Relational Algebra

Carla Teixeira Lopes

Bases de Dados

Licenciatura em Engenharia Informática e Computação, FEUP+FCUP

Based on Jennifer Widom slides

Agenda

Introduction to Relational Algebra

Operators

Alternate notations

Extensions to Relational Algebra

What is algebra?

Mathematical system consisting of:

Operands - variables or values from which new values can be constructed.

Operators - symbols denoting procedures that construct new values from given values.

What is Relational Algebra?

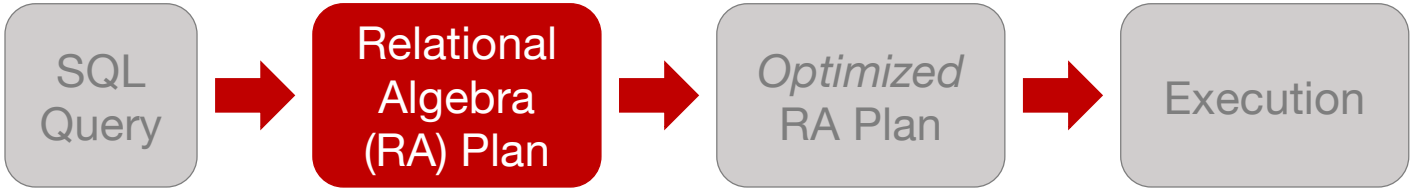
An algebra whose operands are relations or variables that represent relations.

Operators are designed to do the most common things that we need to do with relations in a database.

The result is an algebra that can be used as a query language for relations.

RDBMS Architecture

How does a SQL engine work?



Declarative query (from user)

Translate to relational algebra expression

Find logically equivalent - but more efficient - RA expression

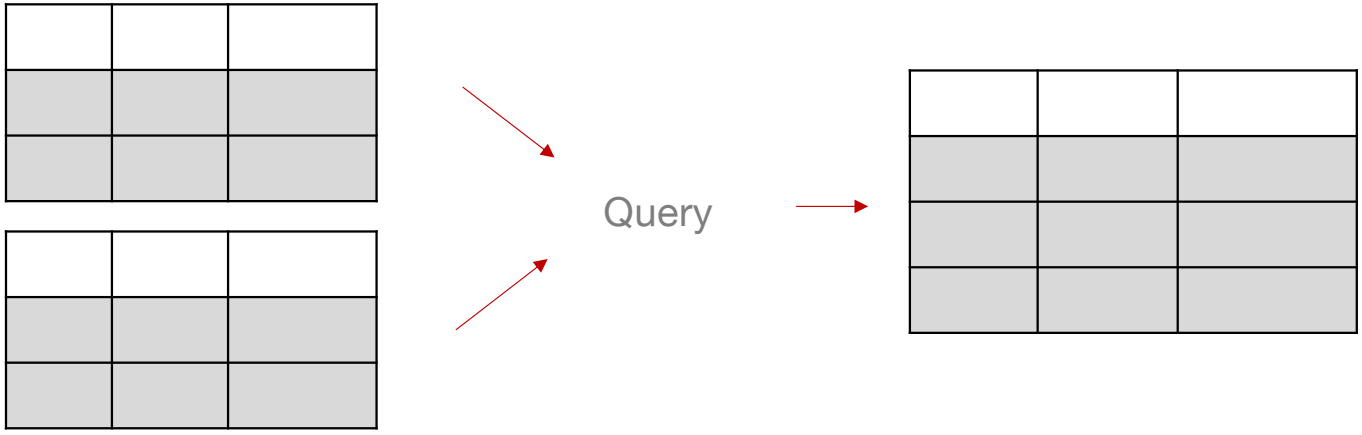
Execute each operator of the optimized plan

Relational Algebra

Formal language

Operates on relations and produce relations as a result

Operators are used to filter, slice and combine



Agenda

~~Introduction to Relational Algebra~~

Operators

Alternate notations

Extensions to Relational Algebra

College Admission Database

College (cName, state, enr)

Student (sID, sName, GPA, HS)

Apply (sID, cName, major, dec)

Demo in Relax: https://dbis-uibk.github.io/relax/

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Simplest query: relation name

Student

Student

sID	sName	GPA	HS

Select operator (σ)

