Relational Model 3: Relational Algebra (Part II)

Gary KL Tam

Department of Computer Science
Swansea University

Relational Algebra (Review)

We have learned the 6 fundamental operations of relational algebra:

- Rename ρ
- Selection σ
- Projection Π
- Set union $\cup$
- Set difference $-$
- Cartesian product $\times$

- The operators of the previous slide can express all queries in relational algebra. However, if we rely on only those operations, some queries common in practice require lengthy expressions.
- To shorten those expressions, people identified the following 4 operations, each of which can be implemented using only the 6 fundamental operators, and can be used to simplify many queries:
  - Natural Join $\bowtie$
  - Assignment $\leftarrow$
  - Set Intersection $\cap$
  - Division $\div$

Cartesian product can be inconvenient

Cartesian Product $\times$

- It can introduce nonsense tuples.
- You can get rid of them with selects.
- But this is so highly common, an operation was defined to make it easier: natural join.

PROF

pid	name	dept	rank	sal
$p1$	Adam	CS	asst	6000
$p2$	Bob	EE	asso	8000
$p3$	Calvin	CS	full	10000
$p4$	Dorothy	EE	asst	5000
$p5$	Emily	EE	asso	8500

TEACH

pid	cid	year
$p1$	$c1$	2011
$p2$	$c2$	2012
$p1$	$c2$	2012

$\mathrm{PROF} \times \mathrm{TEACH}$ returns the table in the next slide.

pid	name	dept	rank	sal	pid	cid	year
$p1$	Adam	CS	asst	6000	p_1	c_1	2011
$p2$	Bob	EE	asso	8000	p_1	c_1	2011
$p3$	Calvin	CS	full	10000	p_1	c_1	2011
$p4$	Dorothy	EE	asst	5000	p_1	c_1	2011
$p5$	Emily	EE	asso	8500	p_1	c_1	2011
$p1$	Adam	CS	asst	6000	p_2	c_2	2012
$p2$	Bob	EE	asso	8000	p_2	c_2	2012
$p3$	Calvin	CS	full	10000	p_2	c_2	2012
$p4$	Dorothy	EE	asst	5000	p_2	c_2	2012
$p5$	Emily	EE	asso	8500	p_2	c_2	2012
$p1$	Adam	CS	asst	6000	p_1	c_2	2012
$p2$	Bob	EE	asso	8000	p_1	c_2	2012
$p3$	Calvin	CS	full	10000	p_1	c_2	2012
$p4$	Dorothy	EE	asst	5000	p_1	c_2	2012
$p5$	Emily	EE	asso	8500	p_1	c_2	2012

Who really taught a course in he past?

Does $p2$ teaches $c1$?

Does $p5$ teaches $c2$?

PROF

pid	name	dept	rank	sal
$p1$	Adam	CS	asst	6000
$p2$	Bob	EE	asso	8000
$p3$	Calvin	CS	full	10000
$p4$	Dorothy	EE	asst	5000
$p5$	Emily	EE	asso	8500

TEACH

pid	cid	year
$p1$	$c1$	2011
$p2$	$c2$	2012
$p1$	$c2$	2012

PROF

pid	name	dept	rank	sal
$p1$	Adam	CS	asst	6000
$p2$	Bob	EE	asso	8000
$p3$	Calvin	CS	full	10000
$p4$	Dorothy	EE	asst	5000
$p5$	Emily	EE	asso	8500

TEACH

pid	cid	year
$p1$	$c1$	2011
$p2$	$c2$	2012
$p1$	$c2$	2012

PROF × TEACH

pid	name	dept	rank	sal	pid	cid	year
$p1$	Adam	CS	asst	6000	p_1	c_1	2011
$p2$	Bob	EE	asso	8000	p_1	c_1	2011
$p3$	Calvin	CS	full	10000	p_1	c_1	2011
$p4$	Dorothy	EE	asst	5000	p_1	c_1	2011
$p5$	Emily	EE	asso	8500	p_1	c_1	2011
$p1$	Adam	CS	asst	6000	p_2	c_2	2012
$p2$	Bob	EE	asso	8000	p_2	c_2	2012
$p3$	Calvin	CS	full	10000	p_2	c_2	2012
$p4$	Dorothy	EE	asst	5000	p_2	c_2	2012
$p5$	Emily	EE	asso	8500	p_2	c_2	2012
$p1$	Adam	CS	asst	6000	p_1	c_2	2012
$p2$	Bob	EE	asso	8000	p_1	c_2	2012
$p3$	Calvin	CS	full	10000	p_1	c_2	2012
$p4$	Dorothy	EE	asst	5000	p_1	c_2	2012
$p5$	Emily	EE	asso	8500	p_1	c_2	2012

