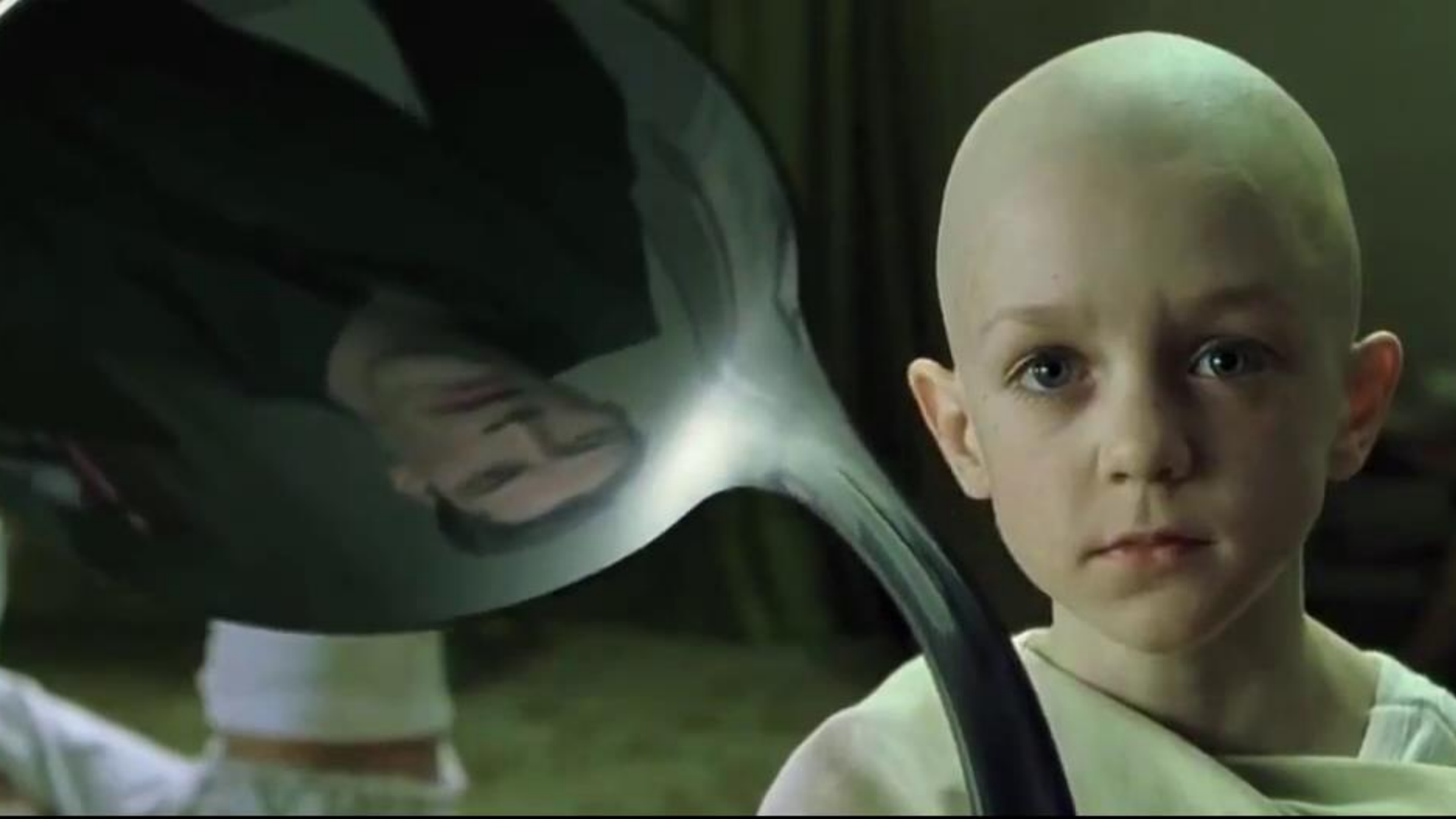
Final Exam (Tentative)

Date Time	Location
Saturday, 10-Dec, 7:00 PM	Odette Building (OB) 104, 112

Last Week × Q4Me

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



Today

Data Modeling
in
RDBMS

Real World Entity

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

Conceptual Level | Logical Level | Relational Model

Conceptual Level | Logical Level | Physical Level | SQL

Conceptual Level | Logical Level | Computable Entity

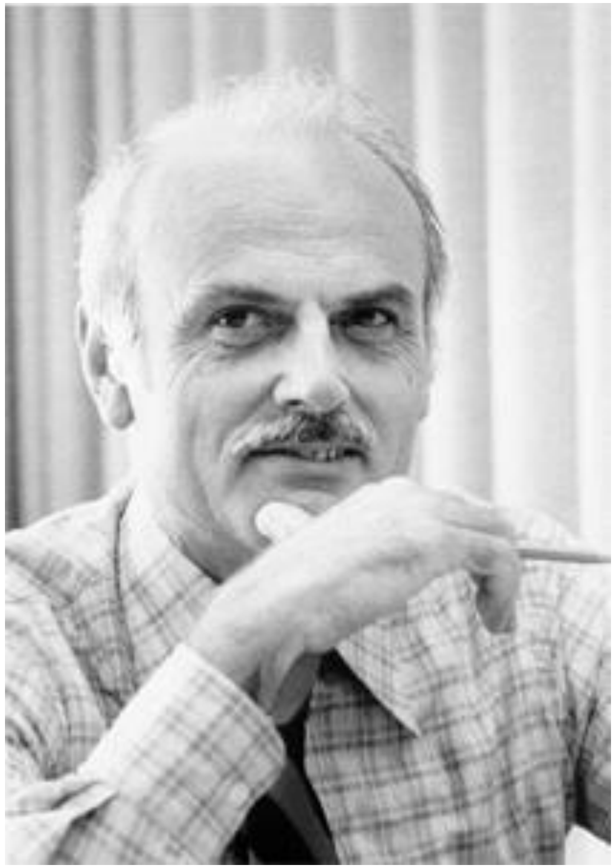
Relational

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 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'?

