Relational Algebra

CSC365

Spring 2019

What is Algebra?

- Algebra consists of operators and operands
- Operators
    - +, -, *, /, etc.
- Operands
    - Constants
        - Numbers such as 1, 2, 3, 4
    - Variables
        - x, y
    - Algebraic expressions
        - (3/2) * (2* x + y)

Relational Algebra

- Operands
  - Relations, Algebraic Expressions
- Operators:
  - σ: Selection
  - π: Projection
  - $\times$: Cartesian Product
  - $\cup$: Union
  - $-$: Difference
  - ρ: Rename
  - $\cap$: Intersection
  - $\bowtie$: Natural Join
  - $\bowtie_c$: Theta Join

Terminologies

- Degree
    - The number of attributes in a relation
- Cardinality
    - The number of tuples in a relation

Stars

name	gender	birthdate
Tom Hanks	male	12/26/86
Kevin Bacon	male	07/08/58
Jennifer Lawrence	female	08/15/90

StarsIn

star	mid
Kevin Bacon	1
Jennifer Lawrence	2
Tom Hanks	3
Tom Hanks	4
Kevin Bacon	3

Movies

id	title	year	length	imdb
1	Footloose	1984	110	6.5
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6
4	Saving Private Ryan	1998	170	8.6

σ: Selection

- $\sigma_C(R)$
  - Applied to a relation R, produces a new relation with a subset of R’s tuples.
  - The tuples in the resulting relation are those that satisfies some condition C on the attributes of R.

Examples

$\sigma_{\text{length >= 120}}$ (Movies)

Movies

id	title	year	length	imdb
1	Footloose	1984	110	6.5
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6
4	Saving Private Ryan	1998	170	8.6

110 >= 120 ? False

142 >= 120 ? True

140 >= 120 ? True

170 >= 120 ? True

id	title	year	length	imdb
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6
4	Saving Private Ryan	1998	170	8.6

Examples

$\sigma_{\text{year >= 2000 and imdb >= 7.0}}$ (Movies)

Movies

id	title	year	length	imdb
1	Footloose	1984	110	6.5
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6
4	Saving Private Ryan	1998	170	8.6

1984 >= 2000 and 6.5 >= 7.0
? False

2012 >= 2000 and 7.2 >= 7.0
? True

1995 >= 2000 and 7.6 >= 7.0
? False

1998 >= 2000 and 8.6 >= 7.0
? False

id	title	year	length	imdb
2	The Hunger Games	2012	142	7.2

π: Projection

- $\pi_{A1,A2,...An}$ (R)
  - Produces from a relation R a new relation that has only some of R's columns.
  - Produces a set of tuples: Duplicate tuples are eliminated.

Examples

π_{star} (StarsIn)

StarsIn

star	mid
Kevin Bacon	1
Jennifer Lawrence	2
Tom Hanks	3
Tom Hanks	4
Kevin Bacon	3

star
Kevin Bacon
Jennifer Lawrence
Tom Hanks

Duplicate Preserving Projection

π^{d}_{star} (StarsIn)

StarsIn

star	mid
Kevin Bacon	1
Jennifer Lawrence	2
Tom Hanks	3
Tom Hanks	4
Kevin Bacon	3

star
Kevin Bacon
Jennifer Lawrence
Tom Hanks
Tom Hanks
Kevin Bacon

×: Cartesian Product

- R x S
  - Produces cross product of two sets R and S.
    - With each row in relation R, every row from relation S is paired.
  - The components from R precede those from S.
  - The name of each relation will be prepended to duplicate attributes for disambiguation.

Examples

R x S

R

A	B
1	2
3	4

X

S

B	C	D
2	5	6
4	7	8
9	10	11

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

$\bowtie_c$: Theta Join

- R $\bowtie_c$ S
  - Pair tuples from two relations on some condition c
  - Equivalent to σ_c (R x S)
    1. Take the product of R and S
    2. Select from the product only those tuples that satisfy the condition c.

