Introduction to Relational Algebra

Introduction

- **Query languages** are specialized languages for asking questions or queries, that involve the data in a database.
- Queries in **algebra** are composed of a collection of operators.
- Every operator in relational algebra accepts (one or two) relation instances as arguments and returns a relation instance as the result.
  - A relational algebra expression is recursively defined to be a relation.
  - Relational algebra is a procedural query language
    - Define a step-by-step procedure for computing the desired answer.

Basic Operators

- Six basic operators :
  - select (σ)
  - project (Π)
  - union (U)
  - set different (-)
  - Cartesian product (x)
  - rename (ρ)
- The operators take one or two relations as inputs and produce a new relation as a result.
- We first discuss relational algebra based on sets (or single-set), and then we discuss how to extend it to multisets.

Sets

- $S = \{a, b, c\}$ -- a set S contains a, b and c.
- $a \in S$ -- a is an element of set S.
- $a \notin S$ -- a is not an element of set S.
- $\{x \in S \mid P(x)\}$ is the set of x from S such that the predicate $P(x)$ is true.

Relations Between Sets

- Let A and B be sets.
  - A is a subset of B such that $A \subseteq B$, if and only if very element of A is also an element of B.
  - A is not a subset of B such that $A \not\subset B$, if some element of A is not an element of B.
  - A is equal to B. $A = B$, if and only if $A \subseteq B$ and $B \subseteq A$.

Your Pronunciation Reference

Ελληνική Αλφάβητος (Α-Ω) 2019

Greek Letter		Word for Letter	English Pronounce	Sounds	Greek Letter		Word for Letter	English Pronounce	Sounds
α	A	άλφα	alfa	apple	ν	N	νι	nee	need
β	B	βήτα	veeta	vet	ξ	Ξ	ξι	ksee	ox
γ	Γ	γάμμα	gamma	yip/y'all	ο	O	όμικρον	omeekron	ore
δ	Δ	δέλτα	thelta	teething	π	Π	πι	pee	pepper
ε	E	έψιλον	epsilon	deck	ρ	P	ρο	rho	rojo
ζ	Z	ζήτα	zeeta	zoo	σ (ς)	Σ	σίγμα	sigma	sing
η	H	ήτα	eeta	bee	τ	T	ταφ	taf	toilet
θ	Θ	θήτα	theta	thousand	υ	Y	ύψιλον	eepseelon	bee
ι	I	ιότα	yota	bee	φ	Φ	φι	fee	feet
κ	K	κάππα	kappa	kazoo	χ	X	χι	hee	heat
λ	Λ	λάμδα	lamtha	lake	ψ	Ψ	ψι	psee	psst/collapse
μ	M	μι	mee	midnight	ω	Ω	ομέγα	omega	ore

https://www.reddit.com/r/GREEK/comments/byr5iz/greek_alphabet_with_english_pronunciation/

Select Operation

- Notation: $\sigma_P(r)$
- P is called the **selection predicate**
- Defined as:
  - $\sigma_P(r) = \{t \mid t \in r \wedge P(t)\}$
  - P is a formula in propositional calculus
- Example of selection:
  - $\sigma_{dept\_name="Phyics"}(instructor)$

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

(a) The *instructor* table

Select Operation

- Notation: $\sigma_P(r)$
- P is called the **selection predicate**
- Defined as:
  - $\sigma_P(r) = \{t \mid t \in r \wedge P(t)\}$
  - P consisting of **terms** connected by $\wedge$ (**and**), $\vee$ (**or**), $\neg$ (**not**).
  - Each **term** is one of:
    - <attribute1> *op* <attribute2>
    - <attribute> op <constant>

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

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

(a) The *instructor* table

Example

- The selection operation σ specifies the tuples to retain through a *selection predicate*.
- Selection predicate is a Boolean expression (can involve multiple conditions using logical connectives e.g., $\wedge$ and $\vee$)

Relation R

a	b	c	d
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

$\sigma_{a=b \wedge d>5}(R)$

a	b	c	d
α	α	1	7
β	β	23	10

Project Operation

- Notation: $\Pi_{A_1,A_2,\ldots,A_k}(r)$
  - $A_1, A_2, \ldots, A_k$ are attribute names
  - r is a relation name.
- The result is defined as the relation of k columns obtained by erasing the columns that are not listed.
- Example: To ignore the *dept_name* attribute of *instructor.*

  $\Pi_{ID,name,salary}(instructor)$

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

(a) The *instructor* table

Project Operation Rephrased

- Projection
  - The projection operator Π allows us to extract columns from a relation.
  - The subscript specifies the fields to be retained.

