#### Relational Algebra

Adapted slides by Manos Athanassoulis

#### Up to now ...

#### we have been discussing how to:

(i) model the requirements(ii) translate them into relational schema(iii) refine the schema

today: execute queries relational algebra

#### Reminders

#### Relation

Schema: relation name, attributes (type & name)

**Students**(*sid*: string, *name*: string, *login*: string, *age*: integer, *gpa*: real)

#### Instance

a table containing rows of such columns

every relation instance is a set (all rows distinct)

### Relational Algebra

Relational Query Languages

Selection & Projection

Union, Set Difference & Intersection

Cross product & Joins

Examples

Division

### Relational Algebra

Relational Query Languages

Selection & Projection

Union, Set Difference & Intersection

Cross product & Joins

Examples

Division

### Relational Query Languages

Query languages: manipulation and retrieval of data

#### 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.

#### 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, very useful for representing execution plans.

Relational Calculus: Lets users describe what they want, rather than how to compute it. (Non-procedural, declarative.)

Understanding Algebra & Calculus is key to understanding SQL, query processing!

#### **Preliminaries**

# Query from a relation instance to a relation instance

input & output schema

<u>different</u> but <u>fixed</u> queries run over any <u>legal</u> instances output schema defined by the query constructs

attribute notation positional & name-field

#### Relational Algebra: 5 Basic Operations

```
<u>Selection</u> (\sigma) Selects a subset of rows from relation (horizontal).
```

<u>Projection</u> ( $\pi$ ) Retains only wanted <u>columns</u> from relation (vertical).

<u>Cross-product</u> (x) Allows us to combine two relations.

<u>Set-difference</u> ( $\overline{\phantom{a}}$ ) Tuples in R<sub>1</sub>, but not in R<sub>2</sub>.

<u>Union</u> (U) Tuples in  $R_1$  and/or in  $R_2$ .

each operation returns a relation : composability (Algebra is "closed")

# Example Instances

sid	bid	day
22	101	10/10/16
58	103	11/12/16

#### **Boats**

<u>bid</u>	bname	color
101	Interlake	blue
102	Interlake	red
103	Clipper	green
104	Marine	red

S

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

**S**<sub>2</sub>

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

### Relational Algebra

Relational Query Languages

Selection & Projection

Union, Set Difference & Intersection

Cross product & Joins

Examples

Division

### Projection

Examples:  $\pi_{age}(S_2)$ ;  $\pi_{sname, rating}(S_2)$  retains only attributes that are in the "projection list"

#### schema of result:

fields of projection list (with the same names)

projection operator has to *eliminate duplicates* why we may have duplicates? why remove them?



Note: systems typically don't do duplicate elimination unless the user explicitly asks for it (Why not?)

# Projection

si	d	sname	rating	aş	ge
2	8	yuppy	9	3	5.0
3	l	lubber	8	5	5.5
4	4	guppy	5	3	5.0
5	8	rusty	10	3	5.0

sname	rating
yuppy	9
lubber	8
guppy	5
rusty	10

 $\pi_{sname,rating}(S_{2})$ 

## Projection

$\mathbf{S}$	id	sna	ıme	rat	ing	age
2	8	yu	ору	9		35.0
3		lut	ber	8	) )	55.5
4	4	gu	ppy	4	5	35.0
5	8	rus	sty	1	0	35.0
-						

sname	rating
yuppy	9
lubber	8
guppy	5
rusty	10

$$\pi_{sname,rating}(S_{2})$$

age
35.0
55.5

$$\pi_{age}(S_2)$$

### Selection (σ)

selects rows that satisfy a *selection condition*result: has the same *schema* as the input relation
do we need to do <u>duplicate elimination</u>?



sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

$$\sigma_{rating>8}(S_{2})$$

### Selection (σ)

selects rows that satisfy a *selection condition* 

result: has the same *schema* as the input relation

do we need to do <u>duplicate elimination</u>?