Returns all tuples which satisfy a condition

Notation: $\sigma_{condition}Relation$

The condition can involve =, <, ≤, >, ≥, <>

Examples

Students with GPA>3.7

$$\sigma_{GPA>3.7}\ Student$$

Students with GPA>3.7 and HS<1000

$$\sigma_{GPA>3.7 \wedge HS<1000}\ Student$$

Applications to Stanford CS major

$$\sigma_{cName='Stanford' \wedge major='CS'} Apply$$

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	500
31	Jane	3.9	1000

Apply

sID	cName	major	dec
12	Stanford	CS	Y
23	MIT	CS	N
12	MIT	CS	N

Project operator (π)

Picks certain columns

Notation: $\pi_{A_1,\ldots,A_n}\ Relation$

sID and decision of all applications

$\pi_{sID,dec}\ Apply$

Apply

sID	cName	major	dec
12	Stanford	CS	Y
23	MIT	CS	N
12	MIT	CS	N

sID	dec
12	Y
23	N
12	N

Combining the Select and Project Operators

ID and name of students with GPA>3.7

$$\pi_{sID,sName}\ (\sigma_{GPA>3.7}\ Student)$$

Redefinition of operators

$$\sigma_{condition}(Expression)$$

$$\pi_{A_1,\ldots,A_n}\ (Expression)$$

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	50
31	Jane	3.9	1000

Sets, Bags and Lists

Sets

- Only one occurrence of each element
- Unordered elements

Bags (or multisets)

- More than one occurrence of an element
- Unordered elements and their occurrences

Lists

- More than one occurrence of an element
- Occurrences are ordered

Duplicates

Relational Algebra

- Eliminates duplicates
- Based on sets (although there is also a multiset relation algebra)

SQL

- Does not eliminate duplicates
- Based on multisets or bags

List of application majors and decisions

$\pi_{major,dec}\ Apply$

Apply

sID	cName	major	dec
12	Stanford	CS	Y
23	MIT	CS	N
12	MIT	CS	N

$\pi_{major,dec}\ Apply$

No duplicates

major	dec
CS	Y
CS	N

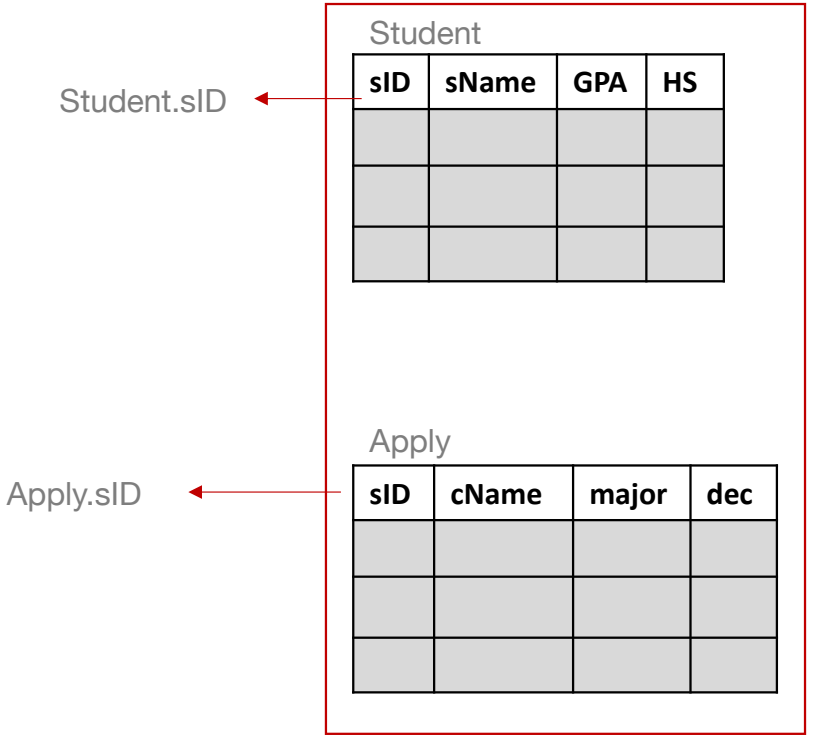
Cross-product

Also known as Cartesian product

Notation: Rel1 x Rel2

Student x Apply

Attributes with the same name are prefaced with the name of the relation



Cross-product

One tuple for every combination of tuples from the student and apply relations

Student

sID	sName	GPA	HS

S tuples

Apply

sID	cName	major	dec

A tuples

Student x Apply

S x A tuples

Example 1

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	50

Apply

sID	cName	major	dec
12	Stanford	CS	Y
23	MIT	CS	N

Student x Apply

Student.sID	sName	GPA	HS	Apply.sID	cName	major	dec
12	Mary	3.5	90	12	Stanford	CS	Y
12	Mary	3.5	90	23	MIT	CS	N
23	John	3.8	50	12	Stanford	CS	Y
23	John	3.8	50	23	MIT	CS	N

