross-product, but only one copy of fields for which equality is specified
- Natural Join: Equijoin on all common fields.









Division



- Not supported as a primitive operator, but useful for expressing queries like: Find sailors who have reserved by boats.
- Let A have 2 fields, x and y; B have only field y:
 $A/B = \{\langle x \rangle | \exists \langle x, y \rangle \in A \ \forall \langle y \rangle \in B\}$
 - i.e., A/B contains all x tuples (sailors) such that/for every y tuple (boat) in B, there is an xy tuple in A.
 - o Or: If the set of y values (boats) associated with an x value (sailor) in A contains all y values in B, the x value is in A/B.
- In general, x and y can be any lists of fields; y is the list of fields in B, and x y is the list of fields of A.



Examples of Division A/B

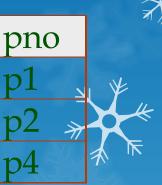
sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

p2	lO
Е	
1	
sno)
s1	
s2	
s3	
s4	

A/B1

p	2 4	
<i>B</i> 2		
	sno	
	s1 s4	
	s4	
A/B2		







sno s1





Expressing A/B Using Basic Operators



- Division is not essential op; just a useful shorthand.
 - (Also true of joins, but joins are so common that systems implement joins specially.)
- Idea: For A/B, compute all x values that are not `disqualified' by some y value in B.



• x value is disqualified if by attaching y value from B, we obtain an xy tuple that is not in A.

Disqualified *x* values

$$\pi_{\chi}((\pi_{\chi}(A)\times B)-A)$$

squalified
$$x$$
 values $\pi_{\chi}((\pi_{\chi}(A) \times B) - A)$

A/B: $\pi_{\chi}(A)$ — all disqualified tuples





Find names of sailors who've reserved boat #103



* Solution 2: ρ (Templ, $\sigma_{bid=103}$ Reserves)

 ρ (Temp2, Temp1 \bowtie Sailors)

 π_{sname} (Temp2)

• Solution 1:

* Solution 3: $\pi_{sname}(\sigma_{bid=103}(\text{Reserves} \bowtie Sailors))$

Find names of sailors who've reserved a red boat



 $\pi_{sname}((\sigma_{color='red'}Boats) \bowtie Reserves \bowtie Sailors)^{\uparrow}$

A more efficient solution:
Information about boat color only available in Boats; so
need an extra join: $\pi_{sname}^{(\pi_{sid}(\pi_{bid}\sigma_{color}=red)} Boats) \bowtie Res) \bowtie Sailors)$

A query optimizer can find this, given the first solution!



Find sailors who've reserved a red or a green boat



 ρ (Temphoats, (σ color='red' \vee color='green' Boats)

 π_{sname} (Temphoats \bowtie Reserves \bowtie Sailors)

Can identify all red or green boats, then find sailors who've reserved one of these boats:



Can also define Tempboats using union! (How?)

* What happens if V is replaced by ^ in this query



Find sailors who've reserved a red <u>and</u> a green boat



Previous approach won't work! Must identify sailors who've reserved red boats, sailors who've reserved green boats, then find the intersection (note that sid is a key for Sailors):

$$\rho$$
 (Tempred, π_{sid} (($\sigma_{color=red}$ Boats) \bowtie Reserves))

$$\rho \; (\textit{Tempgreen}, \, \pi_{\textit{sid}}((\sigma_{\textit{color} = \textit{green'}} \, \textit{Boats}) \bowtie \mathsf{Reserves}))$$

$$\pi_{sname}((Tempred \cap Tempgreen) \bowtie Sailors)$$