si	$\mathbf{d}$	sname	rating	ag	je
2	8	yuppy	9	35	0.
12	<b>—</b>	lubber	8	5/	5
			<u> </u>		
4	+	guppy	3	3.	0.0
5	3	rusty	10	3.	5.0
			((	7 \	

sname	rating
yuppy	9
rusty	10

$$\pi_{sname,rating}(\sigma_{rating>8}(S_2))$$

### Relational Algebra

Relational Query Languages

Selection & Projection

Union, Set Difference & Intersection

Cross product & Joins

Examples

Division

#### Union and Set-Difference

the <u>set</u> operations take two input relations which must be <u>union-compatible</u>

(i) same number of fields

(ii) "corresponding" fields have the same type

for which, if any, is duplicate elimination required? (union/set-difference)

## Union

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S

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
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

$$S_1 \cup S_2$$

#### Set Difference

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

sid	sname	rating	age
22	dustin	7	45.0

$$S_1 - S_2$$

S

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
44	guppy	5	35.0

$$S_2 - S_2$$

#### Compound Operator: Intersection

in addition to the 5 basic operators
several additional <u>compound operators</u>
no new computational power, but useful shorthands
<u>can be expressed solely with the basic ops</u>

intersection takes two <u>union-compatible</u> relations

Q: How to express it using basic operators?

$$R \cap S = R - (R - S)$$



#### Intersection

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

**S**<sub>1</sub>

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
31	lubber	8	55.5
58	rusty	10	35.0

$$S_1 \cap S_2$$

### Relational Algebra

Relational Query Languages

Selection & Projection

Union, Set Difference & Intersection

Cross product & Joins

Examples

Division

#### **Cross-Product**

 $S_1 \times R_1$ : each row of  $S_1$  paired with each row of  $R_1$  how many rows in the result?

*result schema* has one field per field of S<sub>1</sub> and R<sub>1</sub>, with field names "inherited" (if possible)

may have a naming conflict:

both S<sub>1</sub> and R<sub>1</sub> have a field with the same name in this case, can use the *renaming operator*:

$$\rho(C(1 \rightarrow sid_1, 5 \rightarrow sid_2), S_1 \times R_1)$$

#### Cross Product Example

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

sid	bid	day
22	101	10/10/16
58	103	11/12/16

 $R_1$ 

**S**<sub>1</sub>

#### Compound Operator: Join

**Joins** are compound operators :  $\times$ ,  $\sigma$ , (sometimes)  $\pi$ 

frequent type is "<a href="matural join" | (often called "join")</a>

 $R \bowtie S$  conceptually is:

compute R×S

select rows where attributes in both R, S have equal values
project all unique attributes and one copy of the common ones

Note: Usually done much more efficiently than this Useful for putting *normalized* relations back together

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

sid	bid	day
22	101	10/10/16
58	103	11/12/16

 $R_{1}$ 

$$S_1 \bowtie R_1 =$$

sid	sname	rating	age	bid	day
22	dustin	7	45.0	101	10/ 10/ 16
58	rusty	10	35.0	103	11/ 12/ 16

 $S_1 \times R_1 =$ 

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/ 10/ 16
22	dustin	7	45.0	58	103	11/ 12/ 16
31	lubber	8	55.5	22	101	10/ 10/ 16
31	lubber	8	55.5	58	103	11/ 12/ 16
58	rusty	10	35.0	22	101	10/ 10/ 16
58	rusty	10	35.0	58	103	11/ 12/ 16

 $S_1 \times R_1 =$ 

 $^{2}$  O

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/ 10/ 16
-22	dustin	7	45.0	-58	103	<del>11/ 12/ 16</del>
31	lubber	8	55.5	22	101	10/ 10/ 16
31	lubber	8	55.5	58		11/ 12/ 16
58-	rusty	10	35.0			10/ 10/ 16
58	rusty	10	35.0	58	103	11/ 12/ 16

