CPSC 304
Introduction to Database Systems

Formal Relational Languages

Textbook Reference
Database Management Systems: 4 - 4.2
(skip the calculii)

Hassan Khosravi
Borrowing many slides from Rachel Pottinger

Learning Goals

- Identify the basic operators in Relational Algebra (RA).
- Use RA to create queries that include combining RA operators.
- Given an RA query and table schemas and instances, compute the result of the query.

Databases: the continuing saga

When last we left databases…

- We learned that they're excellent things
- We learned how to conceptually model them using ER diagrams
- We learned how to logically model them using relational schemas
- We knew how to normalize our database relations

We're almost ready to use SQL to query it, but first…

Balance, Daniel-san, is key

The mathematical foundations:

- Relational Algebra
  - Clear way of describing core concepts
  - *partially procedural*: describe what you want and how you want it

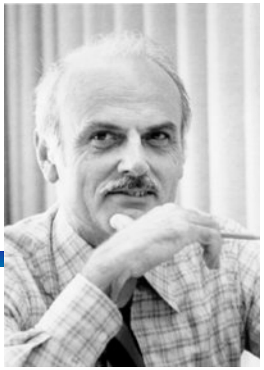
Order of operations matter

Relational Query Languages

- Allow data manipulation and retrieval from a DB
- Relational model supports simple, powerful QLs:
  - Strong formal foundation based on logic
  - Allows for much optimization via *query optimizer*
- Query Languages **!=** Programming Languages
  - QLs not intended for complex calculations
  - QLs provide *easy access* to large datasets
  - Users *do not* need to know how to navigate through complicated data structures

Relational Algebra (RA)

1970



- Basic operations   Just so all in one place
  - *Selection* (σ):   Selects a subset of rows from relation.
  - *Projection* (π):   Deletes unwanted columns from relation.
  - *Cross-product* (x): Allows us to combine two relations.
  - *Set-difference* (-): Tuples in relation 1, but not in relation 2.
  - *Union* (∪): Tuples in relation 1 and in relation 2.
  - *Rename* (ρ): Assigns a (another) name to a relation
- Additional, inessential but useful operations:
  - *Intersection (∩), join (⋈), division (/), assignment(*←)
- All operators take one or two relations as inputs and give a new relation as a result
- For the purposes of relational algebra, relations are sets
- Operations can be ***composed***. (Algebra is "closed")

Example Movies Database

Movie(<u>MovieID</u>, Title, Year)

StarsIn(<u>MovieID, StarID</u>, Character)

MovieStar(<u>StarID</u>, Name, Gender)

Example Instances

Movie:

MovieID	Title	Year
1	Star Wars	1977
2	Gone with the Wind	1939
3	The Wizard of Oz	1939
4	Indiana Jones and the Raiders of the Lost Ark	1981

StarsIn:

MovieID	StarID	Character
1	1	Han Solo
4	1	Indiana Jones
2	2	Scarlett O'Hara
3	3	Dorothy Gale

MovieStar:

StarID	Name	Gender
1	Harrison Ford	Male
2	Vivian Leigh	Female
3	Judy Garland	Female

Selection (σ (sigma))

Greek letters

- Notation: $\sigma_p(r)$
- *p* is called the **selection predicate**
- Defined as:

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

Set of tuples of r satisfying p

Where *p* is a formula in propositional calculus consisting of:

connectives : ∧ (**and**), ∨ (**or**), ¬ (**not**)

and

predicates:

<attribute> *op* <attribute> or

<attribute> *op* <constant>

where *op* is one of: =, ≠, >, ≥, <, ≤

Selection Example

Movie:

MovieID	Title	Year
1	Star Wars	1977
2	Gone with the Wind	1939
3	The Wizard of Oz	1939
4	Indiana Jones and the Raiders of the Lost Ark	1981

$\sigma_{\text{year} > 1940}(\text{Movie})$

MovieID	Title	Year
1	Star Wars	1977
4	Indiana Jones and the Raiders of the Lost Ark	1981