Example 2

Names and GPAs of students with HS>100 who applied to CS and were rejected

Student x Apply — All combinations

$\sigma_{Student.sID=Apply.sID}(Student\ x\ Apply)$ — Combinations that make sense

$\sigma_{Student.sID=Apply.sID\ \wedge\ HS>100\ \wedge\ major='CS'\wedge\ dec='N'}(Student\ x\ Apply)$ — Additional filtering

$\pi_{sName,GPA}(\sigma_{Student.sID=Apply.sID\ \wedge\ HS>100\ \wedge\ major='CS'\wedge\ dec='N'}(Student\ x\ Apply))$

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	5000

Apply

sID	cName	major	dec
12	Stanford	CS	Y
23	MIT	CS	N

Natural Join

Operator: ⋈

Cross product enforcing equality on all attributes with same name

Eliminate one copy of duplicate attributes

College

cName	state	enr

Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Example 1

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	50

Apply

sID	cName	major	dec
12	Stanford	CS	Y
23	MIT	CS	N

$Student \bowtie Apply$

sID	sName	GPA	HS	cName	major	dec
12	Mary	3.5	90	Stanford	CS	Y
23	John	3.8	50	MIT	CS	N

Example 2

Names and GPAs of students with HS>100 who applied to CS and were rejected

$$Student \bowtie Apply$$

$$\sigma_{HS>100 \wedge major='CS' \wedge dec='N'}(Student \bowtie Apply)$$

$$\pi_{sName,GPA}(\sigma_{HS>100 \wedge major='CS' \wedge dec='N'}(Student \bowtie Apply))$$

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	5000

Apply

sID	cName	major	dec
12	Stanford	CS	Y
23	MIT	CS	N

Example 3

Names and GPAs of students with HS>100 who applied to CS at college with enr>10,000 and were rejected

$$Student \bowtie (Apply \bowtie College)$$

$$\sigma_{HS>100 \wedge major='CS' \wedge dec='N' \wedge enr>10{,}000}(Student \bowtie (Apply \bowtie College))$$

$$\pi_{sName,GPA}(\sigma_{HS>100 \wedge major='CS' \wedge dec='N' \wedge enr>10{,}000}(Student \bowtie (Apply \bowtie College)))$$

College

cName	state	enr
MIT	NULL	30000
Stanford	NULL	20000

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	5000

Apply

sID	cName	major	dec
12	Stanford	CS	Y
23	MIT	CS	N

Natural Join

Given R(A, B, C, D), S(A, C, E), what is the schema of R ⋈ S ?

Given R(A, B, C), S(D, E), what is R ⋈ S ?

Given R(A, B), S(A, B), what is R ⋈ S ?

Natural Join does not add expressive power

Can be rewritten using the cross-product

$$Exp1 \bowtie Exp2 \equiv \pi_{schema(E1) \cup schema(E2)}(\sigma_{E1.A1=E2.A1 \wedge E1.A2=E2.A2 \wedge \ldots}(Exp1\ x\ Exp2))$$

It is convenient in terms of notation

Theta Join

A join that involves a predicate

Notation: $\bowtie_\theta$

$Exp_1 \bowtie_\theta Exp_2 \equiv \sigma_\theta(Exp_1 \; x \; Exp_2)$

Basic operation implemented in DBMS

Term “join” often means theta join

θ can be any condition

- If θ is an equality, the join is called an equi-join

Example

Student

ID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	50

Apply

sID	cName	major	dec
12	Stanford	CS	Y
23	MIT	CS	N

$$Student \bowtie_{ID=sID} Apply$$

ID	sName	GPA	HS	sID	cName	major	dec
12	Mary	3.5	90	12	Stanford	CS	Y
23	John	3.8	50	23	MIT	CS	N