Examples

$R \bowtie_{A < D} S$

R

A	B	C
1	2	3
6	7	8
9	7	8

S

B	C	D
2	3	4
2	3	5
7	8	10

A	R.B	R.C	S.B	S.C	D
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

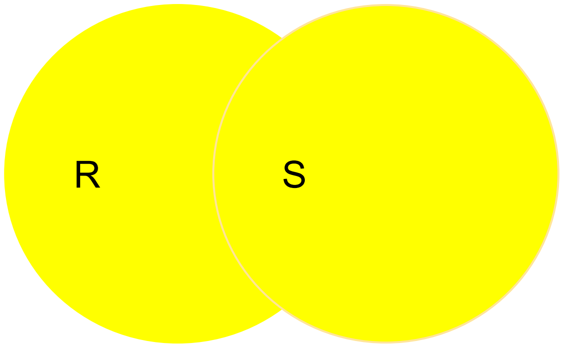
Set Operations on Relations

$R \cup S$, $R \cap S$, $R - S$

- The two relations need to have the same set of attributes
    - The type of each attribute must be the same as well.
- The order of attributes need to match

∪: Union

- R ∪ S
  - Union of two relations R and S
  - The two relations need to have the same set of attributes
  - The order of attributes need to match
  - Duplicates are removed



Examples

$\sigma_{\text{year >= 2000}}$ (Movies) U $\sigma_{\text{length < 120}}$ (Movies)

Can be re-written as

$\sigma_{\text{year >= 2000 OR length < 120}}$ (Movies)

$\sigma_{\text{year >= 2000}}$ (Movies)

id	title	year	length	imdb
2	The Hunger Games	2012	142	7.2

U

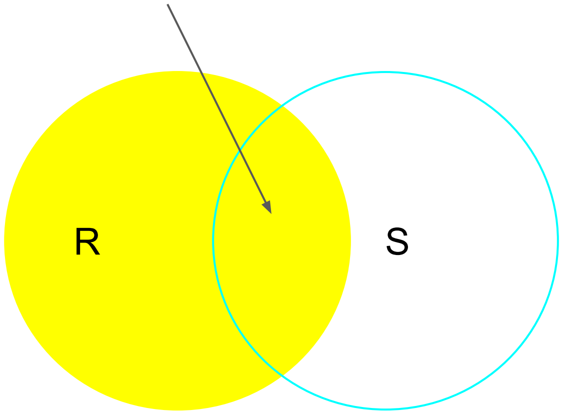
$\sigma_{\text{length < 120}}$ (Movies)

id	title	year	length	imdb
1	Footloose	1984	110	6.5

id	title	year	length	imdb
1	Footloose	1984	110	6.5
2	The Hunger Games	2012	142	7.2

∩: Intersection

- R ∩ S
  - Intersection of two relations R and S
  - The two relations need to have the same set of attributes
  - The order of attributes need to match



Examples

$\sigma_{\text{year} >= 2000}$ (Movies) ∩ $\sigma_{\text{imdb} > 7.0}$ (Movies)

Can be re-written as

$\sigma_{\text{year} >= 2000 \text{ AND imdb} > 7.0}$ (Movies)

$\sigma_{\text{imdb} > 7.0}$ (Movies)

id	title	year	length	imdb
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6
4	Saving Private Ryan	1998	170	8.6

$\sigma_{\text{year} >= 2000}$ (Movies)

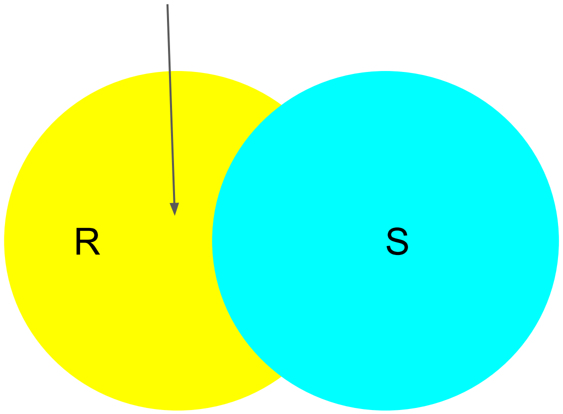
id	title	year	length	imdb
2	The Hunger Games	2012	142	7.2

∩

id	title	year	length	imdb
2	The Hunger Games	2012	142	7.2

- : Difference

- R - S
  - The set of elements that are in R but not in S.
    - R - S is not equivalent to S - R
  - The two relations need to have the same set of attributes
  - The order of attributes need to match



