Lab#4 Extension -3

Lab#4 is extended for another week for

i) Midterm exams
ii) Synchronizing the labs and lectures

Lab#4 becomes a 3-week lab and its due date is March 13, 2019.

All other remaining labs and due dates are pushed back for one week accordingly.

Make-up Class -2

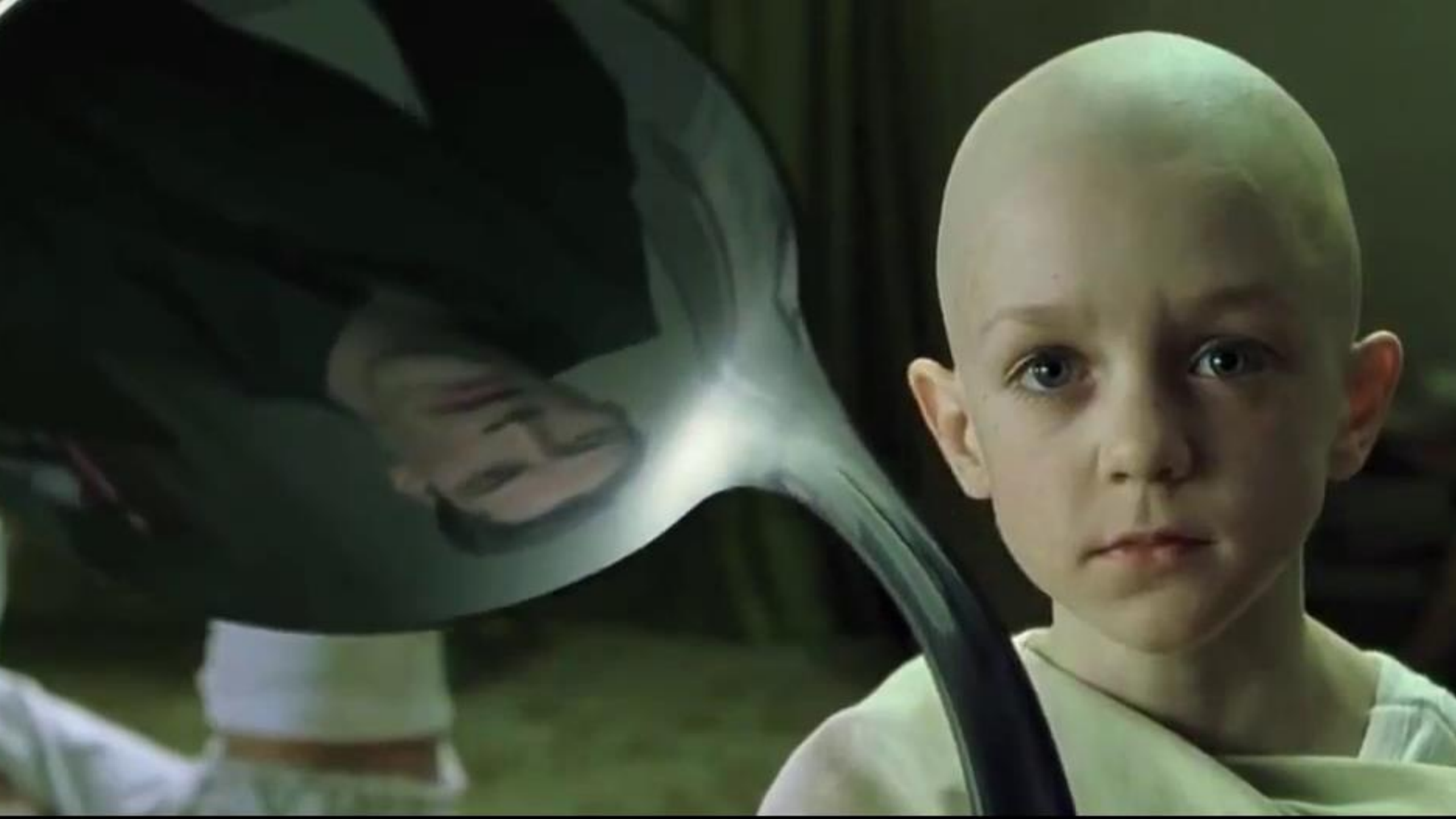
For university closure on Feb. 12, 2019
Within {Feb. 26, 27, Mar. 6, 7, 8}, the only availability is:

Date	Thursday, March 7, 2019
Time	9:00-12:00
Location	VIC608

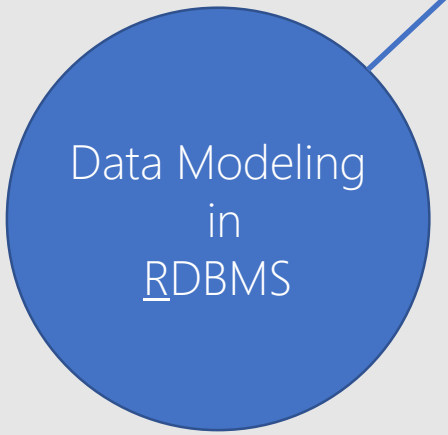
Last Week × Q4Me -1

Book vs. Slides	W09-A: CH02, CH05 (2nd Ed.), CH05 (1st Ed.)
Lab	?
Last Weeks	?

Last Week ×Q4U 0



Today



Real World Entity

Conceptual Level | Entity-Relationship Model (E/R)

| Logical Level | Relational Model

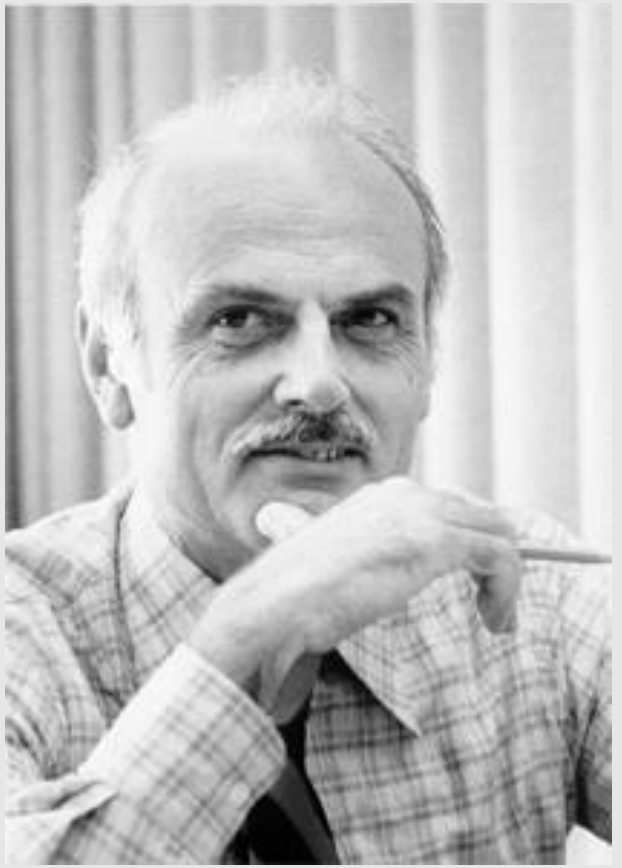
| Physical Level | SQL

Computable Entity

Welcome | Relational Algebra | Unary Op | Binary Op

Relational

Edgar Frank "Ted" Codd, IBM, 1969, 1970



Information Retrieval

A Relational Model of Data for Large Shared Data Banks

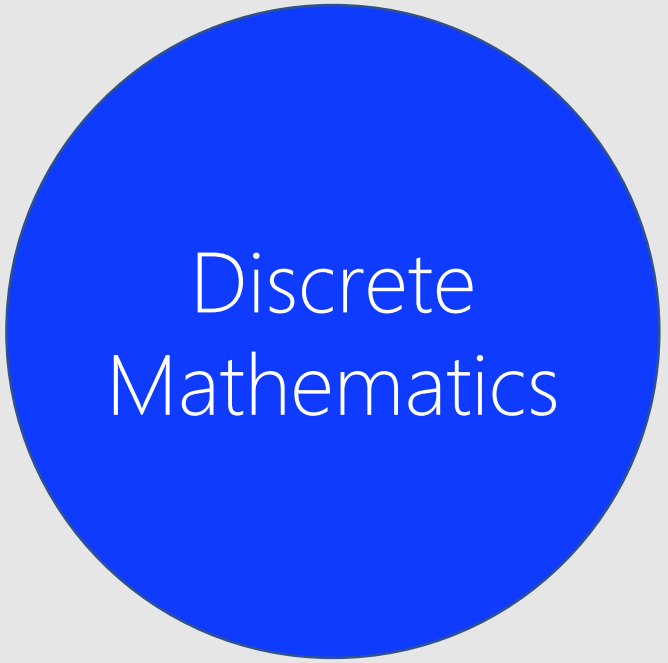
E. F. Codd
IBM Research Laboratory, San Jose, California

Relational Model × Algebra

Given a set, defining operations on elements of the set!

Given Z={integers}={..., -2, -1, 0, 1, 2, ...}
Operators & Operands:
- Unary: -(2), 2!
- Binary: 2+3, 2*3, 2^3

Closure:
- Result is also an element of the set



Relational Model × Algebra

Given relational (table) schema filled with actual data instances (rows):
Operations to SELECT Information FROM Relations

Query in Natural Language → Query in Math Formula

Relational Model × Algebra

Who made 'Pulp Fiction'?

$$\pi_{\substack{\text{Director.FirstName}\\ \text{Director.LastName}}} \left(\sigma_{\substack{\text{Movie.Title='PulpFiction'} \text{ AND}\\ \text{Movie.Id=MovieDirctor.MovieId} \text{ AND}\\ \text{Director.Id=MovieDirctor.DirectorId}}} (\text{Movie} \times \text{MovieDirector} \times \text{Director})\right)$$

Relational Model × Algebra

Who acted in 'Pulp Fiction'?

$$\pi_{\substack{\text{Actor.FirstName}\\ \text{Actor.LastName}}} \left(\sigma_{\substack{\text{Movie.Title='PulpFiction' AND}\\ \text{Movie.Id=StarIn.MovieId AND}\\ \text{Actor.Id=StarIn.ActorId}}} (\text{Movie} \times \text{StarIn} \times \text{Director})\right)$$

Relational Model × Algebra 8

Given relational (table) schema filled with actual data instances (rows):
Operations to SELECT Information FROM Relations

Unary Operation	Binary Operation (Set Theory)
π(R) : Project	R1 $\cup$ R2: Union
σ(R) : Select	R1 $\cap$ R2: Intersection
ρ(R) : Rename	R1 \ R2 : Set Difference
	R1 × R2: Cartesian Product