$$\mathrm{PROF} \times \mathrm{TEACH}$$

pid	name	dept	rank	sal	pid	cid	year
$p1$	Adam	CS	asst	6000	p_1	c_1	2011
$p2$	Bob	EE	asso	8000	p_1	c_1	2011
$p3$	Calvin	CS	full	10000	p_1	c_1	2011
$p4$	Dorothy	EE	asst	5000	p_1	c_1	2011
$p5$	Emily	EE	asso	8500	p_1	c_1	2011
$p1$	Adam	CS	asst	6000	p_2	c_2	2012
$p2$	Bob	EE	asso	8000	p_2	c_2	2012
$p3$	Calvin	CS	full	10000	p_2	c_2	2012
$p4$	Dorothy	EE	asst	5000	p_2	c_2	2012
$p5$	Emily	EE	asso	8500	p_2	c_2	2012
$p1$	Adam	CS	asst	6000	p_1	c_2	2012
$p2$	Bob	EE	asso	8000	p_1	c_2	2012
$p3$	Calvin	CS	full	10000	p_1	c_2	2012
$p4$	Dorothy	EE	asst	5000	p_1	c_2	2012
$p5$	Emily	EE	asso	8500	p_1	c_2	2012

PROF × TEACH

pid	name	dept	rank	sal	pid	cid	year
p1	Adam	CS	asst	6000	p_1	c_1	2011
p2	Bob	EE	asso	8000	p_1	c_1	2011
p3	Calvin	CS	full	10000	p_1	c_1	2011
p4	Dorothy	EE	asst	5000	p_1	c_1	2011
p5	Emily	EE	asso	8500	p_1	c_1	2011
p1	Adam	CS	asst	6000	p_2	c_2	2012
p2	Bob	EE	asso	8000	p_2	c_2	2012
p3	Calvin	CS	full	10000	p_2	c_2	2012
p4	Dorothy	EE	asst	5000	p_2	c_2	2012
p5	Emily	EE	asso	8500	p_2	c_2	2012
p1	Adam	CS	asst	6000	p_1	c_2	2012
p2	Bob	EE	asso	8000	p_1	c_2	2012
p3	Calvin	CS	full	10000	p_1	c_2	2012
p4	Dorothy	EE	asst	5000	p_1	c_2	2012
p5	Emily	EE	asso	8500	p_1	c_2	2012

$$\sigma_{PROF.pid=TEACH.pid}(\text{PROF}\times\text{TEACH})$$

pid	name	dept	rank	sal	pid	cid	year
p1	Adam	CS	asst	6000	p_1	c_1	2011
p2	Bob	EE	asso	8000	p_2	c_2	2012
p1	Adam	CS	asst	6000	p_1	c_2	2012

Natural Join

Denoted by $T_1 \bowtie T_2$

- where T_1 and T_2 are tables.
- The output of the operations T' is formed by
  - Taking the Cartesian product
  - Select to ensure equality on attributes that are in both relations (determined by name)
  - Projecting to remove duplicate attributes.

PROF

pid	name	dept	rank	sal
$p1$	Adam	CS	asst	6000
$p2$	Bob	EE	asso	8000
$p3$	Calvin	CS	full	10000
$p4$	Dorothy	EE	asst	5000
$p5$	Emily	EE	asso	8500

TEACH

pid	cid	year
$p1$	$c1$	2011
$p2$	$c2$	2012
$p1$	$c2$	2012

PROF $\bowtie$ TEACH returns:

pid	name	dept	rank	sal	cid	year
$p1$	Adam	CS	asst	6000	c_1	2011
$p2$	Bob	EE	asso	8000	c_2	2012
$p1$	Adam	CS	asst	6000	c_2	2012

In general:

$$T_1 \bowtie T_2 = \Pi_S(\sigma_{T_1.A_1 = T_2.A_1 \wedge \cdots \wedge T_1.A_d = T_2.A_d}(T_1 \times T_2))$$

where

$$S = (S_1 - S_2) \cup \{T_1.A_1, \ldots, T_1.A_d\} \cup (S_2 - S_1)$$

where S_1 and S_2 are the schemas of T_1 and T_2 respectively, and $A_1, \ldots, A_d$ are the common attributes of T_1 and T_2.

