CS2102
Database Systems

RELATIONAL ALGEBRA

Formal Relational Query Languages

- Two mathematical Query Languages form the basis for practical languages such as SQL
- Relational Algebra
  - Operational, useful for representing execution plans
- Relational Calculus
  - Declarative: Describe what you want, rather than how to compute

Relational Algebra

- Query is composed of a collection of operators called relational operators

- Operators can be composed to form relational algebra expressions

Relational Algebra Operators

- Unary operators – operate on a single relation
  - Selection σ
  - Projection π
  - Renaming ρ
- Binary operators – combine two or more relations
  - Set operators: Union ∪
    Intersection ∩
    Difference —
    Cartesian product ×
  - Join ⊗
  - Division /

Selection

- Select the tuples of a relation *R* that satisfy condition *c*

$$\sigma_c(R)$$

Selection - Example

- Find employees whose salary is less than 100000
- $\sigma_{\text{Salary<100000}}(\text{Employee})$

$\sigma_{\text{Salary<100000}}($

Employee

Name	Salary	Emp_No
Clark	150000	1006
Gates	5000000	1005
Jones	50000	1001
Peters	45000	1002
Phillips	25000	1004
Rowe	35000	1003
Warnock	500000	1007

$) =$

Name	Salary	Emp_No
Jones	50000	1001
Peters	45000	1002
Rowe	35000	1003
Phillips	25000	1004

Selection - Example

- Find movies made after 1997
- $\sigma_{\text{myear}>1997}(\text{Movies})$

Movies

title	director	myear	rating
Fargo	Coen	1996	8.2
Raising Arizona	Coen	1987	7.6
Spiderman	Raimi	2002	7.4
Wonder Boys	Hanson	2000	7.6

$\sigma_{\text{myear}>1997}(\text{Movies})$

title	director	myear	rating
Spiderman	Raimi	2002	7.4
Wonder Boys	Hanson	2000	7.6

Selection Condition

- Selection condition is a boolean combination of terms
- A term is one of the following forms:
  - attribute *op* constant
  - attribute_1 *op* attribute_2
  - $\text{term}_1 \wedge \text{term}_2$
  - $\text{term}_1 \vee \text{term}_2$
  - $\neg\ \text{term}_1$
  - (term_1)
- Operator precedence: ( ), op, $\neg$, $\wedge$, $\vee$

Selection Condition

- Find movies made by Hanson after 1997
- $\sigma_{\text{myear>1997} \wedge \text{director='Hanson'}}$ (Movies)

Movies

title	director	myear	rating
Fargo	Coen	1996	8.2
Raising Arizona	Coen	1987	7.6
Spiderman	Raimi	2002	7.4
Wonder Boys	Hanson	2000	7.6

$\sigma_{\text{myear>1997} \wedge \text{director='Hanson'}}$ **(Movies)**

title	director	myear	rating
Wonder Boys	Hanson	2000	7.6

Projection

- Keep vertical slices of a relation according to a list of attributes L (*a list of columns*) of the relation R

$$\pi_L(\mathbf{R})$$

Projection - Example

- Find all movies and their ratings
- $\pi_{\text{title, rating}}$ (Movies)

Movies

title	director	myear	rating
Fargo	Coen	1996	8.2
Raising Arizona	Coen	1987	7.6
Spiderman	Raimi	2002	7.4
Wonder Boys	Hanson	2000	7.6

$\pi_{\text{title, rating}}$ **(Movies)**

title	rating
Fargo	8.2
Raising Arizona	7.6
Spiderman	7.4
Wonder Boys	7.6

Projection - Example

- Duplicates are removed in the output
- Find all ratings

Movies

title	director	myear	rating
Fargo	Coen	1996	8.2
Raising Arizona	Coen	1987	7.6
Spiderman	Raimi	2002	7.4
Wonder Boys	Hanson	2000	7.6

π_{rating} (Movies)

rating
8.2
7.6
7.4

Renaming

- If attributes or relations have the same name, e.g., when joining a relation with itself, it may be convenient to rename one

$$\rho\ (R'(N_1 \rightarrow N_1', \ldots, N_n \rightarrow N_n'), R)$$

- The new relation R′ has the same instance as R, but its schema has attribute N_i' instead of attribute N_i

Renaming - Example

- ρ (Staff (Name ⟶ Family_Name, Salary ⟶ Gross_salary), Employee)

Employee

Name	Salary	Emp_No
Clark	150000	1006
Gates	5000000	1005
Jones	50000	1001
Peters	45000	1002
Phillips	25000	1004
Rowe	35000	1003
Warnock	500000	1007