    (The other fields are 'projected out').
  - The schema of the result of a projection is determined by the fields that are projected.
  - Duplicated row will be eliminated in the final result.
    - This follows from the definition.

Project Operation

Relation R

a	b	c
α	10	1
α	20	1
β	30	1
β	40	2

$\Pi_{a,c}(R)$

a	c
α	1
α	1
β	1
β	2

Eliminate Duplicated rows

a	c
α	1
β	1
β	2

Practice (1)

Employee

f_name	l_name	id	sex	salary	superid	dno
Joseph	Chan	999999	M	29500	654321	4
Victor	Wong	001100	M	30000	888555	5
Carrie	Kwan	898989	F	26000	654321	4
Joyce	Fong	345345	F	12000	777888	4

Find all employees who works in department 4 and whose salary is greater than 25000?

Practice (1)

Employee

f_name	l_name	id	sex	salary	superid	dno
Joseph	Chan	999999	M	29500	654321	4
Victor	Wong	001100	M	30000	888555	5
Carrie	Kwan	898989	F	26000	654321	4
Joyce	Fong	345345	F	12000	777888	4

Find all employees who works in department 4 and whose salary is greater than 25000?

$\sigma_{dno=4 \wedge salary>25000}(Employee)$

f_name	l_name	id	sex	salary	superid	dno
Joseph	Chan	999999	M	29500	654321	4
Carrie	Kwan	898989	F	26000	654321	4

Practice (2)

Employee

f_name	l_name	id	sex	salary	superid	dno
Joseph	Chan	999999	M	29500	654321	4
Victor	Wong	001100	M	30000	888555	5
Carrie	Kwan	898989	F	26000	654321	4
Joyce	Fong	345345	F	12000	777888	4

Find the employee names and department number of all employees

Practice (2)

Employee

f_name	l_name	id	sex	salary	superid	dno
Joseph	Chan	999999	M	29500	654321	4
Victor	Wong	001100	M	30000	888555	5
Carrie	Kwan	898989	F	26000	654321	4
Joyce	Fong	345345	F	12000	777888	4

Find the employee names and department number of all employees

$$\Pi_{f\_name,l\_name,dno}(Employee)$$

f_name	**l_name**	**dno**
Joseph	Chan	4
Victor	Wong	5
Carrie	Kwan	4
Joyce	Fong	4

Practice (3)

Employee

f_name	l_name	id	sex	salary	superid	dno
Joseph	Chan	999999	M	29500	654321	4
Victor	Wong	001100	M	30000	888555	5
Carrie	Kwan	898989	F	26000	654321	4
Joyce	Fong	345345	F	12000	777888	4

Find the sex and department number of all employees

Practice (3)

Employee

f_name	l_name	id	sex	salary	superid	dno
Joseph	Chan	999999	M	29500	654321	4
Victor	Wong	001100	M	30000	888555	5
Carrie	Kwan	898989	F	26000	654321	4
Joyce	Fong	345345	F	12000	777888	4

Find the sex and department number of all employees

$$\Pi_{sex,dno}\,(\,Employee\,)$$

sex	**dno**
M	4
M	5
F	4

Note

1. Compositing operation gives relational-algebra expr.
2. Result of a relational-algebra expr. is always a relation.

Employee

f_name	l_name	id	sex	salary	superid	dno
Joseph	Chan	999999	M	29500	654321	4
Victor	Wong	001100	M	30000	888555	5
Carrie	Kwan	898989	F	26000	654321	4
Joyce	Fong	345345	F	12000	777888	4

$$\Pi_{f\_name,l\_name} (\sigma_{dno=4\text{^}salary>25000} (Employee))$$

Union Operation (1)

- Notation: $r \cup s$
- Defined as: $r \cup s = \{t \mid t \in r \vee t \in s\}$
- R ∪ S returns a relation instance containing all tuples that occur in either relation R or relation instance S (or both).
- Example: to find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both

$$\prod_{course\_id} (\sigma_{semester=\text{“Fall”} \wedge year=2009} (section)) \cup \prod_{course\_id} (\sigma_{semester=\text{“Spring”} \wedge year=2010} (section))$$

- Also
  - The union operation is commutative: R ∪ S = S ∪ R
  - Duplicate tuples are eliminated.

