Relational Model 2: Relational Algebra

Gary KL Tam

Department of Computer Science
Swansea University

The relational model defines:

1. the format by which data should be stored;
2. the operations for querying the data.

We will focus on the second aspect in this lecture.

Query Language

A database may have many tables

- Need a way to pull out only rows and columns we want

Data are often spread across many tables

- Need a way to combine tables

Relational Algebra

It's an algebra, like elementary algebra in math.

Maths:

- operands like 27.5 and y
- operators like $+$ and $*$

You write expressions describing the value you want.

Relational algebra:

- operands are tables
- operators like "choose the rows that satisfy ..."

You write expressions describing the table you want.

Relational Algebra

- some operators are unary, some binary.
- operands are always relations and result is always a relation
- So you can "compose" expressions and use brackets for precedence, just like in arithmetic expressions.
- Let's see the operations.
  (Remember, a relation/table is a set of tuples)

RA: Questions?

So Relational algebra is mathematics...

Is it a must to learn it?

Why do we learn it?

RA: Questions?

So Relational algebra is mathematics...

- Yes!

Is it a must to learn it?

Why do we learn it?

RA: Questions?

So Relational algebra is mathematics...

- Yes!

Is it a must to learn it?

- Yes!

Why do we learn it?

RA: Questions?

So Relational algebra is mathematics...

- Yes!

Is it a must to learn it?

- Yes!

Why do we learn it?

- Because university teaches theory!

RA: Questions?

So Relational algebra is mathematics...

- Yes!

Is it a must to learn it?

- Yes!

Why do we learn it?

- Because university teaches theory!
- But WHY? (Have you ever asked "why"?)

RA: Questions?

So Relational algebra is mathematics...

- Yes!

Is it a must to learn it?

- Yes!

Why do we learn it?

- Because university teaches theory!
- But WHY? (Have you ever asked "why"?)

WHY?

So that when things change, you know how to adapt...

RA: Questions?

Why learn Concept of OO?

No one can ensure C++/Java will still be the trend.
Python is catching up!

RA: Questions?

Why learn Concept of OO?

No one can ensure C++/Java will still be the trend.
Python is catching up!

RA vs SQL?

No one can ensure SQL will still be the trend.
SQL isn't necessarily the best way to implement that, it's just the best way anyone has come up with so far.

RA: Questions?

Why learn Concept of OO?

No one can ensure C++/Java will still be the trend.
Python is catching up!

RA vs SQL?

No one can ensure SQL will still be the trend.
SQL isn't necessarily the best way to implement that, it's just the best way anyone has come up with so far.

RA

Relational Algebra is ALSO the tool for query optimisation (we will learn later)!

Relational Algebra

Relational algebra is a language for issuing queries on the data stored in a relational database.

Its core consists of 6 fundamental operations:

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

Selection

Denoted by $\sigma_P(T)$

- where T is a table, and P is a predicate on the tuples of T.
- The output is a table T' such that
  - T' has the same schema as T.
  - T' includes all and only the tuples in T satisfying P.

Each predicate can be

- a comparison using the following operators: $=, \neq, <, \leq, >, \geq$.
- multiple comparisons connected by $\wedge$(*and*), $\vee$(*or*) and $\neg$(*not*).

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

$\sigma_{\text{name}=\text{“Bob”}}(\text{PROF})$ returns:

pid	**name**	**dept**	**rank**	**sal**
$p2$	Bob	EE	asso	8000

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

$\sigma_{\text{dept}=\text{``EE''} \wedge \text{sal} > 7000}(\text{PROF})$ returns:

pid	name	dept	rank	sal
$p2$	Bob	EE	asso	8000
$p5$	Emily	EE	asso	8500

Projection

Denoted by $\Pi_A(T)$

- where T is a table, and A is a set of attributes in T.
- The output of the operation is a table T' such that
  - T' has all and only the attributes in A.
  - T' contains all the tuples of T after trimming the attributes not in A.
  - All duplicates (resulting from the trimming) are removed.

Duplicates

- Since relations are sets and the value of an RA expression is a relation, it does not include duplicates.
- Removing columns from a table can introduce duplicate rows. The result of a project has only one copy of each.
- Aside: DBMSs often relax the rule about duplicates, allowing you to specify whether you want them removed.