Selection Example #2

Find all male stars

$$\sigma_{\text{Gender = 'male'}}\text{MovieStar}$$

StarID	Name	Gender
1	Harrison Ford	Male

Projection (π (pi))

- Notation:

  $$\pi_{A1,\ A2,\ \ldots,\ Ak}\ (r)$$

  where *A1, …,Ak* are attributes (the projection list) and *r* is a relation.
- The result: a relation of the *k* attributes A1, A2, …, AK obtained from r by erasing the columns that are not listed
- Duplicate rows removed from result (relations are sets)

Projection Examples

Movie:

MovieID	Title	Year
1	Star Wars	1977
2	Gone with the Wind	1939
3	The Wizard of Oz	1939
4	Indiana Jones and the Raiders of the Lost Ark	1981

$\pi_{\text{Title, Year}}$ (Movie)

Title	Year
Star Wars	1977
Gone with the Wind	1939
The Wizard of Oz	1939
Indiana Jones and the Raiders of the Lost Ark	1981

π_{Year}(Movie)

Year
1977
1939
1981

What is $\pi_{\text{Title,Year}}(\sigma_{\text{year} > 1940}(\text{Movie}))$?

Title	Year
Star Wars	1977
Indiana Jones and the Raiders of the Lost Ark	1981

Projection Example #2

- Find the IDs of actors who have starred in movies

$$\pi_{StarID}(StarsIn)$$

StarID
1
2
3

Clicker Projection Example

- Suppose relation R(A,B,C) has the tuples:

A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6

Compute the projection $\pi_{C,B}(R)$, and identify one of its tuples from the list below.

A. (2,3)
B. (4,2,3)
C. (6,4)
D. (6,5)
E. None of the above

Clicker Projection Example

- Suppose relation R(A,B,C) has the tuples:

A	B	C
1	2	3
4	2	3
4	5	6
2	5	3
1	2	6

Compute the projection $\pi_{C,B}(R)$, and identify one of its tuples from the list below.

A. (2,3) Wrong order

B. (4,2,3) Not projected

C. (6,4) Wrong attributes

D. (6,5) right

E. None of the above

C	B
3	2
6	5
3	5
6	2

Selection and Projection Example

Find the ids of movies made prior to 1950

Movie:

MovieID	Title	Year
1	Star Wars	1977
2	Gone with the Wind	1939
3	The Wizard of Oz	1939
4	Indiana Jones and the Raiders of the Lost Ark	1981

$\pi_{\text{MovieID}} (\sigma_{\text{year} < 1950}$ Movie)

MovieID
2
3

Union, Intersection, Set-Difference

- Notation: $\boldsymbol{r} \cup \boldsymbol{s}$ $\boldsymbol{r} \cap \boldsymbol{s}$ $\boldsymbol{r} - \boldsymbol{s}$
- Defined as:

  $r \cup s = \{t \mid t \in r \text{ or } t \in s\}$

  $r \cap s = \{ t \mid t \in r \textbf{ and } t \in s \}$

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

- For these operations to be well-defined:
  1. *r, s* must have the *same arity* (same number of attributes)
  2. The attribute domains must be *compatible* (e.g., 2nd column of *r* has same domain of values as the 2nd column of *s*)
- What is the schema of the result?

Union, Intersection, and Set Difference Examples

MovieStar

StarID	Name	Gender
1	Harrison Ford	Male
2	Vivian Leigh	Female
3	Judy Garland	Female

Singer

StarID	SName	Gender
3	Judy Garland	Female
4	Christine Lavin	Female

MovieStar ∪ Singer

StarID	Name	Gender
1	Harrison Ford	Male
2	Vivian Leigh	Female
3	Judy Garland	Female
4	Christine Lavin	Female

attributes compatible!