Union Operation (2)

- For $r \cup s$ to be valid
  - r, s must have the *same* **arity** (same number of attributes)
  - R and S, the attribute domains must be *union-compatible*.
    - They have the same number of fields,
    - The corresponding fields have the same domains.
  - The schema of the result is defined to be identical to the schema of R.
    - The field of R $\cup$ S inherit names from R.

Example

R

a	b
α	1
α	2
β	1

S

a	b
α	2
β	3

R U S

a	b
α	1
α	2
β	1
β	3

Set Difference Operation (1)

- Notation: $r - s$
- Defined as: $r - s = \{t \mid t \in r \wedge t \notin s\}$
- Set differences must be taken between **union-compatible** relations.
  - r and s must have the same arity
  - attribute domains of r and s must be compatible
- Example: to find all courses taught in the Fall 2009 semester, but not in the Spring 2010 semester.

$$\prod_{course\_id}(\sigma_{semester=\text{“Fall”} \wedge year=2009}(section)) - \prod_{course\_id}(\sigma_{semester=\text{“Spring”} \wedge year=2010}(section))$$

Set Difference Operation (2)

- *R* – *S* returns a relation instance containing all the tuples that occur in R but not in S.
- Set difference is not commutative:
  - in general, R – S ≠ S – R.
- The schema of the result is defined to be identical to the schema of R.

Example

R

a	b
α	1
α	2
β	1

S

a	b
α	2
β	3

R - S

a	b
α	1
β	1

Cartesian-Product Op. (2)

- Cartesian product (Cross product)
  - R $\times$ S returns a relation instance whose schema contains all the fields of R followed by all the fields of S.
  - The result contains one tuple *<r,s>* (concatenation of tuples *r* and *s*) for each pair of tuples r $\in$ R, s $\in$ S.
  - There is an assumption mentioned later

