CS 4092 Database Design and Development (DDD)

03: Formal Relational Query Language (Part 1)

Seokki Lee

Slides are adapted from:

Relational Query Languages

- What is query language?

Relational Query Languages

- What is query language?
  - A language in which a user requests information from database

Relational Query Languages

- "Pure" languages:
  - Relational algebra
  - Tuple relational calculus
  - Domain relational calculus
- Expressive power of a query language
  - What queries can be expressed in this language?

Topics

- Formal Relational Query Language
  - **Relational Algebra**
  - Tuple Relational Calculus
  - Domain Relational Calculus

Relational Algebra (RA)

- Consists of a set of operators that take one or more relations as inputs and produce new relations as output
  - Composable
- The output of evaluating an expression in RA can be used as input to another relational algebra expression.

Relational Algebra (RA)

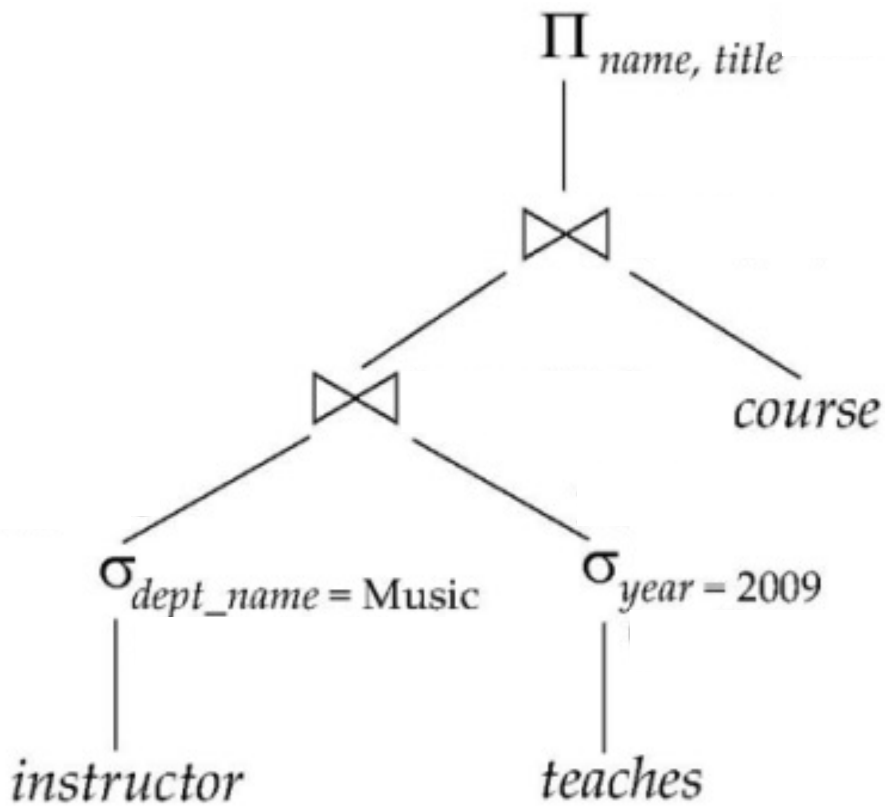
- Six basic operators
  - select: σ
  - project: Π
  - union: $\cup$
  - set difference: –
  - cartesian product: x
  - rename: ρ

Unary

Binary

Relational Algebra (RA)

- Example



Relational Algebra (RA)

- Six basic operators
  - select: σ
  - project: Π
  - union: $\cup$
  - set difference: –
  - cartesian product: x
  - rename: ρ
- Why is RA important?

Relational Algebra (RA)

- Six basic operators
  - select: σ
  - project: Π
  - union: $\cup$
  - set difference: –
  - cartesian product: x
  - rename: ρ
- Why is RA important?
  - Strong formal foundation that is fairly simple
  - Widely used for query optimization

Select Operation – Example

- Relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

- What is the RA expression?

A	B	C	D
α	α	1	7
β	β	23	10

Select Operation

- Filtering out a subset of tuples in a relation *r*
- Notation: $\sigma_{p}(r)$
- *p* is called the **selection predicate**
- Defined as:

$$\sigma_p(r) = \{t \mid t \in r \wedge p(t)\}$$

  - Where *p* is a formula in propositional calculus consisting of **terms** connected by : $\wedge$ (**and**), $\vee$ (**or**), $\neg$ (**not**)
  - Each **term** is one of:

<attribute> *op* <attribute> or <constant>

where *op* is one of: $=, \neq, >, \geq, <, \leq$

Select Operation – Example

- Relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

- What is the RA expression?

A	B	C	D
α	α	1	7
β	β	23	10

Select Operation – Example

- Relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

- $\sigma_{A = B \wedge D > 6}(r)$

A	B	C	D
α	α	1	7
β	β	23	10

Select Operation – Example

- Instructors in the "Physics" department

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Result

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

Select Operation – Example

- Instructors in the "Physics" department
  - $\sigma_{\,dept\_name=\text{“Physics”}}(instructor)$

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Result

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

Select Operation – Example

- Instructors in the “Physics” department with a salary greater than $90,000

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Result

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000

Select Operation – Example

- Instructors in the "Physics" department with a salary greater than $90,000
  - $\sigma_{\textit{dept\_name}=\text{"Physics"} \wedge \textit{salary} > 90000}(\textit{instructor})$

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Result

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000

Select Operation – Example

- Instructors in the "Physics" or "Computer Science" department

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Result

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
33456	Gold	Physics	87000
10101	Srinivasan	Comp. Sci.	65000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000

Select Operation – Example

- Instructors in the "Physics" or "Computer Science" department
  - $\sigma_{\text{dept\_name}=\text{"Physics"} \vee \text{dept\_name}=\text{"Comp.Sci."}}(\textit{instructor})$

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Result

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
33456	Gold	Physics	87000
10101	Srinivasan	Comp. Sci.	65000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000

Project Operation – Example

- Relation *r*

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

- What is the RA expression?