Staff

Family_Name	Gross_Salary	Emp_No
Clark	150000	1006
Gates	5000000	1005
Jones	50000	1001
Peters	45000	1002
Phillips	25000	1004
Rowe	35000	1003
Warnock	500000	1007

Set Operations

- Two relations are <u>union compatible</u> if
  - they have the same arity, and
  - the corresponding attributes have same domains
- Union, Intersection, Set-difference operators require relations to be union compatible

Set Operations

- UNION: $R \cup S = \{ t \mid t \in R \text{ or } t \in S\}$
  - Return a relation containing all tuples that occur in R or S (or both)
- INTERSECTION: $R \cap S = \{ t \mid t \in R \text{ and } t \in S\}$
  - Return a relation containing all tuples that occur in both R and S
- SET-DIFFERENCE: $R - S = \{ t \mid t \in R \text{ and } t \notin S\}$
  - Return a relation containing all tuples that occur in R but not in S
- Schema of the result of *R op S* is identical to the schema of R

UNION - Example

- Find all actors & directors

Actors

actor	ayear
Cage	1964
Hanks	1956
Maguire	1975
McDormand	1957

$\pi_{actor}(Actors)$

actor
Cage
Hanks
Maguire
McDormand

$\pi_{actor}(Actors) \cup \pi_{director}(Directors)$

actor
Cage
Hanks
Maguire
McDormand
Coen
Hanson
Raimi

Directors

director	dyear
Coen	1954
Hanson	1945
Raimi	1959

$\pi_{director}(Directors)$

director
Coen
Hanson
Raimi

INTERSECTION - Example

- Find Coen’s movies with McDormand

Acts

actor	title
Cage	Raising Arizona
Maguire	Spiderman
Maguire	Wonder Boys
McDormand	Fargo
McDormand	Raising Arizona
McDormand	Wonder Boys

$$\xrightarrow{\pi_{\text{title}}(\sigma_{\text{actor='McDormand'}}(\text{Acts}))}$$

title
Fargo
Raising Arizona
Wonder Boys

$$\pi_{\text{title}}(\sigma_{\text{actor='McDormand'}}(\text{Acts}))$$

$$\cap$$

$$\pi_{\text{title}}(\sigma_{\text{director='Coen'}}(\text{Movies}))$$

title
Fargo
Raising Arizona

Movies

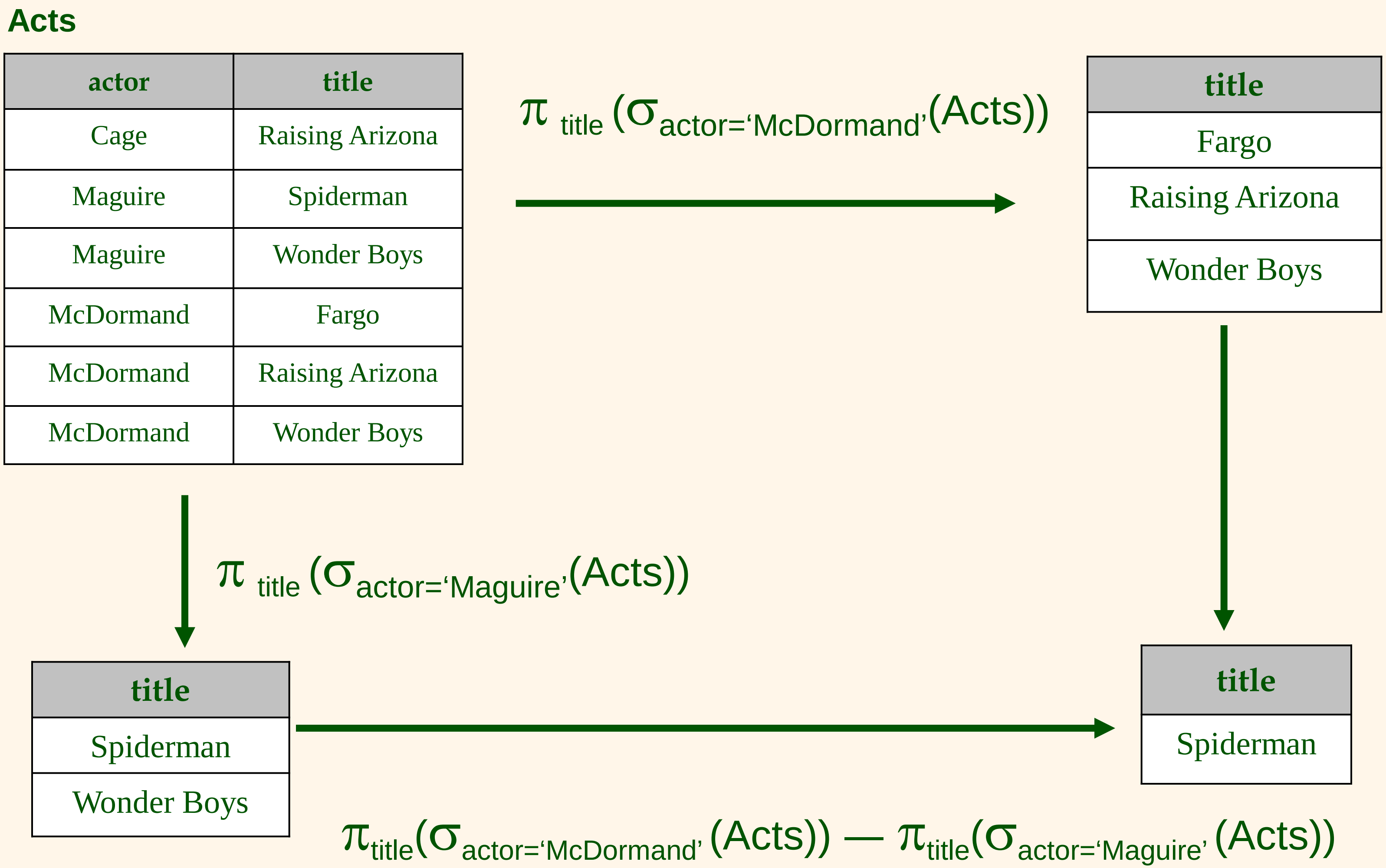
title	director	myear	rating
Fargo	Coen	1996	8.2
Raising Arizona	Coen	1987	7.6
Spiderman	Raimi	2002	7.4
Wonder Boys	Hanson	2000	7.6

