Relational Algebra

COMP9311 25T3; Week 2.2

By Xubo Wang, UNSW

Motivation

- We’ve seen what a relational model is.
- We needed a *formal* language to specify data (tuples) from the relational model.
- **Relational Algebra** (E.F. Codd (1970))

Why Relational Algebra?

- It provides a formal foundation for relational model operations
- It is used as a basis for implementing and optimizing queries in the query processing and optimization
- Some of its concepts are incorporated into SQL

Relational Algebra

Relational Algebra is a procedural data manipulation language (**DML**).

It specifies operations on relations to define new relations:

Unary Relational Operations: Select, Project

Operations from Set Theory: Union, Intersection, Difference, Cartesian Product

Binary Relational Operations: Join, Divide.

1 SELECT

The SELECT operation/predicate is used to select a subset of the tuples of a relation R, satisfying some conditions.

Notation: $\sigma_{<selection\ condition>}(R)$

Intuition: Filters out all tuples that do not satisfy select condition

Selection Condition

The condition is defined by a ***selection clause***:

- <attribute> *operator* <constant>
- < attribute > *operator* < attribute >

Where *operator* is one of =, <, ≤, >, ≥ or ≠ …

Example:

- age ≤ 24
- commission ≥ 24 000

Selection Condition

Selection clauses can also be

- <expression> operator <expression>

With this, we can use **Boolean connectives** as operators

- C1 AND C2
- C1 OR C2
- NOT C

Terms equivalently expressed by $\wedge$ (and), $\vee$(or), $\neg$ (not)

Q: Select the enrolment records for the students whose supervisor is Person 1

ENROLMENT:

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

$\sigma_{(Supervisor=1)}(ENROLMENT)$

The output relation is

Enrolment#	Supervisee	Supervisor	Department	Degree
2	3	1	Comp.Sci	Ph.D.
3	4	1	Comp.Sci	M.Sc.
4	5	1	Comp.Sci	M.Sc.

Q: Select the enrolment records for Person 1's non-Ph.D. students

ENROLMENT:

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

$$\left.\begin{array}{l} \sigma_{(Supervisor=1\ AND\ Degree\neq"Ph.D.")}(ENROLMENT) \\ \sigma_{(Supervisor=1\ AND\ \ NOT\ Degree="Ph.D.")}(ENROLMENT) \end{array}\right\} \text{Same}$$

The output relation is

Enrolment#	Supervisee	Supervisor	Department	Degree
3	4	1	Comp.Sci	M.Sc.
4	5	1	Comp.Sci	M.Sc.

Properties of Selection

Properties:

- Consecutive selects ***can be combined***:

$$\sigma_{<cond1>}\left(\sigma_{<cond2>}(R)\right) = \sigma_{<cond1>\ AND\ <cond2>}(R)$$

- Selection is a ***commutative*** operation:

$$\sigma_{<cond1>}\left(\sigma_{<cond2>}(R)\right) = \sigma_{<cond2>}(\sigma_{<cond1>}(R))$$

2 PROJECT

The PROJECT operation is used to project a subset of the attributes (columns) of a relation, denoted by:

General form: $\pi_{<attribute\ list>}(R)$

Result:

- schema: attribute list ($A_1,\ldots,A_k$)
- instance: the set of all subtuples $t[A_1,\ldots,A_k]$ where $t \in R$

Q: Find **departments** and **degree** requirements for the courses that students enroll.

ENROLMENT:

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

$$\pi_{\{department,\, degree\}}(ENROLMENT)$$

The output relation is

Department	Degree
Psychology	Ph.D.
Comp.Sci	Ph.D.
Comp.Sci	M.Sc.

Duplicates of PROJECT

ENROLMENT:

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

Question: What if we do PROJECTION on only department?

Department
Psychology
Comp.Sci.
Comp.Sci.
Comp.Sci.

or

Department
Psychology
Comp.Sci.

?

Duplicates of PROJECT

ENROLMENT:

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

Question: What if we do PROJECTION on only department?

Answer: Keep only **one** 'Comp.Sci.'.

Department
Psychology
Comp.Sci.

Relational Algebra is based on *sets*, so no duplicates are allowed.