Example

R

a	b
α	1
β	2

S

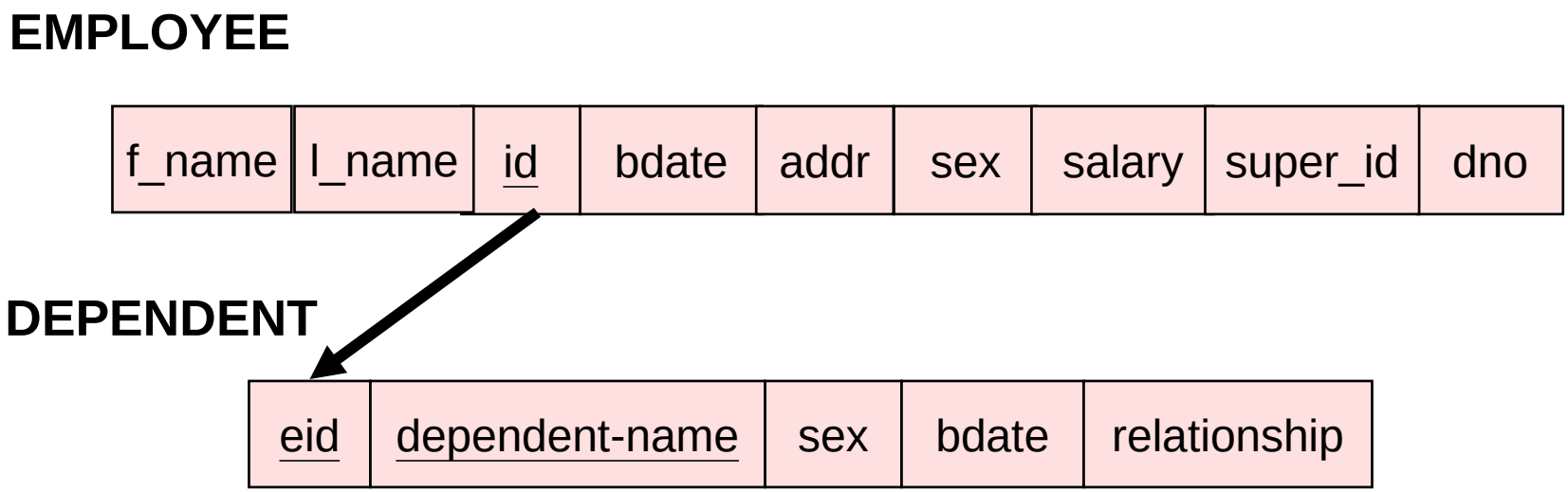
c	d	e
α	10	+
β	10	+
β	20	-
γ	10	-

R × S

a	b	c	d	e
α	1	α	10	+
α	1	β	10	+
α	1	β	20	-
α	1	γ	10	-
β	2	α	10	+
β	2	β	10	+
β	2	β	20	-
β	2	γ	10	-

Practice (1)

Query: To retrieve for each female employee a list of the names of her dependents.



Practice (2)

Female_emps $\leftarrow \sigma_{sex='F'}$ (*Employee*)

Empnames $\leftarrow \Pi_{f\_name,l\_name,id}$ (*Female_emps*)

Female_emps

f_name	l_name	id	bdate	address	sex	salary	superid	dno
Alicia	Chan	998877	2-Jul-70	231, Cai Road, HK	F	9500	654321	4
Jennifer	Wong	654321	20-June-60	342, Cheung Road, HK	F	30000	888555	4
Joyce	Fong	345345	19-Dec-80	23, Young Road, HK	F	12000	777888	5

Empnames

f_name	l_name	id
Alicia	Chan	998877
Jennifer	Wong	654321
Joyce	Fong	345345

Practice (3)

EMPLOYEE

f_name	l_name	id	bdate	addr	sex	salary	super_id	dno

DEPENDENT

eid	dependent-name	sex	bdate	relationship

Empnames

f_name	l_name	id
Alicia	Chan	998877
Jennifer	Wong	654321
Joyce	Fong	345345

Dependents

eid	dep_name	sex	bdate	relationship
334455	Alice	F	5-Apr-90	Daughter
334455	Theodore	M	3-Mar-92	Son
654321	Abner	M	29-Feb-94	Son
123456	Alice	F	2-Nov-97	Daughter

Practice (4)

Emp_dependents ← Empnames x Dependents

Emp_dependents

f_name	l_name	id	eid	dep_name	sex	bdate	relationship
Alicia	Chan	998877	334455	Alice	F	5-Apr-90	Daughter
Alicia	Chan	998877	334455	Theodore	M	3-Mar-92	Son
Alicia	Chan	998877	654321	Abner	M	29-Feb-94	Son
Alicia	Chan	998877	123456	Alice	F	2-Nov-97	Daughter
Jennifer	Wong	654321	334455	Alice	F	5-Apr-90	Daughter
Jennifer	Wong	654321	334455	Theodore	M	3-Mar-92	Son
Jennifer	Wong	654321	654321	Abner	M	29-Feb-94	Son
Jennifer	Wong	654321	123456	Alice	F	2-Nov-97	Daughter
Joyce	Fong	345345	334455	Alice	F	5-Apr-90	Daughter
Joyce	Fong	345345	334455	Theodore	M	3-Mar-92	Son
Joyce	Fong	345345	654321	Abner	M	29-Feb-94	Son
Joyce	Fong	345345	123456	Alice	F	2-Nov-97	Daughter

Practice (5)

Recall: To retrieve for each female employee a list of the names of her dependents

$Actual\_dependents \leftarrow \sigma_{id=eid} (Emps\_dependents)$

$Result \leftarrow \Pi_{f\_name,l\_name,dep\_name} (Actual\_dependents)$

Actual_dependents

f_name	l_name	id	eid	dep_name	sex	bdate	relationship
Jennifer	Wong	654321	654321	Abner	M	29-Feb-94	Son

Result

f_name	l_name	dep_name
Jennifer	Wong	Abner

Noticed Anything?