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Project Operation

- Manipulates attributes
- Notation: $\prod_{A_1, A_2, \ldots, A_k} (r)$

  where A_1, A_2 are attribute names and r is a relation name.
- The result is defined as the relation of ***k*** columns obtained by erasing the columns that are not listed
- **Duplicate rows removed** from result, since relations are sets
- Let A be a subset of the attributes of relation r then:

$$\pi_A(r) = \{t.A \mid t \in r\}$$

Project Operation – Example

- Eliminate the *dept_name* attribute of *instructor*

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Result

ID	*name*	*salary*
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

Project Operation – Example

- Eliminate the *dept_name* attribute of *instructor*
  - $\Pi_{ID,\ name,\ salary}$ (*instructor*)

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Result

ID	*name*	*salary*
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

Composition of Operations – Example

- Find the name of all instructors in the Physics department

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Result

name
Einstein
Gold

Composition of Operations – Example

- Find the name of all instructors in the Physics department
  - $\Pi_{name}(\sigma_{dept\_name=\text{“Physics”}}(instructor))$

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Result

name
Einstein
Gold

Union Operation – Example

- Relations *r, s:*

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

- What is the RA expression?

A	B
α	1
α	2
β	1
β	3

Union Operation

- Combines the tuples of two relations
- Notation: $r \cup s$
- Defined as:

$$r \cup s = \{t \mid t \in r \vee t \in s\}$$

- For $r \cup s$ to be valid

Union Operation

- Combines the tuples of two relations
- Notation: $r \cup s$
- Defined as:

$$r \cup s = \{t \mid t \in r \vee t \in s\}$$

- For $r \cup s$ to be valid
  1. *r, s* must have the *same* **arity** (same number of attributes)
  2. The attribute domains must be **union compatible**
     - Example: 2nd column of *r* deals with the same type of values as does the 2nd column of *s*

Union Operation – Example

- Relations *r, s:*

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

- $r \cup s$

A	B
α	1
α	2
β	1
β	3

Union Operation – Example

- Find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both

Section

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

Result

course_id
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

Union Operation – Example

- Find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both
  - $\Pi_{course\_id}\,(\sigma_{\,semester=\text{“Fall”}\ \wedge\ year=2009}\,(section))$
    $\cup\ \Pi_{course\_id}\,(\sigma_{\,semester=\text{“Spring”}\ \wedge\ year=2010}\,(section))$

Section

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

Result

course_id
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

Set difference of two relations

- Relations *r*, *s*:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- What is the RA expression?

A	B
α	1
β	1

Set Difference Operation

- Allows to find tuples that are in one relation but not in another
- Notation *r* – *s*
- Defined as:

$$r - s = \{t \mid t \in r \wedge t \notin s\}$$

- Set differences must be taken between **compatible** relations.
  - *r* and *s* must have the **same** arity
  - Attribute domains of *r* and *s* must be compatible

Set Difference Operation – Example

- Find all courses taught in the Fall 2009 semester, but not in the Spring 2010 semester

Section

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

Result

course_id
CS-347
PHY-101

Set Difference Operation – Example

- Find all courses taught in the Fall 2009 semester, but not in the Spring 2010 semester
  - $\Pi_{course\_id} (\sigma_{semester="Fall" \land year=2009} (section))$
    $- \Pi_{course\_id} (\sigma_{semester="Spring" \land year=2010} (section))$

Section

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

Result

course_id
CS-347
PHY-101

Cartesian-Product Operation – Example

- Relations *r, s*:

r

A	*B*
α	1
β	2

s

C	*D*	*E*
α	10	a
β	10	a
β	20	b
γ	10	b

- *r* x *s*:

A	*B*	*C*	*D*	*E*
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

Cartesian-Product Operation

- Combines any two relations, computing all combinations of tuples
- Notation *r* x *s*
- Defined as:

$$r \times s = \{t, t' \mid t \in r \wedge t' \in s\}$$

- Assume that attributes of r(R) and s(S) are **disjoint**.
  - $R \cap S = \varnothing$

Cartesian-Product Operation

- Combines any two relations, computing all combinations of tuples
- Notation *r* x *s*
- Defined as:

$$r \times s = \{t, t' \mid t \in r \land t' \in s\}$$

- Assume that attributes of r(R) and s(S) are **disjoint**.
  - $R \cap S = \emptyset$
- If attributes of *r(R)* and *s(S*) are not disjoint
  - then renaming must be used.

Cartesian-Product Operation – Example

instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

Instructor x teaches

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

Composition of Operations

- *r x s*

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

- What is the RA Expression?

A	B	C	D	E
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b

Composition of Operations

- $r \times s$

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

- $\sigma_{A=C}(r \times s)$

A	B	C	D	E
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b

Relational Algebra (RA)

- Six basic operators
  - select: σ
  - project: Π
  - union: $\cup$
  - set difference: –
  - cartesian product: x
  - rename: ρ

Covered

Rename Operation – Example

- Relations *r*

A	B
α	1
β	2

r

- What is the RA expression?

A	D
α	1
β	2

d

Rename Operation

- Allows to name, and therefore to refer to, the results of RA expressions
- Notation: $\rho_X(r)$

  returns the expression *E* under the name *X*
- If an RA expression *E* has arity *n*, then

$$\rho_{x(A_1, A_2, \ldots, A_n)}(r)$$

returns the result of expression *E* under the name *X*, and with the attributes renamed to $A_1, A_2, \ldots, A_n$.

$$\rho_X(r) = \{t(X) \mid t \in r\}$$
$$\rho_{X(A)}(r) = \{t(X).A \mid t \in r\}$$

Rename Operation – Example

- Relations *r*

A	B
α	1
β	2

r

- What is the RA expression?