- The PROJECT operation *removes any duplicate tuples*, so the result of the PROJECT operation is a set of distinct tuples, and this is known as **duplicate elimination**.

Properties of PROJECT

Consider $\pi_{<list1>}(\pi_{<list2>}(R))$

If <list2> contains all the attributes in <list1> :

Then $\pi_{<list1>}\left(\pi_{<list2>}(R)\right) = \pi_{<list1>}(R)$

Else the operation is *not well defined*.

Project Predicate

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

Question: is projection commutative with selection?

i.e., $\pi_X\left(\sigma_B\ (R)\right) = \sigma_B\left(\pi_X(R)\right)$?

Consider the following:

$\pi_{\{degree\}}(\sigma_{(Department='Psychology')}(ENROLMENT))$

Degree
Ph.D.

$\sigma_{(Department='Psychology')}(\pi_{\{degree\}}(ENROLMENT))$ Error as SELECT cannot find Department

Answer: **The attribute used in SELECT must be a subset of the attribute list in PROJECT**

Project Predicate

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

Question: is projection commutative with selection?

i.e., $\pi_X\left(\sigma_B\ (R)\right) = \sigma_B\left(\pi_X(R)\right)$?

Department	Degree
Psychology	Ph.D.

Consider the following:

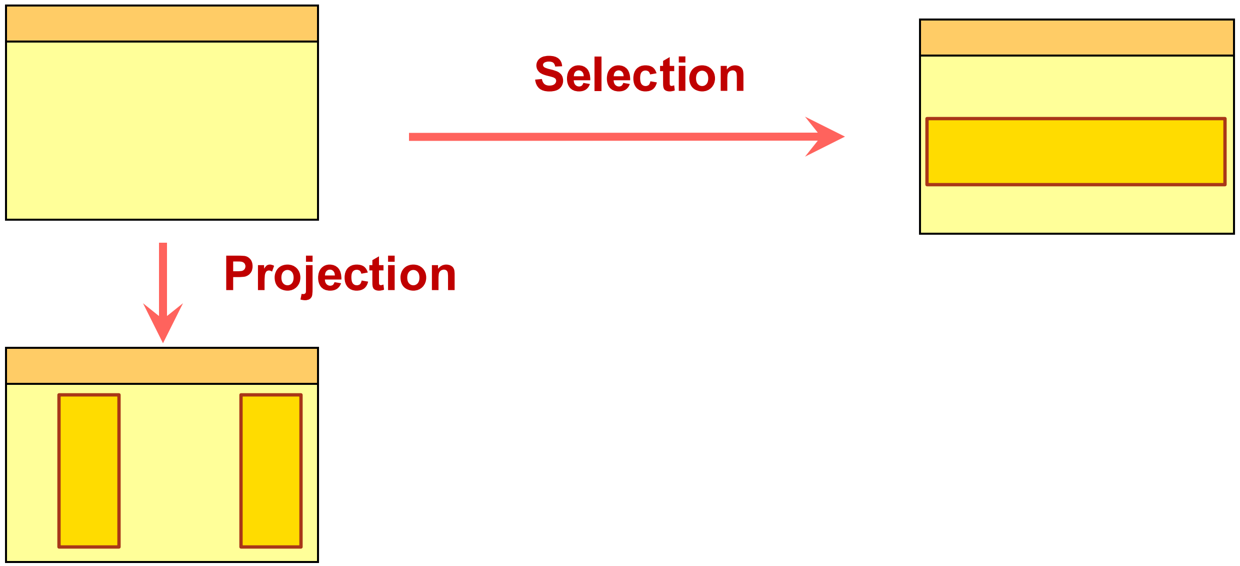
$\pi_{\{Department,Degree\}}(\sigma_{(Department='Psychology')}(ENROLMENT))$

$\sigma_{(Department='Psychology')}(\pi_{\{Department,Degree\}}(ENROLMENT))$

Answer: **The attribute used in SELECT must be a subset of the attribute list in PROJECT**

Intuition: Projection and Selection

1. Selection performs a horizontal decomposition, and
2. Projection performs a vertical decomposition



3 SET UNION

UNION is the set-theoretic union of the tuples of two relations.

$$R \cup S = \{t : t \in R \ or \ t \in S\}$$

Condition: R and S must be **union compatible!**

Union compatibility: there is a 1-1 correspondence between their attributes: the same **name** and same **domain**.

