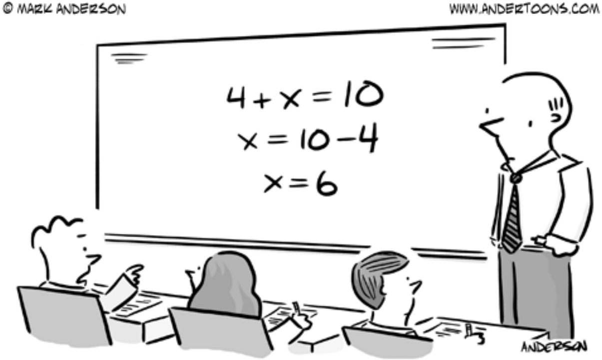
Welcome to Week 7 Workshop



"Hold on. When we learned Roman numerals, X was 10. Now it's 6. What's going on around here?!"

Housekeeping

- The mark and feedback on Assignment 1 (SQL) is available on Wattle.
  - Refer to the sample solutions along with the common issues.
  - Test your queries on moviedb2021 instead of moviedb.

Housekeeping

- The mark and feedback on Assignment 1 (SQL) is available on Wattle.
  - Refer to the sample solutions along with the common issues.
  - Test your queries on moviedb2021 instead of moviedb.

- The specification of Assignment 2 (Database Theory) will be available on Sep 28. The submission via Wattle is due 23:59 Oct 12 (Tuesday, Week 10)
  - **Individual, no group work!**
  - **Do not post any idea/partial solution/result on Wattle.**

SQL ⇒ Relational Algebra

Database users

SQL queries

```
SELECT ...
FROM ...
WHERE ...
...
```

SQL $\Rightarrow$ Relational Algebra

Database users		Database systems
SQL queries		RA queries
SELECT ... FROM ... WHERE	$\rightarrow\mid\rightarrow$ $\rightarrow\mid\rightarrow$	$\sigma,\ \pi,\ \rho$ $\cup,\ \cap,\ -$ $\times,\ \bowtie,\ \ldots$

Why Relational Algebra?

- Make SQL queries easy-to-use ...

Why Relational Algebra?

- Make SQL queries easy-to-use ...

Declarative	vs **Procedural**
Make me a cake	Mix 2 cup flour, 1/2 cup butter, and 2 eggs until well blended. Divide the dough into a 12x2-in. log. Preheat oven to 350° and bake 30-35 minutes.

Why Relational Algebra?

- Make SQL queries easy-to-use ...

Declarative vs **Procedural**	
Make me a cake	Mix 2 cup flour, 1/2 cup butter, and 2 eggs until well blended. Divide the dough into a 12x2-in. log. Preheat oven to 350° and bake 30-35 minutes.

RA bridges the gap between the declarative nature of SQL and the procedure nature of a computer system.

Why Relational Algebra?

- Make SQL queries easy-to-use ...

Declarative vs	**Procedural**
Make me a cake	Mix 2 cup flour, 1/2 cup butter, and 2 eggs until well blended. Divide the dough into a 12x2-in. log. Preheat oven to 350° and bake 30-35 minutes.

RA bridges the gap between the declarative nature of SQL and the procedure nature of a computer system.

- **Expressive:** Each SQL query can be represented by a RA query.
- **Procedural:** Each RA query consists of step-by-step operations.

Why Relational Algebra?

- Make SQL queries run fast ...

Why Relational Algebra?

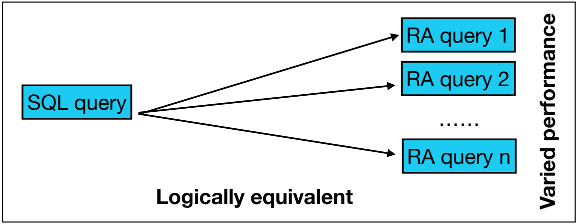
- Make SQL queries run fast ...

RA enables many different ways to implement a SQL query.

Why Relational Algebra?

- Make SQL queries run fast ...

RA enables many different ways to implement a SQL query.



Arithmetic v.s. Algebra

What is the difference between “2+8=8+2” and “a+b=b+a”?

Arithmetic v.s. Algebra

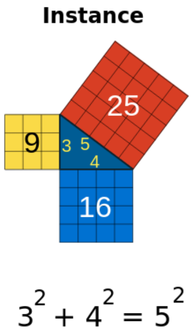
What is the difference between “2+8=8+2” and “a+b=b+a”?

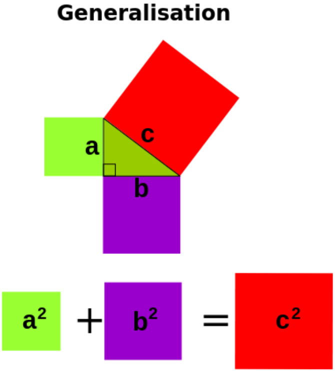
- Arithmetic: “2+8=8+2” is a specific fact.
- Algebra: “a+b=b+a” is a general pattern.

Arithmetic v.s. Algebra

What is the difference between “2+8=8+2” and “a+b=b+a”?

- Arithmetic: “2+8=8+2” is a specific fact.
- Algebra: “a+b=b+a” is a general pattern.





What is an “Algebra”?

- Mathematical system consisting of:
  - **Operands** — variables or values from which new values can be constructed.
  - **Operators** — symbols denoting procedures that construct new values from given values.

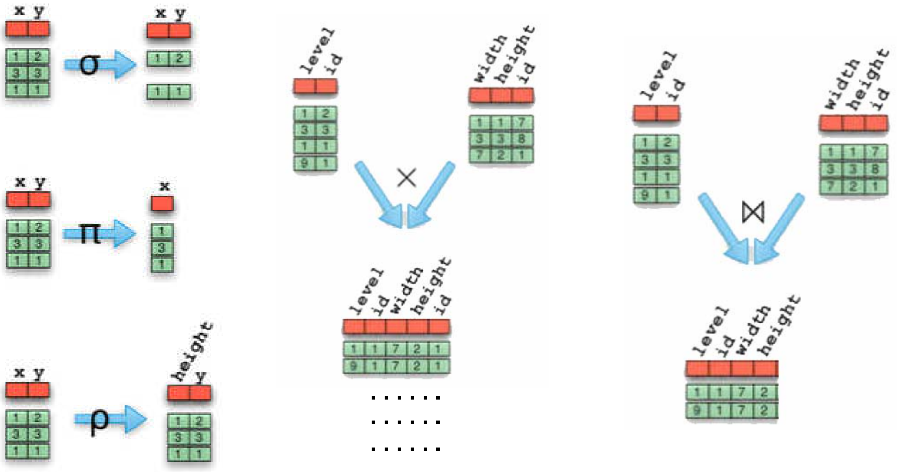
What is an "Algebra"?

- Mathematical system consisting of:
  - **Operands** — variables or values from which new values can be constructed.
  - **Operators** — symbols denoting procedures that construct new values from given values.
- Elementary algebra consisting of:
  - **Operands** — variables X, Y, Z, etc.
  - **Operators** — +, -, $\times$, /

What is an “Algebra”?

- Mathematical system consisting of:
  - **Operands** — variables or values from which new values can be constructed.
  - **Operators** — symbols denoting procedures that construct new values from given values.
- Elementary algebra consisting of:
  - **Operands** — variables X, Y, Z, etc.
  - **Operators** — +, -, $\times$, /
- Relational algebra consisting of:
  - **Operands** — relations R_1, R_2, R_3, etc.
  - **Operators** — $\{\sigma, \pi, \cup, \cap, \bowtie, \ldots\}$

Relational Operators [1]



[1] http://merrigrove.blogspot.com.au/2011/12/another-introduction-to-algebraic-data.html (with some changes)

Summary of Relational Operators

Operator	Notation	Meaning
Selection	$\sigma_{\varphi}(R)$	choose rows
Projection	$\pi_{A_1,\ldots,A_n}(R)$	choose columns
Union **Intersection** **Difference**	$R_1 \cup R_2$ $R_1 \cap R_2$ $R_1 - R_2$	set operations
Cartesian product **Join** **Natural-join**	$R_1 \times R_2$ $R_1 \bowtie_{\varphi} R_2$ $R_1 \bowtie R_2$	combine tables
Renaming	$\rho_{R'(A_1,\ldots,A_n)}(R)$ $\rho_{R'}(R)$ $\rho_{(A_1,\ldots,A_n)}(R)$	rename relation and attributes

Selection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the items with price less than 9 CHF?

Selection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the items with price less than 9 CHF?
  $\sigma_{\varphi}(R)$, $\varphi = Price < 9$, R=SELL $\Rightarrow$ $\sigma_{Price<9}(\textbf{SELL})$.