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

$\Pi_{\text{dept}}(\text{PROF})$ returns:

dept
CS
EE

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

$\Pi_{\text{dept, rank}}(\text{PROF})$ returns:

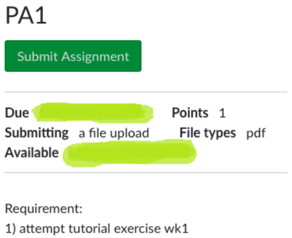
dept	**rank**
CS	asst
EE	asso
CS	full
EE	asst

Peer Assessment

Tutorial Questions

You have all the knowledge to do ex1.pdf
Please submit your attempt to PA1.
It will be due the day before next tutorial.

I will explain what is peer assessment, and how to peer evaluate next week.



Rename

Denoted by $\rho_s(T)$

- where T is a table, s is a string.
- The output of the operation is a table T' that is exactly the same as T, but is named to s.

Denoted by $\rho_{s(a1,a2,...,aN)}(T)$

- Table T is renamed to s with attributes renamed to $a1, a2, ..., aN$.

Denoted by $\rho_{oldname/newname}(T)$

- The table name or an attribute name ($oldname$) of T is renamed to $newname$.

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

$\rho_{\mathrm{LECT}}(\mathrm{PROF})$ returns:

LECT

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

Union

Denoted by $T_1 \cup T_2$

- 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 the tuples of T_1 and T_2, after removing duplicates.

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

$\sigma_{\text{sal} \leq 5000}(\text{PROF}) \cup \sigma_{\text{sal} \geq 10000}(\text{PROF})$ returns:

PROF

pid	name	dept	rank	sal
$p3$	Calvin	CS	full	10000
$p4$	Dorothy	EE	asst	5000

Set Difference

Denoted by $T_1 - T_2$

- 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 the tuples that appear in T_1 but not in T_2, after removing duplicates.

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

$\Pi_{rank}(\sigma_{sal \geq 8000}(\text{PROF})) - \Pi_{rank}(\sigma_{sal \geq 9000}(\text{PROF}))$ returns?

$$\Pi_{rank}(\sigma_{sal \geq 8000}($$

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

$$))$$

$$-$$

$$\Pi_{rank}(\sigma_{sal \geq 9000}($$

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

$$))$$

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	✓

$\Pi_{rank}(\sigma_{sal \geq 8000}($ 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	✓

$\Pi_{rank}(\sigma_{sal \geq 9000}($ PROF $))$

$\Pi_{rank}($

PROF

pid	name	dept	rank	sal
$p2$	Bob	EE	asso	8000
$p3$	Calvin	CS	full	10000
$p5$	Emily	EE	asso	8500
$p6$	Frank	CS	full	9000

$)$

$-$

$\Pi_{rank}($

PROF

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

$)$

$$\begin{array}{c} \textbf{rank} \\ \hline \text{asso} \\ \text{full} \end{array} \quad - \quad \begin{array}{c} \textbf{rank} \\ \hline \text{full} \end{array} \quad \Rightarrow \quad \begin{array}{c} \textbf{rank} \\ \hline \text{asso} \end{array}$$

Cartesian product

Denoted by $T_1 \times T_2$

- where T_1 and T_2 are tables.
- The output of the operation is a table T such that
  - The schema of T includes all the attributes in T_1 and T_2 (if an attribute in T_1 has the same name as an attribute in T_2, they are treated as different attributes in T).
  - For every tuple $t_1 \in T_1$ and $t_2 \in T_2$, T contains a tuple t whose values are the same as t_1 (t_2) on the attributes from T_1 (T_2).

Cartesian product $(T \leftarrow T_1 \times T_2)$ - in another word:
The output of the operations is a table T with every combination of a tuple from T_1 concatenated to a tuple from T_2

- T's schema is every attribute from T_1 followed by every attribute of T_2, in order
- How many tuples are in $T_1 \times T_2$?
- If an attribute occurs in both tables, it occurs twice in the result

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

$\text{PROF} \times \text{TEACH}$ 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

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