Example: Section(course_id, semester, year, instructor)

To find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both:

$$\pi_{\{course\_id\}}(\sigma_{(semester="Fall" \wedge year=2009)}(section)) \cup$$
$$\pi_{\{course\_id\}}(\sigma_{(semester="Spring" \wedge year=2010)}(section))$$

Example

STUDENT:

Person#	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

Example: *STUDENT* ∪ *RESEARCHER* =

Person#	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledhill
5	Ms B.K.Lee
2	Dr R.G.Wilkinson

4 SET INTERSECTION

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

- *INTERSECTION* is an operation that includes all tuples that are present in **both** relations, denoted by

$$R \cap S = \{t : t \in R \text{ and } t \in S\}$$

- Condition: R and S must also be **union compatible!**
- Example:

$$R_1 \leftarrow \sigma_{(supervisor=1)}(ENROLMENT)$$

$$R_2 \leftarrow \sigma_{(degree='Ph.D.')}(ENROLMENT)$$

$$R_1 \cap R_2 =$$

Enrolment#	Supervisee	Supervisor	Department	Degree
2	3	1	Comp.Sci.	Ph.D.

Example of Intersection

STUDENT:

Person#	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

Example: STUDENT ∩ RESEARCHER =

Person#	Name
1	Dr C.C. Chen

5 SET DIFFERENCE

DIFFERENCE is an operation that includes all tuples that are in the left relation but not in the right relation, denoted by

$$R - S = \{t : t \in R \ and \ t \notin S\}$$

Condition: R and S must also be **union compatible!**

STUDENT:

Person#	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

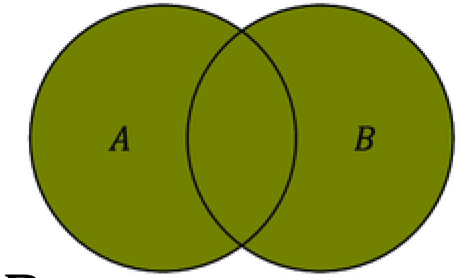
Example: STUDENT – RESEARCHRER =

Person#	Name
3	Ms K. Juliff
4	Ms J. Gledhill
5	Ms B.K. Lee

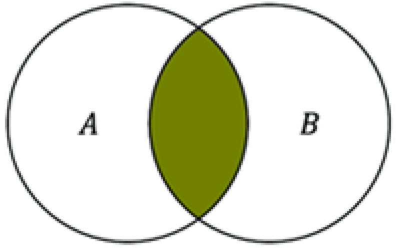
Summary

Operations on Relations

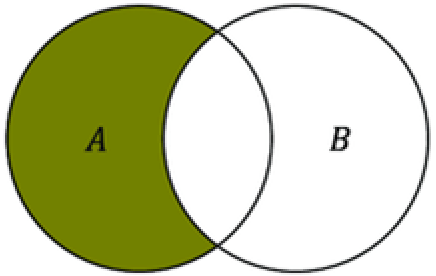
- *UNION:* $A \cup B$



- *INTERSECTION:* $A \cap B$



- *DIFFERENCE:* $A - B$



Express: The names of persons who are either a student **or** a researcher

STUDENT:

Person#	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

Express: The names of persons who are either a student **or** a researcher

STUDENT:

Person#	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

$$\pi_{\{name\}}(STUDENT \cup RESEARCHER)$$

Name
Dr C.C.Chen
Dr R.G.Wilkinson
Ms K.Juliff
Ms J.Gledill
Ms B.K.Lee

Express: The names of persons who are a student **and** a researcher

STUDENT:

Person#	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

Express: The names of persons who are a student **and** a researcher

STUDENT:

Person#	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

$$\pi_{\{name\}}(STUDENT \cap RESEARCHER)$$

Name
Dr C.C.Chen

Express: The names of persons who are a student **but not** a researcher

STUDENT:

Person#	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

Express: The names of persons who are a student **but not** a researcher

STUDENT:

Person#	Name
1	Dr C.C.Chen
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

$$\pi_{\{name\}}(STUDENT - RESEARCHER)$$

Name
Ms K.Juliff
Ms J.Gledill
Ms B.K.Lee

Express: The departments and degrees of Courses which are not enrolled by any student

ENROLMENT:

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

COURSE:

Department	Degree
Psychology	Ph.D.
Comp.Sci.	Ph.D.
Comp.Sci.	M.Sc.
Psychology	M.Sc.