Selection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the items with price less than 9 CHF?
  $\sigma_{\varphi}(R)$, $\varphi = Price < 9$, R=SELL $\Rightarrow$ $\sigma_{Price<9}$**(SELL)**.

Shop	Item	Price
Coop	Ham	8
Migros	Cheese	8

Projection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the available shops and items?

Projection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the available shops and items?
  $\pi_{A_1,\ldots,A_n}(R)$, $\{A_1,\ldots,A_n\} = \{Shop, Item\}$, R=SELL $\Rightarrow$ $\pi_{Shop,Item}$**(SELL)**.

Projection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the available shops and items?

$\pi_{A_1,\ldots,A_n}(R)$, $\{A_1, \ldots, A_n\} = \{Shop, Item\}$, R=SELL $\Rightarrow$ $\pi_{Shop,Item}$**(SELL)**.

Shop	Item
Coop	Cheese
Migros	Cabbage
Coop	Ham
Migros	Cheese

Selection + Projection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the available shops and items with the price less than 9 CHF?

Selection + Projection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the available shops and items with the price less than 9 CHF?

$\pi_{Shop,Item}(\sigma_{Price<9}(\textsc{Sell}))$

Selection + Projection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the available shops and items with the price less than 9 CHF?

$\pi_{Shop,Item}(\sigma_{Price<9}(\text{SELL}))$

Shop	Item	Price
Coop	Ham	8
Migros	Cheese	8

Shop	Item
Coop	Ham
Migros	Cheese

Selection + Projection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the available shops and items with the price less than 9 CHF?

Selection + Projection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the available shops and items with the price less than 9 CHF?

  What about $\sigma_{Price<9}(\pi_{Shop,Item}(\textbf{SELL}))$?

Selection + Projection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the available shops and items with the price less than 9 CHF?

  What about $\sigma_{Price<9}(\pi_{Shop,Item}(\text{SELL}))$?

Shop	Item
Coop	Cheese
Migros	Cabbage
Coop	Ham
Migros	Cheese

Selection + Projection Example

- Consider the relation SELL:

Shop	Item	Price
Coop	Cheese	10
Migros	Cabbage	10
Coop	Ham	8
Migros	Cheese	8

- What if we only want to know all the available shops and items with the price less than 9 CHF?

  What about $\sigma_{Price<9}(\pi_{Shop,Item}(\textsc{Sell}))$?

Shop	Item
Coop	Cheese
Migros	Cabbage
Coop	Ham
Migros	Cheese

Error!
No price attribute available.

Selection and Projection – Properties

- Selections are **commutative**

  $\sigma_{\varphi_1}(\sigma_{\varphi_2}(R)) = \sigma_{\varphi_2}(\sigma_{\varphi_1}(R))$

Selection and Projection – Properties

- Selections are **commutative**

  $\sigma_{\varphi_1}(\sigma_{\varphi_2}(R)) = \sigma_{\varphi_2}(\sigma_{\varphi_1}(R)) = \sigma_{\varphi_1 \wedge \varphi_2}(R)$.

Selection and Projection – Properties

- Selections are **commutative**

  $\sigma_{\varphi_1}(\sigma_{\varphi_2}(R)) = \sigma_{\varphi_2}(\sigma_{\varphi_1}(R)) = \sigma_{\varphi_1 \wedge \varphi_2}(R)$.

- Projections are **not commutative**

  $\pi_{B_1,\ldots,B_m}(\pi_{A_1,\ldots,A_n}(R)) = \pi_{A_1,\ldots,A_n}(\pi_{B_1,\ldots,B_m}(R))$ **does not hold** in general

Selection and Projection – Properties

- Selections are **commutative**

  $\sigma_{\varphi_1}(\sigma_{\varphi_2}(R)) = \sigma_{\varphi_2}(\sigma_{\varphi_1}(R)) = \sigma_{\varphi_1 \wedge \varphi_2}(R).$

- Projections are **not commutative**

  $\pi_{B_1,\ldots,B_m}(\pi_{A_1,\ldots,A_n}(R)) = \pi_{A_1,\ldots,A_n}(\pi_{B_1,\ldots,B_m}(R))$ **does not hold** in general

- Pairs of selection and projection are **not commutative**

  $\pi_{A_1,\ldots,A_n}(\sigma_{\varphi}(R)) = \sigma_{\varphi}(\pi_{A_1,\ldots,A_n}(R))$ **does not hold** in general

Selection and Projection – Properties

- Selections are **commutative**

  $\sigma_{\varphi_1}(\sigma_{\varphi_2}(R)) = \sigma_{\varphi_2}(\sigma_{\varphi_1}(R)) = \sigma_{\varphi_1 \wedge \varphi_2}(R)$.

- Projections are **not commutative**

  $\pi_{B_1,\ldots,B_m}(\pi_{A_1,\ldots,A_n}(R)) = \pi_{A_1,\ldots,A_n}(\pi_{B_1,\ldots,B_m}(R))$ **does not hold** in general

- Pairs of selection and projection are **not commutative**

  $\pi_{A_1,\ldots,A_n}(\sigma_{\varphi}(R)) = \sigma_{\varphi}(\pi_{A_1,\ldots,A_n}(R))$ **does not hold** in general

- Selections will always keep the same number of columns?

Selection and Projection – Properties

- Selections are **commutative**

  $\sigma_{\varphi_1}(\sigma_{\varphi_2}(R)) = \sigma_{\varphi_2}(\sigma_{\varphi_1}(R)) = \sigma_{\varphi_1 \wedge \varphi_2}(R)$.

- Projections are **not commutative**

  $\pi_{B_1,\ldots,B_m}(\pi_{A_1,\ldots,A_n}(R)) = \pi_{A_1,\ldots,A_n}(\pi_{B_1,\ldots,B_m}(R))$ **does not hold** in general

- Pairs of selection and projection are **not commutative**

  $\pi_{A_1,\ldots,A_n}(\sigma_{\varphi}(R)) = \sigma_{\varphi}(\pi_{A_1,\ldots,A_n}(R))$ **does not hold** in general

- Selections will always keep the same number of columns? **Yes**.

Selection and Projection – Properties

- Selections are **commutative**

  $\sigma_{\varphi_1}(\sigma_{\varphi_2}(R)) = \sigma_{\varphi_2}(\sigma_{\varphi_1}(R)) = \sigma_{\varphi_1 \wedge \varphi_2}(R)$.

- Projections are **not commutative**

  $\pi_{B_1,\ldots,B_m}(\pi_{A_1,\ldots,A_n}(R)) = \pi_{A_1,\ldots,A_n}(\pi_{B_1,\ldots,B_m}(R))$ **does not hold** in general

- Pairs of selection and projection are **not commutative**

  $\pi_{A_1,\ldots,A_n}(\sigma_{\varphi}(R)) = \sigma_{\varphi}(\pi_{A_1,\ldots,A_n}(R))$ **does not hold** in general

- Selections will always keep the same number of columns? **Yes**.
- Projections will always keep the same number of rows?

Selection and Projection – Properties

- Selections are **commutative**

  $\sigma_{\varphi_1}(\sigma_{\varphi_2}(R)) = \sigma_{\varphi_2}(\sigma_{\varphi_1}(R)) = \sigma_{\varphi_1 \wedge \varphi_2}(R)$.

- Projections are **not commutative**

  $\pi_{B_1,\ldots,B_m}(\pi_{A_1,\ldots,A_n}(R)) = \pi_{A_1,\ldots,A_n}(\pi_{B_1,\ldots,B_m}(R))$ **does not hold** in general

- Pairs of selection and projection are **not commutative**

  $\pi_{A_1,\ldots,A_n}(\sigma_{\varphi}(R)) = \sigma_{\varphi}(\pi_{A_1,\ldots,A_n}(R))$ **does not hold** in general

- Selections will always keep the same number of columns? **Yes**.
- Projections will always keep the same number of rows? **No** (may introduce duplicates and have to be eliminated).

Selection and Projection – Properties

- Selections are **commutative**

  $\sigma_{\varphi_1}(\sigma_{\varphi_2}(R)) = \sigma_{\varphi_2}(\sigma_{\varphi_1}(R)) = \sigma_{\varphi_1 \wedge \varphi_2}(R)$.

- Projections are **not commutative**

  $\pi_{B_1,\ldots,B_m}(\pi_{A_1,\ldots,A_n}(R)) = \pi_{A_1,\ldots,A_n}(\pi_{B_1,\ldots,B_m}(R))$ **does not hold** in general

- Pairs of selection and projection are **not commutative**

  $\pi_{A_1,\ldots,A_n}(\sigma_{\varphi}(R)) = \sigma_{\varphi}(\pi_{A_1,\ldots,A_n}(R))$ **does not hold** in general

- Selections will always keep the same number of columns? **Yes**.
- Projections will always keep the same number of rows? **No** (may introduce duplicates and have to be eliminated).