PROF				
pid	**name**	**dept**	**rank**	**sal**

TEACH		
pid	**cid**	**year**

PROF $\bowtie$ TEACH

A: pid S1-S2: name, dept, rank, sal S2-S1: cid, year

pid	name	dept	rank	sal	cid	year
$p1$	Adam	CS	asst	6000	c_1	2011
$p2$	Bob	EE	asso	8000	c_2	2012
$p1$	Adam	CS	asst	6000	c_2	2012

PROF				
pid	name	dept	rank	sal

TEACH		
pid	cid	year

PROF $\bowtie$ TEACH

A: pid   S1-S2: name, dept, rank, sal   S2-S1: cid, year

pid	name	dept	rank	sal	cid	year
$p1$	Adam	CS	asst	6000	c_1	2011
$p2$	Bob	EE	asso	8000	c_2	2012
$p1$	Adam	CS	asst	6000	c_2	2012

in comparison...

$\sigma_{PROF.pid=TEACH.pid}$ (PROF$\times$TEACH)

pid	name	dept	rank	sal	pid	cid	year
$p1$	Adam	CS	asst	6000	p_1	c_1	2011
$p2$	Bob	EE	asso	8000	p_2	c_2	2012
$p1$	Adam	CS	asst	6000	p_1	c_2	2012

Properties of Natural Join

- Commutative:

$$T_1 \bowtie T_2 = T_2 \bowtie T_1$$

  (although attribute order may vary)

- Associative:

$$T_1 \bowtie (T_2 \bowtie T_3) = (T_1 \bowtie T_2) \bowtie T_3$$

- So when writing n-ary joins, brackets are irrelevant. We can just write:

$$T_1 \bowtie T_2 \bowtie \cdots \bowtie T_3$$

Special cases of natural join

No tuples match

Dept	**Head**
HR	Boutilier

Employee	**Dept**
Vista	Sales
Kagani	Production
Tzerpos	Production

Result: empty

Special cases of natural join

Relations have exactly the same attributes

Artist	**Name**
9132	William Shatner
8762	Harrison Ford
1868	Angelina Jolie

Artist	**Name**
1234	Brad Pitt
1868	Angelina Jolie
5555	Patrick Stewart

Result:

Artist	**Name**
1868	Angelina Jolie

Special cases of natural join

Relations have no attributes in common

Artist	Name
1234	Brad Pitt
1868	Angelina Jolie
5555	Patrick Stewart

mID	Title	Year
1111	Alien	1979
1234	Sting	1973

Result: same as Cartesian Product

Artist	Name	mID	Title	Year
1234	Brad Pitt	1111	Alien	1979
1868	Angelina Jolie	1111	Alien	1979
5555	Patrick Stewart	1111	Alien	1979
1234	Brad Pitt	1234	Sting	1973
1868	Angelina Jolie	1234	Sting	1973
5555	Patrick Stewart	1234	Sting	1973

Set intersection

Denoted by $T_1 \cap T_2$

- where T_1 and T_2 are tables with the same schema.
- The output of the operation is a table T' such that
  - T' has the same schema as T_1 (and hence, T_2).
  - T' contains all and only the tuples that appear in both T_1 and T_2.

PROF

pid	name	dept	rank	sal
$p1$	Adam	CS	asst	6000
$p2$	Bob	EE	asso	8000
$p3$	Calvin	CS	full	10000
$p4$	Dorothy	EE	asst	5000
$p5$	Emily	EE	asso	8500
$p6$	Frank	CS	full	9000

$$\sigma_{sal \geq 8500}(\mathrm{PROF}) \quad \bigcap \quad \sigma_{dept=CS}(\mathrm{PROF})$$

PROF

pid	name	dept	rank	sal
$p1$	Adam	CS	asst	6000
$p2$	Bob	EE	asso	8000
$p3$	Calvin	CS	full	10000
$p4$	Dorothy	EE	asst	5000
$p5$	Emily	EE	asso	8500
$p6$	Frank	CS	full	9000