Express: The departments and degrees of Courses which are not enrolled by any student

ENROLMENT:

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

COURSE:

Department	Degree
Psychology	Ph.D.
Comp.Sci.	Ph.D.
Comp.Sci.	M.Sc.
Psychology	M.Sc.

$$COURSE - (\pi_{(Department,Degree)}(ENROLMENT))$$

CARTESIAN PRODUCT

$$R \times S = \{t_1 || \, t_2 \, : t_1 \in R \; and \; t_2 \in S\}$$

- Intuition: **every combination of tuples in R with tuples in S**.
- $t_1 \; || \; t_2$ indicates the concatenation of tuples.
- R and S not required to be union compatible, but
- The number of tuples in the output relations is always $\underline{|R| * |S|}$

Usually assumes that attributes of $r(R)$ and $s(S)$ are disjoint. (That is, $R \cap S = \emptyset$). If not, you must devise a naming schema to distinguish between the attribute names if they are the same for example attribute A in $r(A, B)$ and $s(A, C)$, by attaching the relation's name, $r.A$ and $s.A$ (known as dot-notation)

Example of cartesian product

ENROLMENT:

<u>Enrolment#</u>	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

RESEARCHER:

<u>Person#</u>	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

Example of cartesian product

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

ENROLMENT X RESEARCHRER =

E'ment#	S'ee	S'or	D'ment	Degree	Person#	Name
1	1	2	Psych.	Ph.D.	1	Dr C.C. Chen
1	1	2	Psych.	Ph.D.	2	Dr R.G.Wilkinson
2	3	1	Cmp.Sci	Ph.D.	1	Dr C.C. Chen
2	3	1	Cmp.Sci	Ph.D.	2	Dr R.G.Wilkinson
3	4	1	Cmp.Sci	M.Sc.	1	Dr C.C. Chen
3	4	1	Cmp.Sci	M.Sc.	2	Dr R.G.Wilkinson
4	5	1	Cmp.Sci	M.Sc.	1	Dr C.C. Chen
4	5	1	Cmp.Sci	M.Sc.	2	Dr R.G.Wilkinson

There were 4 tuples in ENROLMENT and 2 tuples in RESEARCHER. In the result, there are 8 tuples.

Useful if we add a condition

$$R_1 \leftarrow ENROLMENT \times RESEARCHER$$

E'ment#	S'ee	S'or	D'ment	Degree	Person#	Name
1	1	2	Psych.	Ph.D.	1	Dr C.C. Chen
1	1	2	Psych.	Ph.D.	2	Dr R.G.Wilkinson
2	3	1	Cmp.Sci	Ph.D.	1	Dr C.C. Chen
2	3	1	Cmp.Sci	Ph.D.	2	Dr R.G.Wilkinson
3	4	1	Cmp.Sci	M.Sc.	1	Dr C.C. Chen
3	4	1	Cmp.Sci	M.Sc.	2	Dr R.G.Wilkinson
4	5	1	Cmp.Sci	M.Sc.	1	Dr C.C. Chen
4	5	1	Cmp.Sci	M.Sc.	2	Dr R.G.Wilkinson

In practice it's useful if we give a cartesian product specified condition

$$\sigma_{(Supervisor=Person\#)}(R_1) =$$

E'ment#	S'ee	S'or	D'ment	Degree	Person#	R'cher. Name
1	1	2	Psych.	Ph.D.	2	Dr R.G.Wilkinson
2	3	1	Cmp.Sci.	Ph.D.	1	Dr C.C. Chen
3	4	1	Cmp.Sci.	M.Sc.	1	Dr C.C. Chen
4	5	1	Cmp.Sci.	M.Sc.	1	Dr C.C. Chen