Set Operations

- Relations are sets (of tuples/rows), we have standard operations on sets.
  - **Union**, denoted as $R_1 \cup R_2$, results in a relation that includes all tuples either in R_1 or in R_2. Duplicate tuples are eliminated.
  - **Intersection**, denoted as $R_1 \cap R_2$, results in a relation that includes all tuples that are in both R_1 and R_2.
  - **Difference**, denoted as $R_1 - R_2$, results in a relation that includes all tuples that are in R_1 but not in R_2.

Set Operations

- Relations are sets (of tuples/rows), we have standard operations on sets.
  - **Union**, denoted as $R_1 \cup R_2$, results in a relation that includes all tuples either in R_1 or in R_2. Duplicate tuples are eliminated.
  - **Intersection**, denoted as $R_1 \cap R_2$, results in a relation that includes all tuples that are in both R_1 and R_2.
  - **Difference**, denoted as $R_1 - R_2$, results in a relation that includes all tuples that are in R_1 but not in R_2.
- **Type compatibility**: R_1 and R_2 must have **the same type**, i.e.,
  - the same number of attributes, and
  - the same domains for the attributes (the order is important).

Set Operations

STUDY		
StudentID	CourseNo	Hours
111	COMP2400	120
222	COMP2400	115
333	STAT2001	120
111	BUSN2011	110
111	ECON2102	120
333	BUSN2011	130

- What is the result for $\pi_{StudentID}(\sigma_{CourseNo='COMP2400'}(\text{STUDY})) \cap \pi_{StudentID}(\sigma_{CourseNo='ECON2102'}(\text{STUDY}))$?

$$R_1 = \pi_{StudentID}(\sigma_{CourseNo='COMP2400'}(\text{STUDY}))$$

$$R_2 = \pi_{StudentID}(\sigma_{CourseNo='ECON2102'}(\text{STUDY}))$$

Set Operations

STUDY		
StudentID	CourseNo	Hours
111	COMP2400	120
222	COMP2400	115
333	STAT2001	120
111	BUSN2011	110
111	ECON2102	120
333	BUSN2011	130

- What is the result for $\pi_{StudentID}(\sigma_{CourseNo='COMP2400'}(\text{STUDY})) \cap \pi_{StudentID}(\sigma_{CourseNo='ECON2102'}(\text{STUDY}))$?

$$R_1 = \pi_{StudentID}(\sigma_{CourseNo='COMP2400'}(\text{STUDY}))$$

StudentID
111
222

INTERSECT

$$R_2 = \pi_{StudentID}(\sigma_{CourseNo='ECON2102'}(\text{STUDY}))$$

StudentID
111

Set Operations

STUDY		
StudentID	CourseNo	Hours
111	COMP2400	120
222	COMP2400	115
333	STAT2001	120
111	BUSN2011	110
111	ECON2102	120
333	BUSN2011	130

- What is the result for
$\pi_{StudentID}(\sigma_{CourseNo='COMP2400'}(\text{STUDY})) \cap \pi_{StudentID}(\sigma_{CourseNo='ECON2102'}(\text{STUDY}))$?

$$R_1 = \pi_{StudentID}(\sigma_{CourseNo='COMP2400'}(\text{STUDY}))$$

$$R_1 \cap R_2$$

StudentID
111

$$R_2 = \pi_{StudentID}(\sigma_{CourseNo='ECON2102'}(\text{STUDY}))$$

Set Operations

STUDY		
StudentID	CourseNo	Hours
111	COMP2400	120
222	COMP2400	115
333	STAT2001	120
111	BUSN2011	110
111	ECON2102	120
333	BUSN2011	130

- What is the result for $\pi_{StudentID}(\sigma_{CourseNo='COMP2400'}(\text{STUDY}) \cap \sigma_{CourseNo='ECON2102'}(\text{STUDY}))$?

Set Operations

STUDY		
StudentID	CourseNo	Hours
111	COMP2400	120
222	COMP2400	115
333	STAT2001	120
111	BUSN2011	110
111	ECON2102	120
333	BUSN2011	130

- What is the result for $\pi_{StudentID}(\sigma_{CourseNo='COMP2400'}(\text{STUDY}) \cap \sigma_{CourseNo='ECON2102'}(\text{STUDY}))$?

$$R_1 = \sigma_{CourseNo='COMP2400'}(\text{STUDY})$$

$$\pi_{StudentID}(R_1 \cap R_2)$$

$$R_2 = \sigma_{CourseNo='ECON2102'}(\text{STUDY})$$

Set Operations

STUDY		
StudentID	CourseNo	Hours
111	COMP2400	120
222	COMP2400	115
333	STAT2001	120
111	BUSN2011	110
111	ECON2102	120
333	BUSN2011	130

- What is the result for

$\pi_{StudentID}(\sigma_{CourseNo='COMP2400'}(\text{STUDY}) \cap \sigma_{CourseNo='ECON2102'}(\text{STUDY}))$?

$R_1 = \sigma_{CourseNo='COMP2400'}(\text{STUDY})$

$\pi_{StudentID}(R_1 \cap R_2)$

EMPTY!

$R_2 = \sigma_{CourseNo='ECON2102'}(\text{STUDY})$

Cartesian Product, Join and Natural Join

- **Cartesian product** $R_1 \times R_2$ combines tuples from two relations in a combinatorial fashion.

Cartesian Product, Join and Natural Join

- **Cartesian product** $R_1 \times R_2$ combines tuples from two relations in a combinatorial fashion.
- **Join** $R_1 \bowtie_\varphi R_2$ is introduced as the **combination of Cartesian product and selection**. That is,

$$R_1 \bowtie_\varphi R_2 = \sigma_\varphi(R_1 \times R_2).$$

Cartesian Product, Join and Natural Join

- **Cartesian product** $R_1 \times R_2$ combines tuples from two relations in a combinatorial fashion.
- **Join** $R_1 \bowtie_\varphi R_2$ is introduced as the **combination of Cartesian product and selection**. That is,

$$R_1 \bowtie_\varphi R_2 = \sigma_\varphi(R_1 \times R_2).$$

- **Natural Join** $R_1 \bowtie R_2$
  1. Implicitly apply the join condition on **equality comparisons of attributes that have the same name** in both relations.
  2. Project out one copy of the attributes that have the same name in both relations.

Cartesian Product – Example

COURSE		
No	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

ENROL			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for COURSE $\times$ ENROL?

Cartesian Product – Example

COURSE		
No	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

ENROL			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for COURSE $\times$ ENROL?

  COURSE $\times$ ENROL will have 9 (=3$\times$3) tuples and 7 (=3+4) attributes.

Join – Example

Course		
No	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

Enrol			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for COURSE $\bowtie_{No=CourseNo}$ ENROL?

Join – Example

Course		
No	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

Enrol			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for COURSE $\bowtie_{No=CourseNo}$ ENROL?

No	Cname	Unit	StudentID	CourseNo	Semester	Status
COMP2400	Relational Databases	6	222	**COMP2400**	2016 S1	active
COMP2400	Relational Databases	6	111	**COMP2400**	2016 S2	active
BUSN2011	Management Accounting	6	111	**BUSN2011**	2016 S1	active

Join – Example

COURSE		
No	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

ENROL			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for $\pi_{No,Cname}(\text{COURSE} \bowtie_{No=CourseNo} \text{ENROL})$?

Join – Example

COURSE		
No	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

ENROL			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for $\pi_{No,Cname}(\text{COURSE} \bowtie_{No=CourseNo} \text{ENROL})$?

No	Cname
COMP2400	Relational Databases
BUSN2011	Management Accounting

Natural Join – Example

COURSE		
No	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

ENROL			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for COURSE $\bowtie$ ENROL?

Natural Join – Example

COURSE		
No	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

ENROL			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for COURSE $\bowtie$ ENROL?

  If there are no matching attributes in two tables for NATURAL JOIN, COURSE $\bowtie$ ENROL will become COURSE $\times$ ENROL which outputs 9 (=3$\times$3) tuples and 7 (=3+4) attributes.

Natural Join – Example

COURSE		
CourseNo	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

ENROL			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for COURSE $\bowtie$ ENROL?

Natural Join – Example

Course		
CourseNo	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

Enrol			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for COURSE $\bowtie$ ENROL?

CourseNo	Cname	Unit	StudentID	Semester	Status
COMP2400	Relational Databases	6	222	2016 S1	active
COMP2400	Relational Databases	6	111	2016 S2	active
BUSN2011	Management Accounting	6	111	2016 S1	active

Natural Join – Example

COURSE		
CourseNo	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

ENROL			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for $\sigma_{StudentID=111}(\text{COURSE} \bowtie \text{ENROL})$?

Natural Join – Example

COURSE		
CourseNo	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

ENROL			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for $\sigma_{StudentID=111}(\text{COURSE} \bowtie \text{ENROL})$?