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

Relational Model × Algebra

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

Unary Operation	Binary Operation (Set Theory)
$\pi(R)$: Project	$R1 \cup R2$: Union
$\sigma(R)$: Select	$R1 \cap R2$: Intersection
$\rho(R)$: Rename	$R1 \setminus R2$: Set Difference
	$R1 \times R2$: Cartesian Product

Algebra × Project (π)

π, pi, is used to select a subset of <u>attributes</u> (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 <u>relation</u> including all tuples in R with only attributes in list

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.

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

T

Horizontal Filtering

R is a relation

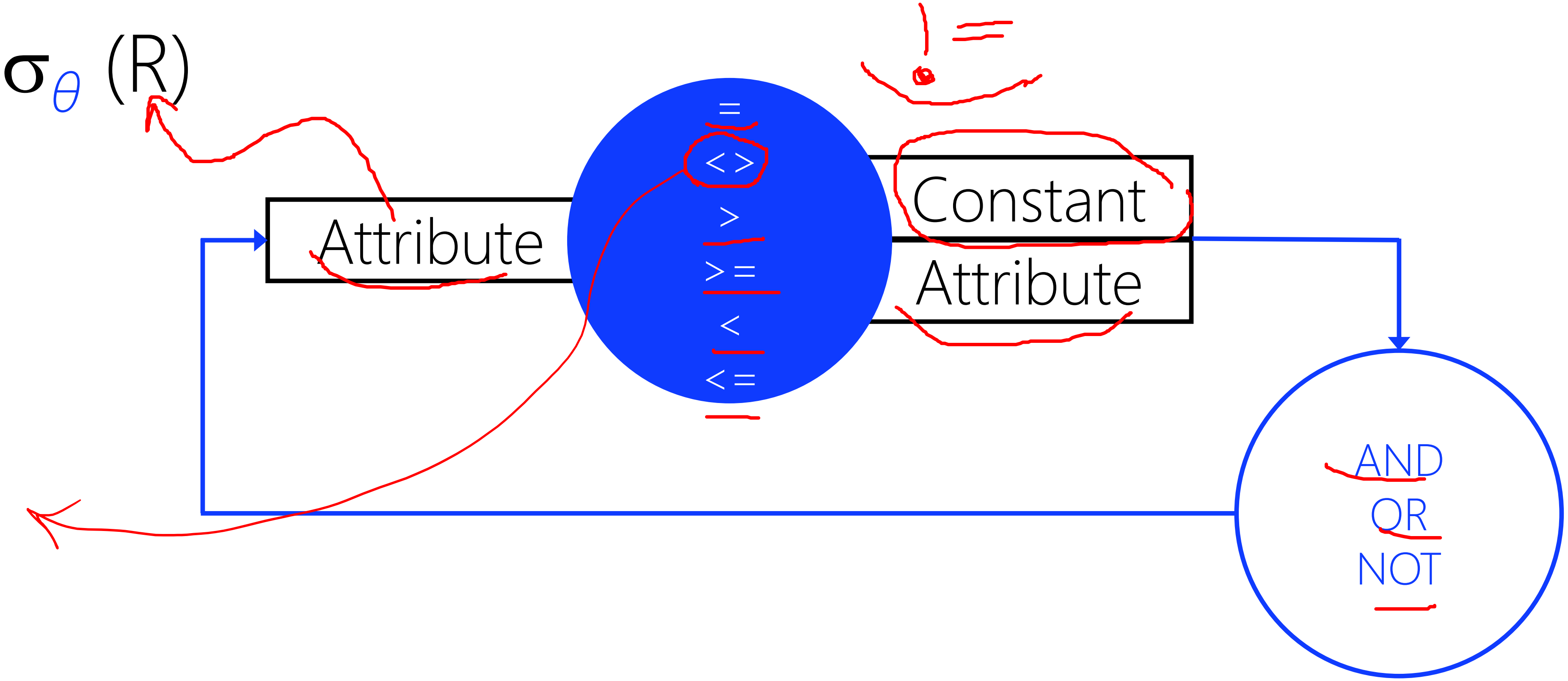
θ is a Boolean expression on the attributes of R

A is a relation including tuples that make θ true

Algebra × Selection ($\boldsymbol{\sigma}$)

θ can be made up of number of Boolean clauses

$$A = \boldsymbol{\sigma}_{\theta}\ (R)$$



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)

$(A \cdot B)' = A' + B'$

Algebra × Selection (σ)

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

Commutative Law

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 × Rename (ρ)