A	D
α	1
β	2

d

Rename Operation – Example

- Relations *r*

A	B
α	1
β	2

r

- $\rho_{d(A,D)}\,(r)$

A	D
α	1
β	2

d

Composition of Operation – Example

- Relations *r, s*

A	D
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

- What is the RA expression *E* ?

A	D
α	10
β	10
β	20

- Which relation are the attributes A and D part of?
- How the attributes and relations are combined?
- What is the condition to be applied to return the output tuples?
- How to return the values over the attributes A and D?

Composition of Operation – Example

- Relations *r, s*

A	D
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

- $\Pi_{A,d.D}\,(\sigma_{A=d.C}\,(r \times \rho_d\,(s)))$

A	D
α	10
β	10
β	20

- A from r; D from s
- Cartesian-Product: Compute all possible combinations
- Predicate *A* = *d.C* in select operation
- List the attributes *A* and *d.D* in project operation

Rename Operation – Example

- Not strictly required
  - $\Pi_{A,s.D}\ (\sigma_{A=C}\ (r \times s))$
  - Sometimes used for alias ($\rightarrow$) to rename an attribute
    - $\Pi_{A,D2}\ (\sigma_{A=C}\ (r \times (\Pi_{C,D \rightarrow D2}(s))))$
- Possible to use a positional notation for attributes
  - $\Pi_{\$4}\ (\sigma_{\$4 < \$8}\ (\text{instructor x instructor}))$
  - \$4 (\$8): attribute salary of the first (second) instructor

instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Formal Definition (Syntax)

- A basic expression in the RA consists of either one of the following:
  - A relation in the database
  - A constant relation: e.g., {(1),(2)}
- Let E_1 and E_2 be RA expressions; the following are all RA expressions:
  - $E_1 \cup E_2$
  - $E_1 - E_2$
  - $E_1 \times E_2$
  - $\sigma_p (E_1)$, P is a predicate on attributes in E_1
  - $\prod_S(E_1)$, S is a list consisting of some of the attributes in E_1
  - $\rho_x (E_1)$, x is the new name for the result of E_1

Formal Definition (Semantics)

- Let E_1 and E_2 be RA expressions.
  - We use [E](I) to denote the evaluation of E over a database instance I
    - For simplicity, often drop I and [ ]

$$[E_1 \cup E_2] = \{t \mid t \in [E_1] \vee t \in [E_2]\}$$
$$[E_1 - E_2] = \{t \mid t \in [E_1] \wedge t \notin [E_2]\}$$
$$[E_1 \times E_2] = \{t, t' \mid t \in [E_1] \wedge t' \in [E_2]\}$$
$$[\sigma_p(E_1)] = \{t \mid t \in [E_1] \wedge p(t)\}$$
$$[\pi_A(E_1)] = \{t.A \mid t \in [E_1]\}$$
$$[\rho_X(E_1)] = \{t(X) \mid t \in [E_1]\}$$

Example Queries

- Find all instructors in the Physics department, along with the *course_id* of all courses they have taught

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

Example Queries

- Find all instructors in the Physics department, along with the *course_id* of all courses they have taught
  - Query1

$$\pi_{instructor.ID, course\_id}(\sigma_{dept\_name='Physics'}(\sigma_{instructor.ID=teaches.ID}(instructor \times teaches)))$$

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Example Queries

- Which queries below are equivalent?
  - Query1

$$\pi_{instructor.ID, course\_id}(\sigma_{dept\_name='Physics'}(\sigma_{instructor.ID=teaches.ID}(instructor \times teaches)))$$

  - Query2

$$\pi_{instructor.ID, course\_id}(\sigma_{instructor.ID=teaches.ID}(\sigma_{dept\_name='Physics'}(instructor \times teaches)))$$

  - Query3

$$\pi_{instructor.ID, course\_id}(\sigma_{instructor.ID=teaches.ID}(\sigma_{dept\_name = \text{“Physics”}}(instructor) \times teaches))$$

Example Queries

- Find all instructors in the Physics department, along with the *course_id* of all courses they have taught
  - Query1

$$\pi_{instructor.ID, course\_id}(\sigma_{dept\_name='Physics'}(\sigma_{instructor.ID=teaches.ID}(instructor \times teaches)))$$

  - Query2

$$\pi_{instructor.ID, course\_id}(\sigma_{instructor.ID=teaches.ID}(\sigma_{dept\_name='Physics'}(instructor \times teaches)))$$

  - Query3

$$\pi_{instructor.ID, course\_id}(\sigma_{instructor.ID=teaches.ID}(\sigma_{dept\_name = \text{“Physics”}}(instructor) \times teaches))$$

Example Queries

- Find all instructors in the Physics department, along with the *course_id* of all courses they have taught
  - Another way?

Coming up as we go through more operations!

Example Queries

- Find the largest salary in the university

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Example Queries

- Find the largest salary in the university
  - Step 1: find instructor salaries that are less than some other instructor salaries (i.e., not maximum)
    - using a copy of *instructor* under a new name *d*
  - Step 2: Find the largest salary

Example Queries

- Find the largest salary in the university
  - Step 1: find instructor salaries that are less than some other instructor salaries (i.e., not maximum)
    - using a copy of *instructor* under a new name *d*

$$\pi_{instructor.salary}(\sigma_{instructor.salary<d.salary}(instructor \times \rho_d(instructor)))$$

  - Step 2: Find the largest salary

Example Queries

- Find the largest salary in the university
  - Step 1: find instructor salaries that are less than some other instructor salaries (i.e., not maximum)
    - using a copy of *instructor* under a new name *d*

$$\pi_{instructor.salary}(\sigma_{instructor.salary<d.salary}(instructor \times \rho_d(instructor)))$$