$$\xrightarrow{\pi_{\text{title}}(\sigma_{\text{director='Coen'}}(\text{Movies}))}$$

title
Fargo
Raising Arizona

SET-DIFFERENCE - Example

- Find movies with Maguire but not McDormand



Acts

actor	title
Cage	Raising Arizona
Maguire	Spiderman
Maguire	Wonder Boys
McDormand	Fargo
McDormand	Raising Arizona
McDormand	Wonder Boys

$\pi_{\text{title}}(\sigma_{\text{actor='McDormand'}}(\text{Acts}))$

title
Fargo
Raising Arizona
Wonder Boys

$\pi_{\text{title}}(\sigma_{\text{actor='Maguire'}}(\text{Acts}))$

title
Spiderman
Wonder Boys

title
Spiderman

$\pi_{\text{title}}(\sigma_{\text{actor='McDormand'}}(\text{Acts})) — \pi_{\text{title}}(\sigma_{\text{actor='Maguire'}}(\text{Acts}))$

Cartesian Product

- Consider relations R(A, B, C) and S(X, Y)
- Cross-product or Cartesian product

R x S returns a relation with attribute list

(A, B, C, X, Y) defined as follows:

$$R \times S = \{(a, b, c, x, y) \mid (a, b, c) \in R, (x, y) \in S\}$$

Cartesian Product

- Combine two relations $\mathbf{R_1 \times R_2}$
- {a, b} x {1,3} = { (a,1), (a,3), (b,1), (b,3) }
- {(a,1), (a,3)} x {(a,1), (a,3)}
  = {((a,1),(a,1)), ((a,1),(a,3)), ((a,3),(a,1)),((a,3),(a,3))}
  = {(a,1,a,1), (a,1,a,3), (a,3,a,1),(a,3,a,3)}

Cartesian Product - Example

- Can_fly × Plane

Emp_No	Model_No
1001	B727
1001	B747
1001	DC10
1002	A320
1002	A340
1002	B757
1002	DC9
1003	A310
1003	DC9

×

Maker	Model_No
Airbus	A310
Airbus	A320
Airbus	A330
Airbus	A340
Boeing	B727
Boeing	B747
Boeing	B757
MD	DC10
MD	DC9

=

Emp_No	Model_No	Maker	Model_No
1001	B727	Airbus	A310
1001	B727	Airbus	A320
1001	B727	Airbus	A330
1001	B727	Airbus	A340
1001	B727	Boeing	B727
1001	B727	Boeing	B747
1001	B727	Boeing	B757
1001	B727	MD	DC10
1001	B727	MD	DC9
1001	B747	Airbus	A310
1001	B747	Airbus	A320
1001	B747	Airbus	A330
1001	B747	Airbus	A340
1001	B747	Boeing	B727
1001	B747	Boeing	B747
1001	B747	Boeing	B757
1001	B747	MD	DC10
1001	B747	MD	DC9
1001	B727	Airbus	A310
1001	B727	Airbus	A320
…	…	…	…

81 tuples!!!