Semijoin

Notation: $\ltimes$

$$Exp_1 \ltimes Exp_2 \equiv \pi_{A1,\ldots,An}\,(Exp_1 \bowtie Exp_2)$$

Where A_1, …, A_n are atributes in Exp_1

Returns the tuples of Exp_1 with a pair in Exp_2

Example

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	50
35	Jane	3.9	60

Apply

sID	cName	major	dec
12	Stanford	CS	Y
23	MIT	CS	N

$Student \ltimes Apply$

sID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	50

Union operator

Operator: ∪

List of college and student names

Can we do it using previous operators?

$\pi_{cName} College \cup \pi_{sName} Student$

Combines information vertically

Technically, the two operands must have the same schema

Not the case in the example above, but we'll correct it later

College

cName	state	enr
MIT	NULL	NULL
Washington	NULL	NULL

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	Washington	3.8	50

Example

Student

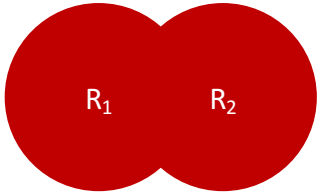
sID	sName	GPA	HS
12	Mary	3.5	90
23	Washington	3.8	50

College

cName	state	enr
MIT	NULL	NULL
Washington	NULL	NULL

$$\pi_{cName} College \cup \pi_{sName} Student$$

cName
Mary
Washington
MIT



Difference operator

Operator: –

IDs of students who didn't apply anywhere

Student

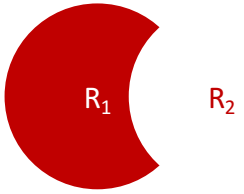
sID	sName	GPA	HS
12	Mary	3.5	90
23	Washington	3.8	50

Apply

sID	cName	major	dec
12	Stanford	CS	Y

$\pi_{sID} Student - \pi_{sID} Apply$

sID
23



Example

Names of students who didn't apply anywhere

$\pi_{sName} Student - \pi_{sID} Apply$?

$\pi_{sName}((\pi_{sID} Student - \pi_{sID} Apply) \bowtie Student)$

Schema equal to the student relation

sID	sName	GPA	HS
12	Mary	3.5	90
23	Washington	3.8	50

Student

sID	cName	major	dec
12	Stanford	CS	Y

Apply

$\pi_{sID} Student - \pi_{sID} Apply$

sID
23

Intersection operator

Operator: ∩

College

cName	state	enr
MIT	NULL	NULL
Washington	NULL	NULL

Names that are both a college name and a student name

$$\pi_{cName} College \cap \pi_{sName} Student$$

Technically, the two operands must have the same schema

Not the case in the example above, but we'll correct it later

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	Washington	3.8	50

Example

Student

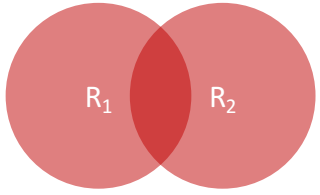
sID	sName	GPA	HS
12	Mary	3.5	90
23	Washington	3.8	50

College

cName	state	enr
MIT	NULL	NULL
Washington	NULL	NULL

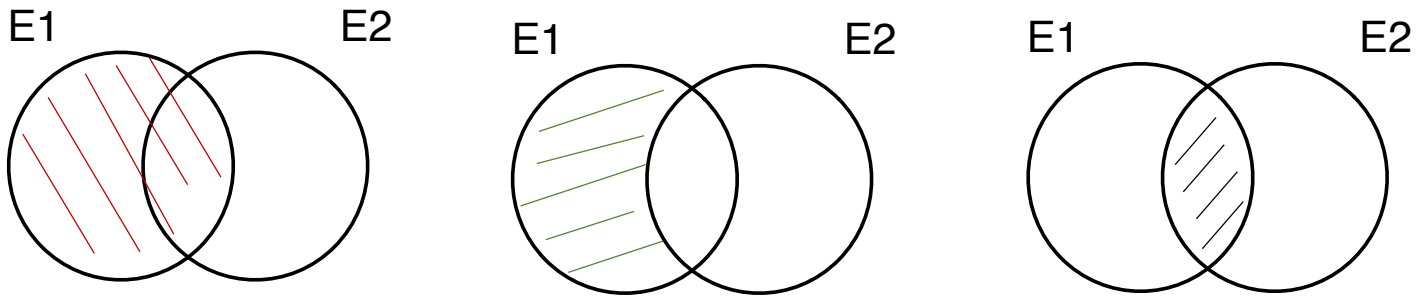
$\pi_{cName} College \cap \pi_{sName} Student$