CourseNo	Cname	Unit	StudentID	Semester	Status
COMP2400	Relational Databases	6	111	2016 S2	active
BUSN2011	Management Accounting	6	111	2016 S1	active

Natural Join – Example

COURSE		
CourseNo	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

ENROL			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for COURSE $\bowtie$ COURSE?

Natural Join – Example

Course		
CourseNo	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

Enrol			
StudentID	CourseNo	Semester	Status
111	BUSN2011	2016 S1	active
222	COMP2400	2016 S1	active
111	COMP2400	2016 S2	active

- What is the result for Course $\bowtie$ Course?

Course		
CourseNo	Cname	Unit
COMP2400	Relational Databases	6
BUSN2011	Management Accounting	6
ECON2102	Macroeconomics	6

Join – More Examples

STUDENT			
StudentID	Name	DoB	Email

COURSE		
No	Cname	Unit

ENROL		
StudentID	CourseNo	Status

- List the email of students who have enrolled in courses and the CourseNo of these courses.

Join – More Examples

STUDENT			
<u>StudentID</u>	Name	DoB	Email

COURSE		
<u>No</u>	Cname	Unit

ENROL		
<u>StudentID</u>	<u>CourseNo</u>	Status

- List the email of students who have enrolled in courses and the CourseNo of these courses.
  1. $\pi_{Email,CourseNo}(\sigma_{Student.StudentID=Enrol.StudentID}(\text{STUDENT} \times \text{ENROL}))$

Join – More Examples

STUDENT			
StudentID	Name	DoB	Email

COURSE		
No	Cname	Unit

ENROL		
StudentID	CourseNo	Status

- List the email of students who have enrolled in courses and the CourseNo of these courses.
  1. $\pi_{Email,CourseNo}(\sigma_{Student.StudentID=Enrol.StudentID}(\text{STUDENT} \times \text{ENROL}))$
  2. $\pi_{Email,CourseNo}(\text{STUDENT} \bowtie_{Student.StudentID=Enrol.StudentID} \text{ENROL})$

Join – More Examples

STUDENT			
StudentID	Name	DoB	Email

COURSE		
No	Cname	Unit

ENROL		
StudentID	CourseNo	Status

- List the email of students who have enrolled in courses and the CourseNo of these courses.
    1. $\pi_{Email,CourseNo}(\sigma_{Student.StudentID=Enrol.StudentID}(\text{STUDENT} \times \text{ENROL}))$
    2. $\pi_{Email,CourseNo}(\text{STUDENT} \bowtie_{Student.StudentID=Enrol.StudentID} \text{ENROL})$
    3. $\pi_{Email,CourseNo}(\text{STUDENT} \bowtie \text{ENROL})$

Join – More Examples

STUDENT			
StudentID	Name	DoB	Email

COURSE		
No	Cname	Unit

ENROL		
StudentID	CourseNo	Status

- List the email of students who have enrolled in courses and the CourseNo of these courses.
  1. $\pi_{Email,CourseNo}(\sigma_{Student.StudentID=Enrol.StudentID}(\text{STUDENT} \times \text{ENROL}))$
  2. $\pi_{Email,CourseNo}(\text{STUDENT} \bowtie_{Student.StudentID=Enrol.StudentID} \text{ENROL})$
  3. $\pi_{Email,CourseNo}(\text{STUDENT} \bowtie \text{ENROL})$
  4. $(\pi_{Email,CourseNo}(\text{STUDENT})) \bowtie \text{ENROL}$

Join – More Examples

STUDENT			
StudentID	Name	DoB	Email

COURSE		
No	Cname	Unit

ENROL		
StudentID	CourseNo	Status

- List the email of students who have enrolled in courses and the CourseNo of these courses.
  1. $\pi_{Email, CourseNo}(\sigma_{Student.StudentID=Enrol.StudentID}(\text{STUDENT} \times \text{ENROL}))$
  2. $\pi_{Email, CourseNo}(\text{STUDENT} \bowtie_{Student.StudentID=Enrol.StudentID} \text{ENROL})$
  3. $\pi_{Email, CourseNo}(\text{STUDENT} \bowtie \text{ENROL})$
  4. $(\pi_{Email, CourseNo}(\text{STUDENT})) \bowtie \text{ENROL}$ **Incorrect!**
  5. $\pi_{Email}(\text{STUDENT}) \bowtie \pi_{CourseNo}(\text{ENROL})$

Join – More Examples

STUDENT			
StudentID	Name	DoB	Email

COURSE		
No	Cname	Unit

ENROL		
StudentID	CourseNo	Status

- List the email of students who have enrolled in courses and the CourseNo of these courses.
  1. $\pi_{Email,CourseNo}(\sigma_{Student.StudentID=Enrol.StudentID}(\text{STUDENT} \times \text{ENROL}))$
  2. $\pi_{Email,CourseNo}(\text{STUDENT} \bowtie_{Student.StudentID=Enrol.StudentID} \text{ENROL})$
  3. $\pi_{Email,CourseNo}(\text{STUDENT} \bowtie \text{ENROL})$
  4. $(\pi_{Email,CourseNo}(\text{STUDENT})) \bowtie \text{ENROL}$ **Incorrect!**
  5. $\pi_{Email}(\text{STUDENT}) \bowtie \pi_{CourseNo}(\text{ENROL})$ **Incorrect!**

Renaming

- **Renaming** is used to rename either the relation name or the attribute names, or both.

Renaming

- **Renaming** is used to rename either the relation name or the attribute names, or both.
- Renaming is denoted as
  - $\rho_{R'(A_1,\ldots,A_n)}(R)$: renaming the relation name to $R^{'}$ and the attribute names to $A_1, \ldots, A_n$,

Renaming

- **Renaming** is used to rename either the relation name or the attribute names, or both.
- Renaming is denoted as
  - $\rho_{R'(A_1,\ldots,A_n)}(R)$: renaming the relation name to $R^{'}$ and the attribute names to $A_1, \ldots, A_n$,
  - $\rho_{R'}(R)$: renaming the relation name to $R^{'}$ and keeping the attribute names unchanged, or

Renaming

- **Renaming** is used to rename either the relation name or the attribute names, or both.
- Renaming is denoted as
  - $\rho_{R'(A_1,\ldots,A_n)}(R)$: renaming the relation name to R' and the attribute names to $A_1, \ldots, A_n$,
  - $\rho_{R'}(R)$: renaming the relation name to R' and keeping the attribute names unchanged, or
  - $\rho_{(A_1,\ldots,A_n)}(R)$: renaming the attribute names to $A_1, \ldots, A_n$ and keeping the relation name unchanged.

Renaming

- **Renaming** is used to rename either the relation name or the attribute names, or both.
- Renaming is denoted as
  - $\rho_{R'(A_1,\ldots,A_n)}(R)$: renaming the relation name to R' and the attribute names to $A_1, \ldots, A_n$,
  - $\rho_{R'}(R)$: renaming the relation name to R' and keeping the attribute names unchanged, or
  - $\rho_{(A_1,\ldots,A_n)}(R)$: renaming the attribute names to $A_1, \ldots, A_n$ and keeping the relation name unchanged.
- Renaming is useful for giving names to the relations that hold the intermediate results.

Rename – Example

- Given the following relation schema:

  STUDENT=\{StudentID, Name, DoB\}

- Find **pairs of** students who have the same birthday. Show their names.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

- What about the following choices?

Rename – Example

- Given the following relation schema:

$$\text{STUDENT}=\{\text{StudentID, Name, DoB}\}$$

- Find **pairs of** students who have the same birthday. Show their names.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

- What about the following choices?
  1. $\pi_{Name,Name}(\sigma_{DoB=DoB}(\text{STUDENT} \times \text{STUDENT}))$
  2. $\pi_{Name,Name}(\text{STUDENT} \bowtie_{DoB=DoB} \text{STUDENT})$
  3. $\pi_{Name,Name}(\text{STUDENT} \bowtie \text{STUDENT})$

Rename – Example

- **(1):** $\pi_{Name,Name}(\sigma_{DoB=DoB}(\text{STUDENT} \times \text{STUDENT}))$.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

Rename – Example

- **(1):** $\pi_{Name,Name}(\sigma_{DoB=DoB}(\text{STUDENT} \times \text{STUDENT}))$.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

STUDENT × STUDENT

StudentID	Name	DoB	StudentID	Name	DoB
457	Lisa	18-Oct-1993	457	Lisa	18-Oct-1993
457	Lisa	18-Oct-1993	458	Mike	16-May-1990
457	Lisa	18-Oct-1993	458	Peter	18-Oct-1993
458	Mike	16-May-1990	457	Lisa	18-Oct-1993
458	Mike	16-May-1990	458	Mike	16-May-1990
458	Mike	16-May-1990	458	Peter	18-Oct-1993
458	Peter	18-Oct-1993	457	Lisa	18-Oct-1993
458	Peter	18-Oct-1993	458	Mike	16-May-1990
458	Peter	18-Oct-1993	458	Peter	18-Oct-1993