Relational Model × Algebra

Given relational (table) schema filled with actual data instances (rows):
Operations to SELECT Information FROM Relations
Operations to write query

Unary Operation

π(R) : Project

σ(R) : Select

ρ(R) : Rename

Binary Operation (Set Theory)

R1 $\cup$ R2: Union

R1 $\cap$ R2: Intersection

R1 \ R2 : Set Difference

R1 × R2: Cartesian Product

Algebra × Project (π)

π, pi, is used to select a subset of attributes (columns) from a relation

$$A = \pi_{<\text{attribute list}>}(R)$$

Vertical Filtering

R is a relation

<attribute list> subset of attributes of R

A is a relation including all tuples in R with only attributes in list

Algebra × Project (π)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

What are directors' name?

$$A = \pi_{\text{FirstName, LastName}}(\text{Director})$$

Algebra × Project (π)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

What are directors' name?

$$A = \pi_{\text{FirstName, LastName}}(\text{Director})$$

Algebra × Project (π)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

How many movies each director made?

$$A = \pi_{\text{FirstName, LastName, MovieCount}}(\text{Director})$$

Algebra × Project (π)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

How many movies each director made?

$$A = \pi_{\text{FirstName, LastName, MovieCount}}(\text{Director})$$

Relational Model × Algebra

Given relational (table) schema filled with actual data instances (rows):
Operations to SELECT Information FROM Relations
Operations to write query

Unary Operation

π(R) : Project

σ(R) : Select

ρ(R) : Rename

Binary Operation (Set Theory)

R1 $\cup$ R2: Union

R1 $\cap$ R2: Intersection

R1 \ R2 : Set Difference

R1 × R2: Cartesian Product

Algebra × Selection (σ)

σ, sigma, is used to select a subset of tuples from a relation based on a condition (θ) over relation's attributes.

$$A = \sigma_{\theta}(R)$$

R is a relation

θ is a Boolean expression on the attributes of R

A is a relation including tuples that make θ true

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which director was born in US?

$$A = \sigma_{\text{PlaceOfBirth}=\text{'USA'}}(\text{Director})$$

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which director was born in US?

$$A = \sigma_{\text{PlaceOfBirth}='\text{USA}'}(\text{Director})$$

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which director make more than 20 movies?

$$A = \sigma_{\text{MovieCount} > 20}(\text{Director})$$

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which director make more than 20 movies?

$$A = \sigma_{\text{MovieCount} > 20}(\text{Director})$$

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which director has same first and last names?

$$A = \sigma_{\text{FirstName} = \text{LastName}} (\text{Director})$$

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which director has same first and last names?

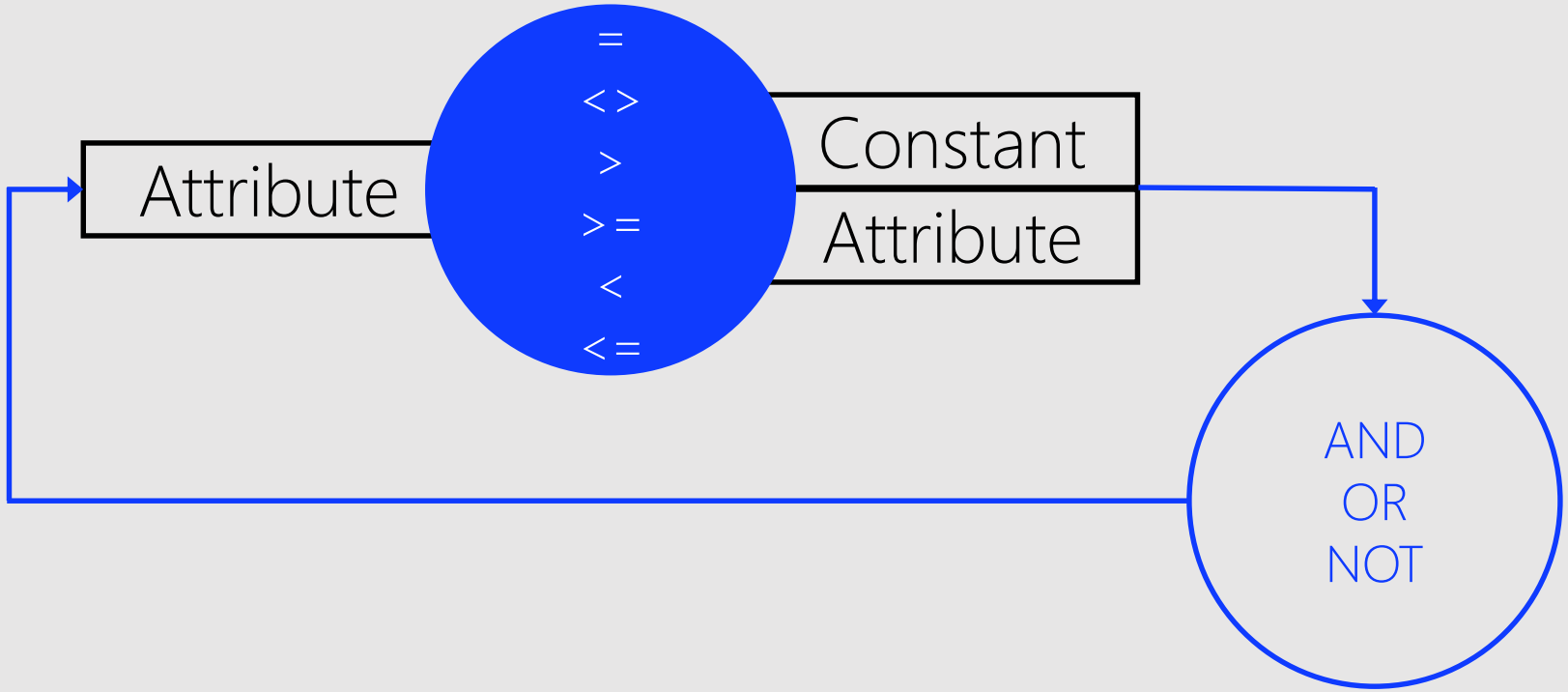
$$A = \sigma_{\text{FirstName} = \text{LastName}} (\text{Director})$$

$$= \emptyset$$

Algebra × Selection (σ)

θ can be made up of number of Boolean clauses

$A = \sigma_{\theta} (R)$



Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which American director made more than 20 movies?

$$A = \sigma_{\text{PlaceOfBirth='USA' AND MovieCount} > 20}(\text{Director})$$

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which American director made more than 20 movies or is not American?

$A1 = \sigma_{\text{PlaceOfBirth='USA' AND MovieCount > 20 OR PlaceOfBirth<>'USA'}}$ (Director)

$A2 = \sigma_{\text{PlaceOfBirth='USA' AND (MovieCount > 20 OR PlaceOfBirth<>'USA')}}$ (Director)

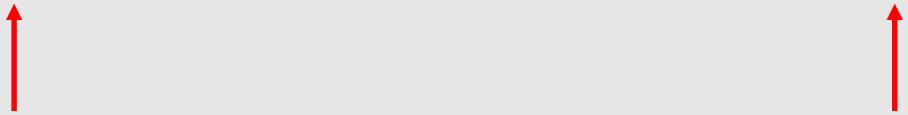
$A3 = \sigma_{\text{(PlaceOfBirth='USA' AND MovieCount > 20) OR (PlaceOfBirth<>'USA')}}$ (Director)

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which American director made more than 20 movies or is not American?

$$A2 = \sigma_{\text{PlaceOfBirth='USA' AND (MovieCount > 20 OR PlaceOfBirth<>'USA')}} (\text{Director})$$

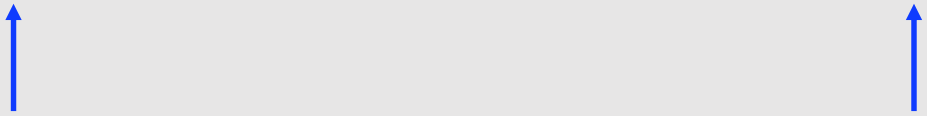


Algebra × Selection (σ)

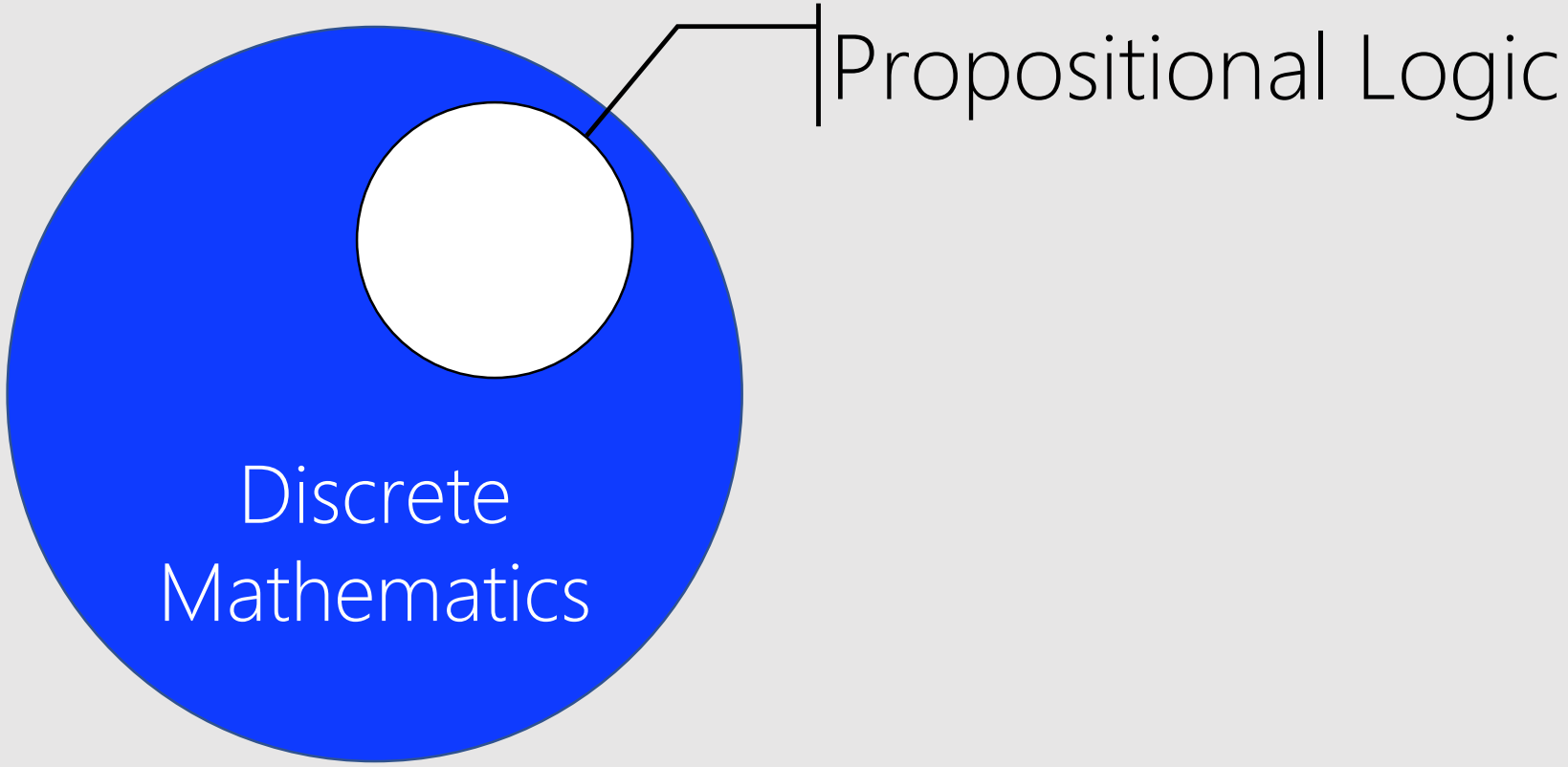
Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which American director made more than 20 movies or is not American?