Examples

$\sigma_{imdb > 7.0}$ (Movies) - $\sigma_{year > 2000}$ (Movies)

$\sigma_{imdb > 7.0}$ (Movies)

id	title	year	length	imdb
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6
4	Saving Private Ryan	1998	170	8.6

-

$\sigma_{year > 2000}$ (Movies)

id	title	year	length	imdb
2	The Hunger Games	2012	142	7.2

id	title	year	length	imdb
3	Apollo 13	1995	140	7.6
4	Saving Private Ryan	1998	170	8.6

ρ: Rename

- $\rho_{s(A1,A2,...An)}(R)$
  - Rename a relation R as S with attributes named as A1,A2,...An
- $\rho_{s}(R)$
  - Rename only the relation name
- $\rho_{(A1,A2,...An)}(R)$
  - Rename only attribute names

Examples

$\rho_{\text{RS(A, B, C, E, F, G)}}(R \bowtie_{A < D} S)$

R

A	B	C
1	2	3
6	7	8
9	7	8

S

B	C	D
2	3	4
2	3	5
7	8	10

RS

A	B	C	E	F	G
1	2	3	2	3	4
1	2	3	2	3	5
1	2	3	7	8	10
6	7	8	7	8	10
9	7	8	7	8	10

How can you get a movie with maximum length?

Equi-Join

- A special case of theta join where the condition c contains only equalities.
- Result schema is similar to cross-product, but only one copy of fields for which equality is specified (projection with optional rename).
- For example
  - $R \bowtie_{B,C} S = \pi_{A,R.B\text{->}B,R.C\text{->}C,D}(R \bowtie_{R.B=S.B \text{ AND } R.C=S.C} S)$

R

A	B	C
1	2	3
6	7	8
9	7	8

S

B	C	D
2	3	4
2	3	5
7	8	10

$\Rightarrow$

A	B	C	D
1	2	3	4
1	2	3	5
6	7	8	10
9	7	8	10

⋈: Natural Join

- R ⋈ S
  - Equi-Join on all common fields.
    - Pairs only those tuples from R and S that agree in attributes which are common to the schemas of R and S
  - Can be re-written as Projection of Selection of the cartesian product of R and S, optionally with Rename applied to common attributes.

Examples

Natural Join: $\pi_{A, R.B \to B,\ R.C \to C,\ D}(\sigma_{R.B = S.B \text{ AND } R.C = S.C}\ (R \times S))$

R

A	B	C
1	2	3
6	7	8
9	7	8

S

B	C	D
2	3	4
2	3	5
7	8	10

A	B	C	D
1	2	3	4
1	2	3	5
6	7	8	10
9	7	8	10

Outer Join

- R ⟗ S : (Full) Outer Join
  - Include everything from both tables, filling missing values from both tables with null.
- R ⟕ S : Left Outer Join
  - Include everything from left table, filling missing values from right table with null
- R ⟖ S : Right Outer Join
  - Include everything from right table, filling missing values from left table with null

Examples: $R ⟗ S$

R

mid	title	year	length	imdb
1	Footloose	1984	110	6.5
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6

S

star	mid
Kevin Bacon	1
Jennifer Lawrence	2
Tom Hanks	4

mid	title	year	length	imdb	Star
1	Footloose	1984	110	6.5	Kevin Bacon
2	The Hunger Games	2012	142	7.2	Jennifer Lawrence
3	Apollo 13	1995	140	7.6	Null
4	Null	Null	Null	Null	Tom Hanks

Examples: R ⟕ S

R

mid	title	year	length	imdb
1	Footloose	1984	110	6.5
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6

S

star	mid
Kevin Bacon	1
Jennifer Lawrence	2
Tom Hanks	4

mid	title	year	length	imdb	Star
1	Footloose	1984	110	6.5	Kevin Bacon
2	The Hunger Games	2012	142	7.2	Jennifer Lawrence
3	Apollo 13	1995	140	7.6	Null

Examples: R ⟖ S

R

mid	title	year	length	imdb
1	Footloose	1984	110	6.5
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6

S

star	mid
Kevin Bacon	1
Jennifer Lawrence	2
Tom Hanks	4

mid	title	year	length	imdb	Star
1	Footloose	1984	110	6.5	Kevin Bacon
2	The Hunger Games	2012	142	7.2	Jennifer Lawrence
4	Null	Null	Null	Null	Tom Hanks

