

Databases: Relational Algebra

Students

Last	First
Potter	Harry
Granger	Hermione
Weasley	Ron
Longbottom	Neville
Malfoy	Draco

Professors

Last	First
McGonagall	Minerva
Snape	Severus
Longbottom	Neville
Dumbledore	Albus

Gryffindors

Last	First
Potter	Harry
Granger	Hermione
Weasley	Ron
McGonagall	Minerva
Longbottom	Neville
Dumbledore	Albus

Name	ID	Major	Age
Alice	1	CS	18
Bob	2	Math	20
Carol	3	CS	19
Dan	4	CS	20
Eva	5	Math	21
Frank	6	Physics	18

Students

Key = ID

CRN	Dept	CourseName	Seats
101	CS	Databases	20
102	CS	Discrete Structures	15
103	CS	Graphics	25
104	Math	Linear Algebra	18
105	Math	Differential Equations	20
106	Music	Piano Lessons	10
107	Physics	Optics	16
108	Music	Music Theory	21
109	Physics	Modern Physics	15
110	Math	Number Theory	20

Courses

Key = CRN

Enrolled

Key = (ID, CRN)

ID	CRN
1	101
1	102
2	104
2	105
3	101
3	104
4	103
5	108
5	105
5	110
6	107
6	110
6	106

Cartesian Product

<i>A</i>	<i>B</i>
1	2
3	4

(a) Relation R

<i>B</i>	<i>C</i>	<i>D</i>
2	5	6
4	7	8
9	10	11

(b) Relation S

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

(c) Result $R \times S$

Natural Join

<i>A</i>	<i>B</i>	<i>C</i>
1	2	3
6	7	8
9	7	8

(a) Relation U

<i>B</i>	<i>C</i>	<i>D</i>
2	3	4
2	3	5
7	8	10

(b) Relation V

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
1	2	3	4
1	2	3	5
6	7	8	10
9	7	8	10

(c) Result $U \bowtie V$

(Natural) joins can be incomplete

- If a tuple from one relation doesn't have a "counterpart" in the other relation, it doesn't contribute to the join ("dangling" tuple):

r_1

Employee	Department
Smith	sales
Black	production
White	production

r_2

Department	Head
production	Mori
purchasing	Brown

$r_1 \bowtie r_2$

Employee	Department	Head
Black	production	Mori
White	production	Mori

(Natural) joins can be empty!

- If no tuple has a counterpart, then the resulting relation is empty.

r_1

Employee	Department
Smith	sales
Black	production
White	production

r_2

Department	Head
marketing	Mori
purchasing	Brown

$r_1 \bowtie r_2$

Employee	Department	Head

New Database!

Database schema:

Person(name, age, school)

name is a key

Frequents(name, pizzeria)

(name, pizzeria) is a key

Eats(name, pizza)

(name, pizza) is a key

Serves(pizzeria, pizza, price)

(pizzeria, price) is a key

school is either “Rhodes” or “U of M”

pizzeria is the name of a pizza restaurant (e.g., “Memphis Pizza Café,” “Aldo’s” etc)

pizza is a type of pizza (e.g., “pepperoni,” “cheese,” “pineapple,” ...)