MovieStar ∩ Singer

StarID	Name	Gender
3	Judy Garland	Female

MovieStar – Singer

StarID	Name	Gender
1	Harrison Ford	Male
2	Vivian Leigh	Female

Set Operator Example

MovieStar

StarID	Name	Gender
1	Harrison Ford	Male
2	Vivian Leigh	Female
3	Judy Garland	Female

Singer

StarID	Name	Gender
3	Judy Garland	Female
4	Christine Lavin	Female

Find the names of stars that are Singers but not MovieStars

π_{Name}(Singers – MovieStars)

Name
Christine Lavin

Cartesian (or Cross)-Product

- Notation: ***r* x *s***
- Defined as:

  $r \times s = \{ t\ q \mid t \in r \textbf{ and } q \in s \}$

- It is possible for r and s to have attributes with the same name, which creates a naming conflict.
  - In this case, the attributes are referred to solely by position.

Cartesian Product Example

MovieStar

StarID	Name	Gender
1	Harrison Ford	Male
2	Vivian Leigh	Female
3	Judy Garland	Female

StarsIn

MovieID	StarID	Character
1	1	Han Solo
4	1	Indiana Jones
2	2	Scarlett O'Hara
3	3	Dorothy Gale

MovieStar x StarsIn

1	Name	Gender	MovieID	5	Character
1	Harrison Ford	Male	1	1	Han Solo
2	Vivian Leigh	Female	1	1	Han Solo
3	Judy Garland	Female	1	1	Han Solo
1	Harrison Ford	Male	4	1	Indiana Jones
2	Vivian Leigh	Female	4	1	Indiana Jones
3	Judy Garland	Female	4	1	Indiana Jones
…	…	…	…	…	…

Rename (ρ (rho))

- Allows us to name results of relational-algebra expressions.
- Notation

$$\rho\ (X, E)$$

returns the expression *E* under the name *X*

- We can rename part of an expression, e.g.,
$\rho((StarID \rightarrow ID), \pi_{StarID,Name}(MovieStar))$

- We can also refer to positions of attributes, e.g.,
$\rho((1 \rightarrow ID)), \pi_{StarID,Name}(MovieStar)$
Is the same as above

Cartesian Product Example

MovieStar

StarID	Name	Gender
1	Harrison Ford	Male
2	Vivian Leigh	Female
3	Judy Garland	Female

StarsIn

MovieID	StarID	Character
1	1	Han Solo
4	1	Indiana Jones
2	2	Scarlett O'Hara
3	3	Dorothy Gale

MovieStar x StarsIn

1	Name	Gender	MovieID	5	Character
1	Harrison Ford	Male	1	1	Han Solo
2	Vivian Leigh	Female	1	1	Han Solo
3	Judy Garland	Female	1	1	Han Solo
1	Harrison Ford	Male	4	1	Indiana Jones
2	Vivian Leigh	Female	4	1	Indiana Jones
3	Judy Garland	Female	4	1	Indiana Jones
…	…	…	…	…	…

ρ((1→StarID1, 5→StarID2), *MovieStar x StarsIn* **)**

Additional Operations

- They can be defined in terms of the primitive operations
- They are added for convenience
- They are:
  - Join (Condition, Equi-, Natural) ($\bowtie$)
  - Division (/)
  - Assignment ($\leftarrow$)

Joins ($\bowtie$)

- *Condition Join*:

$$R \bowtie_c S = \sigma_c (R \times S)$$

- *Result schema* same as cross-product.
- Fewer tuples than cross-product
  - might be able to compute more efficiently
- Sometimes called a *theta-join*.
  - The reference to an attribute of a relation R can be by position (R.i) or by name (R.name)

Condition Join Example

MovieStar

StarID	Name	Gender
1	Harrison Ford	Male
2	Vivian Leigh	Female
3	Judy Garland	Female

StarsIn

MovieID	StarID	Character
1	1	Han Solo
4	1	Indiana Jones
2	2	Scarlett O'Hara
3	3	Dorothy Gale

$\text{MovieStar} \bowtie_{\text{MovieStar.StarID} < \text{StarsIn.StarID}} \text{StarsIn}$

1	Name	Gender	MovieID	5	Character
1	Harrison Ford	Male	2	2	Scarlett O'Hara
1	Harrison Ford	Male	3	3	Dorothy Gale
2	Vivian Leigh	Female	3	3	Dorothy Gale

Condition Join Clicker Example

- Compute R $\bowtie_{R.A < S.C \text{ and } R.B < S.D}$ S where:

R(A,B):

A	B
1	2
3	4
5	6

S(B,C,D):

B	C	D
2	4	6
4	6	8
4	7	9

Assume the schema of the result is (A, R.B, S.B, C, D). Which tuple is in the result?