cName
Washington



Intersection doesn't add expressive power

$E_1 \cap E_2 \equiv E_1 - (E_1 - E_2)$



Intersection doesn't add expressive power

$$E_1 \cap E_2 \equiv E_1 \bowtie E_2$$

Identical schema

Nevertheless, the intersection can be very useful in queries

Division operator

Operator: /

Identifies the attribute values from a relation that are paired with **all** the values from another relation

The division is to the Cartesian product (x) what the division is to multiplication in arithmetic

Necessary to answer queries with “all”

Cross-product and division

Division is the opposite of the cross-product

R1

A
4
8

R2

B
3
1
7

R3 = R1 x R2

R3

A	B
4	3
4	1
4	7
8	3
8	1
8	7

R3 / R2 = R1

R3 / R1 = R2

How to divide

R

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	e

/

S

C	D
c	d
e	f

= ?

Reorder the columns in R so the last ones are the ones in S

Order tuples in R by the first columns

Each R sub-tuple is part of the result if the sub-tuple of the last columns contains the divisor

How to divide

Reorder the columns in R so the last ones are the ones in S

R

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	e

/

S

C	D
c	d
e	f

= ?

How to divide

Order R by the first columns

R

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	e

/

S

C	D
c	d
e	f

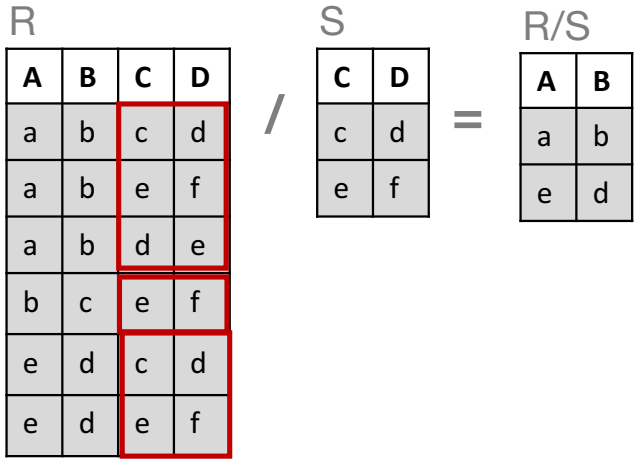
= ?

R

A	B	C	D
a	b	c	d
a	b	e	f
a	b	d	e
b	c	e	f
e	d	c	d
e	d	e	f

How to divide

Each R sub-tuple is part of the result if the sub-tuple of the last columns contains the divisor



Example

Which members are enrolled in all sports?

EnrolledIn

Member	Sport	Payment
6078	GM	25
5819	KB	30
4526	KB	30
4526	SW	20
3955	KB	30
3955	SW	20
3955	GM	25
9876	KB	0

Sports

ID	Name
KB	Kickbox
SW	Swimming
GM	Gimnastics

Example

Which members are enrolled in all sports?

Member	Sport	Payment
6078	GM	25
5819	KB	30
4526	KB	30
4526	SW	20
3955	KB	30
3955	SW	20
3955	GM	25
9876	KB	0

EnrolledIn

ID	Name
KB	Kickbox
SW	Swimming
GM	Gimnastics

Sports

$A = \pi_{Member,Sport}\ EnrolledIn$

Member	Sport
6078	GM
5819	KB
4526	KB
4526	SW
3955	KB
3955	SW
3955	GM
9876	KB

$B = \pi_{ID}\ Sports$

ID
KB
SW
GM

A/B

Member
3955

Division doesn't add expressive power

$R(a_1, ..., a_n, b_1, ..., b_m)$

$S(b_1, ..., b_m)$

$$R/S = \Pi_{a1,...,an}(R) - \Pi_{a1,...,an}[(\Pi_{a1,...,an}(R) \times S) - R]$$

