Relational Algebra

CSCD343- Introduction to databases- A. Vaisman

Formal Relational Query Languages

- Two mathematical Query Languages form the basis for "real" languages (e.g. SQL), and for implementation:
 - <u>*Relational Algebra*</u> More operational(procedural), very useful for representing execution plans.
 - <u>Relational Calculus</u>: Lets users describe what they want, rather than how to compute it. (Nonoperational, <u>declarative</u>.)

CSCD343- Introduction to databases- A. Vaisman

Relational Query Languages

- <u>Query languages</u>: Allow manipulation and retrieval of data from a database.
- Relational model supports simple, powerful QLs:
 - Strong formal foundation based on logic.
 - Allows for much optimization.
- Query Languages != programming languages!
 - QLs not expected to be "Turing complete".
 - QLs not intended to be used for complex calculations.
 - QLs support easy, efficient access to large data sets.

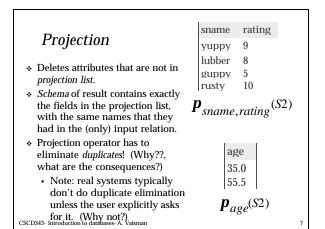
CSCD343- Introduction to databases- A. Vaisman

Preliminaries

- 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 (but query will run regardless of instance!)
 - The schema for the *result* of a given query is also fixed! Determined by definition of query language constructs.
- * Positional vs. named-field notation:
 - Positional notation easier for formal definitions, named-field notation more readable.
 - Both used in SQL

CSCD343- Introduction to databases- A. Vaisman

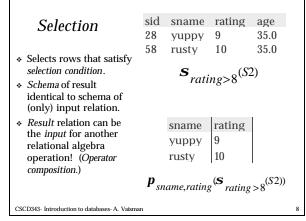
	Example Instan	ces	R1	<u>sid</u> 22 58	<u>bid</u> 101 103	10/1	<u>ay</u> 10/96 12/96	
*	"Sailors" and "Reserves" relations for our examples. "bid"= boats. "sid": sailors	S1	sid 22	snam dustir		~	age 45.0	
*	We'll use positional or named field notation, assume that names of fields		31 58	lubbe rusty		8	55.5 35.0	_
	in query results are `inherited' from names of fields in query input relations.	S2	sid 28 31 44 58	snam yupp lubbe gupp rusty	v g er 8 y 5	3	age 35.0 55.5 35.0 35.0	
c	SCD343- Introduction to databases- A. Vaisman							5



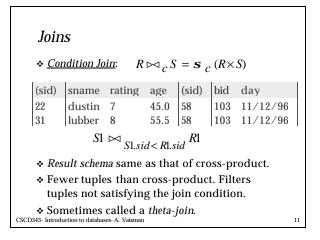
Relational Algebra * Basic operations: * Selects rows that satisfy • <u>Selection</u> (S) Selects a subset of rows from relation. selection condition. • <u>Projection</u> (\boldsymbol{p}) Deletes unwanted columns from relation. * Schema of result identical to schema of • <u>*Cross-product*</u> (\times) Allows us to combine two relations. (only) input relation. • <u>Set-difference</u> (-) Tuples in reln. 1, but not in reln. 2. * Result relation can be • <u>Union</u> (\cup) Tuples in reln. 1 and in reln. 2. the input for another * Additional operations: relational algebra · Intersection, *join*, division, renaming: Not essential, but operation! (Operator (very!) useful. composition.)

6

* Since each operation returns a relation, operations can be *composed*! (Algebra is "closed".) CSCD343- Introduction to databases- A. Vaisman



Union, Intersection, Set-Difference							
	sid	sname	rating	age			
* All of these operations take	22	dustin	7	45.0			
two input relations, which	31	lubber	8	55.5			
must be <u>union-compatible</u> :	58	rusty	10	35.0			
 Same number of fields. 	44	guppy	5	35.0			
 `Corresponding' fields 	28	yuppy	9	35.0			
have the same type.		SIV	$\cup S2$				
* What is the <i>schema</i> of result?							
	sid	sname	rating	age			
sid sname rating age	31	lubber	8	55.5			
22 dustin 7 45.0	58	rusty	10	35.0			
<i>S</i> 1– <i>S</i> 2		S1	$\cap S2$				
CSCD343- Introduction to databases- A. Vaisman							



Cross-Product

- ✤ Each row of S1 is paired with each row of R1.
- *Result schema* has one field per field of S1 and R1, with field names `inherited' if possible.
 - Conflict: Both S1 and R1 have a field called sid.