  - Step 2: Find the largest salary

$$\pi_{salary}(instructor) - \pi_{instructor.salary}(\sigma_{instructor.salary<d.salary}(instructor \times \rho_d(instructor)))$$

Additional Operations

- We define additional operations that *do not add any expressive power* to the relational algebra, but that ***simplify common queries***.
  - Set intersection
  - Join
  - Assignment
  - Outer join

Set-Intersection Operation – Example

- Relation *r*, *s*:

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

- $r \cap s$

A	B
α	2

Set-Intersection Operation

- Common tuples in two relations
- Notation: $r \cap s$
- Defined as:

$$r \cap s = \{t \mid t \in r \wedge t \in s\}$$

- Assume
  - *r*, *s* have the *same arity*
  - Attributes of *r* and *s* are compatible
- Note
  - $r \cap s = r - (r - s)$
  - That is adding intersection to the language does not make it more expressive.

Set-Intersection Operation – Example

- Find all courses taught in Fall 2009 and Spring 2010 semesters
  - $\Pi_{course\_id}\,(\sigma_{\ semester=\text{“Fall”}\ \wedge\ year=2009}\,(section))\ \cap$
    $\Pi_{course\_id}\,(\sigma_{\ semester=\text{“Spring”}\ \wedge\ year=2010}\,(section))$

Section

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

Result

course_id
CS-101

Natural Join Example

- Relations r, s:

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ε

s

- r $\bowtie$ s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

Natural Join Operation

- Notation: $r \bowtie s$
- **Simplifying** certain queries that require a **cross-product**
- Let r and s be relations on schemas R and S respectively. Then, $r \bowtie s$ is a relation on schema $R \cup S$ obtained as follows:
  - Consider each pair of tuples t_r from r and t_s from s.
  - If t_r and t_s have the same value on each of the attributes in $R \cap S$, add a tuple t to the result, where
    - t has the same value as t_r on r
    - t has the same value as t_s on s

Natural Join Operation

- Notation: r ⋈ s
- Example:

  $R = (A, B, C, D)$

  $S = (E, B, D)$

  - Result schema = ?
  - $r \bowtie s$ is defined as ?

Natural Join Operation

- Notation: r $\bowtie$ s
- Example:

  $R = (A, B, C, D)$

  $S = (E, B, D)$

  - Result schema = (A, B, C, D, E)
  - $r \bowtie s$ is defined as:

    $$\Pi_{r.A,\ r.B,\ r.C,\ r.D,\ s.E} (\sigma_{r.B = s.B \wedge r.D = s.D} (r \times s))$$

Natural Join Operation (cont.)

- Let *r* and *s* be relations on schemas *R* and *S* respectively. Then, $r \bowtie s$ is defined as:

$$X = R \cap S$$

$$S' = S - R$$

$$r \bowtie s = \pi_{R,S'}(\sigma_{r.X=s.X}(r \times s))$$

Natural Join Example

- Find the names of all instructors in the Comp. Sci. department together with the course titles of all the courses that the instructors teach

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

course

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

Natural Join Example

- Find the names of all instructors in the Comp. Sci. department together with the course titles of all the courses that the instructors teach
  - $\Pi_{name,\ title}\ (\sigma_{dept\_name = \text{"Comp. Sci."}}\ (instructor \bowtie teaches \bowtie course))$

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

course

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

Natural Join Example

- Find the names of all instructors in the Comp. Sci. department together with the course titles of all the courses of Comp. Sci. that the instructors teach
  - $\Pi_{\textit{name, title}} (\sigma_{\textit{dept\_name}=\text{"Comp. Sci."}} (\textit{instructor} \bowtie \textit{teaches} \bowtie \textit{course}))$

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

course

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

Natural Join Operation

- (*instructor* $\bowtie$ *teaches*) $\bowtie$ *course* vs *instructor* $\bowtie$ (*teaches* $\bowtie$ *course*)

- *instruct* $\bowtie$ *teaches* vs *teaches* $\bowtie$ *instructor*

Natural Join Operation

- Natural join is associative
  - ($instructor \bowtie teaches$) $\bowtie$ $course$ $\equiv$ $instructor \bowtie$ ($teaches \bowtie course$)
- Natural join is commutative (we ignore attribute order)
  - $instruct \bowtie teaches$ $\equiv$ $teaches \bowtie instructor$

Theta Join

- The theta join operation $r \bowtie_\theta s$ is defined as

$$r \bowtie_\theta s = \sigma_\theta(r \times s)$$

- Return the instructor's name with her/his department location (*building*).
  - $\pi_{\text{name,building}}$ (instructor $\bowtie_{\text{instructor.dept\_name = department.dept\_name}}$ department)

$= \pi_{\text{name,building}}$ ($\sigma_{\text{instructor.dept\_name = department.dept\_name}}$ (instructor X department))

instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

department

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

Theta-Join Example

- Find the names of all instructors in the Physics department, along with the *course_id* of all courses they have taught
  - Using Theta-join?