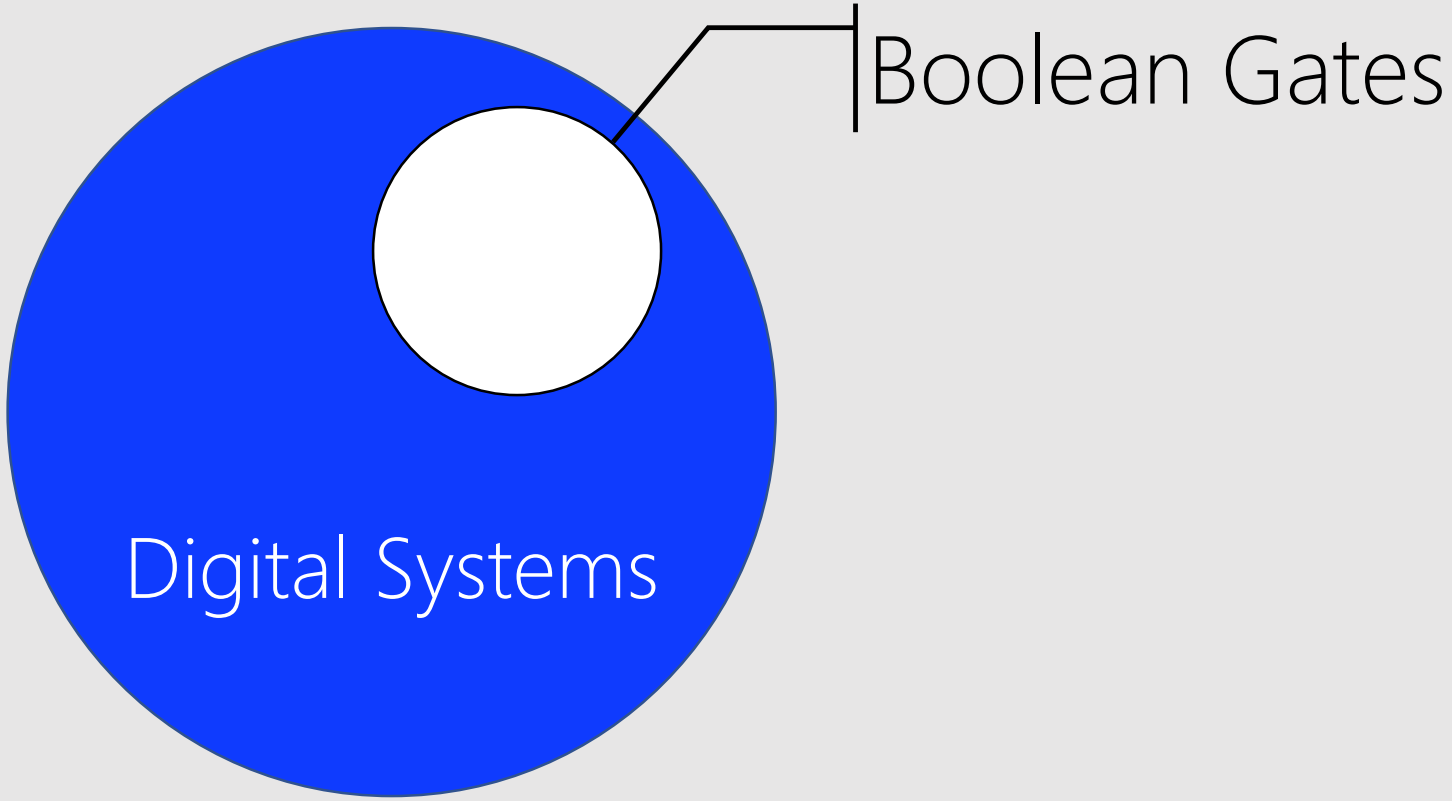
$$A3 = \sigma_{(PlaceOfBirth='USA' \text{ AND } MovieCount > 20) \text{ OR } (PlaceOfBirth<>'USA')}(Director)$$



Algebra × Selection (σ)



Algebra × Selection (σ)



Algebra × Selection (σ)

Operation Precedence

() > § > † > ‡

Commutative Law

A § B = B § A *(we say A commutes with B under §)*

Associative Law

A § (B § C) = (A § B) § C = A § B § C

Distributive Law

A § (B † C) = (A § B) † (A § C)

Algebra × Selection (σ)

Operation Precedence for Logical Operations

() > NOT > AND = OR

Commutative Law

A AND B = B AND A *(we say A commutes with B under AND)*

A OR B = B OR A *(we say A commutes with B under OR)*

Associative Law

A AND (B AND C) = (A AND B) AND C = A AND B AND C

A OR (B OR C) = (A OR B) OR C = A OR B OR C

Distributive Law

A AND (B OR C) = (A AND B) OR (A AND C)

A OR (B AND C) = (A OR B) AND (A OR C)

de Morgan´s Theorem**

NOT (A AND B) = NOT(A) OR NOT(B)

NOT (A OR B) = NOT(A) AND NOT(B)

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which American director made more than 20 movies?

$$A = \sigma_{\text{PlaceOfBirth='USA' AND MovieCount > 20}}(\text{Director})$$

$$= \sigma_{\text{MovieCount > 20 AND PlaceOfBirth='USA'}}(\text{Director})$$

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which American director made more than 20 movies?

$$A1 = \sigma_{PlaceOfBirth='USA'}(Director)$$

$$A = \sigma_{MovieCount > 20}(A1)$$

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which American director made more than 20 movies?

$$A = \sigma_{\text{MovieCount} > 20} \left(\sigma_{\text{PlaceOfBirth}=\text{'USA'}} (\text{Director})\right)$$

Algebra × Selection (σ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Which American director made more than 20 movies?

$$A = \sigma_{\text{PlaceOfBirth='USA'}}(\sigma_{\text{MovieCount} > 20}(\text{Director}))$$

$$= \sigma_{\text{PlaceOfBirth='USA' AND MovieCount} > 20}(\text{Director})$$

$$= \sigma_{\text{MovieCount} > 20 \text{ AND PlaceOfBirth='USA'}}(\text{Director})$$

$$= \sigma_{\text{MovieCount} > 20}\ (\sigma_{\text{PlaceOfBirth='USA'}}(\text{Director}))$$

Algebra × Selection (σ)

$$\sigma_{\theta}(\sigma_{\theta'}(R))$$
$$=$$
$$\sigma_{\theta \text{ AND } \theta'}(R)$$
$$=$$
$$\sigma_{\theta' \text{AND } \theta}(R)$$
$$=$$
$$\sigma_{\theta'}(\sigma_{\theta}(R))$$

Corollary: $\sigma_{\theta}(\sigma_{\theta'}(\sigma_{\theta''\ldots}(\sigma_{\theta'''}(R)) = \sigma_{\theta \text{ AND } \theta' \text{AND } \theta''\ldots \text{ AND } \theta'''}(R)$

Algebra × Selection (σ) × Project (π)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Find directors' name who make more than 20 movies?

Algebra × Selection (σ) × Project (π)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Find directors' name who make more than 20 movies?

$$A1 = \sigma_{MovieCount > 20}(Director)$$

Algebra × Selection (σ) × Project (π)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Find directors' name who make more than 20 movies?

$$A1 = \sigma_{\text{MovieCount} > 20}(\text{Director})$$

$$A = \pi_{\text{FirstName, LastName}}(A1)$$

$$= \pi_{\text{FirstName, LastName}}(\sigma_{\text{MovieCount} > 20}(\text{Director}))$$

Algebra × Selection (σ) × Project (π)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Find directors' name who make more than 20 movies?

✗ $A = \sigma_{MovieCount > 20} (\pi_{FirstName, LastName} (Director))$

Relational Model × Algebra

Given relational (table) schema filled with actual data instances (rows):
Operations to SELECT Information FROM Relations
Operations to write query

Unary Operation

π(R) : Project

σ(R) : Select

ρ(R) : Rename

Binary Operation (Set Theory)

R1 $\cup$ R2: Union

R1 $\cap$ R2: Intersection

R1 \ R2 : Set Difference

R1 × R2: Cartesian Product

Algebra × Rename (ρ)

ρ, rho, is used to rename a relation or its attributes or both

$$A = \rho_{<R'(a'/a,\ b'/b,\ \ldots)>}(R)$$

R is a relation
R' is the new name for R(a, b, ...)
a' is the new name for attribute a of R
b' is the new name for attribute b of R
...

