

Formal Relational Query Languages

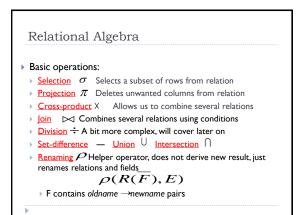
- Two languages form the basis for SQL:
 - Relational Algebra:
 - operational
 - useful for representing execution plans
 - > very relevant as it is used by query optimizers!

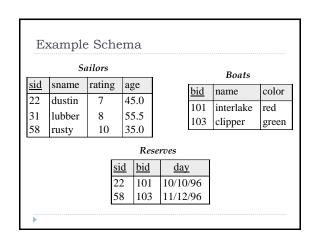
Relational Calculus:

- Lets users describe the result, NOT how to compute it declarative
- We will focus on relational algebra



- A query is applied to relation instances, and the result of a query is also a relation instance
 - Schemas of input relations for a query are fixed
 - The schema for the result of a given query is determined by operand schemas and operator type
- > Each operation returns a relation
 - operations can be composed !
 - Well-formed expression: a relation, or the results of a relational algebra operation on one or two relations





S1			Sı	ailors	<i>S</i> 2				
sid	sname	rating	age	e	<u>sid</u>	snam	e	rating	age
22	dustin	7	45		28	yupp	у	9	35.0
31	lubber	8	55		31	lubbe	r	8	55.5
58	rusty	10	35		44	gupp	y	5	35.0
50	rusty	10	55	.0	58	rusty		10	35.0
		1	R1	Res	erves				
		s	id	bid	da	y			
		2	2	101	10/1	0/96			
		5	8	103	11/1	2/96			

Projection

Unary operator

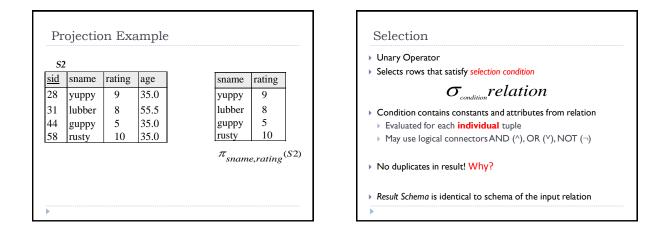
Deletes (projects out) attributes that are not in projection list

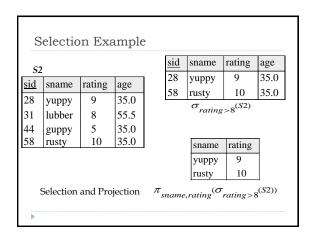
 $\pi_{attr1,attr2,...}$ relation

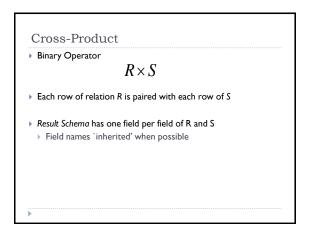
- Result Schema contains the attributes in the projection list
- \blacktriangleright With the same names that they had in the input relation

Projection operator has to eliminate duplicates!

- Real systems typically do not do so by default
- Duplicate elimination is expensive! (sorting)
- User must explicitly asks for duplicate eliminations (DISTINCT)







• •	1	1		_	R	1			1
<u>sid</u>	sname	ratin	° °				sid	bid	<u>day</u>
22	dustin	7	45.0)		1	22	101	10/10/96
31	lubber	8	55.5	5			58	103	11/12/96
58	rusty	10	35.0)			00	100	11/12/20
C =	51 X R1	(sid)	sname	rating	age	(sic	d) t	oid	day
с.	<i>,</i> ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	22	dustin	7	45.0	22	1	101	10/10/96
		22	dustin	7	45.0	58	1	103	11/12/96
		31	lubber	8	55.5	22	1	101	10/10/96
		31	lubber	8	55.5	58	1	103	11/12/96
		58	rusty	10	35.0	22	1	101	10/10/96
		58	rusty	10	35.0	58	1	103	11/12/96

	id1	sname	U	age	sid2	bid	day	ŀ
2		dustin	7	45.0	22	101	10/10/96	
2		dustin	7	45.0	58	103	11/12/96	
3	-	lubber	8	55.5	22	101	10/10/96	
3	1	lubber	8	55.5	58	103	11/12/96	
5	-	rusty	10	35.0	22	101	10/10/96	
5	8	rusty	10	35.0	58	103	11/12/96	
<u>enam</u>	<u>ing</u>	<u>operato</u>	<u>ν</u> ρ(C(1-	<i>sid</i>	l,5→.	sid2),S1>	< <i>R</i> 1