More useful if we add a projection

$R_1 \leftarrow ENROLMENT \times RESEARCHER$

$R_2 \leftarrow \sigma_{(Supervisor=Person\#)}(R_1)$

E'ment#	S'ee	S'or	D'ment	Degree	Person#	R'cher. Name
1	1	2	Psych.	Ph.D.	2	Dr R.G.Wilkinson
2	3	1	Cmp.Sci.	Ph.D.	1	Dr C.C. Chen
3	4	1	Cmp.Sci.	M.Sc.	1	Dr C.C. Chen
4	5	1	Cmp.Sci.	M.Sc.	1	Dr C.C. Chen

$\pi_{\{E'ment\#, S'ee, S'or, Name, D'ment, Degree\}}(R_2)$

E'ment#	S'ee	S'or	Name	D'ment	Degree
1	1	2	Dr R.G.Wilkinson	Psych.	Ph.D.
2	3	1	Dr C.C. Chen	Comp.Sci.	Ph.D.
3	4	1	Dr C.C. Chen	Comp.Sci.	M.Sc.
4	5	1	Dr C.C. Chen	Comp.Sci.	M.Sc.

The two equal attributes occur only once

The last of these is also known as *natural join*, the next to last is *equi-join*.

6 JOIN

- JOIN is used to combine related tuples from two relations into single "longer" tuples.
- **Theta-join**

$$R \bowtie_{<join\ condition>} S = \{t_1 \,||\, t_2 : t_1 \in R \ and \ t_2 \in S \ and < join\ condition >\}$$

- A general join condition is of the form:

  <condition> **AND** <condition> **AND** … **AND** <condition>

<condition> is of the form $A_i \, \theta \, B_j$, A_i is an attribute of R, B_j is an attribute of S, A_i and B_j have the same domain, and θ (theta) is one of the comparison operators {=, <, ≤, >, ≥, ≠}.

6.1 Equi-join

A type of theta-join where the only comparison operator used is "=" is called an Equi-join

Example:

$$ENROLMENT \bowtie_{(Supervisor=Person\#)} RESEARCHER$$

$$R_2 \leftarrow \sigma_{(Supervisor=Person\#)}(R_1)$$

E'ment#	S'ee	S'or	D'ment	Degree	Person#	R'cher. Name
1	1	2	Psych.	Ph.D.	2	Dr R.G.Wilkinson
2	3	1	Cmp.Sci.	Ph.D.	1	Dr C.C. Chen
3	4	1	Cmp.Sci.	M.Sc.	1	Dr C.C. Chen
4	5	1	Cmp.Sci.	M.Sc.	1	Dr C.C. Chen

6.2 Natural Join

ENROLMENT:

Enrolment#	Supervisee	Supervisor	Department	Degree
1	1	2	Psychology	Ph.D.
2	3	1	Comp.Sci.	Ph.D.
3	4	1	Comp.Sci.	M.Sc.
4	5	1	Comp.Sci.	M.Sc.

COURSE:

Department	Degree
Psychology	Ph.D.
Comp.Sci.	Ph.D.
Comp.Sci.	M.Sc.
Psychology	M.Sc.

A type of equi-join that requires each pair of join attributes to have the same name and domain in both relations.

Notes: In a natural join, there may be several valid pairs of join attributes.

$$ENROLMENT \bowtie_{(Department,\, Degree),(Deparment,Degree)} COURSE$$

If there are pairs of joining attributes identically named, we can write

$$ENROLMENT \bowtie COURSE$$

Note: this notion also acceptable if there’s one join attribute

6.2 Natural Join

Intuitions:

- Enforce equality on all attributes with same name
- Eliminate one copy of duplicated attributes

E’ment#	S’ee	S’or	Name	D’ment	Degree
1	1	2	Dr R.G.Wilkinson	Psych.	Ph.D.
2	3	1	Dr C.C. Chen	Comp.Sci.	Ph.D.
3	4	1	Dr C.C. Chen	Comp.Sci.	M.Sc.
4	5	1	Dr C.C. Chen	Comp.Sci.	M.Sc.

JOINS

Remember the differences between the types of joins:

1. Theta JOIN $R \bowtie_{<join\ condition>} S$

2. Equi JOIN

3. Natural JOIN

Note: all denoted with $\bowtie$

Pracs

STUDENT:

Person#	Name
1	Mr J.He
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

COURSE

Depart	Degree
EE	PhD
CS	PhD
EE	MSc
CS	MSc

ENROLMENT:

Enrol#	Supervisee	Supervisor	Depart	Degree
1	1	2	EE	PhD
2	3	1	CS	PhD
3	4	1	CS	MSc
4	5	1	CS	MSc

What are the names of students who are studying for an MSc in CS?