Join

- Combines cross-product, selection and projection
- Join operator is more useful than the plain cross product
- Three types of join:
  - Condition join
  - Equijoin
  - Natural join

Condition Join $R \otimes_c S$

- Condition join = Cross product followed by selection
- Find (director, actor) pairs where director is younger than actor
- $\pi_{\text{director, actor}}$ (Directors $\otimes_{\text{dyear > ayear}}$ Actors)

Directors

director	dyear
Coen	1954
Hanson	1945
Raimi	1959

Actors

actor	ayear
Cage	1964
Hanks	1956
Maguire	1975
McDormand	1957

Directors $\otimes_{\text{dyear > ayear}}$ Actors

Director	dyear	actor	ayear
Raimi	1959	Hanks	1956
Raimi	1959	McDormand	1957

Director	actor
Raimi	Hanks
Raimi	McDormand

Equijoin

- Combines two relations on a condition composed only of equalities of attributes of the first and second relation
- Projects only one of the redundant attribute (since they are equal)

Equijoin

- Find actors who have acted in some Coen's movie
- $\pi_{\text{actor}}\ (\ \sigma_{\text{director='Coen'}}\ (\text{Acts}\ \otimes_{\text{Acts.title = Movies.title}}\ \text{Movies}\ )$

actor	title	director	myear	rating
Cage	Raising Arizona	Coen	1987	7.6
Maguire	Spiderman	Raimi	2002	7.4
Maguire	Wonder Boys	Hanson	2000	7.6
McDormand	Fargo	Coen	1996	8.2
McDormand	Raising Arizona	Coen	1987	7.6
McDormand	Wonder Boys	Hanson	2000	7.6

actor
Cage
McDormand

Natural Join

- Combines two relations on the equality of the attributes with the same names
- Projects only one of the redundant attributes

- Find actors who have acted in some Coen's movie

$$\pi_{\text{actor}} (\sigma_{\text{director='Coen'}} (\text{Acts} \otimes \text{Movies})$$

- Find the name and year of birth of all actors who were in some Coen's movie

$$\pi_{\text{actor, ayear}} (\sigma_{\text{director='Coen'}} (\text{Movies}) \otimes \text{Acts} \otimes \text{Actors})$$

Natural Join

$$\pi_{\text{actor, ayear}} \left(\sigma_{\text{director='Coen'}} (\text{Movies}) \otimes \text{Acts} \otimes \text{Actors} \right)$$

Movies

title	director	myear	rating
Fargo	Coen	1996	8.2
Raising Arizona	Coen	1987	7.6
Spiderman	Raimi	2002	7.4
Wonder Boys	Hanson	2000	7.6

$\otimes$

Acts

actor	title
Cage	Raising Arizona
Maguire	Spiderman
Maguire	Wonder Boys
McDormand	Fargo
McDormand	Raising Arizona
McDormand	Wonder Boys

$\otimes$

Actors

actor	ayear
Cage	1964
Hanks	1956
Maguire	1975
McDormand	1957

Quiz

- Find actors who have acted in all Coen's movie

Movies

title	director	myear	rating
Fargo	Coen	1996	8.2
Raising Arizona	Coen	1987	7.6
Spiderman	Raimi	2002	7.4
Wonder Boys	Hanson	2000	7.6

Actors

actor	ayear
Cage	1964
Hanks	1956
Maguire	1975
McDormand	1957

Acts

actor	title
Cage	Raising Arizona
Maguire	Spiderman
Maguire	Wonder Boys
McDormand	Fargo
McDormand	Raising Arizona
McDormand	Wonder Boys

Directors

director	dyear
Coen	1954
Hanson	1945
Raimi	1959

Division R / S

- Let R have two fields A and B
- Let S have one field B
- **R / S** contains all A tuples, such that for **every** B tuple in S there is a AB tuple in R

R

A	B
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

/

S

B
p2
p4

=

R / S

A
s1
s4

Division

- Compute all possible combinations of the first column of R and S.
- Then remove those rows that exist in R
- Keep only the first column of the result. These are the ***disqualified*** values
- R / S is the first column of R **except** the disqualified values