Algebra × Rename (ρ)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

$$A = \pi_{\text{FirstName, Name}} (\rho_{\text{ActiveDirector(Name/LastName)}} (\sigma_{\text{MovieCount} > 20}(\text{Director})))$$

We'll c its real use ☺

Relational Model × Algebra

Given relational (table) schema filled with actual data instances (rows):
Operations to SELECT Information FROM Relations
Operations to write query

Unary Operation

π(R) : Project

σ(R) : Select

ρ(R) : Rename

Binary Operation (Set Theory)

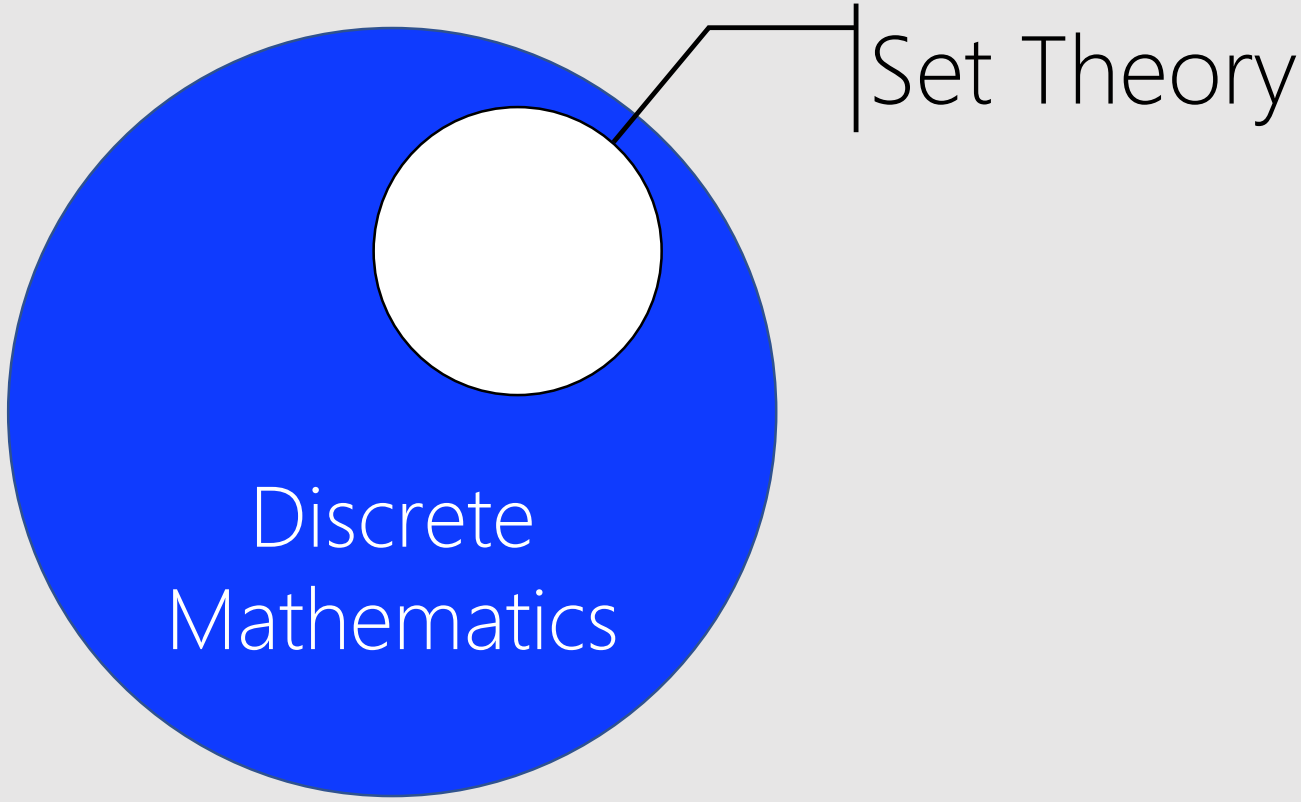
R1 $\cup$ R2: Union

R1 $\cap$ R2: Intersection

R1 \ R2 : Set Difference

R1 × R2: Cartesian Product

Algebra × Binary Operations



Relational Model × Algebra

Given relational (table) schema filled with actual data instances (rows):
Operations to SELECT Information FROM Relations
Operations to write query

Unary Operation

π(R) : Project

σ(R) : Select

ρ(R) : Rename

Binary Operation (Set Theory)

R1 $\cup$ R2: Union

R1 $\cap$ R2: Intersection

R1 \ R2 : Set Difference

R1 × R2: Cartesian Product

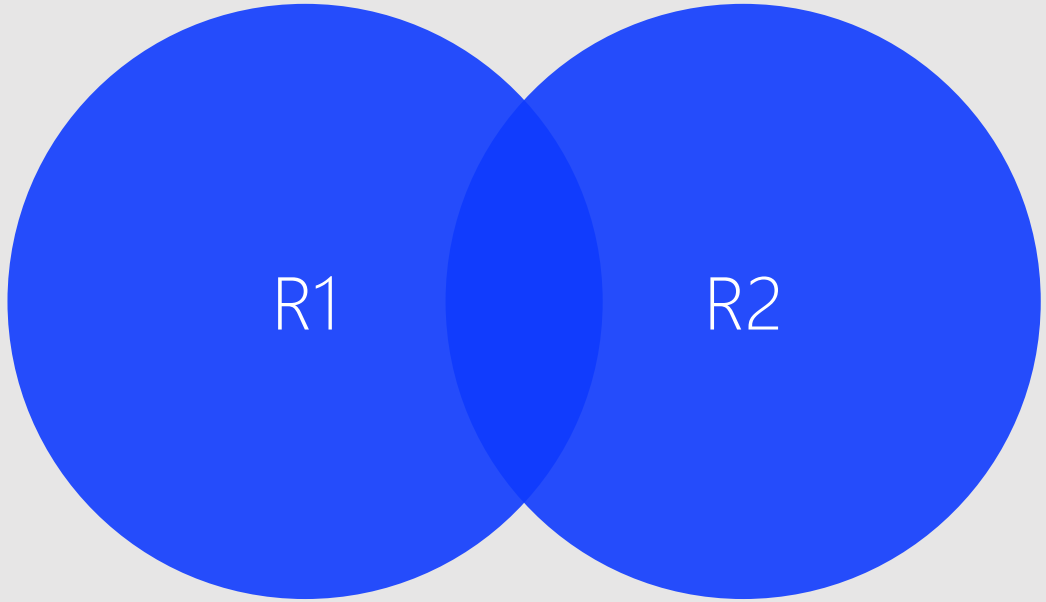
Algebra × Union (∪)

∪, union, is used to include tuples exist in either relations

$$A = R1 \cup R2$$

R1 and R2 are relations
A has all tuples from R1 and R2

No duplicate!



Venn Diagram

Algebra × Union (∪)

∪, union, is used to include tuples exist in either relations

$$A = R1 \cup R2$$

Commutative Law

$R1 \cup R2 = R2 \cup R1$

Associative Law

$R1 \cup (R2 \cup R3) = (R1 \cup R2) \cup R3 = R1 \cup R2 \cup R3$

Algebra × Union (∪)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Actor						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestLine	MovieCount
1	John	Travolta	Feb. 18, 1954	USA	You ...	61
2	Samuel	Jackson	Dec. 21, 1948	USA	Say 'w...	125
3	Uma	Thurman	Apr. 29, 1970	USA	I believe ..	51
4	Clint	Eastwood	May 31, 1930	USA	A good ..	69

Show all American cast and crews' name?

Algebra × Union (∪)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Actor						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestLine	MovieCount
1	John	Travolta	Feb. 18, 1954	USA	You ...	61
2	Samuel	Jackson	Dec. 21, 1948	USA	Say 'w...	125
3	Uma	Thurman	Apr. 29, 1970	USA	I believe ..	51
4	Clint	Eastwood	May 31, 1930	USA	A good ..	69

Actor ∪ Director

Algebra × Union (∪)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Actor						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestLine	MovieCount
1	John	Travolta	Feb. 18, 1954	USA	You ...	61
2	Samuel	Jackson	Dec. 21, 1948	USA	Say 'w...	125
3	Uma	Thurman	Apr. 29, 1970	USA	I believe ..	51
4	Clint	Eastwood	May 31, 1930	USA	A good ..	69

$$\sigma_{\text{PlaceOfBirth}=\text{'USA'}}(\text{Actor} \cup \text{Director})$$

Algebra × Union (∪)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Actor						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestLine	MovieCount
1	John	Travolta	Feb. 18, 1954	USA	You ...	61
2	Samuel	Jackson	Dec. 21, 1948	USA	Say 'w...	125
3	Uma	Thurman	Apr. 29, 1970	USA	I believe ..	51
4	Clint	Eastwood	May 31, 1930	USA	A good ..	69

$$A = \pi_{\text{FirstName, LastName}} (\sigma_{\text{PlaceOfBirth='USA'}} (\text{Actor} \cup \text{Director}))$$

$$= \pi_{\text{FirstName, LastName}} (\sigma_{\text{PlaceOfBirth='USA'}} (\text{Actor}) \cup \sigma_{\text{PlaceOfBirth='USA'}} (\text{Director}))$$

Relational Model × Algebra

Given relational (table) schema filled with actual data instances (rows):
Operations to SELECT Information FROM Relations
Operations to write query

Unary Operation

π(R) : Project

σ(R) : Select

ρ(R) : Rename

Binary Operation (Set Theory)

R1 $\cup$ R2: Union

R1 $\cap$ R2: Intersection

R1 \ R2 : Set Difference

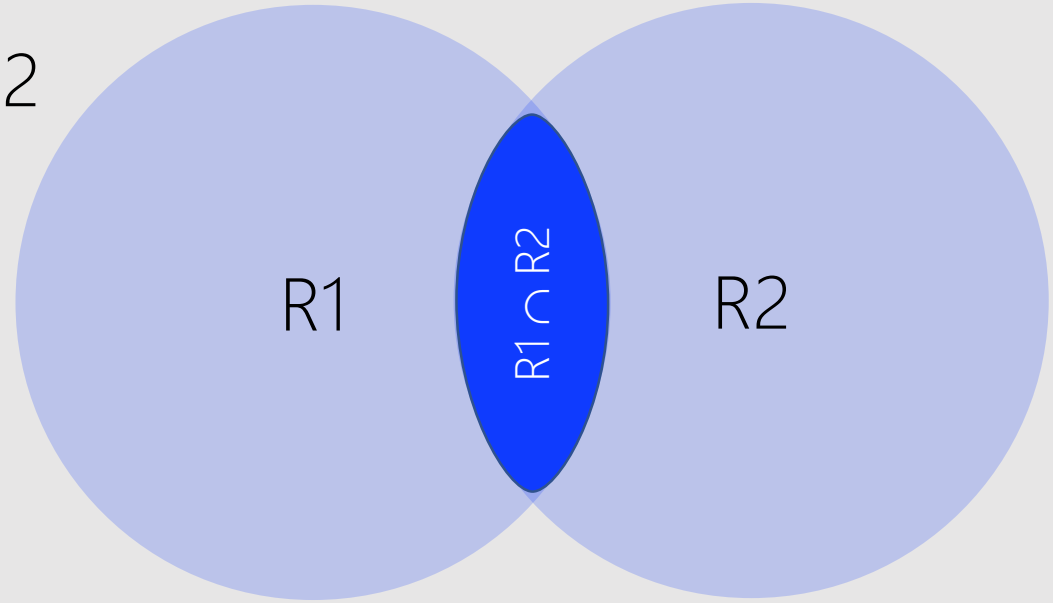
R1 × R2: Cartesian Product

Algebra × Intersection (∩)

∩, intersection, is used to include tuples exist in <u>both</u> relations

$$A = R1 \cap R2$$

R1 and R2 are relations
A has tuples exist both in R1 and R2



Algebra × Intersection (∩)

∩, intersection, is used to include tuples exist in <u>both</u> relations

$$A = R1 \cap R2$$

Commutative Law

$R1 \cap R2 = R2 \cap R1$

Associative Law

$R1 \cap (R2 \cap R3) = (R1 \cap R2) \cap R3 = R1 \cap R2 \cap R3$

Distributive Law

$R1 \cap (R2 \cup R3) = (R1 \cap R2) \cup (R1 \cap R3)$

$R1 \cup (R2 \cap R3) = (R1 \cup R2) \cap (R1 \cup R3)$

Algebra × Intersection (∩)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Actor						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestLine	MovieCount
1	John	Travolta	Feb. 18, 1954	USA	You ...	61
2	Samuel	Jackson	Dec. 21, 1948	USA	Say 'w...	125
3	Uma	Thurman	Apr. 29, 1970	USA	I believe ..	51
4	Clint	Eastwood	May 31, 1930	USA	A good ..	69

Which actor has directed a movie?

