SQL: simple joins

CSUMB

Lecture Objectives

When the lecture is over, students should be able to:

- write simple join statements
- explain the precise meaning of join statements

SQL Review

We have been looking at SQL queries like:

```
select count(*)
  from patients
 where sex = "F";
```

How many tables can go in the “from” part?

Reminder: ‘courses’ tables

the main tables are:

- `instructor (name, dept, salary)`
- `course (title, dept, credits)`
- `dept (name, building, budget)`
- `section (course, section, semester, year, building,…)`
- `teaches (instructor, course, section, semester, year)`

Who are the instructors of each course?

instructor

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

teaches

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-101	1	Spring	2010
15151	MUS-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010

Who are the instructors of each course?

instructor

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

teaches

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-101	1	Spring	2010
15151	MUS-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010


```
select name, course_id
  from instructor, teaches
where instructor.ID = teaches.ID
```



```
Srinivasan,CS-101
Srinivasan,CS-315
Srinivasan,CS-347
Wu,FIN-101
Mozart,MUS-199
Einstein,PHY-101
El Said,HIS-351
Katz,CS-101
Katz,CS-319
Crick,BIO-101
Crick,BIO-301
Brandt,CS-190
Brandt,CS-190
Brandt,CS-319
Kim,EE-181
```

Another way to write the query

```
select name, course_id
  from instructor, teaches
where instructor.ID = teaches.ID
```

the way we just used

```
select name, course_id
  from instructor, teaches
on instructor.ID = teaches.ID
```

using ‘on’ instead of ‘where’

For these kinds of joins, the meaning is the same.

Using ‘on’ is usually preferred for clarity.

Exercise: what is this table operation?

a	b
1	2
3	4
5	6

c	d	e
7	8	9
10	11	12

a	b	c	d	e
1	2	7	8	9
1	2	10	11	12
3	4	7	8	9
3	4	10	11	12
5	6	7	8	9

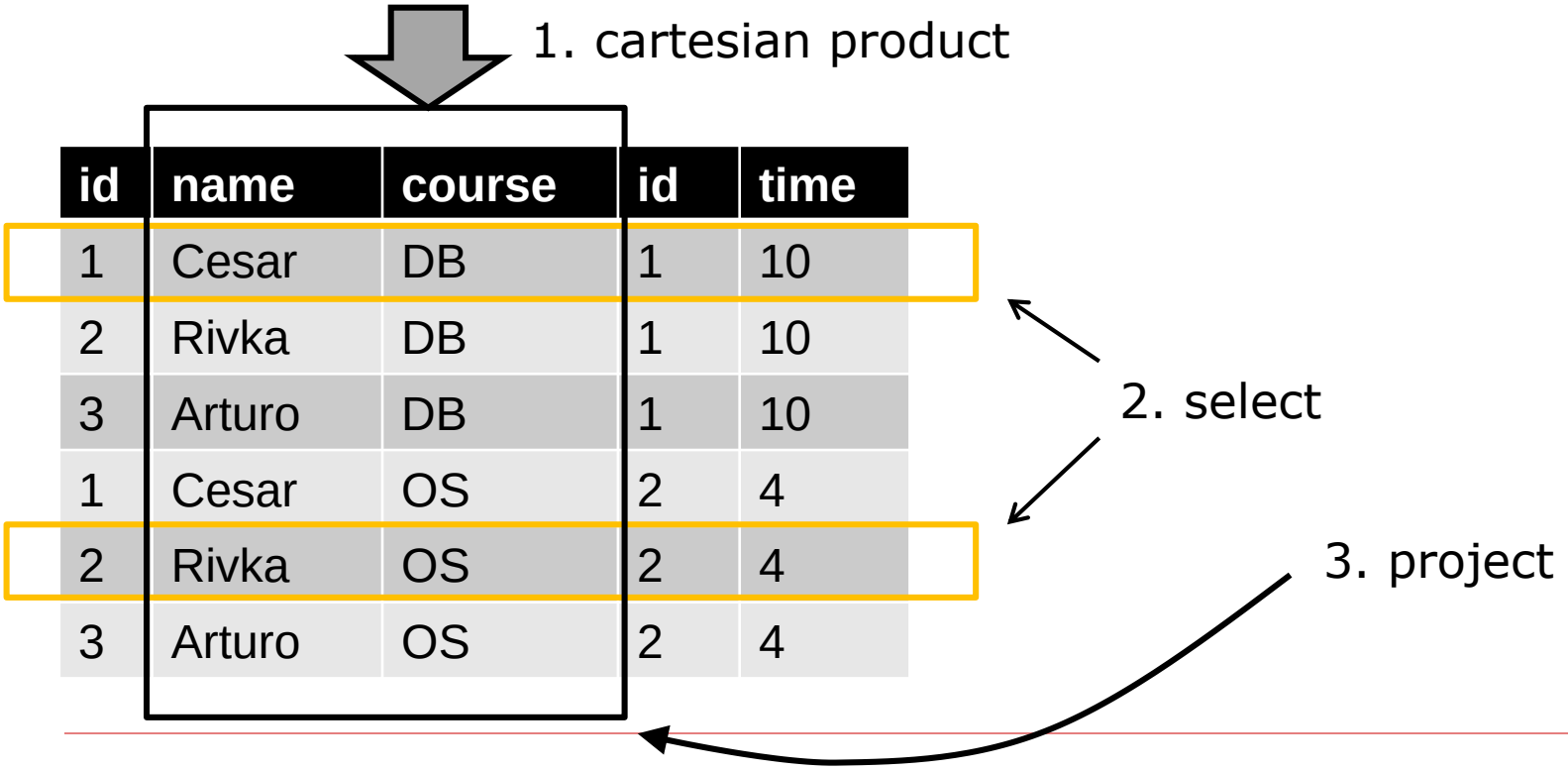
what is the last row?

SQL join

id	name
1	Cesar
2	Rivka
3	Arturo

course	id	time
DB	1	10
OS	2	4

```
select name, course
from instructor, course
where instructor.id = course.id
```



id	name	course	id	time
1	Cesar	DB	1	10
2	Rivka	DB	1	10
3	Arturo	DB	1	10
1	Cesar	OS	2	4
2	Rivka	OS	2	4
3	Arturo	OS	2	4

Exercise: order the steps

```
select name, course_id
  from instructor, teaches
where instructor.ID = teaches.ID
```

where part: **select** rows matching this condition

select part: **project** these columns

from part: take **cartesian product** of these tables

what is the correct order of these steps?

Show courses that are offered all times

course

course_id	title	dept_name	credits
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principals	Physics	4

section

course_id	sec_id	semester	year	building	room_number	time_slot_id
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	B
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

```
select ? from course, section where course.course_id = ?;
```

```
select title from course, section where course.course_id = section.course_id;
```

Two ways to think about join

1. A way to combine information from two or more tables
   - for example: one table has instructor IDs but you need instructor names from another table
2. A way to filter rows from one table based on data in a second table
   - for example: you want only the student IDs in the student table that appear in another table

Summary

- We use join when queries involve two or more tables
- We can understand an SQL join using relational algebra:
  - take **cartesian product** of the tables in “from” part
  - **select** rows using the condition in “where” part
  - **project** using the columns in “select” part