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

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

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
…

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

$\pi(R)$: Project

$\sigma(R)$: Select

$\rho(R)$: Rename

Binary Operation (Set Theory)

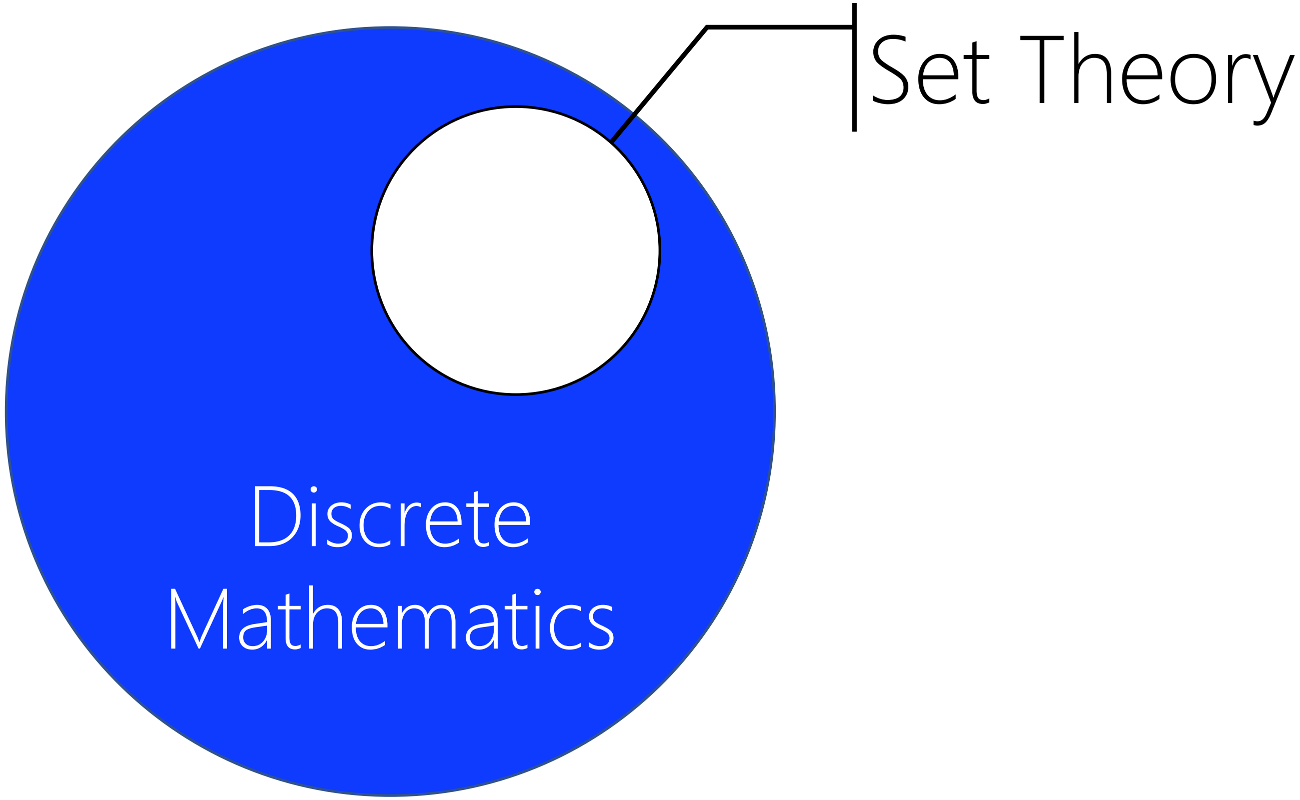
$R1 \cup R2$: Union

$R1 \cap R2$: Intersection

$R1 \setminus R2$: Set Difference

$R1 \times R2$: Cartesian Product

Algebra × Binary Operations



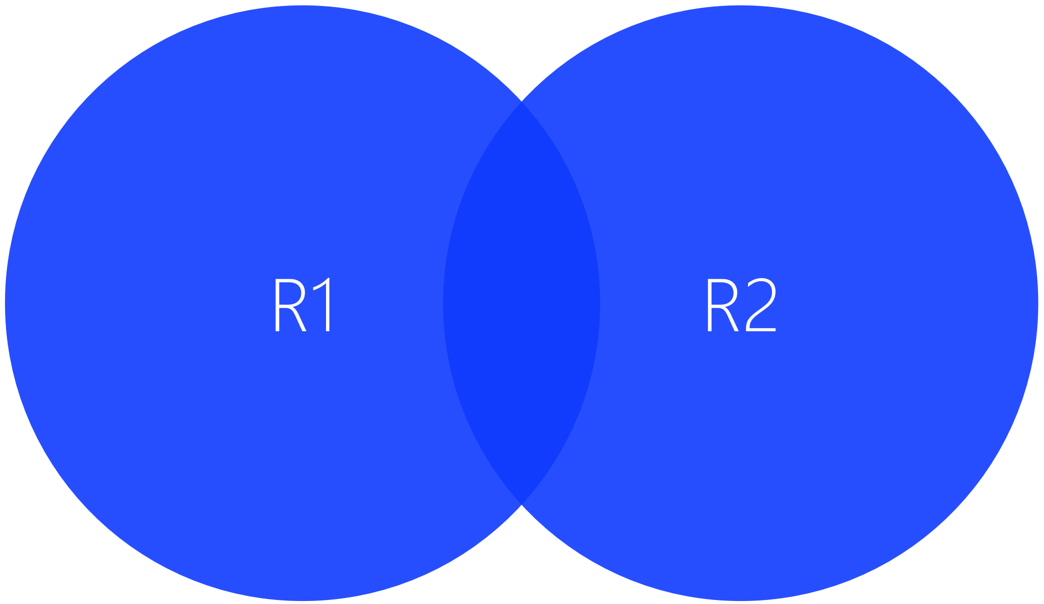
Algebra × Union (∪)

∪, union, is used to include tuples exist in <u>either</u> relations

$$A = R1 \cup R2$$

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

<u>No duplicate</u>!