Rename – Example

- **(1):** $\pi_{Name,Name}(\sigma_{DoB=DoB}(\text{STUDENT} \times \text{STUDENT}))$.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

STUDENT × STUDENT					
StudentID	Name	DoB	StudentID	Name	DoB
457	Lisa	18-Oct-1993	457	Lisa	18-Oct-1993
457	Lisa	18-Oct-1993	458	Mike	16-May-1990
457	Lisa	18-Oct-1993	458	Peter	18-Oct-1993
458	Mike	16-May-1990	457	Lisa	18-Oct-1993
458	Mike	16-May-1990	458	Mike	16-May-1990
458	Mike	16-May-1990	458	Peter	18-Oct-1993
458	Peter	18-Oct-1993	457	Lisa	18-Oct-1993
458	Peter	18-Oct-1993	458	Mike	16-May-1990
458	Peter	18-Oct-1993	458	Peter	18-Oct-1993

- **Incorrect!**

Rename – Example

- **(2):** $\pi_{Name,Name}(\text{STUDENT} \bowtie_{DoB=DoB} \text{STUDENT})$

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

Rename – Example

- **(2):** $\pi_{Name,Name}(\text{STUDENT} \bowtie_{DoB=DoB} \text{STUDENT})$

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

$\text{STUDENT} \bowtie_{DoB=DoB} \text{STUDENT}$?					
StudentID	Name	DoB	StudentID	Name	DoB

Rename – Example

- **(2):** $\pi_{Name,Name}(\text{STUDENT} \bowtie_{DoB=DoB} \text{STUDENT})$

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

STUDENT $\bowtie_{DoB=DoB}$ STUDENT?					
StudentID	Name	DoB	StudentID	Name	DoB

- **Incorrect!**

Rename – Example

- **(3):** $\pi_{Name,Name}(\text{STUDENT} \bowtie \text{STUDENT})$

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

Rename – Example

- **(3):** $\pi_{Name,Name}(\text{STUDENT} \bowtie \text{STUDENT})$

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

(STUDENT ⋈ STUDENT)		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

Rename – Example

- **(3):** $\pi_{Name,Name}(\text{STUDENT} \bowtie \text{STUDENT})$

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

(STUDENT ⋈ STUDENT)		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

- **Incorrect!**

Rename – Example

- Given the following relation schema:

  STUDENT={StudentID, Name, DoB}

- Find **pairs of** students who have the same birthday. Show their names.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

- What about the following choices?

Rename – Example

- Given the following relation schema:

$$\text{STUDENT}=\{\text{StudentID, Name, DoB}\}$$

- Find **pairs of** students who have the same birthday. Show their names.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

- What about the following choices?
  - $\pi_{R_1.Name, R_2.Name}(\sigma_{R_1.DoB=R_2.DoB}(\rho_{R_1}(\text{STUDENT}) \times \rho_{R_2}(\text{STUDENT})))$

Rename – Example

- Given the following relation schema:

$$\text{STUDENT}=\{\text{StudentID, Name, DoB}\}$$

- Find **pairs of** students who have the same birthday. Show their names.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

- What about the following choices?
  - $\pi_{R_1.Name, R_2.Name}(\sigma_{R_1.DoB=R_2.DoB}(\rho_{R_1}(\text{STUDENT}) \times \rho_{R_2}(\text{STUDENT})))$
    Almost correct!

Rename – Example

- Given the following relation schema:

$$\text{STUDENT}=\{\text{StudentID, Name, DoB}\}$$

- Find **pairs of** students who have the same birthday. Show their names.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

- What about the following choices?
  - $\pi_{R_1.Name, R_2.Name}(\sigma_{R_1.DoB=R_2.DoB}(\rho_{R_1}(\text{STUDENT}) \times \rho_{R_2}(\text{STUDENT})))$
    Almost correct!
  - $\pi_{Name, Name'}(\text{STUDENT} \bowtie \rho_{S(StudentID', Name', DoB)}(\text{STUDENT}))$

Rename – Example

- Given the following relation schema:

$$\text{STUDENT}=\{\text{StudentID, Name, DoB}\}$$

- Find **pairs of** students who have the same birthday. Show their names.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

- What about the following choices?
  - $\pi_{R_1.Name, R_2.Name}(\sigma_{R_1.DoB=R_2.DoB}(\rho_{R_1}(\text{STUDENT}) \times \rho_{R_2}(\text{STUDENT})))$
    Almost correct!
  - $\pi_{Name, Name'}(\text{STUDENT} \bowtie \rho_{S(StudentID', Name', DoB)}(\text{STUDENT}))$
    Almost correct!

Rename – Example

- Find **pairs of** students who have the same birthday. Show their names.

  (1). $\pi_{R_1.Name, R_2.Name}(\sigma_{R_1.StudentID < R_2.StudentID}(\sigma_{R_1.DoB = R_2.DoB}($
  $\rho_{R_1}(\text{STUDENT}) \times \rho_{R_2}(\text{STUDENT})))$

  (2). $\pi_{Name, Name'}(\sigma_{StudentID < StudentID'}($
  $\text{STUDENT} \bowtie \rho_{S(StudentID', Name', DoB)}(\text{STUDENT}))$

- If evaluating our queries over the following relation, what will be the result?

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

Rename – Example

- **(1):** $\pi_{R_1.Name, R_2.Name}(\sigma_{R_1.StudentID < R_2.StudentID}(\sigma_{R_1.DoB = R_2.DoB}($
  $\rho_{R_1}(\text{STUDENT}) \times \rho_{R_2}(\text{STUDENT}))).$

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

Rename – Example

- **(1):** $\pi_{R_1.Name, R_2.Name}(\sigma_{R_1.StudentID<R_2.StudentID}(\sigma_{R_1.DoB=R_2.DoB}(\rho_{R_1}(\text{STUDENT}) \times \rho_{R_2}(\text{STUDENT})))).$

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

$\rho_{R_1}(\text{STUDENT}) \times \rho_{R_2}(\text{STUDENT})$

R_1.StudentID	R_1.Name	R_1.DoB	R_2.StudentID	R_2.Name	R_2.DoB
457	Lisa	18-Oct-1993	457	Lisa	18-Oct-1993
457	Lisa	18-Oct-1993	458	Mike	16-May-1990
457	Lisa	18-Oct-1993	458	Peter	18-Oct-1993
458	Mike	16-May-1990	457	Lisa	18-Oct-1993
458	Mike	16-May-1990	458	Mike	16-May-1990
458	Mike	16-May-1990	458	Peter	18-Oct-1993
458	Peter	18-Oct-1993	457	Lisa	18-Oct-1993
458	Peter	18-Oct-1993	458	Mike	16-May-1990
458	Peter	18-Oct-1993	458	Peter	18-Oct-1993

Rename – Example

- **(1):** $\pi_{R_1.Name, R_2.Name}(\sigma_{R_1.StudentID < R_2.StudentID}(\sigma_{R_1.DoB = R_2.DoB}(\rho_{R_1}(\textsc{Student}) \times \rho_{R_2}(\textsc{Student}))).$

Student		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

$R' = \sigma_{R_1.DoB=R_2.DoB}(\rho_{R_1}(\textsc{Student}) \times \rho_{R_2}(\textsc{Student}))$

R_1.StudentID	R_1.Name	R_1.DoB	R_2.StudentID	R_2.Name	R_2.DoB
457	Lisa	18-Oct-1993	457	Lisa	18-Oct-1993
457	Lisa	18-Oct-1993	459	Peter	18-Oct-1993
458	Mike	16-May-1990	458	Mike	16-May-1990
459	Peter	18-Oct-1993	457	Lisa	18-Oct-1993
459	Peter	18-Oct-1993	459	Peter	18-Oct-1993

Rename – Example

- **(1):** $\pi_{R_1.Name, R_2.Name}(\sigma_{R_1.StudentID < R_2.StudentID}(\sigma_{R_1.DoB = R_2.DoB}(\rho_{R_1}(\text{STUDENT}) \times \rho_{R_2}(\text{STUDENT})))$.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

$R' = \sigma_{R_1.DoB = R_2.DoB}(\rho_{R_1}(\text{STUDENT}) \times \rho_{R_2}(\text{STUDENT}))$

R_1.StudentID	R_1.Name	R_1.DoB	R_2.StudentID	R_2.Name	R_2.DoB
457	Lisa	18-Oct-1993	457	Lisa	18-Oct-1993
457	Lisa	18-Oct-1993	459	Peter	18-Oct-1993
458	Mike	16-May-1990	458	Mike	16-May-1990
459	Peter	18-Oct-1993	457	Lisa	18-Oct-1993
459	Peter	18-Oct-1993	459	Peter	18-Oct-1993

$\pi_{R_1.Name, R_2.Name}(\sigma_{R_1.StudentID < R_2.StudentID}(R'))$

R_1.Name	R_2.Name
Lisa	Peter

Rename – Example

- **(2):** $\pi_{Name,Name'}(\sigma_{StudentID<StudentID'}($
  $\text{STUDENT} \bowtie \rho_{S(StudentID',Name',DoB)}(\text{STUDENT})).$

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

Rename – Example

- **(2):** $\pi_{Name,Name'}(\sigma_{StudentID<StudentID'}($
  $\text{STUDENT} \bowtie \rho_{S(StudentID',Name',DoB)}(\text{STUDENT}))$.