A. (1,2,2,6,8)

B. (1,2,4,4,6)

C. (5,6,2,4,6)

D. All are valid

E. None are valid

Condition Join Clicker Example

- Compute R $\bowtie_{R.A < S.C \wedge R.B < S.D}$ S where:

R(A,B):

A	B
1	2
3	4
5	6

S(B,C,D):

B	C	D
2	4	6
4	6	8
4	7	9

Assume the schema of the result is (A, R.B, S.B, C, D). Which tuple is in the result?

A. (1,2,2,6,8) — (2,6,8) would have to be in S

B. (1,2,4,4,6) — (4,4,6) would have to be in S

C. (5,6,2,4,6) — Violates R.A < SC & R.B < S.D (5 > 2, and 6 = 6)

D. All are valid

E. None are valid — Correct

Equi-Join & Natural Join

- *Equi-Join*: A special case of condition join where condition contains only ***equalities***
  - *Result schema:* similar to cross-product, but contains only one copy of fields for which equality is specified
- *Natural Join*: Equijoin on *all* common attributes
  - *Result schema:* similar to cross-product, but has only one copy of each common attribute
  - No need to show the condition
  - If the two attributes have no common attributes, this would be the same as cross product.

Equi and Natural Join Examples

MovieStar

StarID	Name	Gender
1	Harrison Ford	Male
2	Vivian Leigh	Female
3	Judy Garland	Female

StarsIn

MovieID	StarID	Character
1	1	Han Solo
4	1	Indiana Jones
2	2	Scarlett O'Hara
3	3	Dorothy Gale

$\text{MovieStar} \bowtie_{\text{MovieStar.StarID} = \text{StarsIn.StarID}} \text{StarsIn}$ *or*

$\text{MovieStar} \bowtie \text{StarsIn}$

StarID	Name	Gender	MovieID	Character
1	Harrison Ford	Male	1	Han Solo
1	Harrison Ford	Male	4	Indiana Jones
3	Judy Garland	Female	3	Dorothy Gale
2	Vivian Leigh	Female	2	Scarlett O'Hara

Join Example

- Find the names of all Movie Stars who were in any Movie

$$\pi_{name}(\text{MovieStar} \bowtie \text{StarsIn})$$

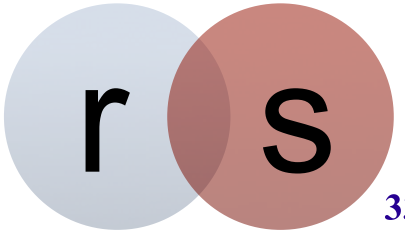
Name
Harrison Ford
Vivian Leigh
Judy Garland

- What if you wanted to find Movie Stars who were in all movies?

Assignment Operation

- Notation: $t \leftarrow E$
  assigns the result of expression E to a temporary relation t.
- Used to break complex queries to small steps.
- Assignment is always made to a temporary relation variable.
- Example: Write $r \cap s$ in terms of $\cup$ and $-$

$$temp1 \leftarrow r - s$$
$$result \leftarrow r - temp1$$



Find names of actors who have been in "Indiana Jones"

$(\sigma_{Title\ =\ \text{"Indiana Jones"}}\ Movie)$

MovieID	Title	Year
4	Indiana Jones and the Raiders of the Lost Ark	1981

$((\sigma_{Title\ =\ \text{"Indiana Jones"}}\ Movie) \bowtie StarsIn)$

MovieID	Title	Year	StarID	Character
4	Indiana Jones and the Raiders of the Lost Ark	1981	1	Indiana Jones