Semi-join

- R $\ltimes$ S : Left semi-join
  - Returns tuples from the left table that join with the right table.
  - Columns from the right table do not appear
- R $\rtimes$ S : Right semi-join
  - Returns tuples from the right table that join with the left table.
  - Columns from the left table do not appear

Examples

R

mid	title	year	length	imdb
1	Footloose	1984	110	6.5
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6

S

star	mid
Kevin Bacon	1
Jennifer Lawrence	2
Tom Hanks	4

$R \ltimes S$

mid	title	year	length	imdb
1	Footloose	1984	110	6.5
2	The Hunger Games	2012	142	7.2

Examples

R

mid	title	year	length	imdb
1	Footloose	1984	110	6.5
2	The Hunger Games	2012	142	7.2
3	Apollo 13	1995	140	7.6

S

star	mid
Kevin Bacon	1
Jennifer Lawrence	2
Tom Hanks	4

$R \ltimes S$

star	mid
Kevin Bacon	1
Jennifer Lawrence	2

Exercise Instances

Boats

bid	bname	bcolor
101	Gippy	red
103	Fullsail	green

Sailors

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

Reserves

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

Find names of sailors who've reserved boat #103

- Solution 1: $\pi_{sname}((\sigma_{bid=103} \text{Reserves}) \bowtie \mathit{Sailors})$

- Solution 2: $\rho\ (\mathit{Temp}1,\ \sigma_{bid=103} \text{Reserves})$

  $\rho\ (\mathit{Temp}2,\ \mathit{Temp}1 \bowtie \mathit{Sailors})$

  $\pi_{sname}(\mathit{Temp}2)$

- Solution 3: $\pi_{sname}(\sigma_{bid=103}(\text{Reserves} \bowtie \mathit{Sailors}))$

Find names of sailors who've reserved a red boat

- Information about boat color only available in Boats; so need an extra join:

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

- A more efficient solution:

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

- *A query optimizer can find this given the first solution!*

Find sailors who've reserved a red or a green boat

$\rho_{\text{rgBoats}}(\sigma_{\text{bcolor='red' OR bcolor='green'}}(\text{Boats}))$

$\pi_{\text{sname}}(\text{rgBoats} \bowtie_{\text{rgBoats.bid=RS.bid}} (\rho_{\text{RS}}(\text{Reserves} \bowtie_{\text{Reserves.sid=Sailors.sid}} \text{Sailors})))$

Find sailors names who've reserved a red <u>and</u> a green boat

- Previous approach won't work! Must identify sailors who've reserved red boats, sailors who've reserved green boats, then find the intersection (note that *sid* is a key for Sailors):

$$\rho\ (Tempred, \pi_{sid}((\sigma_{color='red'} Boats) \bowtie \text{Reserves}))$$

$$\rho\ (Tempgreen, \pi_{sid}((\sigma_{color='green'} Boats) \bowtie \text{Reserves}))$$

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

Duplicate Elimination

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

R

A	B
1	2
3	4
1	2

δ(R)

A	B
1	2
3	4

Grouping Operator

- $\gamma_L(R)$
  - L is a list of elements that are either:
    - Individual (grouping) attributes
    - AGG(A), where AGG is one of the aggregation operators and A is an attribute.
  - Possible aggregates: SUM, AVG, COUNT, MIN, and MAX
  - You can use -> to rename attributes.

Applying $\gamma_L(R)$

- Group R according to all the grouping attributes on list L.
    - That is, form one group for each distinct list of values for those attributes in R.
- Within each group, compute AGG(A) for each aggregation on list L.
- Results has grouping attributes and aggregations as attributes.

The count aggregate

- count(*)
  - Counts the number of tuples
- count(A)
  - Counts the number of tuples (including duplicates) where the attribute A is not null.