 $S_1 \times R_1 =$ 

2

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/ 10/ 16
-22	dustin	7	45.0	-58	103	<del>11/ 12/ 16</del>
31_	lubber	88	55.5	22	101	10/ 10/ 16
31_	lubber	8	55.5	-58-	103	<del>11/ 12/ 16</del>
58-	rusty	10	35.0	22	101	10/ 10/ 16
58	rusty	10	35.0	58	103	11/ 12/ 16



$$S_1 \bowtie R_1 =$$

sid	sname	rating	age	bid	day
22	dustin	7	45.0	101	10/ 10/ 16
58	rusty	10	35.0	103	11/ 12/ 16

#### Other Types of Joins

condition join (or "theta-join")

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

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	58	103	11/ 12/ 16
31	lubber	8	55.5	58	103	11/ 12/ 16

$$S_1 \square S_1 sid < R_1 sid R_1$$

result schema same as that of cross-product may have fewer tuples than cross-product

<u>Equi-Join</u>: Special case: condition c contains only conjunction of equalities.

### Relational Algebra

Relational Query Languages

Selection & Projection

Union, Set Difference & Intersection

Cross product & Joins

**Examples** 

Division

# Examples

#### Reserves

sid	bid	<u>day</u>
22	101	10/10/16
58	103	11/12/16

#### **Sailors**

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

#### **Boats**

bid	bname	color
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	Red

Sailors (sid, sname, rating, age)

Boats (bid, bname, color)

#### Find names of sailors who have reserved boat #103

Solution 1:

$$\pi_{sname}((\sigma_{bid=103} \text{Reserves}) \coprod Sailors)$$

Solution 2:

$$\pi_{sname}(\sigma_{bid=103}(\text{Reserves} \square Sailors))$$

another solution?



Reserves (sid, bid, day) **Sailors** (sid, sname, rating, age) **Boats** (bid, bname, color)

Find names of sailors who have reserved a red boat

boat color only available in Boats; need an <u>extra</u> join:

$$\pi_{sname}((\sigma_{color=red}, Boats) \coprod Reserves \coprod Sailors)$$

a <u>more efficient</u> solution: why more efficient?



$$\pi_{sname}(\pi_{sid}((\pi_{bid}\sigma_{color='red'},Boats) \coprod Res) \coprod Sailors)$$

a <u>query optimizer</u> can find this given the first solution!

Reserves (sid, bid, day) Sailors (sid, sname, rating, age)
Boats (bid, bname, color)

Find sailors who have reserved a red or a green boat

identify all red or green boats first

$$\rho$$
 (Tempboats, ( $\sigma_{color=red' \lor color=green'}$  Boats))

then find sailors who have reserved one of these boats:

$$\pi_{sname}$$
(Temphoats  $\coprod$  Reserves  $\coprod$  Sailors)

Find sailors who have reserved a red and a green boat

Previous approach will not work! **Why?** identify sailors who have reserved <u>red</u> boats

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

sailors who have reserved green boats

$$\rho$$
 (Tempgreen,  $\pi_{sid}((\sigma_{color=green}, Boats) \coprod Reserves))$ 

then find the intersection (sid is a key for Sailors)

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

# More examples – Your turn! 7



- 1. Find (the name of) all sailors whose rating is above 9
- 2. Find all sailors who reserved a boat prior to November 1, 2016
- 3. Find (the names of) all boats that have been reserved at least once
- 4. Find all pairs of sailors with the same rating
- 5. Find all pairs of sailors in which the <u>older</u> sailor has a <u>lower</u> rating

(1) Find (the name of) all sailors whose rating is above 9

$$\pi_{sname}(\sigma_{rating>9}(Sailors))$$

(2) Find all sailors who reserved a boat prior to November 1, 2016

$$\pi_{sname}$$
(Sailors $\square \sigma_{day < 11/1/16}$ (Reserves))

Reserves (sid, bid, day)

Sailors (sid, sname, rating, age)

Boats (bid, bname, color)