R

A	B
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

$\pi_A(R)$

A
s1
s2
s3
s4

S

B
p2
p4

$\pi_A(R) \times S$

A	B
s1	p2
s1	p4
s2	p2
s2	p4
s3	p2
s3	p4
s4	p2
s4	p4

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

A	B
s2	p4
s3	p4

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

A
s1
s4

Division

- Given $R(X_1, X_2, \ldots, X_m, Y_1, Y_2, \ldots, Y_n)$ and $S(Y_1, Y_2, \ldots, Y_n)$
- $R / S = T(X_1, X_2, \ldots, X_m)$ where

$$(x_1, x_2, \ldots, x_m) \in T \Leftrightarrow \{(x_1, x_2, \ldots, x_m)\} \times S \subseteq R$$

- Actors who have acted in all Coen's movie = Acts / CMovies

Acts

actor	title
Cage	Raising Arizona
Maguire	Spiderman
Maguire	Wonder Boys
McDormand	Fargo
McDormand	Raising Arizona
McDormand	Wonder Boys

CMovies

title
Fargo
Raising Arizona

Answer

actor
McDormand

Composability

- The result of an expression is a relation
- $\pi_{\text{Emp\_No, Num}}(\sigma_{\text{Num}>150}(\text{Employee}))$

Employee

Emp_No	Dep_Date	Num
1001	Nov 1	100
1001	Oct 31	100
1002	Nov 1	200

→

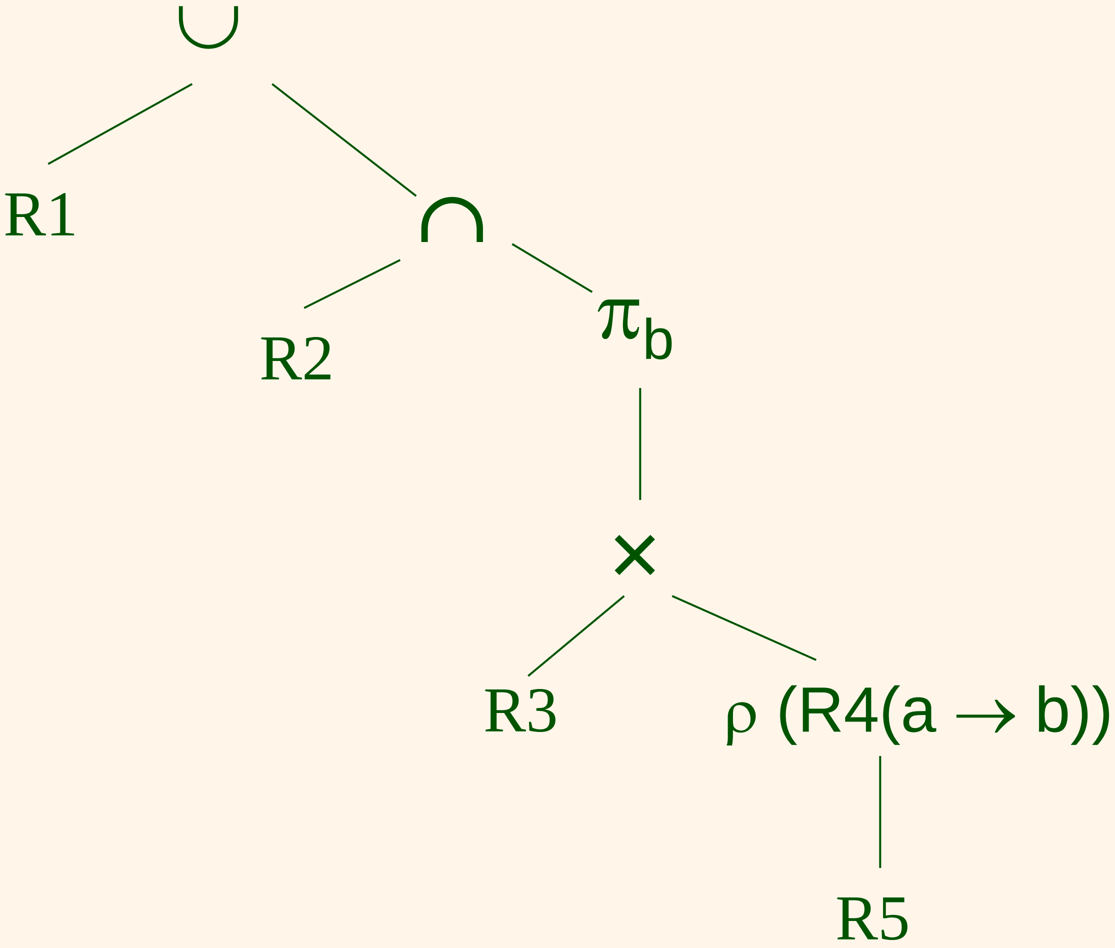
Emp_No	Num
1002	200

- $\sigma_{\text{Num}>150}(\pi_{\text{Emp\_No, Num}}(\text{Employee}))$

Can I always exchange the order of $\boldsymbol{\sigma}$ and $\boldsymbol{\pi}$?