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

$R' = \text{STUDENT} \bowtie \rho_{S(StudentID',Name',DoB)}(\text{STUDENT})$

StudentID	Name	DoB	StudentID'	Name'
457	Lisa	18-Oct-1993	459	Peter
459	Peter	18-Oct-1993	457	Lisa
459	Peter	18-Oct-1993	459	Peter
457	Lisa	18-Oct-1993	457	Lisa
458	Mike	16-May-1990	458	Mike

Rename – Example

- **(2):** $\pi_{Name,Name'}(\sigma_{StudentID<StudentID'}($
  $\text{STUDENT} \bowtie \rho_{S(StudentID',Name',DoB)}(\text{STUDENT})).$

STUDENT		
StudentID	Name	DoB
457	Lisa	18-Oct-1993
458	Mike	16-May-1990
459	Peter	18-Oct-1993

$R' = \text{STUDENT} \bowtie \rho_{S(StudentID',Name',DoB)}(\text{STUDENT})$

StudentID	Name	DoB	StudentID'	Name'
457	Lisa	18-Oct-1993	459	Peter
459	Peter	18-Oct-1993	457	Lisa
459	Peter	18-Oct-1993	459	Peter
457	Lisa	18-Oct-1993	457	Lisa
458	Mike	16-May-1990	458	Mike

$\pi_{Name,Name'}(\sigma_{StudentID<StudentID'}(R'))$

Name	Name'
Lisa	Peter

Relational Algebra (RA) – example

Which awards are there in USA? List these award names.

Relational Algebra (RA) – example

Which awards are there in USA? List these award names.

Which relation schema(s) will be used?

- AWARD(*award_name*, *institution*, *country*)
  `primary key` : {*award_name*}

Relational Algebra (RA) – example

Which awards are there in USA? List these award names.

Which relation schema(s) will be used?

- $\text{AWARD}(\textit{award\_name}, \textit{institution}, \textit{country})$
  `primary key` : $\{\textit{award\_name}\}$

$\pi_{\text{award\_name}}(\sigma_{\textit{country}='\textit{USA}'}(\text{AWARD}))$

Relational Algebra (RA) – example

Find the titles of the comedy movies (i.e. the major genre of the movie is comedy) which were produced in 1994.

Relational Algebra (RA) – example

Find the titles of the comedy movies (i.e. the major genre of the movie is comedy) which were produced in 1994.

Which relation schema(s) will be used?

- MOVIE(*title*, *production_year*, *country*, *run_time*, *major_genre*)
  `primary key` : {*title*, *production_year*}

Relational Algebra (RA) – example

Find the titles of the comedy movies (i.e. the major genre of the movie is comedy) which were produced in 1994.

Which relation schema(s) will be used?

- $\textsc{Movie}(\textit{title}, \textit{production\_year}, \textit{country}, \textit{run\_time}, \textit{major\_genre})$
  $\texttt{primary key}: \{\textit{title}, \textit{production\_year}\}$

$$\pi_{\text{title}}(\sigma_{(production\_year=1994)\wedge(major\_genre='comedy')}(\text{MOVIE}))$$

Relational Algebra (RA) – example

Find the titles of the comedy movies (i.e. the major genre of the movie is comedy) which were produced in 1994.

Which relation schema(s) will be used?

- MOVIE(*title*, *production_year*, *country*, *run_time*, *major_genre*)
  `primary key` : {*title*, *production_year*}

$\pi_{\text{title}}(\sigma_{(production\_year=1994)\wedge(major\_genre='comedy')}(\text{MOVIE}))$

Is the following RA also correct?

$\pi_{\text{title}}(\sigma_{production\_year=1994}(\text{MOVIE})) \cap \pi_{\text{title}}(\sigma_{major\_genre='comedy'}(\text{MOVIE}))$

Relational Algebra (RA) – example

Find the titles of the comedy movies (i.e. the major genre of the movie is comedy) which were produced in 1994.

Which relation schema(s) will be used?

- MOVIE(*title*, *production_year*, *country*, *run_time*, *major_genre*)
  `primary key` : {*title*, *production_year*}

$\pi_{\text{title}}(\sigma_{(production\_year=1994)\wedge(major\_genre='comedy')}(\text{MOVIE}))$

Is the following RA also correct?

$\pi_{\text{title}}(\sigma_{production\_year=1994}(\text{MOVIE}))\cap\pi_{\text{title}}(\sigma_{major\_genre='comedy'}(\text{MOVIE}))$

It is not correct. Consider two movies, Robot (1994, action), Robot (2001, comedy).

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Which relation schema(s) will be used?

- MOVIE(*title*, *production_year*, *country*, *run_time*, *major_genre*)
  `primary key` : {*title*, *production_year*}
- PERSON(*id*, *first_name*, *last_name*, *year_born*)
  `primary key` : {*id*}
- ROLE(*id*, *title*, *production_year*, *description*, *credits*)
  `primary key` : {*title*, *production_year*, *description*}
  `foreign keys` : [*title*, *production_year*] $\subseteq$ MOVIE[*title*, *production_year*]
  [*id*] $\subseteq$ PERSON[*id*]

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Which of the following RAs are correct?

- $\pi_{\text{ROLE.id, first\_name, last\_name}}(\sigma_{(production\_year=1995)\wedge(\text{ROLE}.id=\text{PERSON}.id)}(\text{ROLE} \times \text{PERSON}))$
- $\pi_{\text{ROLE.id, first\_name, last\_name}}(\sigma_{production\_year=1995}(\text{ROLE} \bowtie_{\text{ROLE}.id=\text{PERSON}.id} \text{PERSON}))$
- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{production\_year=1995}(\text{ROLE} \bowtie \text{PERSON}))$
- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{production\_year=1995}(\text{MOVIE} \bowtie \text{ROLE} \bowtie \text{PERSON}))$

  All the above RAs are correct. The last RA is also correct although the natural join of MOVIE is not needed.

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Which about the following RAs?

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{(\textit{production\_year}=1995)\wedge(\text{ROLE}.\textit{id}=\text{PERSON}.\textit{id})}(\text{ROLE} \times \text{PERSON}))$

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Which about the following RAs?

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{(production\_year=1995)\wedge(\text{ROLE}.id=\text{PERSON}.id)}(\text{ROLE} \times \text{PERSON}))$

  We need to specify id (from ROLE or PERSON) under π

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Which about the following RAs?

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{(\textit{production\_year}=1995)\wedge(\text{ROLE}.\textit{id}=\text{PERSON}.\textit{id})}(\text{ROLE} \times \text{PERSON}))$

  We need to specify id (from ROLE or PERSON) under π

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{\textit{production\_year}=1995}(\text{ROLE} \bowtie_{\text{ROLE}.\textit{id}=\text{PERSON}.\textit{id}} \text{PERSON}))$

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Which about the following RAs?

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{(production\_year=1995)\wedge(\text{ROLE}.id=\text{PERSON}.id)}(\text{ROLE} \times \text{PERSON}))$

  We need to specify id (from ROLE or PERSON) under π

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{production\_year=1995}(\text{ROLE} \bowtie_{\text{ROLE}.id=\text{PERSON}.id} \text{PERSON}))$

  We need to specify id (from ROLE or PERSON) under π

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Which about the following RAs?

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{(production\_year=1995)\wedge(\text{ROLE}.id=\text{PERSON}.id)}(\text{ROLE} \times \text{PERSON}))$

  We need to specify id (from ROLE or PERSON) under π

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{production\_year=1995}(\text{ROLE} \bowtie_{\text{ROLE}.id=\text{PERSON}.id} \text{PERSON}))$

  We need to specify id (from ROLE or PERSON) under π

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{production\_year=1995}(\text{ROLE} \bowtie \text{PERSON}))$

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Which about the following RAs?