Division doesn't add expressive power

$$R/S = \Pi_{a1,\ldots,an}(R) - \Pi_{a1,\ldots,an}[(\Pi_{a1,\ldots,an}(R) \times S) - R]$$

R

Member	Sport
5819	KB
4526	KB
4526	SW
3955	KB
3955	SW
3955	GM

S

ID
KB
SW
GM

$\Pi_{a1,\ldots,an}(R)$

Member
5819
4526
3955

All tuples from the first n attributes of R

Division doesn't add expressive power

$$R/S = \Pi_{a1,\ldots,an}(R) - \Pi_{a1,\ldots,an}[(\Pi_{a1,\ldots,an}(R) \times S) - R]$$

$\Pi_{a1,\ldots,an}(R)$

Member
5819
4526
3955

S

ID
KB
SW
GM

$\Pi_{a1,\ldots,an}(R) \times S$

Member	ID
5819	KB
5819	SW
5819	GM
4526	KB
4526	SW
4526	GM
3955	KB
3955	SW
3955	GM

All combinations of the first n attributes of R with the tuples of S

Division doesn't add expressive power

$$R/S = \Pi_{a1,\ldots,an}(R) - \Pi_{a1,\ldots,an}[(\Pi_{a1,\ldots,an}(R) \times S) - R]$$

$\Pi_{a1,\ldots,an}(R) \times S$

Member	ID
5819	KB
5819	SW
5819	GM
4526	KB
4526	SW
4526	GM
3955	KB
3955	SW
3955	GM

R

Member	Sport
5819	KB
4526	KB
4526	SW
3955	KB
3955	SW
3955	GM

$\Pi_{a1,\ldots,an}(R) \times S - R$

Member	ID
5819	SW
5819	GM
4526	GM

All combinations of the first n attributes of R with the tuples of S excluding the tuples that are present in R

Division doesn't add expressive power

$$R/S = \Pi_{a1,\ldots,an}(R) - \Pi_{a1,\ldots,an}[(\Pi_{a1,\ldots,an}(R) \times S) - R]$$

$\Pi_{a1,\ldots,an}(R) \times S - R$

Member	ID
5819	SW
5819	GM
4526	GM

$\Pi_{a1,\ldots,an}[(\Pi_{a1,\ldots,an}(R) \times S) - R]$

Member
5819
4526

First n attributes of all combinations of the first n attributes of R with the tuples of S excluding the tuples that are present in R

R

Member	Sport
5819	KB
4526	KB
4526	SW
3955	KB
3955	SW
3955	GM

S

ID
KB
SW
GM

Division doesn't add expressive power

$$R/S = \Pi_{a1,\ldots,an}(R) - \Pi_{a1,\ldots,an}[(\Pi_{a1,\ldots,an}(R) \times S) - R]$$

R

Member	Sport
5819	KB
4526	KB
4526	SW
3955	KB
3955	SW
3955	GM

$\Pi_{a1,\ldots,an}[(\Pi_{a1,\ldots,an}(R) \times S) - R]$

Member
5819
4526

$\Pi_{a1,\ldots,an}(R) - \Pi_{a1,\ldots,an}[(\Pi_{a1,\ldots,an}(R) \times S) - R]$

Member
3955

Rename operator

Changes the schema, not the instance

General Notation

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

Abbreviated notation to only change the relation's name

$\rho_R(E)$

Abbreviated notation to only change attribute names

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

Example

$\rho_{Student2(ID,Name,Grade,HighSchool)}(Student)$

Student

sID	sName	GPA	HS

Student2

ID	Name	Grade	HighSchool

Rename operator in use

To unify schemas for set operators

List of colleges and student names

$$\rho_{C(name)}(\pi_{cName} College) \cup \rho_{C(name)}(\pi_{sName} Student)$$

College

cName	state	enr

Student

sID	sName	GPA	HS

Rename operator in use

For disambiguation in “self-joins”

Pairs of colleges in the same state

$\sigma_{state=state}(College \times College)$?

$\sigma_{s1=s2}(\rho_{c_1(n_1,s_1,e_1)}(College) \times \rho_{c_2(n_2,s_2,e_2)}(College))$

$\rho_{c_1(n_1,s_1,e_1)}(College) \bowtie \rho_{c_2(n_2,s_2,e_2)}(College)$?