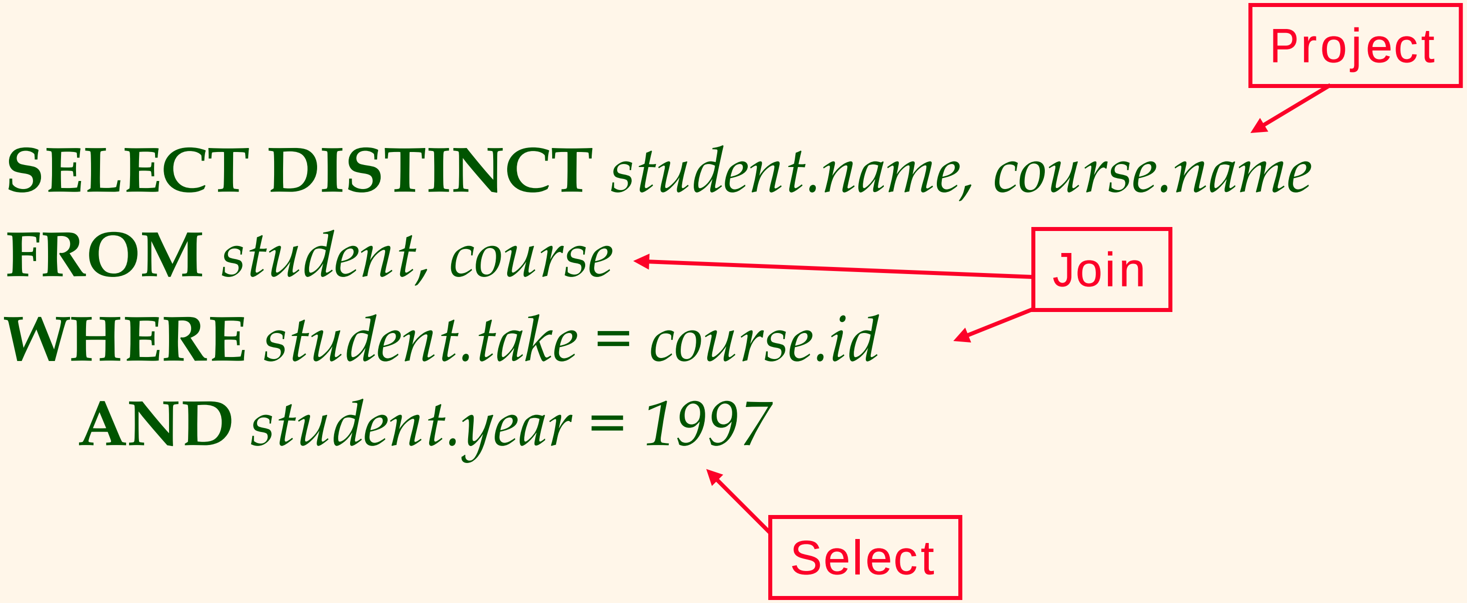
Complex Expression

- $R_1 \cup (R_2 \cap \pi_b(R_3 \times \rho(R_4(a \rightarrow b), R_5)))$



Simple SQL and Algebra queries

- Also called **P**roject **S**elect **J**oin queries (**PSJ**)



SELECT DISTINCT *student.name, course.name*
FROM *student, course*
WHERE *student.take = course.id*
    AND *student.year = 1997*

Simple SQL and Algebra queries - Selection

- $\sigma_{\text{Salary<100000}}(\text{Employee})$

```
SELECT *
FROM Employee
WHERE Salary < 100000
```

Simple SQL and Algebra queries - Projection

- Is $\pi_{\text{Emp\_No, Num}}(\text{Assigned\_To})$ equivalent to

  SELECT Emp_No, Num
  FROM Assigned_To

$\pi_{\textbf{Emp\_No, Num}}($

Emp_No	Dep_Date	Num
1001	Nov 1	100
1001	Oct 31	100
1002	Nov 1	100

$) =$

Emp_No	Num
1001	100
1002	100

- **No.** Relational algebra works with sets (i.e., no duplicates)