(3) Find (the names of) all boats that have been reserved at least once

 $\pi_{bname}(Boats \coprod Reserves)$ 

Reserves (sid, bid, day)

Sailors (sid, sname, rating, age)

Boats (bid, bname, color)

### (4) Find all pairs of sailors with the same rating

$$\rho(S_1(1 \longrightarrow sid_1, 2 \longrightarrow sname_1, 3 \longrightarrow rating_1, 4 \longrightarrow age_1), Sailors)$$
  
 $\rho(S_2(1 \longrightarrow sid_2, 2 \longrightarrow sname_2, 3 \longrightarrow rating_2, 4 \longrightarrow age_2), Sailors)$ 

$$\pi_{sname_1}$$
,  $sname_2$   $(S_1 \square rating_1 = rating_2 \land sid_1 \neq sid_2$  is this ok?  $sid_1 < sid_2$ 

(5) Find all pairs of sailors in which the older sailor has a lower rating



$$\pi_{sname_1,sname_2}(S_1 \square rating_1 < rating_2 \land age_1 > age_2 S_2)$$

# Relational Algebra

Relational Query Languages

Selection & Projection

Union, Set Difference & Intersection

Cross product & Joins

Examples

Division

## **Last Compound Operator: Division**

useful for expressing "<u>for all</u>" queries like: "find sids of sailors who have reserved <u>all</u> boats"

for A/B attributes of B are subset of attributes of A may need to "project" to make this happen. e.g., let A have 2 fields, x and y; B have only field y:

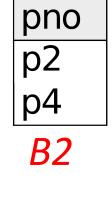
$$A/B = \{ \langle x \rangle | \forall \langle y \rangle \in B(\exists \langle x, y \rangle \in A) \}$$

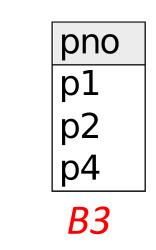
A/B contains all x tuples such that for every y tuple in B, there is an xy tuple in A

# Examples of Division A/B

sno	pno
s1	p1
s1 (	p2
s1	р3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

pno	
p2	
<i>B</i> 1	





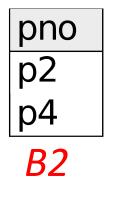
sno s1 s2 s3 s4

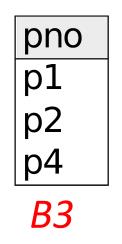
A/B1

# Examples of Division A/B

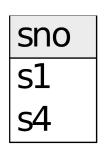
sno	pno
s1	p1
sl (	p2
s1	<b>2</b> 0
s1 (	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4
A	

pno	
p2	
B1	





sno	
s1	
s2	
s3	
s4	

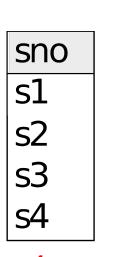


A/B2

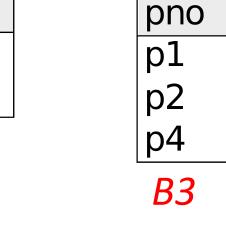
## Examples of Division A/B

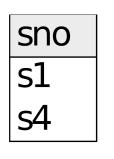
sno	pno
s1	p1
s1	p2
s1	<b>5</b> 3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

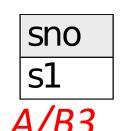
pno	
p2	
<i>B</i> 1	



pno	
p2	
p4	
<i>B2</i>	







A

A/B1

*A/B2* 

### Expressing A/B Using Basic Operators

division is not essential op; just a shorthand

(true for joins, but so common that are implemented specially)

*Idea*: For *A/B*, compute all *x* <u>values that are not "disqualified"</u> 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: 
$$\pi_{\chi}((\pi_{\chi}(A) \times B) - A)$$