Venn Diagram

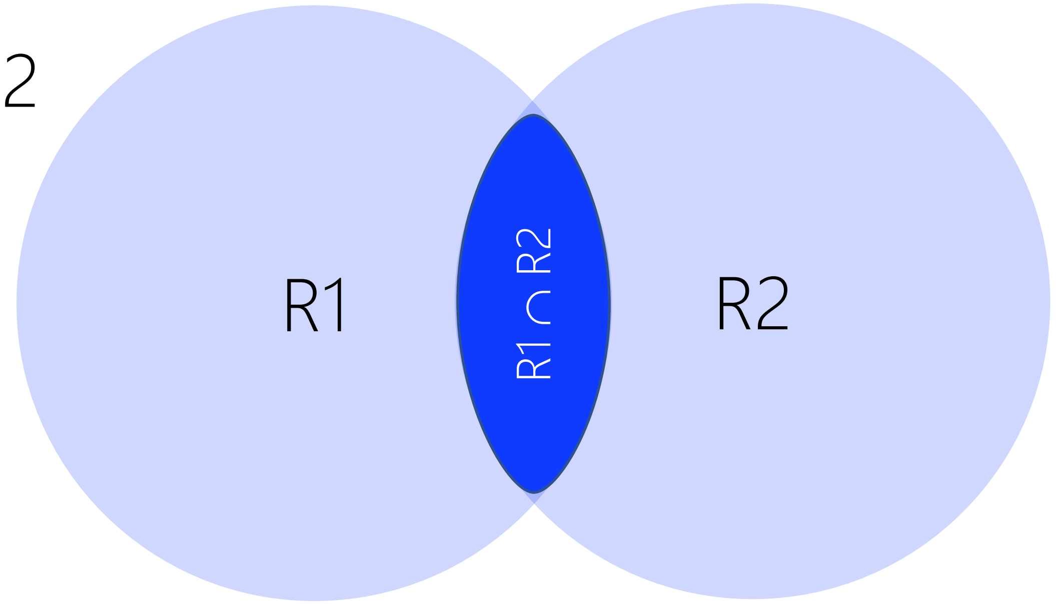
Algebra × Intersection (∩)

∩, 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



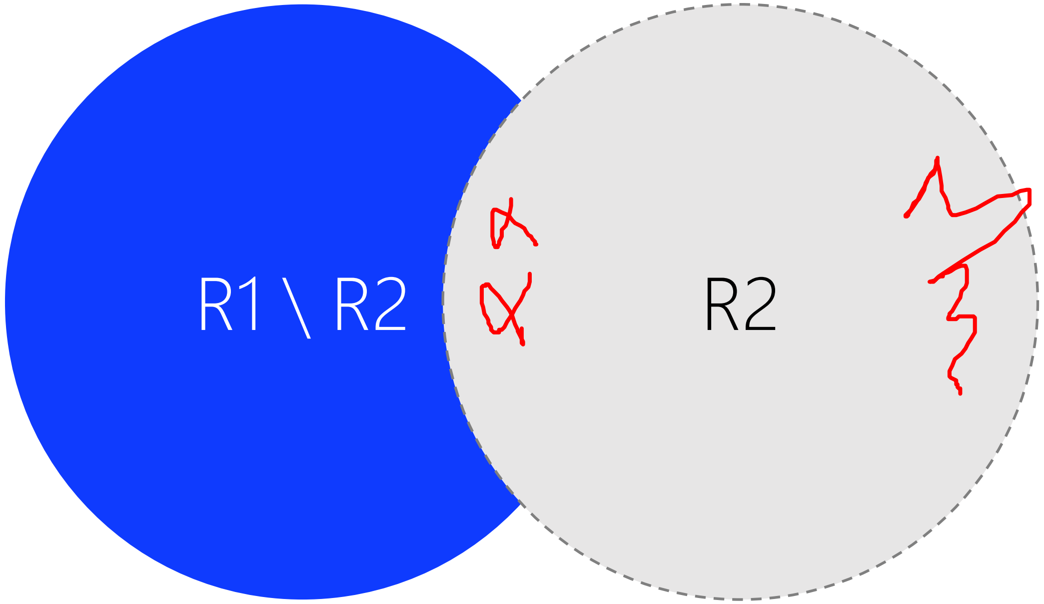
Algebra × Set Difference (\ )

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

R1(a)={(2)}
R2(b)

$$A = R1 \setminus R2$$

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



✗ Commutative Law

R1 \ R2 ≠ R2 \ R1

$2-(3-5) \neq (2-3)-5$

✗ Associative Law

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

Algebra × Set Compatibility

$\pi_{a,b}(R_1)$ $\pi_{c,d}(R_2)$

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

iii) Semantics 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

$\pi(R)$: Project

$\sigma(R)$: Select

$\rho(R)$: Rename

Binary Operation (Set Theory)