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{(\mathit{production\_year}=1995)\wedge(\text{ROLE}.\mathit{id}=\text{PERSON}.\mathit{id})}(\text{ROLE} \times \text{PERSON}))$

  We need to specify id (from ROLE or PERSON) under π

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{\mathit{production\_year}=1995}(\text{ROLE} \bowtie_{\text{ROLE}.\mathit{id}=\text{PERSON}.\mathit{id}} \text{PERSON}))$

  We need to specify id (from ROLE or PERSON) under π

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{\mathit{production\_year}=1995}(\text{ROLE} \bowtie \text{PERSON}))$

  There is no need to specify id under π

Relational Algebra (RA) – example

List the ids, first names, and last names of the persons who played at least one role in the movies produced in 1995.

Which about the following RAs?

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{(production\_year=1995)\wedge(\text{ROLE}.id=\text{PERSON}.id)}(\text{ROLE} \times \text{PERSON}))$

  We need to specify id (from ROLE or PERSON) under π

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{production\_year=1995}(\text{ROLE} \bowtie_{\text{ROLE}.id=\text{PERSON}.id} \text{PERSON}))$

  We need to specify id (from ROLE or PERSON) under π

- $\pi_{\text{id, first\_name, last\_name}}(\sigma_{production\_year=1995}(\text{ROLE} \bowtie \text{PERSON}))$

  There is no need to specify id under π

- Note the difference between Cartesian Product, Inner Join and Natural Join.

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which relation schema(s) will be used?

- $\text{MOVIE}(title, production\_year, country, run\_time, major\_genre)$
  `primary key` : $\{title, production\_year\}$
- $\text{DIRECTOR}(id, title, production\_year)$
  `primary key` : $\{title, production\_year\}$
  `foreign keys` : $[title, production\_year] \subseteq \text{MOVIE}[title, production\_year]$
  $[id] \subseteq \text{PERSON}[id]$
- $\text{WRITER}(id, title, production\_year, credits)$
  `primary key` : $\{id, title, production\_year\}$
  `foreign keys` : $[title, production\_year] \subseteq \text{MOVIE}[title, production\_year]$
  $[id] \subseteq \text{PERSON}[id]$

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which of the following RAs are correct?

- $\pi_{\text{DIRECTOR.id}}(\sigma_{(\text{DIRECTOR}.\textit{id}=\text{WRITER}.\textit{id})\wedge(\text{DIRECTOR}.\textit{title}=\text{WRITER}.\textit{title})\wedge(\text{DIRECTOR}.\textit{production\_year}=\text{WRITER}.\textit{production\_year})}(\text{DIRECTOR} \times \text{WRITER}))$
- $\pi_{\text{DIRECTOR.id}}(\text{DIRECTOR} \bowtie_{(\text{DIRECTOR}.\textit{id}=\text{WRITER}.\textit{id})\wedge(\text{DIRECTOR}.\textit{title}=\text{WRITER}.\textit{title})\wedge(\text{DIRECTOR}.\textit{production\_year}=\text{WRITER}.\textit{production\_year})} \text{WRITER})$
- $\pi_{\text{id}}(\text{DIRECTOR} \bowtie \text{WRITER})$

All the above RAs are correct.

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which about the following RAs?

- $\pi_{\text{DIRECTOR.id}}(\sigma_{(\text{DIRECTOR}.id=\text{WRITER}.id)\wedge(\text{DIRECTOR}.title=\text{WRITER}.title)}(\text{DIRECTOR} \times \text{WRITER}))$

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which about the following RAs?

- $\pi_{\text{DIRECTOR.id}}(\sigma_{(\text{DIRECTOR}.id=\text{WRITER}.id)\wedge(\text{DIRECTOR}.title=\text{WRITER}.title)}(\text{DIRECTOR} \times \text{WRITER}))$

  We need to compare *production_year*

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which about the following RAs?

- $\pi_{\text{DIRECTOR.id}}(\sigma_{(\text{DIRECTOR}.id=\text{WRITER}.id)\wedge(\text{DIRECTOR}.title=\text{WRITER}.title)}(\text{DIRECTOR} \times \text{WRITER}))$

  We need to compare *production_year*

- $\pi_{\text{DIRECTOR.id}}(\sigma_{\text{DIRECTOR}.id=\text{WRITER}.id}(\text{DIRECTOR} \times \text{WRITER}))$

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which about the following RAs?

- $\pi_{\text{DIRECTOR.id}}(\sigma_{(\text{DIRECTOR}.id=\text{WRITER}.id)\wedge(\text{DIRECTOR}.title=\text{WRITER}.title)}(\text{DIRECTOR} \times \text{WRITER}))$

  We need to compare *production_year*

- $\pi_{\text{DIRECTOR.id}}(\sigma_{\text{DIRECTOR}.id=\text{WRITER}.id}(\text{DIRECTOR} \times \text{WRITER}))$

  This query lists ids of the directors who have written at least one movie.

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which about the following RAs?

- $\pi_{\text{DIRECTOR.id}}(\sigma_{(\text{DIRECTOR}.id=\text{WRITER}.id)\wedge(\text{DIRECTOR}.title=\text{WRITER}.title)}(\text{DIRECTOR} \times \text{WRITER}))$

  We need to compare *production_year*

- $\pi_{\text{DIRECTOR.id}}(\sigma_{\text{DIRECTOR}.id=\text{WRITER}.id}(\text{DIRECTOR} \times \text{WRITER}))$

  This query lists ids of the directors who have written at least one movie.

- $\pi_{\text{id}}(\text{DIRECTOR}) \cap \pi_{\text{id}}(\text{WRITER})$

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which about the following RAs?

- $\pi_{\text{DIRECTOR.id}}(\sigma_{(\text{DIRECTOR}.id=\text{WRITER}.id)\wedge(\text{DIRECTOR}.title=\text{WRITER}.title)}(\text{DIRECTOR} \times \text{WRITER}))$

  We need to compare *production_year*

- $\pi_{\text{DIRECTOR.id}}(\sigma_{\text{DIRECTOR}.id=\text{WRITER}.id}(\text{DIRECTOR} \times \text{WRITER}))$

  This query lists ids of the directors who have written at least one movie.

- $\pi_{\text{id}}(\text{DIRECTOR}) \cap \pi_{\text{id}}(\text{WRITER})$

  This query lists ids of the directors who have written at least one movie.

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which about the following RAs?

- $\pi_{\text{DIRECTOR.id}}(\sigma_{(\text{DIRECTOR}.id=\text{WRITER}.id)\wedge(\text{DIRECTOR}.title=\text{WRITER}.title)}(\text{DIRECTOR} \times \text{WRITER}))$

  We need to compare *production_year*

- $\pi_{\text{DIRECTOR.id}}(\sigma_{\text{DIRECTOR}.id=\text{WRITER}.id}(\text{DIRECTOR} \times \text{WRITER}))$

  This query lists ids of the directors who have written at least one movie.

- $\pi_{\text{id}}(\text{DIRECTOR}) \cap \pi_{\text{id}}(\text{WRITER})$

  This query lists ids of the directors who have written at least one movie.

- $\pi_{\text{id}}(\pi_{\text{id,title,production\_year}}(\text{DIRECTOR}) \cap \pi_{\text{id,title,production\_year}}(\text{WRITER}))$

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which about the following RAs?

- $\pi_{\text{DIRECTOR.id}}(\sigma_{(\text{DIRECTOR}.\mathit{id}=\text{WRITER}.\mathit{id})\wedge(\text{DIRECTOR}.\mathit{title}=\text{WRITER}.\mathit{title})}$ $(\text{DIRECTOR} \times \text{WRITER}))$

  We need to compare *production_year*

- $\pi_{\text{DIRECTOR.id}}(\sigma_{\text{DIRECTOR}.\mathit{id}=\text{WRITER}.\mathit{id}}(\text{DIRECTOR} \times \text{WRITER}))$

  This query lists ids of the directors who have written at least one movie.

- $\pi_{\text{id}}(\text{DIRECTOR}) \cap \pi_{\text{id}}(\text{WRITER})$

  This query lists ids of the directors who have written at least one movie.

- $\pi_{\text{id}}(\pi_{\text{id,title,production\_year}}(\text{DIRECTOR}) \cap \pi_{\text{id,title,production\_year}}(\text{WRITER}))$

  Correct.

Relational Algebra (RA) – example

List the ids of the directors who have directed at least one movie written by themselves.

Which about the following RAs?

- $\pi_{\text{DIRECTOR.id}}(\sigma_{(\text{DIRECTOR}.id=\text{WRITER}.id)\wedge(\text{DIRECTOR}.title=\text{WRITER}.title)}(\text{DIRECTOR} \times \text{WRITER}))$

  We need to compare *production_year*

- $\pi_{\text{DIRECTOR.id}}(\sigma_{\text{DIRECTOR}.id=\text{WRITER}.id}(\text{DIRECTOR} \times \text{WRITER}))$

  This query lists ids of the directors who have written at least one movie.