- The **assignment operation** (←)
  - Result of the expression to the right of ← is assigned to the relation variable on the left of ←
- 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.

Rename Operation

- Allow us to name, and therefore to refer to, the results of relational algebra expressions.
- Allow us to refer to a relation by more than one name.
- Example: $\rho_X(E)$

  returns the expression E under the name X.
- If a relational-algebra expression E has arity n, then

  $$\rho_{X(A_1, A_2, \ldots, A_n)}(E)$$

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

Example Query

Find the largest salary in the university

- Step 1: find instructor salaries that are less than some other instructor salary (i.e. not maximum)

$$\Pi_{instructor.salary}\ (\sigma_{instructor.salary < d.salary}\ (instructor \times \rho_{d}\ (instructor)))$$

- Step 2: Find the largest salary

$$\Pi_{salary}\ (instructor) - \Pi_{instructor.salary}\ (\sigma_{instructor.salary < d.salary}\ (instructor \times \rho_{d}\ (instructor)))$$

instructor

ID	name	dept_name	salary
8	ABC	SEEM	100
7	XYZ	SEEM	120

I.ID	I.name	I.dept_name	I.salary	d.ID	d.name	d.salary	d.salary
8	ABC	SEEM	100	8	ABC	SEEM	100
8	ABC	SEEM	100	7	XYZ	SEEM	120
7	XYZ	SEEM	120	8	ABC	SEEM	100
7	XYZ	SEEM	120	7	XYZ	SEEM	120

Example Queries

- Find the names of all instructors in the Physics department, along with the *course_id* of all courses they have taught.
  - *Query 1*

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

  - *Query 2*

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

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

teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009

Rename (ρ)

- Rename (ρ)
  - ρ (R(F), E) or ρ (R, E)
    - where E is an arbitrary relation algebra expression
  - result: relation named R,
  - R = E except that the fields may be renamed according to F.
  - F is called the renaming list:
  - oldname → newname or position → newname.

Example: We may rename fields:

ρ_d (*instructor*)

ρ (*C* (*sid* $\rightarrow$ *identity*), *E*)

ρ (*C* (*sid* $\rightarrow$ *identity*, *child* $\rightarrow$ *dependent*), *E*)

Example

ρ (*C* (*3* $\rightarrow$ *identity*), *E*)

Result is C

3rd attribute in E is renamed as "*identity*" in C

Cartesian-Product Operation

- Notation $r \times s$
- Defined as: $r \times s = \{pq \mid p \in r \wedge q \in s\}$
- Assume that attributes of $r(R)$ and $s(S)$ are disjoint. (That is, $R \cap S = \emptyset$).
- $w(W)$ implies that a relation w following the schema W.
- If attributes of $r(R)$ and $s(S)$ are not disjoint, then *renaming* must be used. Since it happens, we devise a naming schema to distinguish between the attribute names if they are the same in $r(A, B)$ and $s(A, C)$, by attaching the relation name, $r.A$ and $s.A$ (known as dot-notation).

Additional Operations

Set-Intersection Operation

- 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* and attributes of r and s are compatible

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- Note: $r \cap s = r - (r - s)$

$r \cap s$

A	B
α	2

Set-Intersection Rephrased

- Set-Intersection
  - R ∩ S returns a relation instance containing all tuples that occur in both R and S.
  - The relations R and S must be union-compatible.
  - The schema of the result is defined to be identical to the schema of R.

Natural-Join Operation

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

  $R = (A, B, C, D), S = (E, B, D)$
  - Result schema is (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 and Theta Join

- Find the names of all instructors in the Comp. Sci. department together with the course titles of all the courses that the instructors teach
  - $\Pi_{name,\ title}(\sigma_{dept\_name=\text{“Comp. Sci.”}}(instructor \bowtie teaches \bowtie course))$
- Natural join is associative
  - $(instructor \bowtie teaches) \bowtie course$ is equivalent to $instructor \bowtie (teaches \bowtie course)$
- Natural join is commutative
  - $instruct \bowtie teaches$ is equivalent to $teaches \bowtie instructor$

Natural Join

- The **theta join** operation $\mathrm{r} \bowtie_\theta s$ is defined by combing a selection and a Cartesian product into a single operation as
  - $\mathrm{r} \bowtie_\theta s = \sigma_\theta(r \times s)$
  - The most general version of join operation accepts a *join condition c*.
  - The join condition is identical to a selection condition in form.