A/B: 
$$\pi_{\chi}(A)$$
 — Disqualified x values

## Expressing A/B: $\pi_{sno}(A) - \pi_{sno}((\pi_{sno}(A) \times B) - A)$

sno	pno
s1	p1
s1	p2
s1	р3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

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

 $\begin{array}{c|c}
\hline
sno \\
s1 \\
s2 \\
s3 \\
s4
\end{array} \times \begin{bmatrix}
pno \\
p1 \\
p2 \\
p4 \\
B
\end{array}$ 

A

$$T1 = \pi \sum_{SNO} (A) \times B$$

## Expressing A/B: $\pi_{sno}(A) - \pi_{sno}((\pi_{sno}(A) \times B) - A)$

sno	pno
s1	p1
s1	p2
s1	р3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4
s4	p4

sno	pno
<u>5</u> 1	<u>n1</u>
<u>51</u>	p2
_1	' 
<del>51</del>	<del>p4</del>
<u>\$2</u>	p1
	•
<u>\$2</u>	p2_
s2	n/l
52	p4
s3	p1
<del>53</del> —	<del>-</del>
32	<del>p2</del>
s3	p4
	-
s4	p1
54	<del>p2</del>
_	<b>1 1 1</b>
<del>54  </del>	<del>1)'1</del>
$\overline{\mathcal{L}}$	$\rightarrow R$

sno	pno			
s2	p <mark>4</mark>			
<u>53</u>	n1			
s3	p4			
s4	p <mark>1</mark>			
T1-A				

pno	
p1	
p2	
p4	
R	

$$T2=\pi \overline{sno}(T1-A)$$

# Expressing A/B: $\pi_{sno}(A) - \pi_{sno}((\pi_{sno}(A) \times B) - A)$

		sno	pno	S	no	pno			
sno	pno	<u>51</u>	p1	S	2	p <mark>4</mark>			
s1	p1	s1	<del>p2</del>	S		<del>p</del> 1		pno	
s1	p2	<del>51</del>	<del>p4</del>	S		p4		p1	
s1	р3	s2	p1	S		p1			
s1	p4	<u>\$2</u>	p2			-		p2 p4	
s2	p1	s2	p4		<i>I</i> 1.	-A			
s2	p2	s3	-		1			В	
s3	p2		p1	sno	-		1		,
1	p2	<del>53</del>	<del>p2</del>	s1		sno		sno	
s4 s4	p4	s3	p4	s2	_	s2	=	s1	
ЭТ	РТ	s4	p1	s3		s3	A/I	3=	_
-	4	54	<del>p2</del>	s4		s4	$\pi$	$no^{(A)}$	-T2
	<i>-</i>	<b>54</b>	<del>p4</del>	T2-	au	no(T1	-4) $Si$	$no^{(21)}$	1 4
	$T1=\pi$	$S_{sno}(A)$	$A)\times B$	1 2-	SI	10 (1 1	<b>11</b> <i>j</i>		

### Find the names of sailors who have reserved all boats

use division; schemas of the input relations to / must be carefully chosen (why?)

$$\rho \; (\textit{Tempsids}, (\pi_{\textit{sid}, \textit{bid}} \text{Reserves}) \, / \, (\pi_{\textit{bid}} \textit{Boats})) \\ \pi_{\textit{sname}} (\textit{Tempsids} \coprod \textit{Sailors})$$

To find sailors who have reserved all "Interlake" boats:

.... 
$$/\pi_{bid}$$
 ( $\sigma_{bname=Interlake}$  Boats)

#### Find the names of sailors who have reserved all boats

use division; schemas of the input relations to / must be carefully chosen (why?)

$$\rho$$
 (Tempsids, ( $\pi$  sid,bid Reserves) / ( $\pi$  bid Boats))
$$\pi$$
 sname (Tempsids  $\coprod$  Sailors)

what if we divided Reserves /  $\pi_{hid}$  (Boats)?



this would return the pairs of (sid,date) that have a value for evey boat, i.e., the <u>sids</u> that <u>rented **every** boat</u>, <u>**every** day</u> they made any reservation!!!! Not so useful!