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

Theta-Join Example

- Find the names of all instructors in the Physics department, along with the *course_id* of all courses they have taught

$$\pi_{instructor.ID, course\_id}($$

$$(\sigma_{dept\_name = \text{“Physics”}}(instructor)) \bowtie_{instructor.ID = teaches.ID} teaches$$

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017
12121	FIN-201	1	Spring	2018
15151	MU-199	1	Spring	2018
22222	PHY-101	1	Fall	2017
32343	HIS-351	1	Spring	2018
45565	CS-101	1	Spring	2018
45565	CS-319	1	Spring	2018
76766	BIO-101	1	Summer	2017
76766	BIO-301	1	Summer	2018
83821	CS-190	1	Spring	2017
83821	CS-190	2	Spring	2017
83821	CS-319	2	Spring	2018
98345	EE-181	1	Spring	2017

Example Queries

- Which one performs better?
  - Query1

$$\pi_{instructor.ID, course\_id}(\sigma_{dept\_name='Physics'}(\sigma_{instructor.ID=teaches.ID}(instructor \times teaches)))$$

  - Query3

$$\pi_{instructor.ID, course\_id}(\sigma_{instructor.ID=teaches.ID}(\sigma_{dept\_name=\text{“Physics”}}(instructor) \times teaches))$$

  - Query4

$$\pi_{instructor.ID, course\_id}((\sigma_{dept\_name=\text{“Physics”}}(instructor)) \bowtie_{instructor.ID=teaches.ID} teaches)$$

Outer Join

- An extension of the join operation that avoids loss of information.
  - Left or Right or Full Outer Join
- Computes the join and then adds tuples form one relation that does not match tuples in the other relation to the result of the join.
- Uses *null* values:
  - *null* signifies that the value is unknown or does not exist
  - All comparisons involving *null* are (roughly speaking) false by definition

Outer Join – Example

- Relation *instructor*

ID	*name*	*dept_name*
10101	Srinivasan	Comp. Sci.
12121	Wu	Finance
15151	Mozart	Music

- Relation *teaches*

ID	*course_id*
10101	CS-101
12121	FIN-201
76766	BIO-101

Outer Join – Example

- Join

  instructor $\bowtie$ *teaches*

ID	*name*	*dept_name*	*course_id*
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201

- Left Outer Join

  instructor $⟕$ *teaches*

ID	*name*	*dept_name*	*course_id*
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	null

Outer Join – Example

- Right Outer Join

  instructor ⟖ *teaches*

ID	*name*	*dept_name*	*course_id*
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
76766	null	null	BIO-101

- Full Outer Join

  instructor ⟗ *teaches*

ID	*name*	*dept_name*	*course_id*
10101	Srinivasan	Comp. Sci.	CS-101
12121	Wu	Finance	FIN-201
15151	Mozart	Music	null
76766	null	null	BIO-101

Defining Outer Join using Join

- Outer join can be expressed using basic operations

$r ⟕ s =$

$r ⟖ s =$

$r ⟗ s =$

Defining Outer Join using Join

- Outer join can be expressed using basic operations

$$r ⟕ s = (r \bowtie s) \cup ((r - \Pi_R(r \bowtie s)) \times \{(null, \ldots, null)\})$$

$$r ⟖ s =$$

$$r ⟗ s =$$

Defining Outer Join using Join

- Outer join can be expressed using basic operations

$$r ⟕ s = (r \bowtie s) \cup ((r - \Pi_R(r \bowtie s)) \times \{(null, \ldots, null)\})$$

$$r ⟖ s = (r \bowtie s) \cup (\{(null, \ldots, null)\} \times (s - \Pi_S(r \bowtie s)))$$

$$\begin{aligned} r ⟗ s = {} & (r \bowtie s) \cup ((r - \Pi_R(r \bowtie s)) \times \{(null, \ldots, null)\}) \\ & \cup (\{(null, \ldots, null)\} \times (s - \Pi_S(r \bowtie s))) \end{aligned}$$

Semijoin

- Example

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017

Result

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000

Semijoin

- Returning rows from the first table where one or more matches are found in the second table

- Formal expression

$$R \ltimes_C S = \Pi_{A1,\ldots,An} (R \bowtie_C S)$$

  - The attributes $A_1, \ldots, A_n$ are from R

- Difference with conventional join?

Semijoin

- Example
  - Return instructors who have taught courses since 2018

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017

Result

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000

Semijoin

- Example
  - Return instructors who have taught courses since 2018
    - $instructor \ltimes_{instructor.ID = teaches.ID} (\sigma_{year > 2017} (teaches))$

Instructor

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2017
10101	CS-315	1	Spring	2018
10101	CS-347	1	Fall	2017

Result

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000

Semijoin

- Techniques for query processing in this distributed database systems
  - Why?
  - Possible strategy?

Semijoin

- Techniques for query processing in this distributed database systems
  - Computing query with minimum amount of data transfer
  - An example strategy
    - Compute and send common attribute values to another location (*temp1*)
    - Evaluate join using *temp1* and send the result to the original node (*orig*)
    - Compute the result over *orig*

Assignment Operation

- The **assignment** operation ($\leftarrow$) provides a convenient way to express complex queries.
  - Write query as a sequential program consisting of
    - A **series** of assignments
    - Followed by an expression whose value is displayed as a **result of the query**.
  - Assignment must always be made to a **temporary relation** variable.

$$E_1 \leftarrow \sigma_{salary>40000}(instructor)$$
$$E_2 \leftarrow \sigma_{salary<10000}(instructor)$$
$$E_3 \leftarrow E_1 \cup E_2$$

Assignment Operation

- Find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both
  - $\Pi_{course\_id}\,(\sigma_{\,semester=\text{“Fall”}\ \land\ year=2009}\,(section))$
    $\cup\ \Pi_{course\_id}\,(\sigma_{\,semester=\text{“Spring”}\ \land\ year=2010}\,(section))$

Section

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

Result

course_id
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

Assignment Operation

- Find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both
  - courses_fall_2009 ← $\Pi_{course\_id}\ (\sigma_{\ semester=\text{“Fall”}\ \wedge\ year=2009}\ (section))$
  - courses_spring_2010 ← $\Pi_{course\_id}\ (\sigma_{\ semester=\text{“Spring”}\ \wedge\ year=2010}\ (section))$
  - output ← courses_fall_2009 ∪ courses_spring_2010