(sid)	sname	rating	age	(sid)	bid	dav
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	rustv	10	35.0	22	101	10/10/96
58	rustv	10	35.0	58	103	11/12/96

• <u>Renaming operator</u>: $r(C(1 \rightarrow sidl, 5 \rightarrow sid2), S1 \times R1)$ CSCD343- Introduction to databases-A. Vaisman

Joins

10

 <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		
one	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$							
CSCD343- I								

Division

Not supported as a primitive operator, but useful for expressing queries like: *Find sailors who have reserved <u>all</u> boats.*

 $\Delta = \frac{1}{2} \frac{1}{2}$

- Precondition: in A/B, the attributes in B must be included in the schema for A. Also, the result has attributes A-B.
 - SALES(supId, prodId);
 - PRODUCTS(prodId);
 - Relations SALES and PRODUCTS must be built using projections.
 - SALES/PRODUCTS: the ids of the suppliers supplying

CSCD34AInterproducesbases-A. Vaisman

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. Division is NOT implemented in SQL).
- Idea: For SALES/PRODUCTS, compute all products such that there exists at least one supplier not supplying it.
 - *x* value is *disqualified* if by attaching *y* value from *B*, we obtain an *xy* tuple that is not in *A*

$$A = \mathbf{p}_{sid}((\mathbf{p}_{sid}(Sales) \times \text{Products}) - Sales))$$

15

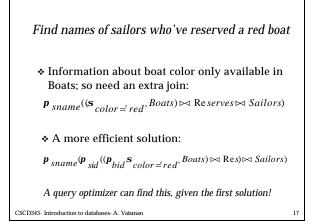
The answer is $\pi_{
m sid}(
m Sales)$ - A

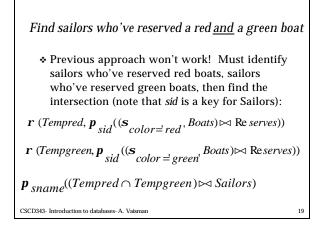
CSCD343- Introduction to databases- A. Vaisman

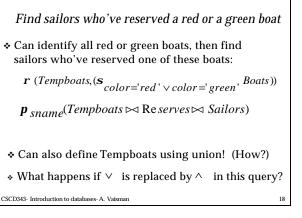
13

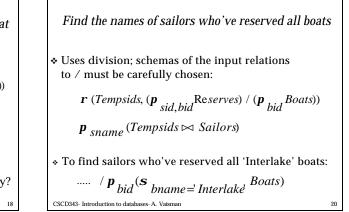
Exa	mple	es of Divisio	on A/B	
sno s1 s1 s1 s1 s1 s2	pno p1 p2 p3 p4 p1	pno p2 B1 sno	pno p2 p4 <i>B2</i>	pno p1 p2 p4 <i>B3</i>
s2 s3 s4 s4	p2 p2 p2 p4	s1 s2 s3 s4 <i>A/B1</i>	sno s1 s4 <i>A/B2</i>	sno s1 A/B3
343- Introdu	iction to dat	abases- A. Vaisman		

Find names of sailors who've reserved boat #103						
Solution 1:	$\boldsymbol{p}_{sname}((\boldsymbol{s}_{bid=103} \text{ Reserves}) \bowtie \text{ Sailors})$					
Solution 2:	\boldsymbol{r} (Templ, $\boldsymbol{s}_{bid=103}$ Reserves)					
	r (Temp2, Temp1 ⋈ Sailors)					
	p _{sname} (Temp2)					
Solution 3:	$p_{sname}(s_{bid=103}(\text{Reserves} \bowtie Sailors))$					
CSCD343- Introduction to o	databases- A. Vaisman 16					









Summary

- The relational model has rigorously defined query languages that are simple and powerful.
- Relational algebra is more operational; useful as internal representation for query evaluation plans.
- Several ways of expressing a given query; a query optimizer should choose the most efficient version.

CSCD343- Introduction to databases- A. Vaisman

21