Condition Join (Theta-join) $R \bowtie_{\theta} S = \sigma_{\theta}(R \times S)$ • Result Schema same as that of cross-product

X R1	sid1	sname	rating	age	sid2	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
		S	$S \bowtie_{S}$	1.sid	< <u>R1.s</u>		
	sid1	sname	rating	age	sid2	bid	day
	22	dustin	7	45.0	58	103	11/12/96
	31	lubber	8	55.5	58	103	11/12/96

Equi-Join

 A special case of condition join where the condition contains only equalities

$$R \bowtie Rattr1 = Sattr2^S$$

 Result Schema similar to cross-product, but only one copy of fields for which equality is specified.

							1					
51 X R1	sid1	sna	me r	ating	age	:	sid	2	bid		day	
	22	dus	tin	7	45.	0	22		101		10/10/9	96
	22	dus	tin	7	45.	0	58		103	_	11/12/5	76
	31	lub		8	55.	5	22		101		10/10/9	×
	21	11.1		0	55	~	50		102		11/12/0	
	51	100	501	10	25		20		105		10/10/0	
	50	rust	5	10	55.	0	22		101		10/10/2	
	58	rust	y	10	35.	0	58		103		11/12/9	96
				S	'l⊳	1 s	id ¹	R1				
	s	sid	snam	ne rat	ing	ag	,e	bi	d	da	y	
	2	22	dusti	in 🗆	7	45	5.0	1()1	10	/10/96	
	4	58	rusty	/ 1	0	35	5.0	10)3	11	/12/96	

Natural Join

- ▶ Equijoin on *all* common fields $R \bowtie S$
- ▶ Common fields are NOT duplicated in the result

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 domain (type)
- What is the schema of result?

U	nion E	xamp	le					
S1				_				
sid	sname	rating	age					
22	dustin	7	45.0]	sid	sname	rating	age
31	lubber	8	55.5		22	dustin	7	45.0
58	rusty	10	35.0		31	lubber	8	55.5
S2			1	1	58	rusty	10	35.0
sid	sname	rating	age	1	44	guppy	5	35.0
28	yuppy	9	35.0		28	yuppy	9	35.0
31	lubber	8	55.5			S1	$\cup S2$	
	lubber	~						
44	guppy	5	35.0					
58	rusty	10	35.0					
•								

	In	itersec	tion E	Exam	ple				
Sid sname rating age 31 lubber 8 55.5 58 rusty 10 35.0 Sid sname rating age 28 yuppy 9 35.0 31 lubber 8 55.5 44 guppy 5 35.0	S1	_	_		_				
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	sid	sname	rating	age					
58 rusty 10 35.0 52 sid sname rating age sid sname rating age 28 yuppy 9 35.0 31 lubber 8 55.5 31 lubber 8 55.5 44 guppy 5 35.0 $S1 \cap S2$	22	dustin	7	45.0					
s2 31 lubber 8 55.5 sid sname rating age 28 yuppy 9 35.0 31 lubber 8 55.5 44 guppy 5 35.0	31	lubber	8	55.5					
sid sname rating age 28 yuppy 9 35.0 31 lubber 8 55.5 44 guppy 5 35.0	58	rusty	10	35.0		sid	sname	rating	age
28 yuppy 9 35.0 31 lubber 8 55.5 44 guppy 5 35.0	S2					31	lubber	8	55.5
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	sid	sname	rating	age]	58	rusty	10	35.0
31 lubber 8 55.5 44 guppy 5 35.0	28	yuppy	9	35.0	1		<u>S1</u>	52	
8-11/	31	lubber	8	55.5			317	52	
58 rusty 10 35.0	44	guppy	5	35.0					
	58	rusty	10	35.0					

Se	et-Diffe	erence	e Exa
S1			
sid	sname	rating	age
2	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
S2			
sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
14	guppy	5	35.0
58	rusty	10	35.0