Algebra × Intersection (∩)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Actor						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestLine	MovieCount
1	John	Travolta	Feb. 18, 1954	USA	You ...	61
2	Samuel	Jackson	Dec. 21, 1948	USA	Say 'w...	125
3	Uma	Thurman	Apr. 29, 1970	USA	I believe ..	51
4	Clint	Eastwood	May 31, 1930	USA	A good ..	69

$$A = \pi_{\text{FirstName, LastName}} (\text{Actor} \cap \text{Director})$$

Algebra × Intersection (∩)

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Actor						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestLine	MovieCount
1	John	Travolta	Feb. 18, 1954	USA	You ...	61
2	Samuel	Jackson	Dec. 21, 1948	USA	Say 'w...	125
3	Uma	Thurman	Apr. 29, 1970	USA	I believe ..	51
4	Clint	Eastwood	May 31, 1930	USA	A good ..	69

$$A = \pi_{\text{FirstName, LastName}} (\text{Actor} \cap \text{Director})$$

Algebra × Intersection (∩)

FirstName	LastName
Stanley	*Kubrick*
Alfred	*Hitchcock*
Clint	*Eastwood*

FirstName	LastName
John	*Travolta*
Samuel	*Jackson*
Uma	*Thurman*
Clint	*Eastwood*

$\pi_{\text{FirstName, LastName}}(\text{Director})$

Algebra × Intersection (∩)

FirstName	LastName
Stanley	*Kubrick*
Alfred	*Hitchcock*
Clint	*Eastwood*

FirstName	LastName
John	*Travolta*
Samuel	*Jackson*
Uma	*Thurman*
Clint	*Eastwood*

$\pi_{\text{FirstName, LastName}}$ (Actor)

Algebra × Intersection (∩)

FirstName	LastName
Stanley	*Kubrick*
Alfred	*Hitchcock*
Clint	*Eastwood*

FirstName	LastName
John	*Travolta*
Samuel	*Jackson*
Uma	*Thurman*
Clint	*Eastwood*

$$A = (\pi_{\text{FirstName, LastName}}(\text{Actor})) \cap (\pi_{\text{FirstName, LastName}}(\text{Director}))$$

Relational Model × Algebra

Given relational (table) schema filled with actual data instances (rows):
Operations to SELECT Information FROM Relations
Operations to write query

Unary Operation

π(R) : Project

σ(R) : Select

ρ(R) : Rename

Binary Operation (Set Theory)

R1 $\cup$ R2: Union

R1 $\cap$ R2: Intersection

R1 \ R2 : Set Difference

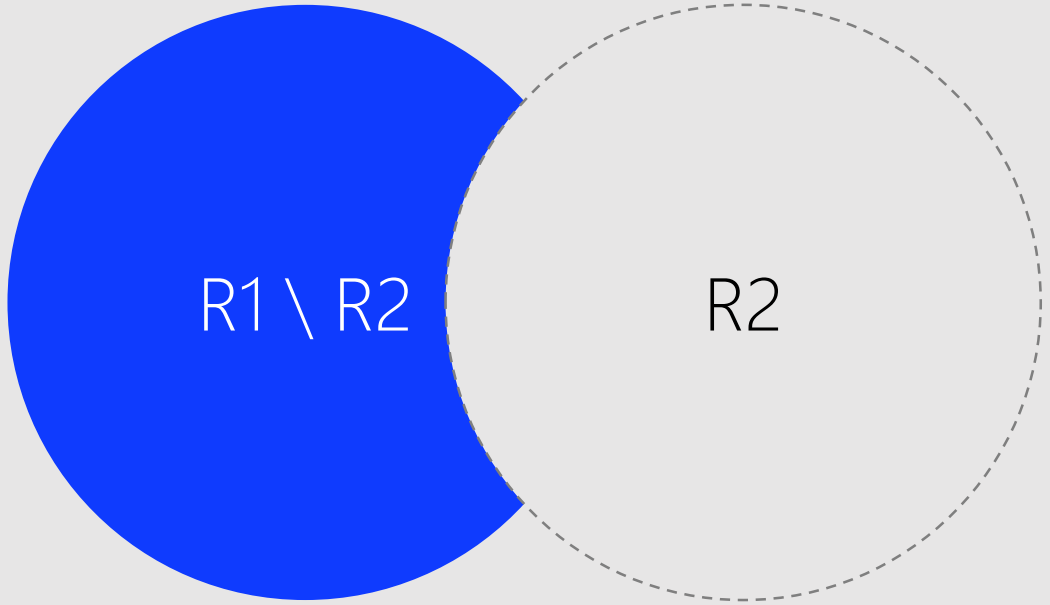
R1 × R2: Cartesian Product

Algebra × Set Difference (\ )

\ , minus, is used to exclude tuples from a relation

$$A = R1 \setminus R2$$

R1 and R2 are relations
A has tuples in R1 but not in R2



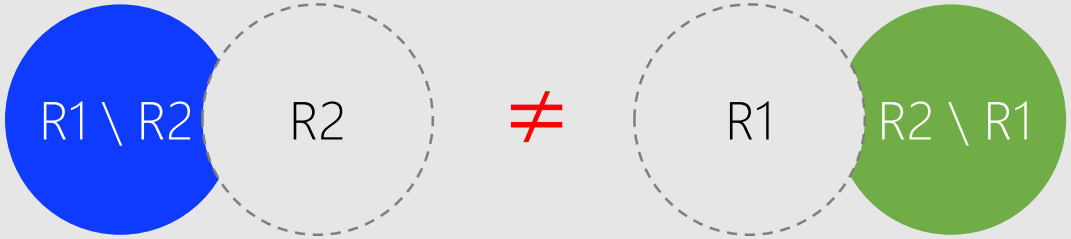
Algebra × Set Difference (\ )

\ , minus, is used to exclude tuples from a relation

$$A = R1 \setminus R2$$

Commutative Law

$R1 \setminus R2 \neq R2 \setminus R1$



Algebra × Set Difference (\ )

\ , minus, is used to exclude tuples from a relation

$$A = R1 \setminus R2$$

Associative Law

R1 \ (R2 \ R3) ≠ (R1 \ R2) \ R3 (Why?)

Algebra × Set Difference (\ )

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Actor						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestLine	MovieCount
1	John	Travolta	Feb. 18, 1954	USA	You ...	61
2	Samuel	Jackson	Dec. 21, 1948	USA	Say 'w...	125
3	Uma	Thurman	Apr. 29, 1970	USA	I believe ..	51
4	Clint	Eastwood	May 31, 1930	USA	A good ..	69

Which director never appeared in a movie?

Algebra × Set Difference (\ )

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Actor						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestLine	MovieCount
1	John	Travolta	Feb. 18, 1954	USA	You ...	61
2	Samuel	Jackson	Dec. 21, 1948	USA	Say 'w...	125
3	Uma	Thurman	Apr. 29, 1970	USA	I believe ..	51
4	Clint	Eastwood	May 31, 1930	USA	A good ..	69

$$A = \pi_{\text{FirstName, LastName}} (\text{Director} \setminus \text{Actor})$$

Algebra × Set Difference (\ )

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	Stanley	Kubrick	Jul. 26, 1928	USA	1	13
2	Alfred	Hitchcock	Aug. 13, 1899	England	203	47
3	Clint	Eastwood	May 31, 1930	USA	803	35

Actor						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestLine	MovieCount
1	John	Travolta	Feb. 18, 1954	USA	You ...	61
2	Samuel	Jackson	Dec. 21, 1948	USA	Say 'w...	125
3	Uma	Thurman	Apr. 29, 1970	USA	I believe ..	51
4	Clint	Eastwood	May 31, 1930	USA	A good ..	69

Director \ Actor

Algebra × Set Difference (\ )

FirstName	LastName
Stanley	*Kubrick*
Alfred	*Hitchcock*
Clint	*Eastwood*

FirstName	LastName
John	*Travolta*
Samuel	*Jackson*
Uma	*Thurman*
Clint	*Eastwood*

$\pi_{\text{FirstName, LastName}}$ (Director)

Algebra × Set Difference (\ )

FirstName	LastName
Stanley	*Kubrick*
Alfred	*Hitchcock*
Clint	*Eastwood*

FirstName	LastName
John	*Travolta*
Samuel	*Jackson*
Uma	*Thurman*
Clint	*Eastwood*

$\pi_{\text{FirstName, LastName}}$ (Actor)

Algebra × Set Difference (\ )

FirstName	LastName
Stanley	*Kubrick*
Alfred	*Hitchcock*
Clint	*Eastwood*

FirstName	LastName
John	*Travolta*
Samuel	*Jackson*
Uma	*Thurman*
Clint	*Eastwood*

$$A = (\pi_{\text{FirstName, LastName}}(\text{Director})) \setminus (\pi_{\text{FirstName, LastName}}(\text{Actor}))$$

Algebra × Set Compatibility

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Movie			
Id	Title	Language	RunningTime
1	*2001: A Space Odyssey*	*English*	*142*
2	*Rosemary's Baby*	*English*	*136*

Director and Movie do not share same

i) Number of attributes

ii) Datatype in attributes

A = Director ∪ Movie

A = Director ∩ Movie

A = Director \ Movie

Algebra × Set Compatibility

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Movie			
Id	Title	Language	RunningTime
1	*2001: A Space Odyssey*	*English*	*142*
2	*Rosemary's Baby*	*English*	*136*

Find movies which are NOT the best movies of any directors?