$(\pi_{Name}((\sigma_{Title\ =\ \text{"Indiana Jones"}}\ Movie) \bowtie StarsIn \bowtie MovieStar))$

Name
Harrison Ford

Find names of actors who have been in "Indiana Jones" or "Star Wars"

$(\sigma_{\text{Title = "Indiana Jones" v title = "Star Wars"}}$ Movie)

MovieID	Title	Year
1	Star Wars	1977
4	Indiana Jones and the Raiders of the Lost Ark	1981

$(\pi_{\text{Name}}((\sigma_{\text{Title = "Indiana Jones" v title = "Star Wars"}}$ Movie)
$\bowtie$ StarsIn $\bowtie$ MovieStar)

Name
Harrison Ford

Find the name of actors who have been in "Indiana Jones" and "Star Wars"

$Indy \leftarrow \pi_{starID}((\sigma_{Title = \text{"Indiana Jones"}}\ Movie) \bowtie StarsIn)$

$StarWars \leftarrow \pi_{starID}((\sigma_{Title = \text{"Star Wars"}}\ Movie) \bowtie StarsIn)$

$CoolPeople \leftarrow Indy \cap StarWars$

$\pi_{name}(CoolPeople \bowtie MovieStar)$

In-class Exercise

Find the names of actors who have been in a movie with the same title as the actor's name

Clicker Exercise

Find the names of actors who have been in a movie with the same title as the actor's name

Which of the following does *not* do that correctly:

A. $\pi_{Name}((Movie \bowtie StarsIn) \bowtie_{title = name \wedge StarID = MovieStar.StarID} MovieStar)$

B. $\pi_{Name}(MovieStar \bowtie_{Name = title \wedge MovieStar.StarID = StarID} (StarsIn \bowtie Movie))$

C. $\pi_{Name}((StarsIn \bowtie (\pi_{StarID,Name} MovieStar)) \bowtie_{MovieID = Movie.MovieID \wedge title = name} Movie)$

D. All are correct

E. None are correct

Clicker Exercise

Find the names of actors who have been in a movie with the same title as the actor's name

Which of the following does *not* do that correctly:

A. $\pi_{Name}((Movie \bowtie StarsIn) \bowtie_{title = name \wedge StarID = MovieStar.StarID} MovieStar)$

B. $\pi_{Name}(MovieStar \bowtie_{Name = title \wedge MovieStar.StarID = StarID} (StarsIn \bowtie Movie))$

C. $\pi_{Name}((StarsIn \bowtie (\pi_{StarID,Name} MovieStar)) \bowtie_{MovieID = Movie.MovieID \wedge title = name} Movie)$

D. All are correct

E. None are correct

All are correct (D)

Division

- Notation: ***r / s* or *r ÷ s***
- Useful for expressing queries that include a "**for all**" or "**for every**" phrase, e.g., *Find movie stars who were in **all** movies*.
- Let *r* and *s* be relations on schemas R and S respectively where
  - $r = (A_1, \ldots, A_m, B_1, \ldots, B_n)$
  - $s = (B_1, \ldots, B_n)$

  Then r / s is a relation on schema

  $r / s = (A_1, \ldots, A_m)$

  defined as

  $$r / s = \{ t \mid t \in \prod_{r-s}(r) \land \forall u \in s\ (tu \in r) \}$$

  - i.e., ***A/B* contains all *x* tuples (MovieStars) such that for *every* *y* tuple (movies) in *B*, there is an *x,y* tuple in *A*.**

Examples of Division A/B

A

sno	pno
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

B1

pno
p2

B2

pno
p2
p4

B3

pno
p1
p2
p4

A/B1

sno
s1
s2
s3
s4

A/B2

sno
s1
s4

A/B3

sno
s1

Division Clicker Question
Consider the relations

R

C	D	E
1	2	1
2	2	1
3	2	1

S

A	B
1	2
2	2
3	2
1	1

T

???
2

Which of the following is a possible expression for creating T?