$\rho_{c_1(n_1,s,e_1)}(College) \bowtie \rho_{c_2(n_2,s,e_2)}(College)$

College

cName	state	enr

Rename operator in use

Pairs of **different** colleges in the same state

$$\sigma_{n_1 \neq n_2}(\rho_{c_1(n_1,s,e_1)}(College) \bowtie \rho_{c_2(n_2,s,e_2)}(College))$$

~~MIT MIT~~

Pairs of **different** colleges in the same state without repeated pairs

$$\sigma_{n_1 > n_2}(\rho_{c_1(n_1,s,e_1)}(College) \bowtie \rho_{c_2(n_2,s,e_2)}(College))$$

~~Stanford Berkeley~~
Berkeley Stanford

College

cName	state	enr
MIT	Massachusetts	NULL
Stanford	California	NULL
Berkeley	California	NULL

Agenda

~~Introduction to Relational Algebra~~

~~Operators~~

Alternate notations

Extensions to Relational Algebra

Assignment statements

Pairs of different colleges in the same state

$$\sigma_{n_1<n_2}(\rho_{c_1(n_1,s,e_1)}(College) \bowtie \rho_{c_2(n_2,s,e_2)}(College))$$

Alternative notation

$$C1 := \rho_{c_1(n_1,s,e_1)}(College)$$

$$C2 := \rho_{c_2(n_2,s,e_2)}(College)$$

$$CP := C1 \bowtie C2$$

$$Answ := \sigma_{n_1<n_2} CP$$

College

cName	state	enr

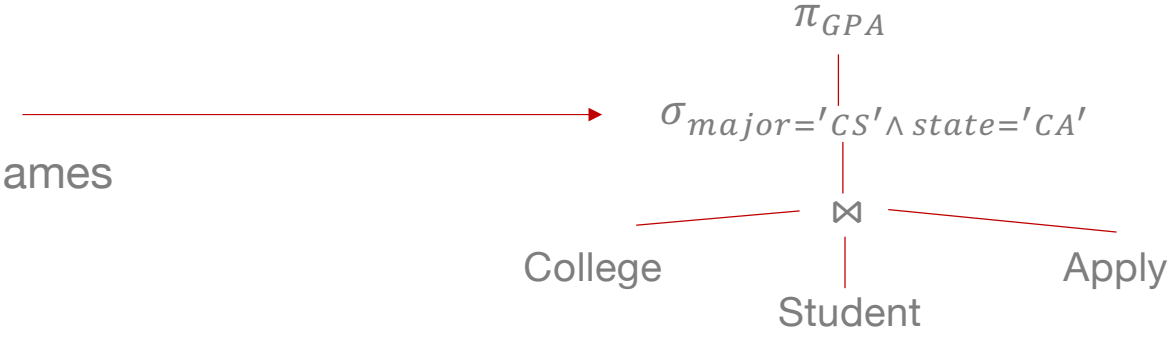
Expression trees

GPAs of students applying to CS in CA

$$\pi_{GPA}(\sigma_{major='CS' \wedge state='CA'}(Student \bowtie Apply \bowtie College))$$