Examples

R

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

$\gamma_{A,B,COUNT(C)}(R)$ =

First, group R

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

Then, count C within groups

A	B	COUNT(C)
1	2	2
4	5	1

Examples

R

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

$\gamma_{A,B,AVG(C)}(R) =$

First, group R

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

Then, average C within groups

A	B	AVG(C)
1	2	4
4	5	6

Examples

R

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

$\gamma_{A,B,MAX(C)}(R) =$

First, group R

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

Then, find max of C within groups

A	B	MAX(C)
1	2	5
4	5	6

Examples

R

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

$\gamma_{A,B,MIN(C)}(R)$ =

First, group R

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

Then, find min of C within groups

A	B	MIN(C)
1	2	3
4	5	6

Sorting

- $\tau_{A1\ [asc|desc],\ A2[asc|desc],\ \dots}\ (R)$
  - Sort tuples on specified attributes in either:
    - asc
      - ascending order (default)
    - desc
      - descending order

Examples

R

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

$T_{A,B}(R)$ =

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

$T_{A,B,C}(R)$ =

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

$T_{A\ desc,B\ desc,C\ desc}(R)$ =

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

Division

R / S

- Not an essential operation
- A shorthand for combinations of basic operators
- Disqualifier(x) := $\pi_x((\pi_x(R) \times S) - R)$
  - Compute all x values that are not ‘disqualified’ by some y value in S.
  - x value is disqualified if by attaching y value from S, we obtain an xy tuple that is not in R.
- $\pi_x(R)$ - Disqualifier

Linear Notation for Expressions

- Invent new names for intermediate relations, and assign them values that are algebraic expressions.
- Renaming of attributes implicit in schema of new relation.

Examples: R/S

R

A	B
1	1
1	2
2	3
2	4
3	2
4	2
4	4

S

B
2

$\pi_A(R) \times S$

A	B
1	2
2	2
3	2
4	2

$(\pi_A(R) \times S) - R$

A	B
2	2

$D(A) := \pi_A((\pi_A(R) \times S) - R)$

A
2

$R/S = \pi_A(R) - D$

A
1
3
4

Examples: R/S2

R

A	B
1	1
1	2
2	3
2	4
3	2
4	2
4	4

S2

B
2
4

$\pi_A(R) \times S2$

A	B
1	2
1	4
2	2
2	4
3	2
3	4
4	2
4	4

$(\pi_A(R) \times S2) - R$

A	B
1	4
2	2
3	4

$D(A) := \pi_A((\pi_A(R) \times S2) - R)$

A
1
2
3

$R/S2 = \pi_A(R) - D$

A
4

Example: Taken / Prerequisite

T

student	course
A	101
A	202
B	101
B	202
C	101

P

course
101
202

$\pi_{student}(T) \times P$

student	course
A	101
A	202
B	101
B	202
C	101
C	202

$(\pi_{student}(T) \times P) - T$

student	course
C	202

$D(student) := (\pi_{student}(\pi_{student}(T) \times P) - T))$

student
C

$T/P = \pi_A(p) - D$

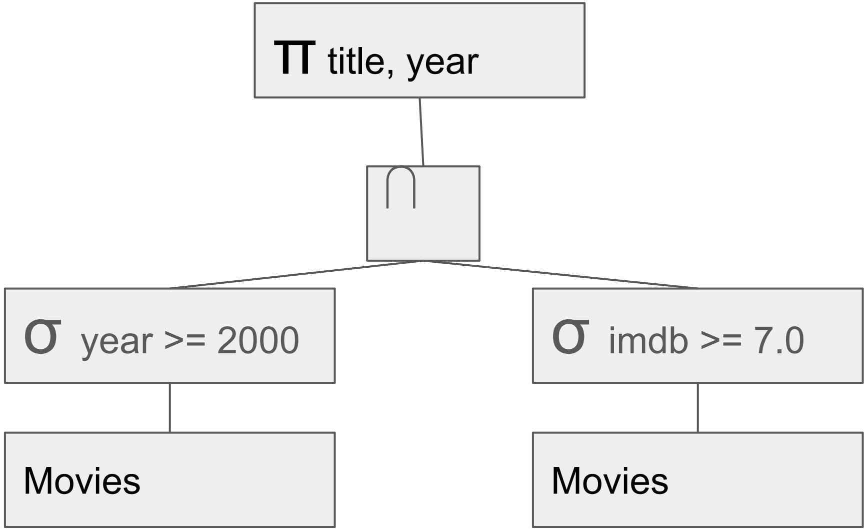
student
A
B

Combining Operations to Form Queries

- Relational Algebra allows us to form expressions of arbitrary complexity by applying operations to the result of other operations.
- It is also possible to represent expressions as expression trees.