Section

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

Result

course_id
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

Division Operator

- Given relations r(R) and s(S),
  - Such that $S \subseteq R$
    - Every attribute in S is a subset of R
  - $r \div s$ produces all tuples from r that all their extensions on $R \cap S$ with tuples from s exist in R
  - Both of two conditions hold
    - $t \in \prod_{R\text{-}S}(r)$
    - For every tuple t_s in s, there is a tuple t_r in r satisfying both of the following
      - $t_r[S] = t_s[S]$
      - $t_r[R - S] = t$

Division Operator Example

- Let $r(ID, course\_id) = \prod_{ID,\ course\_id} (takes)$ and
  $s(course\_id) = \prod_{course\_id} (\sigma_{dept\_name=\text{"Biology"}}(course))$

→ r ÷ s = students who have taken all courses in the Biology department

r

ID	course_id
00128	CS-101
00128	CS-347
98988	BIO-101
98988	BIO-301

÷

s

course_id
BIO-101
BIO-301

→

ID
98988

Division Operator

- Can write $r \div s$ as

$$E_1 \leftarrow$$

$$E_2 \leftarrow$$

$$r \div s = E_1 - E_2$$

r

ID	*course_id*
00128	CS-101
00128	CS-347
98988	BIO-101
98988	BIO-301

s

course_id
BIO-101
BIO-301

Division Operator

- Can write $r \div s$ as

$$E_1 \leftarrow \Pi_{R-S}(r)$$
$$E_2 \leftarrow$$
$$r \div s = E_1 - E_2$$

r

ID	*course_id*
00128	CS-101
00128	CS-347
98988	BIO-101
98988	BIO-301

s

course_id
BIO-101
BIO-301

E_1

ID
00128
98988

Division Operator

- Can write $r \div s$ as

$$E_1 \leftarrow \Pi_{R-S}(r)$$

$$E_2 \leftarrow \Pi_{R-S}\left(\overset{1}{\boxed{(E_1 \times s)}} - \overset{2}{\boxed{\Pi_{R-S,S}(r \bowtie s)}}\right)$$

$$r \div s = E_1 - E_2$$

All combinations of students and Biology courses

Students who have taken a course in Bioogy depertment

r

ID	*course_id*
00128	CS-101
00128	CS-347
98988	BIO-101
98988	BIO-301
98999	BIO-101

s

course_id
BIO-101
BIO-301

E_1

ID
00128
98988
98999

E_2

ID
00128
98999

$E_{2.1}$

ID	course_id
00128	BIO-101
00128	BIO-301
98988	BIO-101
98988	BIO-301
98999	BIO-101
98999	BIO-301

$E_{2.2}$

ID	course_id
98988	BIO-101
98988	BIO-301
98999	BIO-101

Division Operator Example

- Return the name of all persons who read all newspapers

reads

name	*newspaper*
Peter	Times
Bob	Wall Street
Alice	Times
Alice	Wall Street

newspaper

newspaper
Times
Wall Street

$$E_1 \leftarrow \Pi_{R-S}(r)$$

$$E_2 \leftarrow \Pi_{R-S}((E_1 \times s) - \Pi_{R-S,S}(r \bowtie s))$$

$$r \div s = E_1 - E_2$$

Division Operator Example

- Return the name of all persons who read all newspapers

reads

name	*newspaper*
Peter	Times
Bob	Wall Street
Alice	Times
Alice	Wall Street

newspaper

newspaper
Times
Wall Street

$$E_1 \leftarrow \Pi_{name}(reads)$$

$$E_2 \leftarrow \Pi_{name}((E_1 \times newspaper) - \Pi_{name,newspaper}(reads \bowtie newspaper)$$

$$reads \div newspaper = E_1 - E_2$$

$$[reads \div newspaper] = \{(Alice)\}$$

Extended Relational-Algebra Operations

- Generalized Projection
- Aggregate Functions
- Duplicate Elimination
- Sorting

Generalized Projection

- Extends the projection operation by allowing arithmetic functions to be used in the projection list.

$$\pi_{F_1,\ldots,F_n}(E)$$

- *E* is any RA expression
- Each of F_1, F_2, …, F_n are arithmetic expressions and function calls involving constants and attributes in the schema of *E*.
- Given relation *instructor(ID, name, dept_name,* salary) where salary is annual salary, get the same information but with monthly salary

$$\Pi_{ID,\ name,\ dept\_name,\ salary/12}\ (instructor)$$

- Adding functions increases expressive power?

Generalized Projection

- Extends the projection operation by allowing arithmetic functions to be used in the projection list.

$$\pi_{F_1, \ldots, F_n}(E)$$

- E is any RA expression
- Each of F_1, F_2, …, F_n are arithmetic expressions and function calls involving constants and attributes in the schema of E.
- Given relation *instructor(ID, name, dept_name,* salary) where salary is annual salary, get the same information but with monthly salary

$$\Pi_{ID,\ name,\ dept\_name,\ salary/12}\ (instructor)$$

- Adding functions increases expressive power!
  - In standard RA, there is no way to change attribute values.

Aggregate Functions and Operations

- Aggregation function takes a set of values and returns a single value as a result.

  avg: average value
  min: minimum value
  max: maximum value
  sum: sum of values
  count: number of values

- Aggregate operation in relational algebra

$$_{G_1, G_2, \ldots, G_m}\mathcal{G}_{F_1(A_1), F_2(A_2), \ldots, F_n(A_n)}(E)$$

  E is any RA expression

  - G_1, G_2 …, G_m is a list of attributes on which to group (can be empty)
  - Each F_i is an aggregate function
  - Each A_i is an attribute name

- Note: Some books/articles use γ instead of $\mathcal{G}$ (Calligraphic G)

Aggregate Operation – Example

- Relation *r*

A	*B*	*C*
α	α	7
α	β	7
β	β	3
β	β	10

- $\mathcal{G}_{\textbf{sum(c)}}(\mathrm{r})$

sum(*c*)
27

Aggregate Operation – Example

- Find the average salary of each department

ID	*name*	*dept_name*	*salary*
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

dept_name	*avg_sal*
Biology	72000
Comp. Sci.	77333
Elec. Eng.	80000
Finance	85000
History	61000
Music	40000
Physics	91000

Aggregate Operation – Example

- Find the average salary of each department

$$_{dept\_name}\mathcal{G}_{\textbf{avg}(salary)}(instructor)$$

ID	*name*	*dept_name*	*salary*
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

dept_name	*avg(salary)*
Biology	72000
Comp. Sci.	77333
Elec. Eng.	80000
Finance	85000
History	61000
Music	40000
Physics	91000

Aggregate Functions (Cont.)