S

sid	sname	rating	age
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.5

R

sid	bid	day
22	101	10/10/96
58	103	11/12/96

$$S \bowtie_{S.sid<R.sid} R$$

(s.sid)	sname	rating	age	(r.sid)	bid	day
22	Dustin	7	45.0	58	103	11/12/96
31	Lubber	8	55.5	58	103	11/12/96

S

sid	sname	rating	age
22	Dustin	7	45.0
31	Lubber	8	55.5
58	Rusty	10	35.5

R

sid	bid	day
22	101	10/10/96
58	103	11/12/96

$$S \rhd\lhd_{S.sid=R.sid} R$$

sid	sname	rating	age	bid	day
22	Dustin	7	45.0	101	10/10/96
58	Rusty	10	35.5	103	11/12/96

Example

Query: *Find the names of employees with the highest salary*.

$$\Pi_{l\_name,f\_name}(Employee) -$$

$$\Pi_{Employee.l\_name,\ Employee.f\_name}($$

$$Employee \bowtie_{Employee.salary<F.salary} \rho(F, Employee))$$

Note: We assume that <l_name,f_name> is a key in this relation.

Join (3)

- Equi-join
  - A special case of the theta-join operation is when the join condition consists solely of equalities (connected by $\wedge$) of the form

    R.name1 = S.name2
  - In the resulting relation, S.name2 will be dropped by an additional projection operation.

R

a	b	c
α	1	α
β	5	γ
γ	4	β
α	1	γ
δ	2	β

S

b	e	f
1	X	α
3	X	β
1	X	γ
2	Y	δ
3	Y	ε

$R \bowtie_{R.b=S.b} S$

a	b	c	e	f
α	1	α	X	α
α	1	α	X	γ
α	1	γ	X	α
α	1	γ	X	γ
δ	2	β	Y	δ

Summary

- The natural join operation $R \bowtie S$ is an equijoin in which equalities are specified on **all** fields having the same names in R and S.
- We can simply omit the join condition.
- The resulting schema contains the attributes of R followed by the attributes in S that are not in R.
- If the two relations have no attributes in common, $R \bowtie S$ is simply the cross-product.

R

a	b	c	d
α	1	α	X
β	2	γ	X
γ	4	β	Y
α	1	γ	Y
δ	2	β	Y

S

b	d	e
1	X	α
3	X	β
1	X	γ
2	Y	δ
3	Y	ε

R ⋈ S

a	b	c	d	e
α	1	α	X	α
α	1	α	X	γ
δ	2	β	Y	δ

Division (/)

- Division
  - The division operation is useful for expressing certain kinds of queries, for example, “find the names of sailors who have reserved all boats.

Division Rephrased

- Example
  - Consider two relations A and B.
  - A has exactly two fields x and y.
  - B has just one field y, with the same domain as in A.
  - The division operation A/B is the set of all x values (in the form of unary tuples) such that for every y value in a tuple of B, there is a tuple <x,y> in A.
- The division operation A/B is the set of all x values (in the form of unary tuples) such that for every y value in a tuple of B, there is a tuple <x,y> in A.

Division (/)

A

x	y
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ε	1
ε	2

B

y
1
2

Division (/)

A

x	y
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ε	1
ε	2

B

y
1
2

A / B

x
α
ε

Division (/)

A

x	y
α	1
α	2
α	3
β	1
γ	1
δ	1
δ	3
δ	4
ε	1
ε	2

B

y
1
2

Attributes in A but not in B. Note that it is not a valid notation in relational algebra.

$Temp1 \leftarrow \Pi_{A-B}(A)$

$Temp2 \leftarrow \Pi_{A-B}((Temp1 \times B) - A)$

$Result = Temp1 - Temp2$

Temp1

x
α
β
γ
δ
ε

$(Temp1 \times B) - A$

x	y
β	2
γ	2
δ	2

Temp2

A / B

x
α
ε

Check Understanding

Employee

fname	lname	id	bdate	address	salary	sid	dno

Works_on

id	pno

Query: Retrieve the names of employees who work on all the projects that `John Sung' works on.