A = Movie \ Director

Algebra × Set Compatibility

BestMovieId
1
203
803

Id
1
2

Find movies which are NOT the best movies of any directors?

$$A = (\pi_{Id}(Movie)) \setminus (\pi_{BestMovieId}(Director))$$

Relational Model × Algebra

Given relational (table) schema filled with actual data instances (rows):
Operations to SELECT Information FROM Relations
Operations to write query

Unary Operation

π(R) : Project

σ(R) : Select

ρ(R) : Rename

Binary Operation (Set Theory)

R1 $\cup$ R2: Union

R1 $\cap$ R2: Intersection

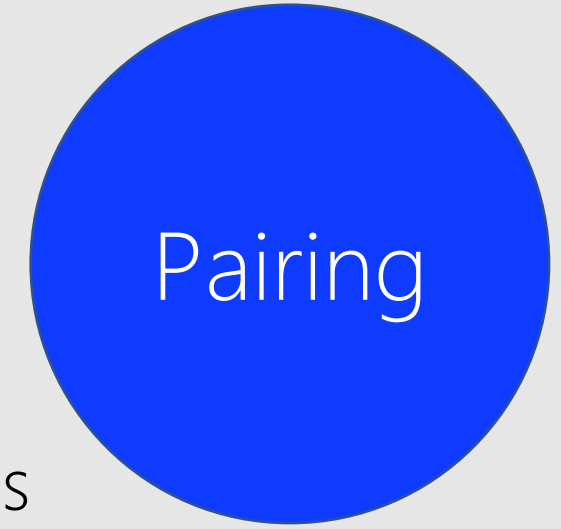
R1 \ R2 : Set Difference

R1 × R2: Cartesian Product

Algebra × Cartesian Product (×)

×, product, is used to pair attributes (columns) and tuples (rows) of two relations

$$A = R1 \times R2$$



$R1(a_1, a_2, \ldots, a_n)$ and $R2(a'_1, a'_2, \ldots, a'_m)$ are relations

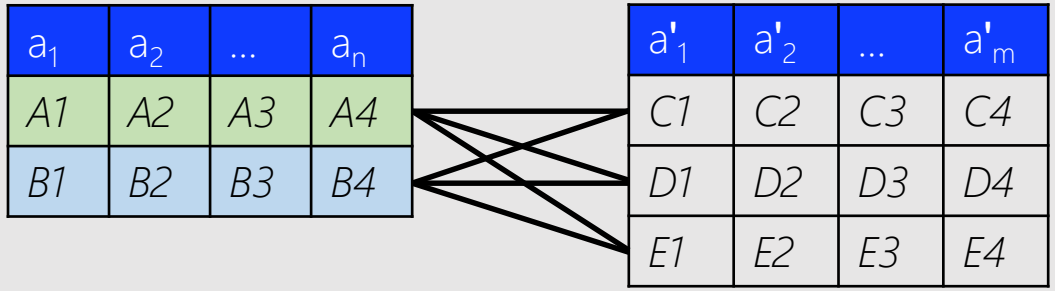
A is a relation with all attributes in R1 & R2:

$A(a_1, a_3, \ldots, a_n, a'_1, a'_2, \ldots, a'_m)$

In A, <u>each</u> tuple of R1 is paired with <u>all</u> tuples of R2

Algebra × Cartesian Product (×)

R1 × R2 =



a_1	a_2	...	a_n
A1	*A2*	*A3*	*A4*
B1	*B2*	*B3*	*B4*

a'_1	a'_2	...	a'_m
C1	*C2*	*C3*	*C4*
D1	*D2*	*D3*	*D4*
E1	*E2*	*E3*	*E4*

A

a_1	a_2	...	a_n	a'_1	a'_2	...	a'_m
A1	*A2*	*A3*	*A4*	*C1*	*C2*	*C3*	*C4*
A1	*A2*	*A3*	*A4*	*D1*	*D2*	*D3*	*D4*
A1	*A2*	*A3*	*A4*	*E1*	*E2*	*E3*	*E4*
B1	*B2*	*B3*	*B4*	*C1*	*C2*	*C3*	*C4*
B1	*B2*	*B3*	*B4*	*D1*	*D2*	*D3*	*D4*
B1	*B2*	*B3*	*B4*	*E1*	*E2*	*E3*	*E4*

Algebra × Cartesian Product (×)

×, product, is used to pair attributes (columns) and tuples (rows) of two relations

$$A = R1 \times R2$$

attributes in A, $n + m$

tuples in A, $|A| = |R1| \times |R2|$

Commutative Law

$R1 \times R2 = R2 \times R1$

Associative Law

$R1 \times (R2 \times R3) = (R1 \times R2) \times R3 = R1 \times R2 \times R3$

Algebra × Cartesian Product (×)

×, product, is used to pair attributes (columns) and tuples (rows) of two relations

$$A = R1 \times R2$$

R1 and R2 do not have to be set compatible

Any two relations can be paired.

Algebra × Cartesian Product (×)

×, product, is used to pair attributes (columns) and tuples (rows) of two relations

$$A = R1 \times R2$$

R1 and R2 do not have to be set compatible

Any two relations can be paired.

But if you want to pair two or more relations, which ones?

Algebra × Cartesian Product (×)

Movie			
Id	Title	Language	RunningTime
1	2001: A Space Odyssey	English	142
2	Rosemary's Baby	English	136

User		
Id	Username	Password
1	fani	***
2	cjason	***
3	h_f492	***

What meaningful queries can be answered by pairing these two relations?

Algebra × Cartesian Product (×)

Movie			
Id	Title	Language	RunningTime
1	2001: A Space Odyssey	English	142
2	Rosemary's Baby	English	136

MovieGenre	
MovieId	GenreId
1	1
1	3
2	6
2	7
2	2

Genre	
Id	Title
1	Sci-fi
2	Action
3	Adventure
4	Comedy
5	Crime
6	Drama
7	Horror

What meaningful queries can be answered by pairing these two relations?

Algebra × Cartesian Product (×)

Movie × MovieGenre					
Id	Title	Language	RunningTime	MovieId	GenreId
1	2001: A Space Odyssey	English	142	1	1
1	2001: A Space Odyssey	English	142	1	3
1	2001: A Space Odyssey	English	142	2	6
1	2001: A Space Odyssey	English	142	2	7
1	2001: A Space Odyssey	English	142	2	2
2	Rosemary's Baby	English	136	1	1
2	Rosemary's Baby	English	136	1	3
2	Rosemary's Baby	English	136	2	6
2	Rosemary's Baby	English	136	2	7
2	Rosemary's Baby	English	136	2	2

Genre	
Id	Title
1	Sci-fi
2	Action
3	Adventure
4	Comedy
5	Crime
6	Drama
7	Horror

Algebra × Cartesian Product (×)

Movie × MovieGenre					
Id	Title	Language	RunningTime	MovieId	GenreId
1	2001: A Space Odyssey	English	142	1	1
1	2001: A Space Odyssey	English	142	1	3
1	2001: A Space Odyssey	English	142	2	6
1	2001: A Space Odyssey	English	142	2	7
1	2001: A Space Odyssey	English	142	2	2
2	Rosemary's Baby	English	136	1	1
2	Rosemary's Baby	English	136	1	3
2	Rosemary's Baby	English	136	2	6
2	Rosemary's Baby	English	136	2	7
2	Rosemary's Baby	English	136	2	2

Genre	
Id	Title
1	Sci-fi
2	Action
3	Adventure
4	Comedy
5	Crime
6	Drama
7	Horror

Algebra × Cartesian Product (×)

Id	Title	Language	RunningTime	MovieId	GenreId
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*1*
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*3*

σ Id=MovieId AND
Title='2001: A Space Odyssey'

(Movie×MovieGenre)

Genre	
Id	Title
1	*Sci-fi*
2	*Action*
3	*Adventure*
4	*Comedy*
5	*Crime*
6	*Drama*
7	*Horror*

Algebra × Cartesian Product (×)

Id	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	2001: A Space Odyssey	English	142	1	1	1	Sci-fi
1	2001: A Space Odyssey	English	142	1	1	2	Action
1	2001: A Space Odyssey	English	142	1	1	3	Adventure
1	2001: A Space Odyssey	English	142	1	1	4	Comedy
1	2001: A Space Odyssey	English	142	1	1	5	Crime
1	2001: A Space Odyssey	English	142	1	1	6	Drama
1	2001: A Space Odyssey	English	142	1	1	7	Horror
1	2001: A Space Odyssey	English	142	1	3	1	Sci-fi
1	2001: A Space Odyssey	English	142	1	3	2	Action
1	2001: A Space Odyssey	English	142	1	3	3	Adventure
1	2001: A Space Odyssey	English	142	1	3	4	Comedy
1	2001: A Space Odyssey	English	142	1	3	5	Crime
1	2001: A Space Odyssey	English	142	1	3	6	Drama
1	2001: A Space Odyssey	English	142	1	3	7	Horror

$$(\sigma_{\text{Id=MovieId AND Title='2001: A Space Odyssey'}}(\text{Movie} \times \text{MovieGenre}) \times \text{Genre})$$

Algebra × Cartesian Product (×)

Id	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*1*	*1*	*Sci-fi*
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*3*	*3*	*Adventure*

$$A = \sigma_{Id=GenreId}(\sigma_{Id=MovieId \text{ AND } Title='2001: A Space Odyssey'}(Movie \times MovieGenre)) \times Genre)$$

Algebra × Cartesian Product (×)

×, product, is used to pair attributes (columns) and tuples (rows) of two relations

$$A = R1 \times R2$$

R1 and R2 do not have to be set compatible

Any two relations can be paired.

But if you want to pair two or more relations, which ones?

- Those have relationship in E/R diagram
- Those have linked by foreign keys (FK) in relational model

Algebra × Cartesian Product (×)

Id	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*1*	*1*	*Sci-fi*
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*3*	*3*	*Adventure*

$$A = \sigma_{Id=GenreId}(\sigma_{Id=MovieId \text{ AND } Title='2001: A Space Odyssey'}(Movie \times MovieGenre)) \times Genre)$$

Ambiguous name reference! Solution?

Algebra × Cartesian Product (×)

Id	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*1*	*1*	*Sci-fi*
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*3*	*3*	*Adventure*

$$A = \sigma_{\text{Genre.Id=GenreId}}(\sigma_{\text{Movie.Id=MovieId AND Title='2001: A Space Odyssey'}}(\text{Movie} \times \text{MovieGenre})) \times \text{Genre})$$

Ambiguous name reference! Solution?