$R1 \cup R2$: Union

$R1 \cap R2$: Intersection

$R1 \setminus R2$: Set Difference

$R1 \times R2$: Cartesian Product

Algebra × Cartesian Product (×)

×, product, is used to pair two relations

$$A = R1 \times R2$$

Pairing

$R1(a_1, a_2, \ldots, a_n)$ 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

R1

a_1	a_2	…	a_n
A1	A2	A3	A4
B1	B2	B3	B4

R2

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

R2 × R1

Algebra × Cartesian Product (×)

×, product, is used to pair two relations

$$A = R1 \times R2$$

\# attributes in A, #attributes in R1 + #attributes in R2

\# 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 two relations

$$A = R1 \times R2$$

R1 and R2 do <u>not</u> have to be set compatible

Any two relations can be paired!
Any two relations can be paired!
Any two relations can be paired!
Any two relations can be paired!

Algebra × Cartesian Product (×)

×, product, is used to pair two relations

$$A = R1 \times R2$$

R1 and R2 do <u>not</u> 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 has Genre

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 three 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*
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*

$\sigma_{\text{Id=MovieId}}$ AND (Movie × MovieGenre)

Title = ' '

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*

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

Id = 1

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

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

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	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

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

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

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

Algebra × Cartesian Product (×)

×, product, is used to pair two relations

$$A = R1 \times R2$$

R1 and R2 do <u>not</u> 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*

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

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*

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

Ambiguous name reference! Solution?

., dot, namespace operator

Algebra × Cartesian Product (×)

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

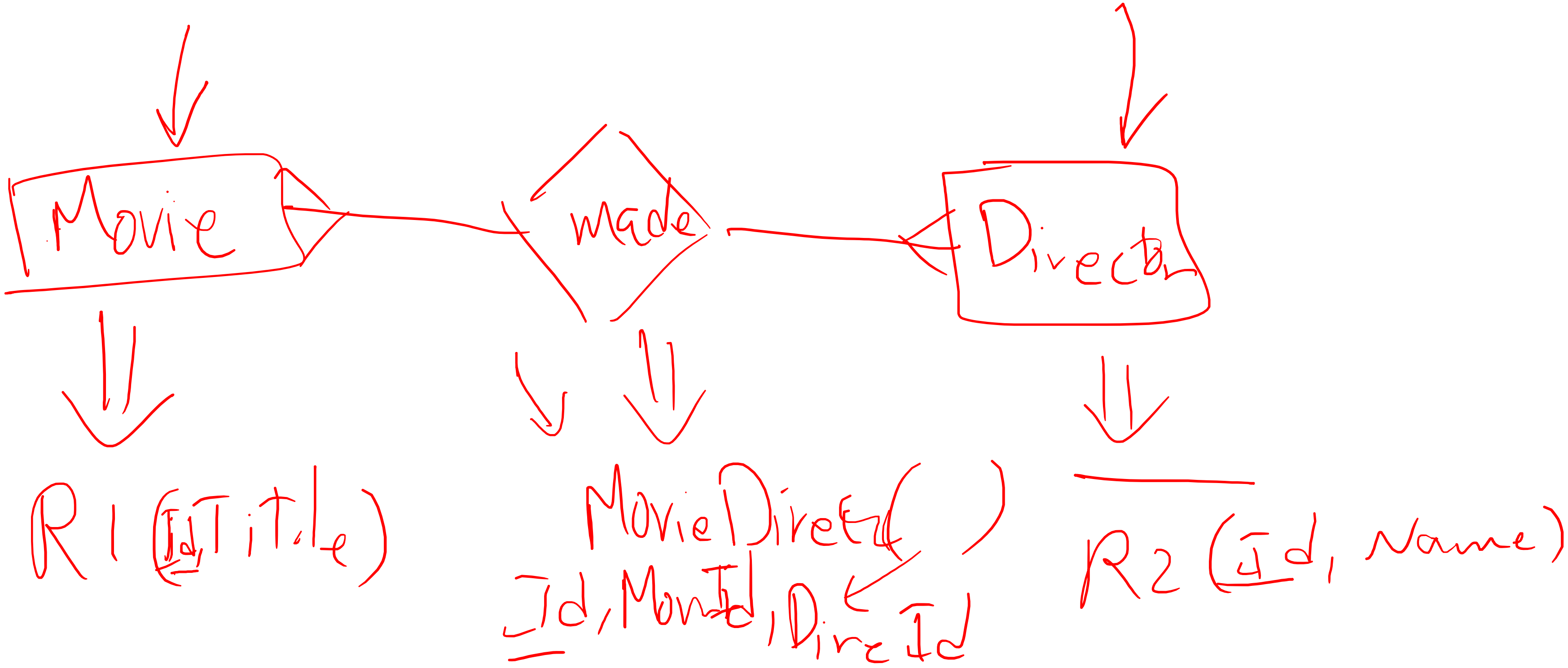
$$A = \sigma_{\text{Id=GenreId}}(\sigma_{\substack{\text{MId=MovieId AND} \\ \text{Title='2001: A Space Odyssey'}}}(\rho_{\text{R(MId/Id)}}(\text{Movie})) \times \text{MovieGenre}) \times \text{Genre})$$

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'?