  SELECT **DISTINCT** Emp_No, Num
  FROM Assigned_To

Simple SQL and Algebra queries - Renaming

- ρ (Staff (Name → Family_Name, Salary → Gross_salary), Employee)

```
SELECT    Name AS Family_Name,
          Salary AS Gross_salary
FROM  Employee  Staff
```

Simple SQL and Algebra queries – Set Operations

- $Plane_1 \cup Plane_2$

SELECT DISTINCT *
FROM Plane1
UNION
SELECT DISTINCT *
FROM Plane2

Maker	Model_No
Airbus	A310
Airbus	A320
Airbus	A330
Airbus	A340
MD	DC10
MD	DC9

∪

Maker	Model_No
Boeing	B727
Boeing	B747
Boeing	B757
MD	DC10
MD	DC9

↓

Maker	Model_No
Airbus	A310
Airbus	A320
Airbus	A330
Airbus	A340
Boeing	B727
Boeing	B747
Boeing	B757
MD	DC10
MD	DC9

Simple SQL and Algebra queries – Set Operations

- $\text{Plane}_1 \cap \text{Plane}_2$

SELECT DISTINCT *

FROM Plane1

INTERSECT

SELECT DISTINCT *

FROM Plane2

Maker	Model_No
Airbus	A310
Airbus	A320
Airbus	A330
Airbus	A340
MD	DC10
MD	DC9

∩

Maker	Model_No
Boeing	B727
Boeing	B747
Boeing	B757
MD	DC10
MD	DC9

↓

Maker	Model_No
MD	DC10
MD	DC9

Simple SQL and Algebra queries – Set Operations

- $\text{Plane}_1 - \text{Plane}_2$

```
SELECT DISTINCT *
FROM Plane1
EXCEPT
SELECT DISTINCT *
FROM Plane2
```

Maker	Model_No
Airbus	A310
Airbus	A320
Airbus	A330
Airbus	A340
MD	DC10
MD	DC9

—

Maker	Model_No
Boeing	B727
Boeing	B747
Boeing	B757
MD	DC10
MD	DC9

↓

Maker	Model_No
Airbus	A310
Airbus	A320
Airbus	A330
Airbus	A340

Simple SQL and Algebra queries – Cross Product

- Can_fly × Plane

SELECT DISTINCT *
FROM Can_Fly, Plane

Emp_No	Model_No
1001	B727
1001	B747
1001	DC10
1002	A320
1002	A340
1002	B757
1002	DC9
1003	A310
1003	DC9

×

Maker	Model_No
Airbus	A310
Airbus	A320
Airbus	A330
Airbus	A340
Boeing	B727
Boeing	B747
Boeing	B757
MD	DC10
MD	DC9

=

Emp_No	Model_No	Maker	Model_No
1001	B727	Airbus	A310
1001	B727	Airbus	A320
1001	B727	Airbus	A330
1001	B727	Airbus	A340
1001	B727	Boeing	B727
1001	B727	Boeing	B747
1001	B727	Boeing	B757
1001	B727	MD	DC10
1001	B727	MD	DC9
1001	B747	Airbus	A310
1001	B747	Airbus	A320
1001	B747	Airbus	A330
1001	B747	Airbus	A340
1001	B747	Boeing	B727
1001	B747	Boeing	B747
1001	B747	Boeing	B757
1001	B747	MD	DC10
1001	B747	MD	DC9
1001	B727	Airbus	A310
1001	B727	Airbus	A320
…	…	…	…

Simple SQL and Algebra queries - Join

- Can_fly $\otimes_{\text{Can\_fly.Model\_No = Plane.Model\_No}}$ Plane

```
SELECT DISTINCT *
FROM Can_Fly, Plane
WHERE Can_Fly.Model_No = Plane.Model_No
```

Emp_No	Model_No
1001	B727
1001	B747
1001	DC10
1002	A320
1002	A340
1002	B757
1002	DC9
1003	A310
1003	DC9

$\otimes$

Maker	Model_No
Airbus	A310
Airbus	A320
Airbus	A330
Airbus	A340
Boeing	B727
Boeing	B747
Boeing	B757
MD	DC10
MD	DC9

=

Emp_No	Can_Fly.Model_No	Maker	Plane.Model_No
1003	A310	Airbus	A310
1002	A320	Airbus	A320
1002	A340	Airbus	A340
1001	B727	Boeing	B727
1001	B747	Boeing	B747
1002	B757	Boeing	B757
1001	DC10	MD	DC10
1002	DC9	MD	DC9
1003	DC9	MD	DC9