$Sung \leftarrow \sigma_{fname=\text{“John”} \wedge lname=\text{“Sung”}} (Employee)$

$Sung\_pnos \leftarrow \Pi_{pno} (Works\_on \bowtie Sung)$

fname	lname	id	bdate	address	salary	sid	dno	pno

$Result\_id \leftarrow Works\_on / Sung\_pnos$

$Result \leftarrow \Pi_{fname,lname} (Result\_id \bowtie Employee)$

Exercise

Consider the schema below, the primary key fields are underlined.

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

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

Example (1)

Query 1: Find the names of sailors who have reserved boat with bid = 103.

- Solution 1:

Π_{sname} (($\sigma_{bid=103}$ Reserves) $\bowtie$ Sailors)

- Solution 2:

ρ (Temp1, $\sigma_{bid=103}$ Reserves)

ρ (Temp2, Temp1 $\bowtie$ Sailors)

Π_{sname} (Temp2)

- Solution 3:

Π_{sname} ($\sigma_{bid=103}$ (Reserves $\bowtie$ Sailors))

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

Example (2)

Query 2: Find the names of sailors who have reserved at least a red boat.

- Solution 1:

$$\Pi_{\text{sname}} ((\sigma_{\text{color='red'}} \text{Boats}) \bowtie \text{Reserves} \bowtie \text{Sailors})$$

- Solution 2:

$$\Pi_{\text{sname}} (\Pi_{\text{sid}} ((\Pi_{\text{bid}} \ \sigma_{\text{color='red'}} \text{Boats}) \bowtie \text{Reserves}) \bowtie \text{Sailors})$$

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

Example (3)

Query 3: Find the names of sailors who have reserved at least a red or a green boats. We can identify all red or green boats, then find sailors who have reserved one of these boats.

— Solution 1:

ρ (Tempboats, ($\sigma_{\text{color='red' V color='green'}}$ Boats))

Π_{sname} (Tempboats $\bowtie$ Reserves $\bowtie$ Sailors)

What happens if V is replaced by $\wedge$ in this query?

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

Example (3)

Query 3: Find the names of sailors who have reserved at least a red or a green boats. We can identify all red or green boats, then find sailors who have reserved one of these boats.

— Solution 2:

$$\Pi_{sname}((\sigma_{color='red'}Boats) \bowtie Reserves) \bowtie Sailors) \cup$$
$$\Pi_{sname}((\sigma_{color='green'}Boats) \bowtie Reserves) \bowtie Sailors)$$

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

Example (4)

Query 4: Find the names of sailors who have reserved at least a red boat and at least a green boat.

The previous solution 1 would not work. We must identify sailors who have reserved red boats, sailors who have reserved green boats, then find the intersection.

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

Example (4)

Note that sid is a key for Sailors, but sname is not a key.

ρ (Tempred, Π_{sid} (($\sigma_{color='red'}$ Boats) $\bowtie$ Reserves))

ρ (Tempgreen, Π_{sid} (($\sigma_{color='green'}$ Boats) $\bowtie$ Reserves))

Π_{sname} ((Tempred $\cap$ Tempgreen) $\bowtie$ Sailors)

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

Example (5)

Query 5: Find the names of sailors who have reserved all boats.

— Solution:

ρ (Tempsids, ($\Pi_{sid,bid}$ Reserves) / (Π_{bid} Boats))

Π_{sname} (Tempsids $\bowtie$ Sailors)

What if we simply do Reserves / (Π_{bid} Boats) ?

date	sid	bid
2-3-2002	007	A
3-7-2002	007	B

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

Example (6)

Query 6: To find sailors who have reserved all red boats:

ρ (Tempsids, (Π $_{sid,bid}$ Reserves) / (Π $_{bid}$ (σ $_{color='red'}$ Boats))

Π $_{sname}$ (Tempsids $\bowtie$ Sailors)

Remarks

- Only standard relational algebra is covered in the lecture
  - Relation is assumed to be a set of records (no duplicated records)
  - Extended operators such as sorting operators and aggregation operators are not covered.

Reference

- Some slides adapted from Database System Concepts – 6th Edition by Silberschatz, Korth and Sudarshan