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

Relational Model × Algebra

Who made 'Pulp Fiction'?

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

Selection (σ) is commutative

Relational Model × Algebra

Who made 'Pulp Fiction'?

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

$$=$$

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

Relational Model × Algebra

Who acted in 'Pulp Fiction'?

Relational Model × Algebra

Who acted in 'Pulp Fiction'?

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

Algebra × Cartesian Product (×)

×, product, is used to pair 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

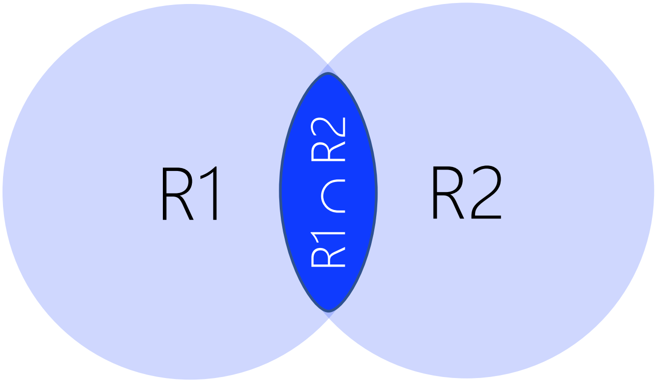
Algebra × Complete Set of Operators

$\pi(R)$	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.

$R1 \cap R2$ No Need for Intersection!

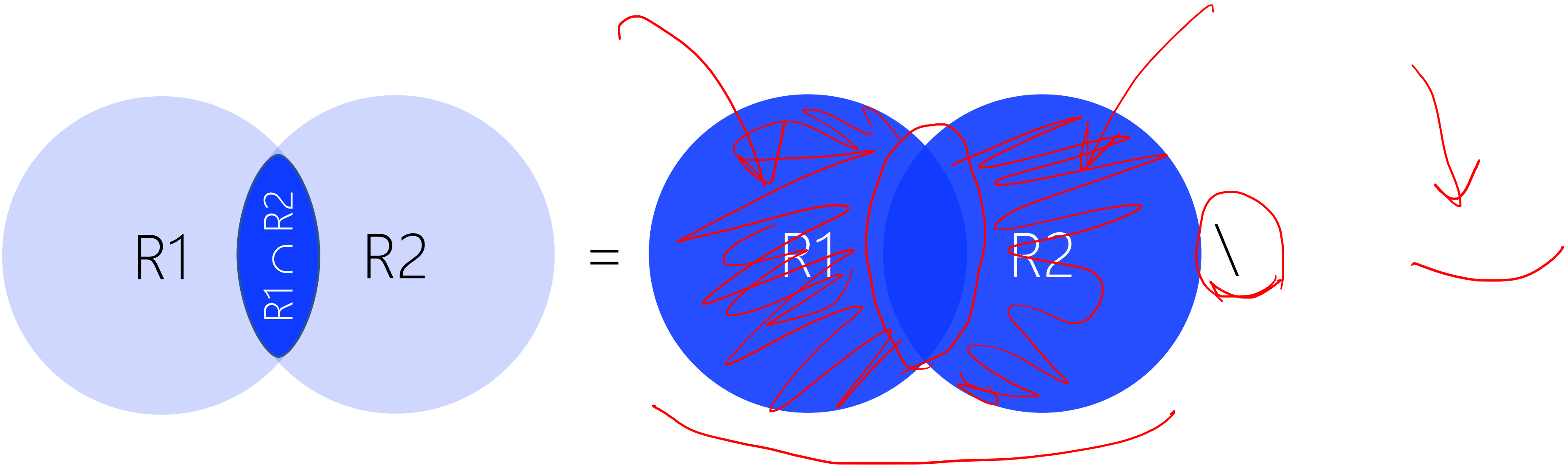
Algebra × Complete Set of Operators



=

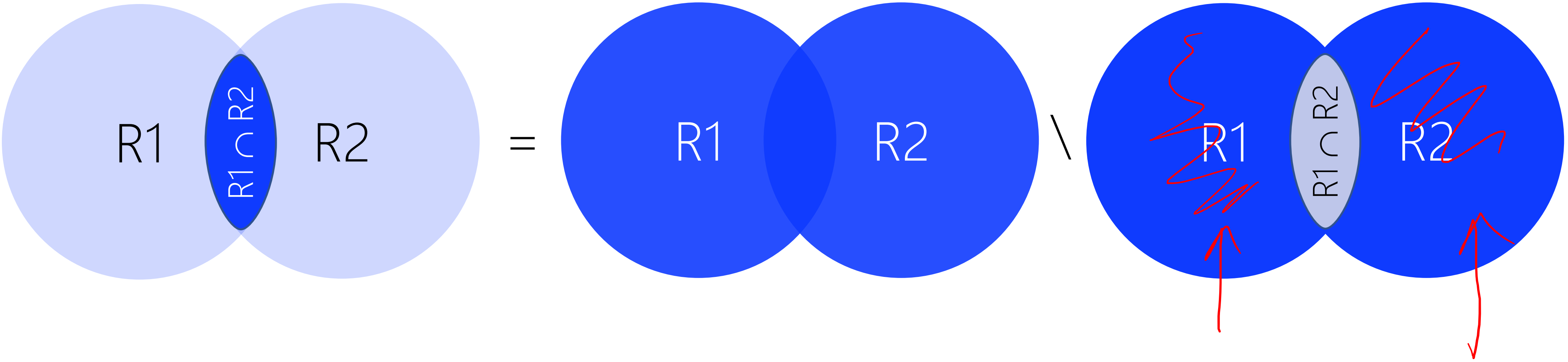
R1 ∩ R2 = What mixture of other operators?