$\sigma_{sal \geq 8500}($ PROF $)$

$\cap$

PROF

pid	name	dept	rank	sal
$p1$	Adam	CS	asst	6000
$p2$	Bob	EE	asso	8000
$p3$	Calvin	CS	full	10000
$p4$	Dorothy	EE	asst	5000
$p5$	Emily	EE	asso	8500
$p6$	Frank	CS	full	9000

$\sigma_{dept = CS}($ PROF $)$

PROF

pid	name	dept	rank	sal
$p3$	Calvin	CS	full	10000
$p5$	Emily	EE	asso	8500
$p6$	Frank	CS	full	9000

$$\bigcap$$

PROF

pid	name	dept	rank	sal
$p1$	Adam	CS	asst	6000
$p3$	Calvin	CS	full	10000
$p6$	Frank	CS	full	9000

Remember, union $\bigcup$ and intersect $\bigcap$:

The two operands have the same schema!

PROF

pid	name	dept	rank	sal
$p1$	Adam	CS	asst	6000
$p2$	Bob	EE	asso	8000
$p3$	Calvin	CS	full	10000
$p4$	Dorothy	EE	asst	5000
$p5$	Emily	EE	asso	8500
$p6$	Frank	CS	full	9000

$\sigma_{\text{sal} \geq 8500}(\text{PROF}) \cap \sigma_{\text{dept} = \text{CS}}(\text{PROF})$ returns:

pid	name	dept	rank	sal
$p3$	Calvin	CS	full	10000
$p6$	Frank	CS	full	9000

In general:

$$T_1 \cap T_2 \quad = \quad T_1 - (T_1 - T_2)$$

Division

Denoted by $T_1 \div T_2$

- where T_1 and T_2 are tables such that the schema of T_2 is a subset of the schema of T_1.
- The output of the operation is a table T' such that
  - The schema of T' includes all the columns that are in T_1, but not in T_2.
  - T' contains all and only the tuples t such that:
    - for every tuple $t_2 \in T_2$, $t_1 = (t, t_2)$ is a tuple in T_1, where (t, t_2) represents a tuple that concatenates the attributes of t with those of t_2.

T_1

pid	**cid**
$p1$	$c1$
$p1$	$c2$
$p1$	$c3$
$p2$	$c2$
$p2$	$c3$
$p3$	$c1$
$p4$	$c1$
$p4$	$c2$
$p4$	$c3$

T_2

cid
$c1$
$c2$
$c3$

$T_1 \div T_2$ returns:

pid
$p1$
$p4$

Division Tip: good for answering query like

Find in T1 those who/which takes ALL in T2

Example - Division

Get the names of students who take ALL modules taught by Gary.

$T1 \leftarrow \Pi_{studentID,courses}(Takes)$

$T2 \leftarrow \Pi_{course}(\sigma_{lecturer='Gary'} Teaches)$

$Answer \leftarrow T1 \div T2$

In general:

$$T_1 \div T_2 \quad = \quad \Pi_{S_1 - S_2}(T_1) - \Pi_{S_1 - S_2}\Big(\Pi_{S_1 - S_2}(T_1) \times T_2 - T_1\Big)$$

where S_1 and S_2 are the schemas of T_1 and T_2 respectively.

- Remember it.
- It will becomes useful when we come to SQL.
- More explanation later.

Assignment

Denoted by $T \leftarrow [\textit{expression}]$

- where $[\textit{expression}]$ is a relational algebra expression, and T is a table variable.
- The assignment stores in T the table output by $[\textit{expression}]$.

Assignments are often used to increase clarity by cutting a long query into multiple steps, each of which can be described by a short line.

Assignment 2

Alternative Notation: $T'(s_1, \cdots, s_n) \leftarrow [expression]$

- Let's you name all the attributes of the new relation (not necessarily the same name they would get from Expression).
- T' must be a temporary variable, not one of the relations in the schema.

  Ie, you are not updating the content of a relation!