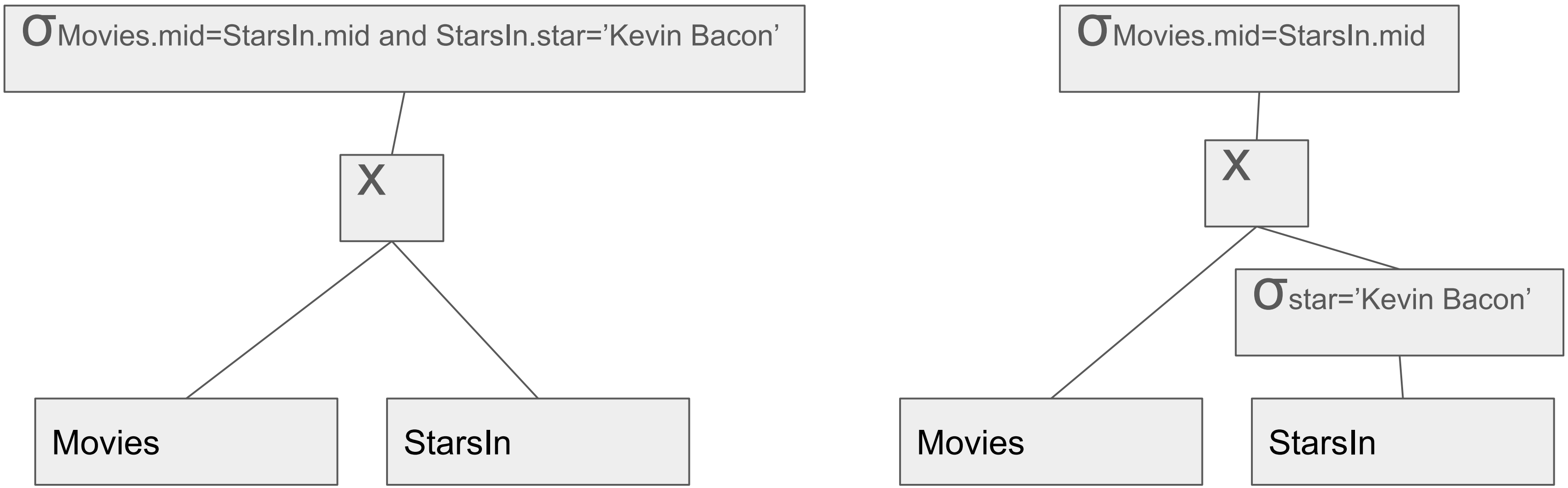
Examples: $\pi_{\text{title, year}}(\sigma_{\text{year >= 2000}}(\text{Movies}) \cap \sigma_{\text{imdb >= 7.0}}(\text{Movies}))$

1. Select tuples from Movies that have year >= 2000
2. Select tuples from Movies that have imdb score >= 7.0
3. Computer the intersection of 1 and 2
4. Project the relation from 3 onto attribute title and year



Examples

$$\sigma_{\text{Movies.mid=StarsIn.mid and StarsIn.star='Kevin Bacon'}}\ (\text{Movies x StarsIn})$$



Equivalent Expressions and Query Optimization

- An expression can be converted to other equivalent expression.
- An important job of the query optimizer is to replace one expression of relational algebra by an equivalent expression that is more efficiently evaluated.

Summary

- Operands
  - Relations, Algebraic Expressions
- Operators:
  - σ: Selection (selecting rows)
  - π: Projection (selecting columns)
  - $\times$: Cartesian Product
  - $\cup$: Union
  - $-$: Difference
  - $\cap$: Intersection
  - $\bowtie$: Natural Join
  - $\bowtie_c$: Theta Join

Summary

- Operators:
  - ⟗, ⟖, ⟕: Outer Joins - Include all tuples from one or both relations
  - ⋉, ⋊: Semi Joins - the joined relation include attributes from only one relation
  - ρ: Rename
  - δ: Duplicate Elimination
  - /: Division
  - γ: Grouping Operator
  - COUNT, SUM, AVG, MAX, MIN: Aggregation Operators
  - τ : Sorting

Summary

- Degree
  - The number of attributes in a relation
- Cardinality
  - The number of tuples in a relation