Example 1

- Find the employment numbers of pilots who can fly Airbus planes

$$\pi_{\text{Emp\_No}}\ (\sigma_{\text{Maker} = \text{‘Airbus’}}\ (\text{Can\_Fly} \otimes \text{Plane}\ )\ )$$

Can_Fly

Emp_No	Model_No
1001	B727
1001	B747
1001	DC10
1002	A320
1002	A340
1002	B757
1002	DC9
1003	A310
1003	DC9

Plane

Maker	Model_No
Airbus	A310
Airbus	A320
Airbus	A330
Airbus	A340
Boeing	B727
Boeing	B747
Boeing	B757
MD	DC10
MD	DC9

Emp_No
1002
1003

Example 2

- Find the employment numbers of pilots who can fly Boeing **or** MD planes

$$\pi_{\text{Emp\_No}} \left(\sigma_{\text{Maker='Boeing'} \vee \text{Maker='MD'}} \left(\text{Can\_Fly} \otimes \text{Plane} \right) \right)$$

Can_Fly

Emp_No	Model_No
1001	B727
1001	B747
1001	DC10
1002	A320
1002	A340
1002	B757
1002	DC9
1003	A310
1003	DC9

Plane

Maker	Model_No
Airbus	A310
Airbus	A320
Airbus	A330
Airbus	A340
Boeing	B727
Boeing	B747
Boeing	B757
MD	DC10
MD	DC9

Emp_No
1001
1002
1003

Example 3

- Find the employment numbers of pilots who can fly Boeing **and** MD planes

Can_Fly

Emp_No	Model_No
1001	B727
1001	B747
1001	DC10
1002	A320
1002	A340
1002	B757
1002	DC9
1003	A310
1003	DC9

$$\pi_{\text{Emp\_No}}(\sigma_{\text{Maker='Boeing'} \wedge \text{Maker='MD'}}\ (\text{Can\_Fly} \otimes \text{Plane}))$$

WRONG!

The correct expression is:

$$\pi_{\text{Emp\_No}}(\sigma_{\text{Maker='Boeing'}}(\text{Can\_Fly} \otimes \text{Plane}))$$

$$\cap$$

$$\pi_{\text{Emp\_No}}(\sigma_{\text{Maker='MD'}}(\text{Can\_Fly} \otimes \text{Plane}))$$

Emp_No
1001
1002

Example 4

- Find the employment numbers of pilots who can fly **at least two** Boeing planes

Can_Fly

Emp_No	Model_No
1001	B727
1001	B747
1001	DC10
1002	A320
1002	A340
1002	B757
1002	DC9
1003	A310
1003	DC9

$$\rho(\text{BP1}, \sigma_{\text{Maker='Boeing'}}(\text{Can\_Fly} \otimes \text{Plane}))$$

$$\rho(\text{BP2}, \text{BP1})$$

$$\pi_{\text{BP1.Emp\_No}}(\sigma_{\text{BP1.Model\_No} \neq \text{BP2.Model\_No} \wedge \text{BP1.Emp\_No=BP2.Emp\_No}}(\text{BP1} \times \text{BP2}))$$

Emp_No
1001

Example 4 – cont'd

- Find the employment numbers of pilots who can fly **at least two** Boeing planes

Can_Fly

Emp_No	Model_No
1001	B727
1001	B747
1001	DC10
1002	A320
1002	A340
1002	B757
1002	DC9
1003	A310
1003	DC9

```
SELECT Emp_No
FROM Can_Fly, Plane
WHERE Maker = 'Boeing'
AND Plane.Model_No = Can_Fly.Model_No
GROUP BY Emp_No
HAVING COUNT(*) >= 2
```

Algebra does not support aggregations

Emp_No
1001

Example 5 - Division

- Find the employment numbers of pilots who can fly **all** MD planes

$\rho\ (\ B,\ \pi_{\text{Model\_No}}\ (\ \sigma_{\text{Maker='MD'}}\ (\ \text{Plane}\ )))$

$\rho\ (\ A,\ \text{Can\_Fly})$

$\pi_{\text{Emp\_No}}\ (A) - \pi_{\text{Emp\_No}}\ ((\ \pi_{\text{Emp\_No}}\ (A) \times B\ ) - A)$

What is the corresponding SQL?

Summary

- Relational algebra is a simple and powerful query language
- Basic operators: σ, π, $\cup$, $-$, $\times$
- Additional operators: ρ, $\cap$, $\otimes$, /
- Relational algebra is closed – operator's output is a relation
- Relational operators can be composed to form complex relational algebra expressions