- $\pi_{\text{id}}(\text{DIRECTOR}) \cap \pi_{\text{id}}(\text{WRITER})$

  This query lists ids of the directors who have written at least one movie.

- $\pi_{\text{id}}(\pi_{\text{id,title,production\_year}}(\text{DIRECTOR}) \cap \pi_{\text{id,title,production\_year}}(\text{WRITER}))$

  Correct.

Relational Algebra (RA) – example

List the ids of the directors who have never played any roles in the movies directed by themselves.

- List ids of all directors.

Relational Algebra (RA) – example

List the ids of the directors who have never played any roles in the movies directed by themselves.

- List ids of all directors.
  $D_1 = \pi_{\text{id}}(\text{DIRECTOR})$

Relational Algebra (RA) – example

List the ids of the directors who have never played any roles in the movies directed by themselves.

- List ids of all directors.
  $D_1 = \pi_{\text{id}}(\text{DIRECTOR})$
- List ids of director who have played at least one role in the movies directed by themselves.

Relational Algebra (RA) – example

List the ids of the directors who have never played any roles in the movies directed by themselves.

- List ids of all directors.
  $D_1 = \pi_{\text{id}}(\text{DIRECTOR})$
- List ids of director who have played at least one role in the movies directed by themselves.
  $D_2 = \pi_{\text{id}}(\text{DIRECTOR} \bowtie \text{ROLE})$

Relational Algebra (RA) – example

List the ids of the directors who have never played any roles in the movies directed by themselves.

- List ids of all directors.
  $D_1 = \pi_{\text{id}}(\text{DIRECTOR})$
- List ids of director who have played at least one role in the movies directed by themselves.
  $D_2 = \pi_{\text{id}}(\text{DIRECTOR} \bowtie \text{ROLE})$
- List the ids of the directors who have never played any roles in the movies directed by themselves.

Relational Algebra (RA) – example

List the ids of the directors who have never played any roles in the movies directed by themselves.

- List ids of all directors.
  $D_1 = \pi_{\text{id}}(\text{DIRECTOR})$
- List ids of director who have played at least one role in the movies directed by themselves.
  $D_2 = \pi_{\text{id}}(\text{DIRECTOR} \bowtie \text{ROLE})$
- List the ids of the directors who have never played any roles in the movies directed by themselves.
  $Result = D_1 - D_2$.

Relational Algebra (RA)

- Relational algebra is a query language with **RA operators**:

Relational Algebra (RA)

- Relational algebra is a query language with **RA operators**:

σ selection
π projection
ρ renaming

$\Uparrow$
Unary operator

Relational Algebra (RA)

- Relational algebra is a query language with **RA operators**:

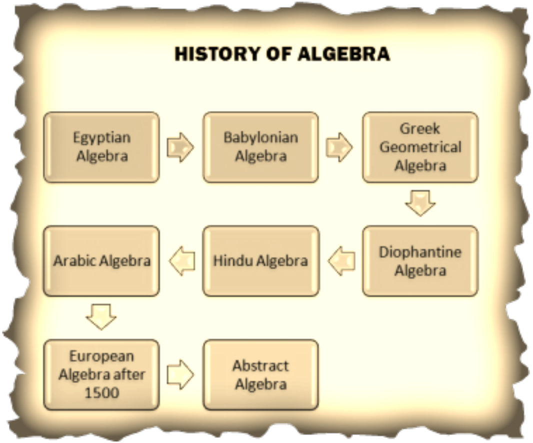
Unary operator	Binary operator (Type compatible)
σ selection	$\cup$ union
π projection	$\cap$ intersection
ρ renaming	$-$ difference

Relational Algebra (RA)

- Relational algebra is a query language with **RA operators**:

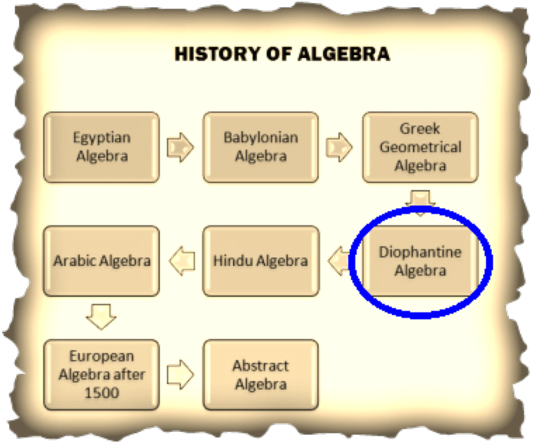
Unary operator	Binary operator (Type compatible)	Binary operator
σ selection π projection ρ renaming	$\cup$ union $\cap$ intersection $-$ difference	$\times$ cartesian product $\bowtie$ natural Join $\bowtie_{\phi}$ Inner Join

(credit cookie) History of Algebra



http://historyofpythagoreantheorem.weebly.com/algebra.html

(credit cookie) History of Algebra



http://historyofpythagoreantheorem.weebly.com/algebra.html

(credit cookie) Diophantus of Alexandria

‘Here lies Diophantus’, the wonder behold.
Through art algebraic, the stone tells how old:
‘God gave him his boyhood one-sixth of his life,
One twelfth more as youth while whiskers grew rife;
And then yet one-seventh ere marriage begun;
In five years there came a bouncing new son.
Alas, the dear child of master and sage
After attaining half the measure of his father’s life chill fate took him.
After consoling his fate by the science of numbers for four years,
he ended his life’.

(credit cookie) Diophantus of Alexandria

‘Here lies Diophantus’, the wonder behold.
Through art algebraic, the stone tells how old:
‘God gave him his boyhood one-sixth of his life,
One twelfth more as youth while whiskers grew rife;
And then yet one-seventh ere marriage begun;
In five years there came a bouncing new son.
Alas, the dear child of master and sage
After attaining half the measure of his father’s life chill fate took him.
After consoling his fate by the science of numbers for four years,
he ended his life’.

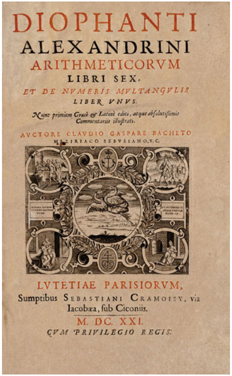
$x = x/6 + x/12 + x/7 + 5 + x/2 + 4$

(credit cookie) Diophantus of Alexandria

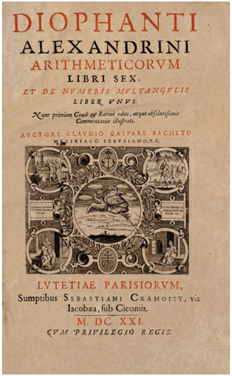
'Here lies Diophantus', the wonder behold.
Through art algebraic, the stone tells how old:
'God gave him his boyhood one-sixth of his life,
One twelfth more as youth while whiskers grew rife;
And then yet one-seventh ere marriage begun;
In five years there came a bouncing new son.
Alas, the dear child of master and sage
After attaining half the measure of his father's life chill fate took him.
After consoling his fate by the science of numbers for four years,
he ended his life'.

$$x = x/6 + x/12 + x/7 + 5 + x/2 + 4 \Rightarrow x = 84$$

(credit cookie) Arithmetica and Margin-writing by Fermat

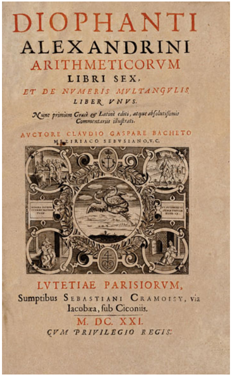


(credit cookie) Arithmetica and Margin-writing by Fermat



"If an integer n is greater than 2, then $a^n + b^n = c^n$ has no solutions in non-zero integers a, b, and c. I have a truly marvelous proof of this proposition which this margin is too narrow to contain."
—Pierre de Fermat (1607-1665)

(credit cookie) Arithmetica and Margin-writing by Fermat



DIOPHANTI
ALEXANDRINI
ARITHMETICORVM
LIBRI SEX,
ET DE NVMERIS MVLTANGVLIS
LIBER VNVS.
Nunc primùm Græcè & Latinè editi, atque absolutissimis
Commentariis illustrati.
AVCTORE CLAVDIO GASPARE BACHETO
MEZIRIACO SEBVSIANO, V.C.

LVTETIAE PARISIORVM,
Sumptibus SEBASTIANI CRAMOISY, via
Iacobæa, sub Ciconiis.
M. DC. XXI.
CVM PRIVILEGIO REGIS.

"If an integer n is greater than 2, then $a^n + b^n = c^n$ has no solutions in non-zero integers a, b, and c. I have a truly marvelous proof of this proposition which this margin is too narrow to contain."
—Pierre de Fermat (1607-1665)

Fermat's Last Theorem was proved by Andrew Wiles in 1994.