•, dot, namespace operator

Algebra × Cartesian Product (×)

MId	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*1*	*1*	*Sci-fi*
1	*2001: A Space Odyssey*	*English*	*142*	*1*	*3*	*3*	*Adventure*

$$A = \sigma_{Id=GenreId}(\sigma_{\substack{MId=MovieId \text{ AND} \\ Title='2001: A Space Odyssey'}}(\rho_{R(MId/Id)}(Movie)) \times MovieGenre)) \times Genre)$$

Ambiguous name reference! Solution?

ρ, rho, rename operator

Relational Model × Algebra

Who made 'Pulp Fiction'?

Relational Model × Algebra

Who made 'Pulp Fiction'?

i) Find the movie in Movie relation
ii) Find directors in Director relation
iii) Find who made a movie in MovieDirector relation

Relational Model × Algebra

Who made 'Pulp Fiction'?

$$\sigma_{\text{Director.Id}=\text{MovieDirctor.DirectorId}}\left(\sigma_{\text{Movie.Title}=\text{'PulpFiction'}\ \text{AND}\ \text{Movie.Id}=\text{MovieDirctor.MovieId}}(\text{Movie} \times \text{MovieDirector})\right) \times \text{Director}$$

Relational Model × Algebra

Who made 'Pulp Fiction'?

$$\sigma_{\text{Director.Id=MovieDirctor.DirectorId}}\left(\sigma_{\text{Movie.Title='PulpFiction' AND Movie.Id=MovieDirctor.MovieId}}(\text{Movie} \times \text{MovieDirector})\right) \times \text{Director}$$

Selection (σ) is commutative

Relational Model × Algebra

Who made 'Pulp Fiction'?

$$\sigma_{\text{Director.Id=MovieDirctor.DirectorId}}(\sigma_{\text{Movie.Title='PulpFiction' AND Movie.Id=MovieDirctor.MovieId}}(\text{Movie} \times \text{MovieDirector})) \times \text{Director}$$

=

$$\sigma_{\text{Movie.Title='PulpFiction' AND Movie.Id=MovieDirctor.MovieId AND Director.Id=MovieDirctor.DirectorId}}(\text{Movie} \times \text{MovieDirector} \times \text{Director})$$

Relational Model × Algebra

Who acted in 'Pulp Fiction'?

Relational Model × Algebra

Who acted in 'Pulp Fiction'?

$$\pi_{\text{Actor.FirstName, Actor.LastName}}\left(\sigma_{\text{Movie.Title='PulpFiction' AND Movie.Id=StarIn.MovieId AND Actor.Id=StarIn.ActorId}}(\text{Movie} \times \text{StarIn} \times \text{Director})\right)$$

Algebra × Cartesian Product (×)

×, product, is used to pair attributes (columns) and tuples (rows) of two relations

$$A = R1 \times R2$$

P.S.

I) Product is the most important operator in relational model

II) Product is the most common operator in relational model

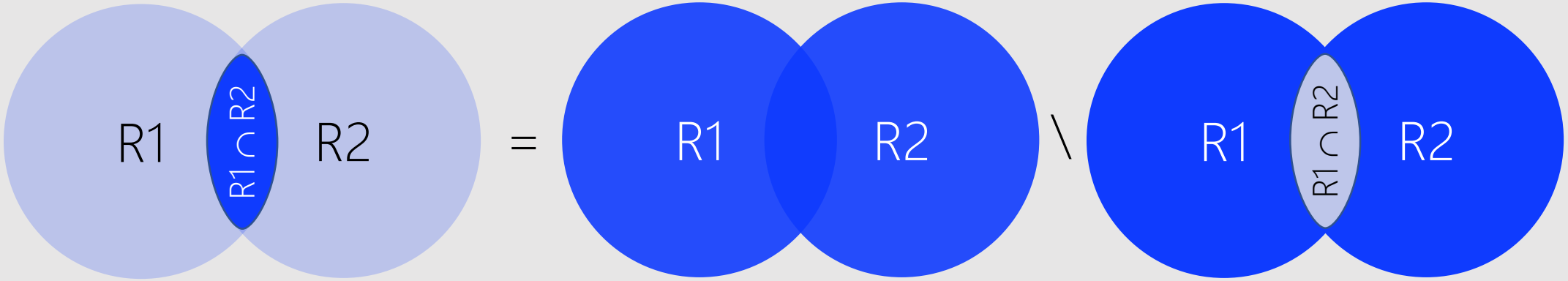
III) Product is the most expensive operator in relational model

Algebra × Complete Set of Operators 100

$\pi(R)$	Project
$\sigma(R)$	Select
$\rho(R)$	Rename
$R1 \cup R2$	Union
$R1 \setminus R2$	Set Difference
$R1 \times R2$	Cartesian Product

Any other relational algebra expression can be expressed by a combination of these operations.

Algebra × Complete Set of Operators 101



$$R1 \cap R2 = (R1 \cup R2) \setminus ((R1 \setminus R2) \cup (R2 \setminus R1))$$

Algebra × Complete Set of Operators 101

$$R1 \cap R2 = R1 \setminus (R1 \setminus R2)$$

Algebra × θ-Join

$\bowtie_\theta$, θ-join, is product (×) of relations followed by selection (σ)

$$R1 \bowtie_\theta R2 = \sigma_\theta (R1 \times R2)$$

Algebra × θ-Join

Movies and their directors?

$$\sigma_{\text{Genre.Id=GenreId}}(\sigma_{\text{Movie.Id=MovieId}}(\text{Movie}\times\text{MovieGenre}))\times\text{Genre})$$

Algebra × θ-Join

Movies and their directors?

$$\sigma_{\text{Genre.Id=GenreId}}(\sigma_{\text{Movie.Id=MovieId}}(\text{Movie}\times\text{MovieGenre}))\times\text{Genre})$$

$$\text{Movie} \bowtie_{\text{Movie.Id=MovieId}} \text{MovieGenre}$$

Algebra × θ-Join

Movies and their directors?

$$\sigma_{\text{Genre.Id=GenreId}}(\sigma_{\text{Movie.Id=MovieId}}(\text{Movie}\times\text{MovieGenre}))\times\text{Genre})$$

$$(\text{Movie} \bowtie_{\text{Movie.Id=MovieId}} \text{MovieGenre}) \bowtie_{\text{Genre.Id=GenreId}} \text{Genre}$$

Algebra × θ-Join

⋈, natural join, is product (×) of relations followed by selection (σ)

$$R1 \bowtie_{\theta} R2 = \sigma_{\theta}(R1 \times R2)$$

Commutative?

Associative?

Algebra × θ-Join

Movies and their directors?

$$\sigma_{\text{Movie.Id=MovieId AND Genre.Id=GenreId}}(\text{Movie} \times \text{MovieGenre} \times \text{Genre})$$
$$\sigma_{\text{Movie.Id=MovieId AND Genre.Id=GenreId}}(\text{Movie} \times \text{Genre} \times \text{MovieGenre})$$
$$\sigma_{\text{Movie.Id=MovieId AND Genre.Id=GenreId}}(\text{Genre} \times \text{Movie} \times \text{MovieGenre})$$
$$\sigma_{\text{Genre.Id=GenreId AND Movie.Id=MovieId}}(\text{Movie} \times \text{MovieGenre} \times \text{Genre})$$

Commutative and Associative law for AND and Product (×)

Algebra × θ-Join

Movies and their directors?

$$\sigma_{\text{Movie.Id=MovieId AND Genre.Id=GenreId}}(\text{Movie} \times \text{MovieGenre} \times \text{Genre})$$

$$(\text{Movie} \bowtie_{\text{Movie.Id=MovieId}} \text{MovieGenre}) \bowtie_{\text{Genre.Id=GenreId}} \text{Genre}$$
$$(\text{MovieGenre} \bowtie_{\text{Movie.Id=MovieId}} \text{Movie}) \bowtie_{\text{Genre.Id=GenreId}} \text{Genre}$$

$$\text{MovieGenre} \bowtie_{\text{Movie.Id=MovieId}} (\text{Movie} \bowtie_{\text{Genre.Id=GenreId}} \text{Genre})$$
$$(\text{Movie} \bowtie_{\text{Genre.Id=GenreId}} \text{MovieGenre}) \bowtie_{\text{Movie.Id=MovieId}} \text{Genre}$$

Algebra × θ-Join

$\bowtie$, natural join, is product (×) of relations followed by selection (σ)

$$R1 \bowtie_\theta R2 = \sigma_\theta (R1 \times R2)$$

Commutative: $R1 \bowtie_\theta R2 = R2 \bowtie_\theta R1$

Associative: $(R1 \bowtie_\theta R2) \bowtie_{\theta'} R3 \neq R1 \bowtie_\theta (R2 \bowtie_{\theta'} R3)$ (Why?)

Algebra × Natural Join

⋈, *natural* join, is θ-Join equating all shared attributes (same name)

$$R1 \bowtie R2 = R1 \bowtie_{\theta} R2 = \sigma_{\theta} (R1 \times R2)$$

where θ: R1.a = R2.a AND R1.b = R2.b AND … AND R1.z = R2.z

Algebra × Natural Join

Director						
Id	FirstName	LastName	DateOfBirth	PlaceOfBirth	BestMovieId	MovieCount
1	*Stanley*	*Kubrick*	*Jul. 26, 1928*	*USA*	*1*	*13*
2	*Alfred*	*Hitchcock*	*Aug. 13, 1899*	*England*	*203*	*47*
3	*Clint*	*Eastwood*	*May 31, 1930*	*USA*	*803*	*35*

Movie			
Id	Title	Language	RunningTime
1	*2001: A Space Odyssey*	*English*	*142*
2	*Rosemary's Baby*	*English*	*136*

What are directors' best movie name?