A. X(D)← $\pi_A S$
   R(C,D)/X

B. Y(A)←$\pi_C R$
   S(B,A)/Y

C. Z(C)←$\pi_A S$
   R(E,C)/Z

D. All of the above

E. None of the above

Division Clicker Question Answer A exposed

R

C	D	E
1	2	1
2	2	1
3	2	1

S

A	B
1	2
2	2
3	2
1	1

T

???
2

Which of the following is a possible expression for creating T?

A. X(D)← π_AS
R(C,D)/X

B. Y(A)←π_CR
S(B,A)/Y

C. Z(C)←π_AS
R(E,C)/Z

D. All of the above

E. None of the above

C	D
1	2
2	2
3	2

D
1
2
3

Division Clicker Question Answer B exposed

R

C	D	E
1	2	1
2	2	1
3	2	1

S

A	B
1	2
2	2
3	2
1	1

T

???
2

Which of the following is a possible expression for creating T?

A. X(D)← π_AS
R(C,D)/X

B. Y(A)←π_CR
S(B,A)/Y

C. Z(C)←π_AS
R(E,C)/Z

D. All of the above

E. None of the above

B	A
2	1
2	2
2	3
1	1

A
1
2
3

Division Clicker Question Answer C exposed

R

C	D	E
1	2	1
2	2	1
3	2	1

S

A	B
1	2
2	2
3	2
1	1

T

???
2

Which of the following is a possible expression for creating T?

A. $X(D) \leftarrow \pi_A S$
R(C,D)/X

B. $Y(A) \leftarrow \pi_C R$
S(B,A)/Y

C. $Z(C) \leftarrow \pi_A S$
R(E,C)/Z

D. All of the above

E. None of the above

E	C
1	1
1	2
1	3

C
1
2
3

Division Clicker Question
Consider the relations

R

C	D	E
1	2	1
2	2	1
3	2	1

S

A	B
1	2
2	2
3	2
1	1

T

???
2

Which of the following is a possible expression for creating T?

A. X(D)← π_AS
R(C,D)/X

nothing

B. Y(A)←π_CR
S(B,A)/Y

right

C. Z(C)←π_AS
R(E,C)/Z

No, 1

D. All of the above

E. None of the above

Find the name of actors who have been in all movies

Uses division; schemas of the input relations *must be carefully chosen*:

$$\text{InAll} \leftarrow \pi_{\text{StarID, MovieID}}\ \text{StarsIn} / \pi_{\text{MovieID}}(\text{Movie})$$

$$\pi_{\text{Name}}(\text{InAll} \bowtie \text{MovieStar})$$

Find the names of actors who have been in all movies after 1950

$$\text{LateMovieIds} \leftarrow \pi_{\text{MovieID}}(\sigma_{\text{year} > 1950}(\text{Movie}))$$

$$\text{InAll} \leftarrow (\pi_{\text{StarID, MovieID}}(\text{StarsIn}) / \text{LateMovieIds})$$

$$\pi_{\text{Name}}(\text{InAll} \bowtie \text{MovieStar})$$

Clicker Exercise

Find the names of actors who have been in a movie with the same title as the actor's name

Which of the following does *not* do that correctly:

A. $\pi_{Name}((Movie \bowtie StarsIn) \bowtie_{title = name \wedge StarID = MovieStar.StarID} MovieStar)$

B. $\pi_{Name}(MovieStar \bowtie_{Name = title \wedge MovieStar.StarID = StarID} (StarsIn \bowtie Movie))$

C. $\pi_{Name}((StarsIn \bowtie (\pi_{StarID,Name} MovieStar)) \bowtie_{MovieID = Movie.MovieID \wedge title = name} Movie)$

D. All are correct

E. None are correct

All are correct (D)

Learning Goals Revisited

- Identify the basic operators in RA.
- Use RA to create queries that include combining RA operators.
- Given an RA query and table schemas and instances, compute the result of the query.