Alternative notation

- The leaves are always relation names



College

cName	state	enr

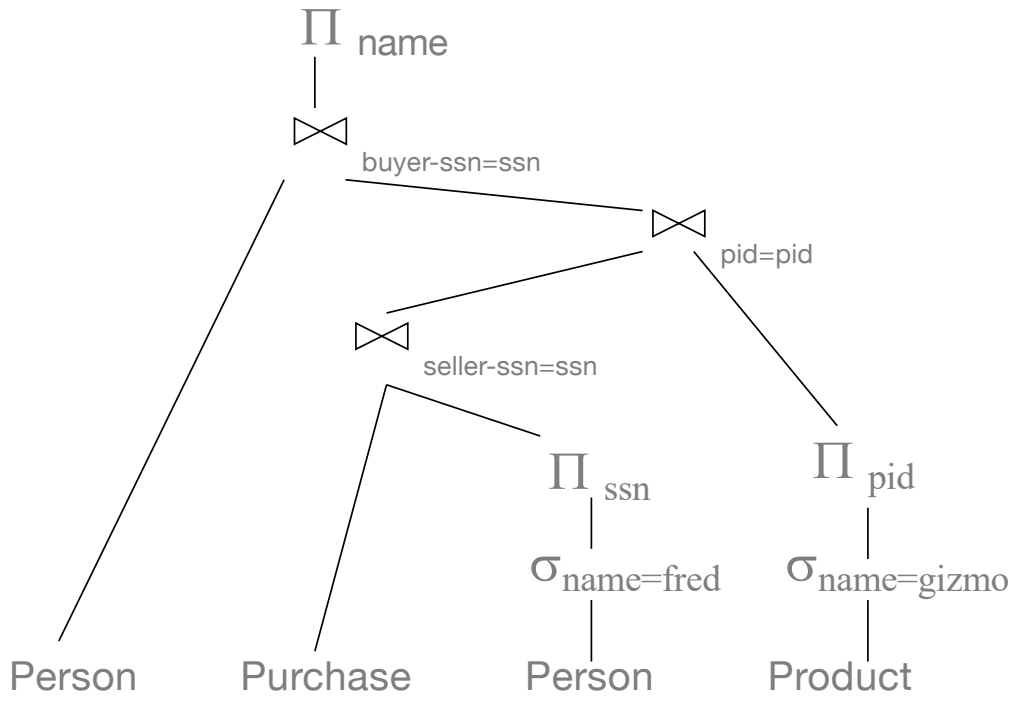
Student

sID	sName	GPA	HS

Apply

sID	cName	major	dec

Expressions can get complex



Agenda

~~Introduction to Relational Algebra~~

~~Operators~~

~~Alternate notations~~

Extensions to Relational Algebra

Extensions to Relational Algebra

Arithmetic expressions in projections

Student

sID	sName	GPA	HS
12	Mary	3.5	90
23	John	3.8	50
31	Jane	3.9	1000

List the student ids with the GPA increased in 0.2

$\pi_{sID,new=GPA+0.2}\ Student$

In the parameters of the projection

- Left side can only be the name of a new attribute
- Right side can only involve attributes of the involved relation

Extensions to Relational Algebra

Aggregation operators

cnt, sum, avg, max, min

$\pi_{cnt(*)}\ R$

Relation with 1 tuple and 1 attribute containing the number of tuples in R

College

cName	state	enr
MIT	Massachusetts	30000
Stanford	California	20000
Berkeley	California	10000
Harvard	Massachusetts	NULL

$\pi_{cnt(*)}\ College$

cnt(*)
4

Extensions to Relational Algebra

$\pi_{cnt(B)}\ R$

Relation with 1 tuple and 1 attribute containing the number of tuples in R with non-null values in B

College

cName	state	enr
MIT	Massachusetts	30000
Stanford	California	20000
Berkeley	California	10000
Harvard	Massachusetts	NULL

$\pi_{cnt(enr)}\ College$

cnt(enr)
3

Extensions to Relational Algebra

$\pi_{max(B)}\ R$

Relation with 1 tuple and 1 attribute containing the maximum value in B

College

cName	state	enr
MIT	Massachusetts	30000
Stanford	California	20000
Berkeley	California	10000
Harvard	Massachusetts	NULL

$\pi_{max(enr)}\ College$

max(enr)
30000

Extensions to Relational Algebra

$\pi_{A,\max(B)}\ R$

Relation with 1 tuple per each value of A and 2 attributes: the value of A and the maximum value of B for that value of A

College

cName	state	enr
MIT	Massachusetts	30000
Stanford	California	20000
Berkeley	California	10000
Harvard	Massachusetts	NULL

$\pi_{state,max(enr)}\ College$

state	max(enr)
Massachusetts	30000
California	20000

Relational Algebra Operators Summary

Core operators

- R
- $\sigma_{condition} E$
- $\pi_{A_1,\ldots,A_n} E$
- $E_1 \times E_2$
- $E_1 \cup E_2$
- $E_1 - E_2$
- $\rho_{R(A_1,\ldots,A_n)}(E)$

Derived operators

- $E_1 \bowtie E_2$
- $E_1 \bowtie_\theta E_2$
- $E_1 \cap E_2$
- $E_1 \ltimes E_2$
- $E_1 \,/\, E_2$

Parentheses are used for disambiguation

Readings

Jeffrey Ullman, Jennifer Widom, A first course in Database Systems 3rd Edition

- Section 2.4 – An Algebraic Query Language
- Section 5.2 – Extended Operators of Relational Algebra