PROF

pid	**name**	**dept**	**rank**	**sal**
$p1$	Adam	CS	asst	6000
$p2$	Bob	EE	asso	8000
$p3$	Calvin	CS	full	10000
$p4$	Dorothy	EE	asst	5000
$p5$	Emily	EE	asso	8500
$p6$	Frank	CS	full	9000

$$T_1 \leftarrow \Pi_{\mathrm{rank}}(\sigma_{\mathrm{sal}\,\geq\,8000}(\mathrm{PROF}))$$
$$T_2 \leftarrow \Pi_{\mathrm{rank}}(\sigma_{\mathrm{sal}\,\geq\,9000}(\mathrm{PROF}))$$
$$T_1 - T_2$$

returns:

rank
asso

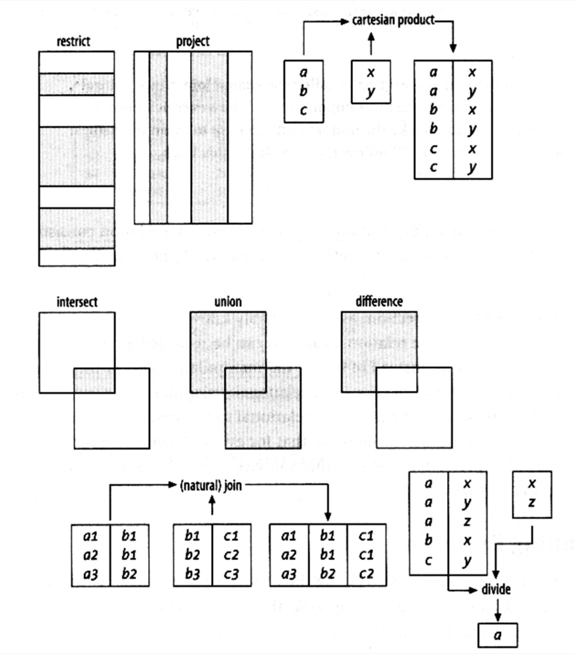
Example

Given tables Q, R, S:

- $\text{Temp1} \leftarrow \text{Q} \bowtie \text{R}$
- $\text{Temp2} \leftarrow \sigma_{a=99}(\text{Temp1}) \bowtie \text{S}$
- $\text{Answer(part, price)} \leftarrow \Pi_{b,c}\ (\text{Temp2})$

- Whether / how small to break things down is up to you. It's all for readability.
- As we saw, assignment can be used not only to break things down, but also to change the names of relations [and attributes].

Summary for Relational Algebra



Building complex expressions

- Complex expressions can be composed recursively, just as in arithmetic.
- Parentheses and precedence rules define the order of evaluation.
- Precedence, from highest to lowest, is:
  σ, Π, ρ
  $\times$, $\bowtie$
  $\cap$, $\div$
  $\cup$, $-$
- Unless very sure, use brackets!

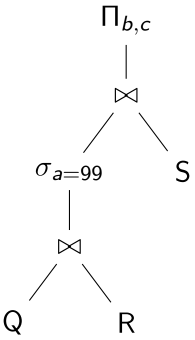
Breaking down complex expressions

- Complex nested expressions can be hard to read.
- Two alternative notations allow us to break them down:
  1. Sequences of assignment statements
  2. Expression trees (operator trees)

Expression tree

Earlier Example

- Temp1 $\leftarrow$ Q $\bowtie$ R
- Temp2 $\leftarrow \sigma_{a=99}(\text{Temp1}) \bowtie$ S
- Answer(part, price) $\leftarrow \Pi_{b,c}$ (Temp2)



Tips for Relational Algebra

- Ask yourself which relations need to be involved.
  Ignore the rest.
- Every time you combine relations, confirm that
  1. attributes that should match will be made to match and
  2. attributes that will be made to match should match
- Break the answer down. Define intermediate relations using assignment.
  - Use good names for the new relations.
  - Name the attributes on the LHS each time, so you don't forget what you have in hand.
  - Add a comment to explain exactly what the relation contains.

Next lecture: SQL

SQL is not based on sets

- Although the relational model is based on sets, SQL is not.
- Reason: getting rid of duplicates is expensive!
- Instead, SQL generally leaves duplicates in unless you ask it not to.
- SQL is based on "bags" (or "multisets"):
  just like sets, but duplicates are allowed.
- $\{6, 2, 56, 1, 9\}$ is a set, and a bag;
  $\{6, 2, 6, 56, 1, 9\}$ is not a set, but is a bag.