Algebra × Complete Set of Operators



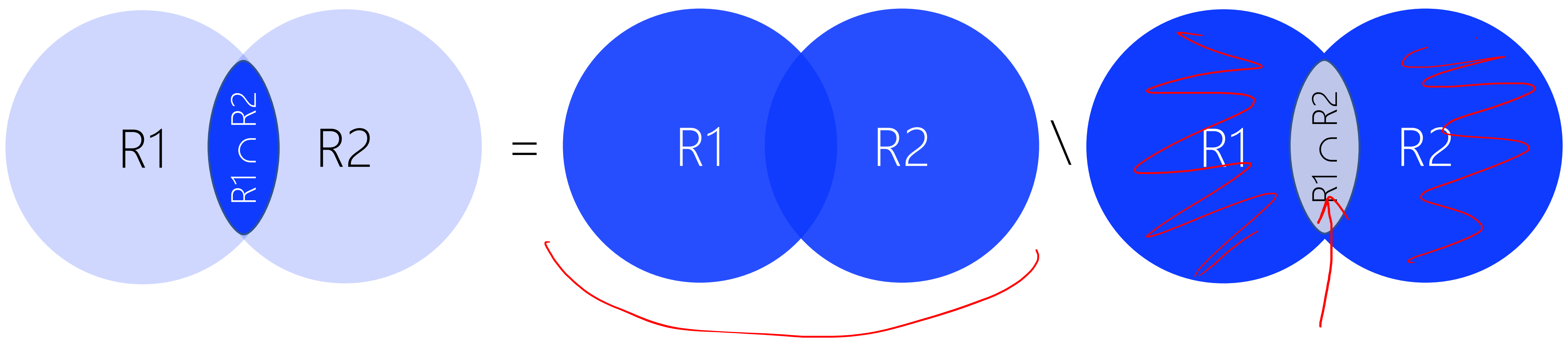
$R1 \cap R2 = (R1 \cup R2) \setminus ?)$

Algebra × Complete Set of Operators



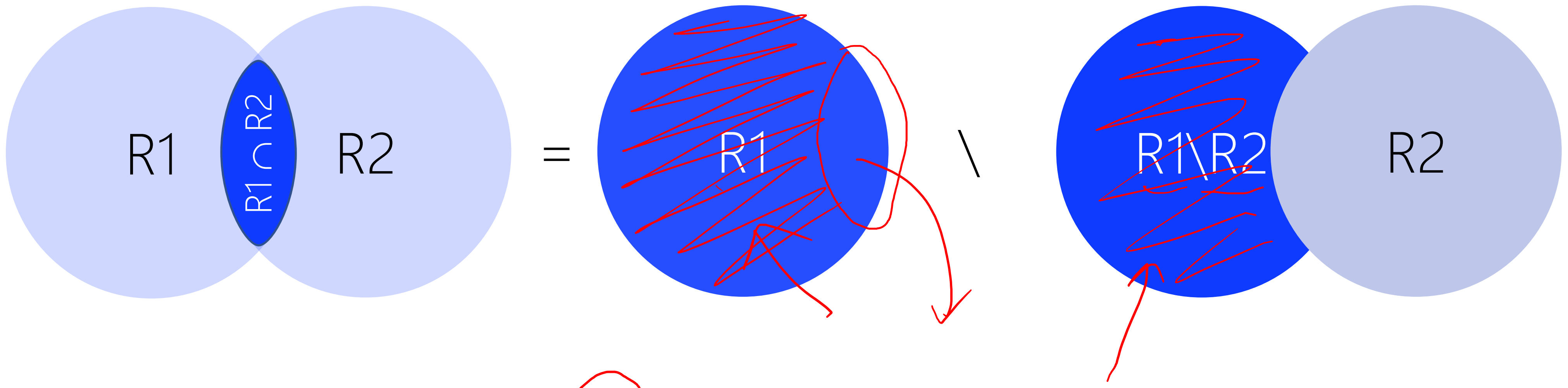
$R1 \cap R2 = (R1 \cup R2) \setminus ?)$

Algebra × Complete Set of Operators



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

Algebra × Complete Set of Operators



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

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 genres?

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

Algebra × θ-Join

Movies and their directors?

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

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

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)

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

$R2 \bowtie_{\theta} R$

✓ Commutative?

✓ Associative?

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 × θ-Join

Movies and their genres?

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

Commutative and Associative law for AND and Product (×)

Algebra × θ-Join

Movies and their genres?

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

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

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

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

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

Algebra × Natural Join

X

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

$$R1 \bowtie R2 = R1 \bowtie_{\theta} R2 = \boldsymbol{\sigma}_{\theta}(R1 \times R2)$$

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

R1 X R2

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?