- What are the names for attributes in aggregation results?

Aggregate Functions (Cont.)

- What are the names for attributes in aggregation results?
  - Need some convention!
    - E.g., use the expression as a name **avg**(salary)
  - For convenience, we **permit renaming** as part of aggregate operation

$$\textit{dept\_name}\ \mathcal{G}\ \textbf{avg}(\textit{salary})\ \textbf{as}\ \textit{avg\_sal}\ (\textit{instructor})$$

OR

$$\textit{dept\_name}\ \mathcal{G}\ \textbf{avg}(\textit{salary}) \rightarrow \textit{avg\_sal}\ (\textit{instructor})$$

Aggregate Functions (Cont.)

- Find instructors whose salary is larger than the average salary.

ID	*name*	*dept_name*	*salary*
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

Aggregate Functions (Cont.)

- Find instructors whose salary is larger than the average salary.

$$avgSalary \leftarrow \mathcal{G}\ \textbf{avg}(salary)\ \textbf{as}\ avg\_sal\ (instructor)$$

$$\sigma_{salary > avg\_sal}(instructor\ X\ avgSalary)$$

ID	*name*	*dept_name*	*salary*
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
22222	Einstein	Physics	95000

Operations on Bags

- A **bag** = a set with repeated elements
  - Relational engines work on bags
- All operations need to be defined carefully on bags
  - {a,b,b,c} ∪ {a,b,b,b,e,f,f} = {a,a,b,b,b,b,b,c,e,f,f}
  - {a,b,b,b,c,c} - {b,c,c,c,d} = ?

Operations on Bags

- A **bag** = a set with repeated elements
  - Relational engines work on bags
- All operations need to be defined carefully on bags
  - {a,b,b,c} ∪ {a,b,b,b,e,f,f} = {a,a,b,b,b,b,b,c,e,f,f}
  - {a,b,b,b,c,c} - {b,c,c,c,d} = {a,b,b}
  - …
- Select, project, and join work for bags as well as sets.
  - σ: preserve the number of occurrences
  - π: no duplicate elimination
  - Cartesain product, join: no duplicate elimination

Operations on Bags

- Some familiar laws continue to hold for bags.
  - Union and intersection are still **commutative** and **associative**
- But other laws, e.g., distributivity?
  - $R \cap (S \cup T) \equiv (R \cap S) \cup (R \cap T)$

Operations on Bags

- Some familiar laws continue to hold for bags.
  - Union and intersection are still **commutative** and **associative**
- But other laws, e.g., distributivity, that hold for sets do not hold for bags
  - $R \cap (S \cup T) \equiv (R \cap S) \cup (R \cap T)$ holds for sets
    - Let R, S, and T each be the bag $\{1\}$
    - Left side
      - $(S \cup T) = ?$
      - $R \cap (S \cup T) = ?$
    - Right side
      - $R \cap S = R \cap T = ?$
      - $(R \cap S) \cup (R \cap T) = ?$

Operations on Bags

- Some familiar laws continue to hold for bags.
  - Union and intersection are still **commutative** and **associative**
- But other laws, e.g., distributivity, that hold for sets do not hold for bags
  - $R \cap (S \cup T) \equiv (R \cap S) \cup (R \cap T)$ holds for sets
    - Let R, S, and T each be the bag {1}
    - Left side
      - $(S \cup T) = \{1,1\}$
      - $R \cap (S \cup T) = \{1\}$
    - Right side
      - $R \cap S = R \cap T = ?$
      - $(R \cap S) \cup (R \cap T) = ?$

Operations on Bags

- Some familiar laws continue to hold for bags.
  - Union and intersection are still **commutative** and **associative**
- But other laws, e.g., distributivity, that hold for sets do not hold for bags
  - $R \cap (S \cup T) \equiv (R \cap S) \cup (R \cap T)$ holds for sets
    - Let R, S, and T each be the bag $\{1\}$
    - Left side
      - $(S \cup T) = \{1,1\}$
      - $R \cap (S \cup T) = \{1\}$
    - Right side
      - $R \cap S = R \cap T = \{1\}$
      - $(R \cap S) \cup (R \cap T) = \{1,1\}$

Duplicate Elimination

- δ(R) = relation with one copy of each tuple that appears one or more times in R

R

A	B
1	2
3	4
1	2

δ(R)

A	B
1	2
3	4

Sorting

- τ_L(R) = list of tuples of R, ordered according to attributes on list L
- Note that result type is **outside the normal types** (set or bag) for relational algebra
  - Consequence: τ cannot be followed by other relational operators

R

A	B
1	3
3	4
5	2

τ_B(R) = ?