Algebra × Natural Join

A1		
FirstName	LastName	~~BestMovieId~~ → Id
Stanley	*Kubrick*	*1*
Alfred	*Hitchcock*	*203*
Clint	*Eastwood*	*803*

Movie			
Id	Title	Language	RunningTime
1	*2001: A Space Odyssey*	*English*	*142*
2	*Rosemary's Baby*	*English*	*136*

What are directors' best movie name?

$$A1 = \rho_{Movie(Id/BestMovieId)} (\pi_{FirstName,\ LastName,\ BestMovieId} (Movie))$$

Algebra × Natural Join

A						
FirstName	LastName	~~BestMovieId~~ → Id	Id	Title	Language	RunningTime
Stanley	*Kubrick*	*1*	*1*	*2001: A Space Odyssey*	*English*	*142*
Alfred	*Hitchcock*	*203*				
Clint	*Eastwood*	*803*				
			2	*Rosemary's Baby*	*English*	*136*

What are directors' best movie name?

$$A = A1 \bowtie Movie$$
$$= A1 \bowtie_{A1.Id=Movie.Id} Movie$$
$$= \sigma_{A1.Id=Movie.Id} (A1 \times Movie)$$

Algebra × Left Outer Join (⟕)

A						
FirstName	LastName	~~BestMovieId~~ → Id	Id	Title	Language	RunningTime
Stanley	*Kubrick*	*1*	*1*	*2001: A Space Odyssey*	*English*	*142*
Alfred	*Hitchcock*	*203*	NULL	NULL	NULL	NULL
Clint	*Eastwood*	*803*	NULL	NULL	NULL	NULL
			2	*Rosemary's Baby*	*English*	*136*

What are directors' best movie name if any?

$$A = A1 ⟕ Movie$$
$$= (A1 ⋈_{A1.Id=Movie.Id} Movie) \quad ∪ (?)$$
$$= (\sigma_{A1.Id=Movie.Id} (A1 \times Movie)) \quad ∪ (?)$$

Algebra × Right Outer Join (⋈)

A						
FirstName	LastName	~~BestMovieId~~ → Id	Id	Title	Language	RunningTime
Stanley	*Kubrick*	*1*	*1*	*2001: A Space Odyssey*	*English*	*142*
Alfred	*Hitchcock*	*203*				
Clint	*Eastwood*	*803*				
NULL	NULL	NULL	*2*	*Rosemary's Baby*	*English*	*136*

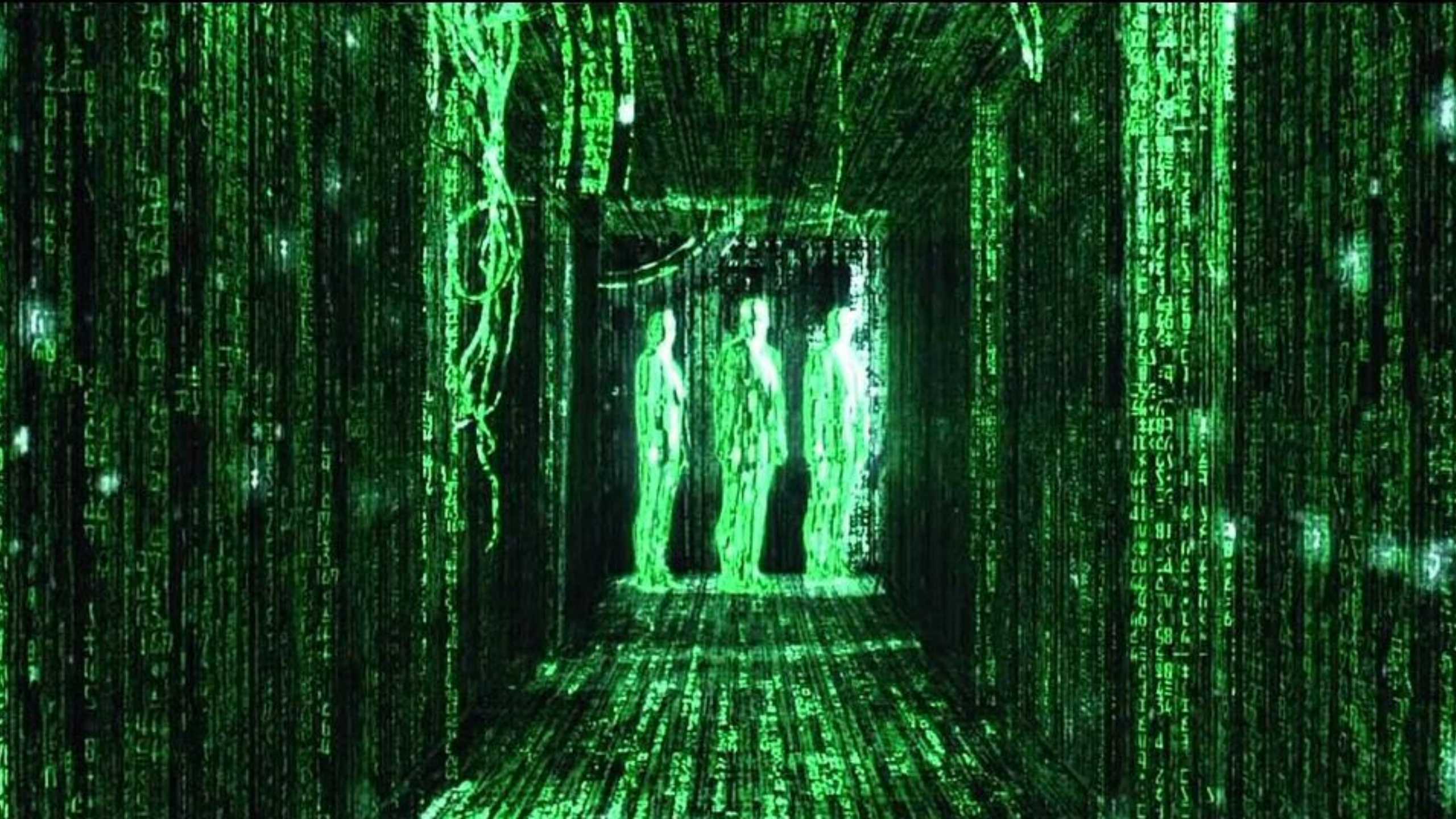
List all movies and identify whether each one is the best of its director?

$$A = A1 ⟕ Movie$$
$$= (A1 ⋈_{A1.Id=Movie.Id} Movie) \quad ∪ (?)$$
$$= (σ_{A1.Id=Movie.Id} (A1 × Movie)) \quad ∪ (?)$$

Algebra × Full Outer Join (⟗)

A						
FirstName	LastName	~~BestMovieId~~ → Id	Id	Title	Language	RunningTime
Stanley	*Kubrick*	*1*	*1*	*2001: A Space Odyssey*	*English*	*142*
Alfred	*Hitchcock*	*203*	NULL	NULL	NULL	NULL
Clint	*Eastwood*	*803*	NULL	NULL	NULL	NULL
NULL	NULL	NULL	*2*	*Rosemary's Baby*	*English*	*136*

$$A = A1 ⟗ Movie$$
$$= A1 ⋈ Movie \cup (A1 ⟗ Movie)$$



Algebra × More

Relational algebra has more complex operators such as:

R1 ▷ R2 Antijoin
R1 / R2 Division

→ *https://en.wikipedia.org/wiki/Relational_algebra*

Algebra × More × Division (/) 111

/, division, is used to find tuples (rows) in R1 which matched with ALL tuples (rows) in R2 :

A = R1 / R2

R1 and R2 are relations
A is a relation with all attributes in R1

Algebra × More × Division (/)

What movie(s) belong to ALL genres?

A1							
Id	Title	Language	RunningTime	MovieId	GenreId	Id	Title
1	2001: A Space Odyssey	English	142	1	1	1	Sci-fi
1	2001: A Space Odyssey	English	142	1	3	3	Adventure
2	Rosemary's Baby	English	136	2	6	6	Drama
2	Rosemary's Baby	English	136	2	7	7	Horror
2	Rosemary's Baby	English	136	2	2	2	Action
2	Rosemary's Baby	English	136	2	1	1	Sci-fi
2	Rosemary's Baby	English	136	2	3	3	Adventure
2	Rosemary's Baby	English	136	2	4	4	Comedy
2	Rosemary's Baby	English	136	2	5	5	Crime

Genre	
Id	Title
1	Sci-fi
2	Action
3	Adventure
4	Comedy
5	Crime
6	Drama
7	Horror

$$A1 = \sigma_{\text{Movie.Id=MovieId AND Genre.Id=GenreId}} (\text{Movie} \times \text{MovieGenre} \times \text{Genre})$$

Algebra × More × Division (/)

What movie(s) belong to ALL genres?

A2		
MovieTitle	Id	Title
2001: A Space Odyssey	*1*	*Sci-fi*
2001: A Space Odyssey	*3*	*Adventure*
Rosemary's Baby	*6*	*Drama*
Rosemary's Baby	*7*	*Horror*
Rosemary's Baby	*2*	*Action*
Rosemary's Baby	*1*	*Sci-fi*
Rosemary's Baby	*3*	*Adventure*
Rosemary's Baby	*4*	*Comedy*
Rosemary's Baby	*5*	*Crime*

Genre	
Id	Title
1	*Sci-fi*
2	*Action*
3	*Adventure*
4	*Comedy*
5	*Crime*
6	*Drama*
7	*Horror*

$$A2 = \rho_{A1(MovieTitle/Movie.Title)}(\pi_{Movie.Title,\ Genre.Id,\ Genre.Title}\ (A1)$$

Algebra × More × Division (/)

What movie(s) belong to ALL genres?

A2		
MovieTitle	Id	Title
2001: A Space Odyssey	*1*	*Sci-fi*
2001: A Space Odyssey	*3*	*Adventure*
Rosemary's Baby	*6*	*Drama*
Rosemary's Baby	*7*	*Horror*
Rosemary's Baby	*2*	*Action*
Rosemary's Baby	*1*	*Sci-fi*
Rosemary's Baby	*3*	*Adventure*
Rosemary's Baby	*4*	*Comedy*
Rosemary's Baby	*5*	*Crime*

A = A2 / Genre

Genre	
Id	Title
1	*Sci-fi*
2	*Action*
3	*Adventure*
4	*Comedy*
5	*Crime*
6	*Drama*
7	*Horror*

Algebra × More × Division (/)

What movie(s) belong to ALL genres?

A2		
MovieTitle	Id	Title
2001: A Space Odyssey	*1*	*Sci-fi*
2001: A Space Odyssey	*3*	*Adventure*
Rosemary's Baby	*6*	*Drama*
Rosemary's Baby	*7*	*Horror*
Rosemary's Baby	*2*	*Action*
Rosemary's Baby	*1*	*Sci-fi*
Rosemary's Baby	*3*	*Adventure*
Rosemary's Baby	*4*	*Comedy*
Rosemary's Baby	*5*	*Crime*

$$A = A2 / Genre$$
$$= A2 \{\pi, \sigma, \rho, \cup, \setminus, \times\} Genre$$

Genre	
Id	Title
1	*Sci-fi*
2	*Action*
3	*Adventure*
4	*Comedy*
5	*Crime*
6	*Drama*
7	*Horror*

Algebra × More × Division (/) 116

Has been a movie won ALL Oscar awards?

Algebra × Extensions

Relational algebra accepts some extensions to support SQL from physical level such as:

Outerjoin
Aggregation Functions (SUM, AVG, MAX, MIN)
Grouping
Sorting

We will cover them in more details when explaining SQL.