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

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?

$$A = A1 ⟕ Movie$$
$$= (A1 \bowtie_{A1.Id=Movie.Id} Movie) \cup (?)$$
$$= (\sigma_{A1.Id=Movie.Id}(A1 \times Movie)) \cup (?)$$

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?

$$A = A1 ⟖ \text{Movie}$$

$$= (A1 \bowtie_{A1.Id=Movie.Id} \text{Movie}) \cup (?)$$

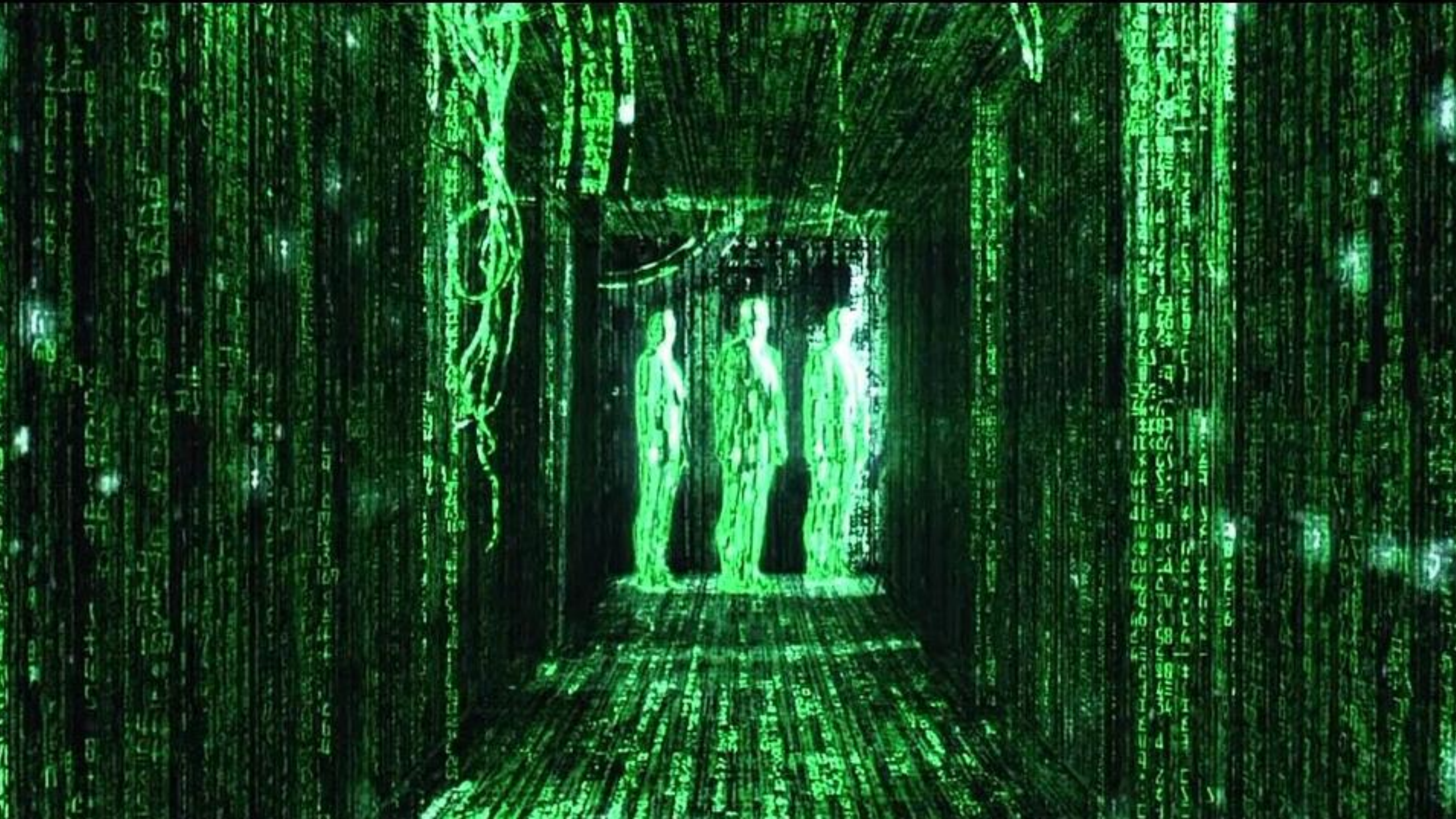
$$= (\sigma_{A1.Id=Movie.Id} (A1 \times \text{Movie})) \cup (?)$$

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 ∪ (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 (/)

/, 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 (/)

Mov ÷ / G

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*

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

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 (/)

Has been a movie won ALL Oscar awards?
Students who should graduate? (passed ALL the courses)
Girls who have collaborated with ALL boys in a class?
Boys who have collaborated with ALL girls in a class?

A (Fully Experienced) = R1 (Reality) / R2 (All Possibilities)

https://en.wikipedia.org/wiki/Relational_algebra#Division_(%C3%B7)

Algebra × More × Division (/)

Has been a movie won ALL Oscar awards?
Students who should graduate? (passed ALL the courses)
Girls who have collaborated with ALL boys in a class?
Boys who have collaborated with ALL girls in a class?

$$\text{Fully Experienced} = \text{Reality} - \underbrace{(\text{All Possibilities} - \text{Reality})}_{\text{Yet to Experience}}$$

https://en.wikipedia.org/wiki/Relational_algebra#Division_(%C3%B7)

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.