Pracs

STUDENT:

Person#	Name
1	Mr J.He
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

COURSE

Depart	Degree
EE	PhD
CS	PhD
EE	MSc
CS	MSc

ENROLMENT:

Enrol#	Supervisee	Supervisor	Depart	Degree
1	1	2	EE	PhD
2	3	1	CS	PhD
3	4	1	CS	MSc
4	5	1	CS	MSc

What are the names of students who are studying for an MSc in CS?

$$\pi_{\{name\}}(\sigma_{(degree=MSc\ and\ Depart=CS)}\ ENROLMENT \bowtie_{supervisee=person\#} Student)$$

Pracs

STUDENT:

Person#	Name
1	Mr J.He
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

COURSE

Depart	Degree
EE	PhD
CS	PhD
EE	MSc
CS	MSc

ENROLMENT:

Enrol#	Supervisee	Supervisor	Depart	Degree
1	1	2	EE	PhD
2	3	1	CS	PhD
3	4	1	CS	MSc
4	5	1	CS	MSc

The IDs of students who are supervised by Dr C.C.Chen

Pracs

STUDENT:

Person#	Name
1	Mr J.He
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

COURSE

Depart	Degree
EE	PhD
CS	PhD
EE	MSc
CS	MSc

ENROLMENT:

Enrol#	Supervisee	Supervisor	Depart	Degree
1	1	2	EE	PhD
2	3	1	CS	PhD
3	4	1	CS	MSc
4	5	1	CS	MSc

The IDs of students who are supervised by Dr C.C.Chen

$$R1 = \text{ENROLMENT} \bowtie \text{(supervisor=person\#) RESEARCHER}$$
$$R2 = \sigma_{(\text{name=Dr C.C.Chen})}R1$$
$$R3 = \boldsymbol{\pi}_{\{\text{supervisee}\}}R2$$

Divide

- The DIVISION operation is applied to two Relations R and S, where the attributes of S are a subset of the attributes of R.
- The relation returned by the division operator will have attributes = (All attributes of R – All Attributes of S)
- Return all tuples from relation R which are associated to every S's tuple.

R

A	B
a_1	b_1
a_1	b_2
a_2	b_1
a_3	b_2
a_4	b_1
a_5	b_1
a_5	b_2

S

B
b_1
b_2

$R \div S =$

A
a_1
a_5

Divide

Typical use: which departments offer all degrees?

$$Course \div (\pi_{Degree} Course)$$

COURSE

<u>Depart</u>	<u>Degree</u>
EE	PhD
CS	PhD
EE	MSc
CS	MSc

Pracs

STUDENT:

Person#	Name
1	Mr J.He
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

COURSE

Depart	Degree
EE	PhD
CS	PhD
EE	MSc
CS	MSc

ENROLMENT:

Enrol#	Supervisee	Supervisor	Depart	Degree
1	1	2	EE	PhD
2	3	1	CS	PhD
3	4	1	CS	MSc
4	5	1	CS	MSc

The names of supervisor who supervises both MSc and PhD students

Pracs

STUDENT:

Person#	Name
1	Mr J.He
3	Ms K.Juliff
4	Ms J.Gledill
5	Ms B.K.Lee

RESEARCHER:

Person#	Name
1	Dr C.C.Chen
2	Dr R.G.Wilkinson

COURSE

Depart	Degree
EE	PhD
CS	PhD
EE	MSc
CS	MSc

ENROLMENT:

Enrol#	Supervisee	Supervisor	Depart	Degree
1	1	2	EE	PhD
2	3	1	CS	PhD
3	4	1	CS	MSc
4	5	1	CS	MSc

The names of supervisor who supervises both MSc and PhD students

$R1 = \pi_{\{SUPERVISOR,DEGREE\}}$ ENROLMET $\div$ $\pi_{\{DEGREE\}}$ COURSE

$R2 = \pi_{\{Name\}}$ (R1 $\bowtie_{(supervisor=person\#)}$ RESEARCH)

Exercise

R:

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_1	c_3
a_1	b_2	c_2
a_2	b_1	c_1
a_2	b_2	c_2
a_3	b_1	c_1
a_3	b_2	c_1
a_3	b_2	c_2

S:

B	C
b_1	c_1
b_1	c_2

Write relational algebra that retrieves:

1. Find A of *R* that contains all *S*.

Exercise

R:

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_1	c_3
a_1	b_2	c_2
a_2	b_1	c_1
a_2	b_2	c_2
a_3	b_1	c_1
a_3	b_2	c_1
a_3	b_2	c_2

S:

B	C
b_1	c_1
b_1	c_2

Write relational algebra that retrieves:

1. Find A of *R* that contains all *S*.

$$R \div S$$

A
a_1

Exercise

R:

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_1	c_3
a_1	b_2	c_2
a_2	b_1	c_1
a_2	b_2	c_2
a_3	b_1	c_1
a_3	b_2	c_1
a_3	b_2	c_2

S:

B	C
b_1	c_1
b_1	c_2

Write relational algebra that retrieves:

2. Find (A, B) of *R* that contains all C of *S*.

Exercise

R:

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_1	c_3
a_1	b_2	c_2
a_2	b_1	c_1
a_2	b_2	c_2
a_3	b_1	c_1
a_3	b_2	c_1
a_3	b_2	c_2

S:

B	C
b_1	c_1
b_1	c_2

Write relational algebra that retrieves:

2. Find (A, B) of *R* that contains all C of *S*.

$$R \div \pi_{\{C\}}(S)$$

A	B
a_1	b_1
a_3	b_2

Rename Operator

- The ***rename*** operator ρ changes the name of one or more attributes
- Change the names in a schema
- Does not affect instance of the target relation

Family

Father	Child
Adam	Abel
Adam	Cain
Abraham	Isaac

$\rho_{(Parent,\ Child)}$ (Family)

Parent	Child
Adam	Abel
Adam	Cain
Abraham	Isaac

$\rho_{S(A1,A2,\ldots,An)}(R)$ or $\rho_S(R)$ or $\rho_{(A1,A2,\ldots,An)}(R)$

- Why might this be useful? To be included in relational algebra?

Why RENAME Operator?

- To unify schemas for set operators
  - **Union compatibility**: there is a 1-1 correspondence between their attributes: the same **name** and same **domain**.

$$R(zid, name), S(studentid, name)$$

$$R \cup \rho_{(zid,name)} S$$

- For disambiguation in "self-join"

$$STUDENT \bowtie_{STUDENT.major = STUDENT.major} STUDENT$$

$$\rho_{S1}(STUDENT) \bowtie_{S1.major = S2.major} \rho_{S2}(STUDENT)$$

Basic vs Extended Operators

Note: $\{\sigma,\ \pi,\ \cup,\ -,\ \times\}$ (*and rename*) are sufficient to define all these operations: this is a relationally complete set of operators. These are the ***basic operators*** of the Relational Algebra.

What about *JOIN*, *INTERSECTION* and *DIVIDE*?
They are ***extended operators*** because they can be derived from the basic operators.

$$R \bowtie_{<\text{ condition }>} S \;=\; \sigma_{<\text{ condition }>}(R \times S)$$

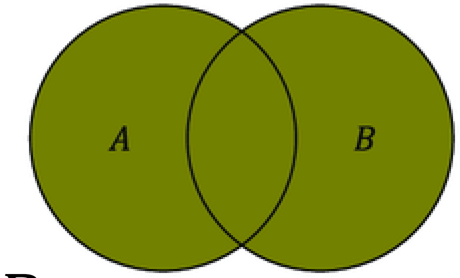
$$A \cap B = A - (A - B)$$

Summary

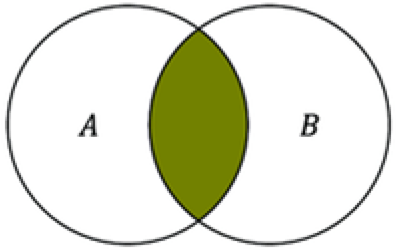
$A \cap B = A - (A - B)$

Operations on Relations

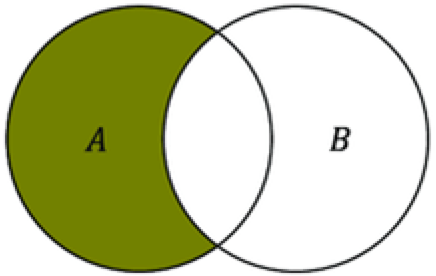
- *UNION:* $A \cup B$



- *INTERSECTION:* $A \cap B$



- *DIFFERENCE:* $A - B$



Basic vs Extended Operators

Note: $\{\sigma, \pi, \cup, -, \times\}$ (*and rename*) are sufficient to define all these operations: this is a relationally complete set of operators. These are the ***basic operators*** of the Relational Algebra.