Sorting

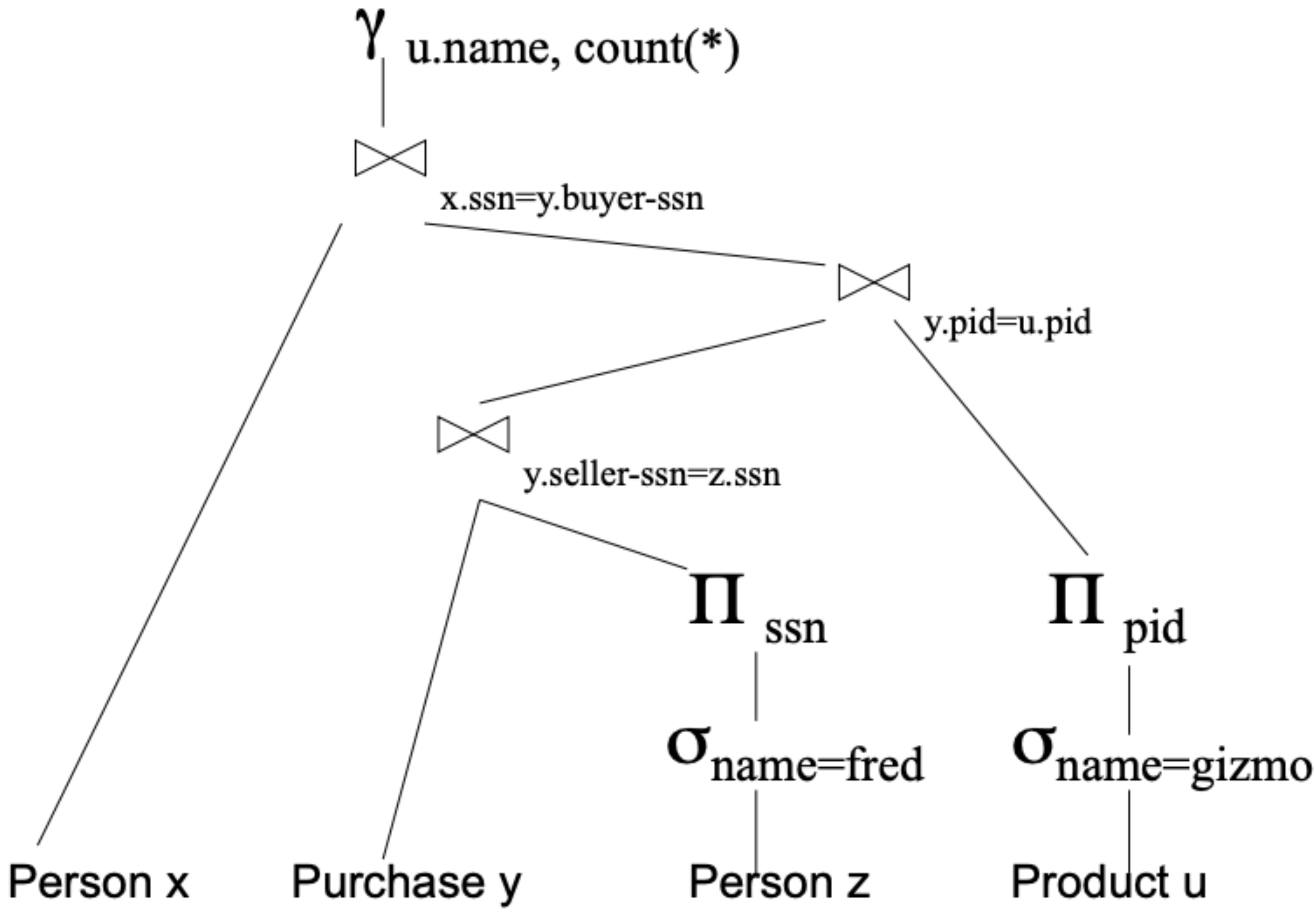
- $\tau_L(R)$ = list of tuples of R, ordered according to attributes on list L
- Note that result type is **outside the normal types** (set or bag) for relational algebra
  - Consequence: τ cannot be followed by other relational operators

R

A	B
1	3
3	4
5	2

$\tau_B(R) = [(5,2), (1,3), (3,4)]$

Example Complex RA Expression



Modification of the Database

- The content of the database may be modified using the following operations:
  - Deletion
  - Insertion
  - Updating
- All these operations can be expressed using the assignment operator
- <u>Example</u>: Delete instructors with salary over $1,000,000

Modification of the Database

- The content of the database may be modified using the following operations:
  - Deletion
  - Insertion
  - Updating
- All these operations can be expressed using the assignment operator
- <u>Example</u>: Delete instructors with salary over $1,000,000

$$R \leftarrow R - (\sigma_{salary > 1000000}(R))$$

Restrictions for Modification

- Consider a modification where R=(A,B) and S=(C)

$$R \leftarrow \sigma_{C>5}(S)$$

Restrictions for Modification

- Consider a modification where R=(A,B) and S=(C)

$$R \leftarrow \sigma_{C>5}(S)$$

- This would change the schema of R!
  - Should not be allowed

Restrictions for Modification

- Consider a modification where R=(A,B) and S=(C)

$$R \leftarrow \sigma_{C>5}(S)$$

- This would change the schema of R!
  - Should not be allowed

- Requirements for modifications
  - The name **R** on the left-hand side of the assignment operator refers to an ***existing relation*** in the database schema
  - The expression on the right-hand side of the assignment operator should be ***union-compatible*** with **R**

Recursion?

- Find all direct and indirect relatives of Fred

Name1	Name2	Relationship
Fred	Mary	Father
Mary	Joe	Cousin
Mary	Bill	Spouse
Nancy	Lou	Sister

Limitations of RA

- Find all direct and indirect relatives of Fred
  - Cannot compute “transitive closure”
  - Cannot express in RA

Name1	Name2	Relationship
Fred	Mary	Father
Mary	Joe	Cousin
Mary	Bill	Spouse
Nancy	Lou	Sister

Corresponding Reading Materials

- Relational Algebra
  - Database System Concepts 7th Edition
    - Chapter 2.6
  - Database System Concepts 6th Edition
    - Chapter 6.1
- Resource
  - http://dbis-uibk.github.io/relax/landing