What about *JOIN*, *INTERSECTION* and *DIVIDE*?
They are ***extended operators*** because they can be derived from the basic operators.

E.g., We can write $R \div S$ as

$$TEMP1 \leftarrow \pi_{R-S}(R)$$

$$TEMP2 \leftarrow \pi_{R-S}((TEMP1 \times S) - R)$$

$$RESULT = TEMP1 - TEMP2$$

- *R-S: attributes in R not in S*
- The result to the right of ← is assigned to the relation variable on the left of ←.
- May use variable in subsequent expressions.

Aggregate Operators

What if we want a relation with information about "sum of salaries" of employees, or the "average age" of students?

We need more expressive power. We can use ***aggregation functions*** to summarize information from multiple tuples into ***aggregate values.***

We can use an **aggregation operator γ** and a function such as *SUM, AVG, MIN, MAX, or COUNT*.

In general, duplicates a t considered in the aggregation.

If R =

A	B
1	2
3	4
3	5
1	1

, then $\gamma_{SUM(A)}(R)$ =

SUM(A)
8

and $\gamma_{AVG(B)}(R)$ =

AVG(B)
3

Aggregate Operators

We can also retrieve aggregate values for groups, like the “sum of employee salaries” *per department* or the “average student age” *per faculty*.

We give γ additional arguments to specify that the aggregation behavior should be based on *groups* (defined by a set of attributes).

If $R =$

a	b
1	2
3	4
3	5
1	3

, then $\gamma_{a,SUM(b)}(R) =$

a	SUM(b)
1	5
3	9

Formal Definition

A basic relational algebra expression is one of the following:

- A relation in the database
- (could also be a) constant relation
  - (fixed set of tuples, e.g., {(1,2), (1,3), (2,3)})

A general relational algebra expression is constructed out of smaller subexpressions.

Let E_1 and E_2 be relational algebra expressions; the following are all relational-algebra expressions:

- $E_1 \cup E_2$
- $E_1 - E_2$
- $E_1 \times E_2$
- $\sigma_P(E_1)$ where P is predicate on attributes in E_1
- $\pi_S(E_1)$ where S is a set of attributes in E_1
- $\rho_X(E_1)$ where X is the new name for the result of E_1

OPERATION	PURPOSE	NOTATION
SELECT	Selects all tuples that satisfy the selection condition from a relation R	$\sigma_{<selection\ condition>}(R)$
PROJECT	Produces a new relation with only some of the attributes of R and removes duplicate tuples.	$\pi_{<attribute\ list>}(R)$
THETA-JOIN	Produces all combinations of tuples from R and S that satisfy the join condition.	$R \bowtie_{<join\ condition>} S$
EQUI-JOIN	Produces all the combinations of tuples from R and S that satisfy a join condition with only equality comparisons.	$R \bowtie_{<join\ condition>} S$
NATURAL-JOIN	Same as EQUIJOIN except that the join attributes of S are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R \bowtie_{<join\ condition>} S$
UNION	Produces a relation that includes all the tuples in R or S or both R and S; R and S must be union compatible.	$R \cup S$
INTERSECTION	Produces a relation that includes all the tuples in both R and S; R and S must be union compatible.	$R \cap S$
DIFFERENCE	Produces a relation that includes all the tuples in R that are not in S; R and S must be union compatible.	$R - S$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R and S and includes as tuples all possible combinations of tuples from R and S.	$R \times S$
DIVISION	Produces a relation T(X) that includes all tuples t[X] in R(Z) that appear in R in combination with every tuple from S(Y), where Z = X ∪ Y.	$R(Z) \div S(Y)$