Outer joins

Credit: Dr. Bruns

Using join with 'on'

student

ID	name	dept_name	tot_cred
128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56

takes

ID	course_id	sec_id	semester	year	grade
128	CS-101	1	Fall	2009	A
128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A

```
select *
   from student, takes
  where student.ID = takes.ID and year=2009;

select *
   from student inner join takes
  where student.ID = takes.ID and year=2009;

select *
   from student inner join takes
     on student.ID = take.ID
  where year=2009;
```

'on' modifies 'join'

these all do the same thing…

… but the last version is recommended

All the joins

This looks terribly confusing:

- inner joins
- outer joins
- natural joins
- left outer joins
- right outer joins
- natural right outer joins
- full outer joins
- … !

Don’t worry: only three concepts are needed, and we know two already!

The ‘student’ and ‘takes’ tables

student

ID	name	dept_name	tot_cred
128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

takes

ID	course_id	sec_id	semester	year	grade
128	CS-101	1	Fall	2009	A
128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B
23121	FIN-201	1	Spring	2010	C+
44553	PHY-101	1	Fall	2009	B-
45678	CS-101	1	Fall	2009	F
45678	CS-101	1	Spring	2010	B+
45678	CS-319	1	Spring	2010	B
54321	CS-101	1	Fall	2009	A-
54321	CS-190	2	Spring	2009	B+
55739	MU-199	1	Spring	2010	A-
76543	CS-101	1	Fall	2009	A
76543	CS-319	2	Spring	2010	A
76653	EE-181	1	Spring	2009	C
98765	CS-101	1	Fall	2009	C-
98765	CS-315	1	Spring	2010	B
98988	BIO-101	1	Summer	2009	A
98988	BIO-301	1	Summer	2010	

Get info about classes students take

student

ID	name	dept_name	tot_cred
128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46

takes

ID	course_id	sec_id	semester	year	grade
128	CS-101	1	Fall	2009	A
128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B

```
sqlite> select * from student natural join takes;
00128,Zhang,Comp. Sci.,102,CS-101,1,Fall,2009,A
00128,Zhang,Comp. Sci.,102,CS-347,1,Fall,2009,A-
12345,Shankar,Comp. Sci.,32,CS-101,1,Fall,2009,C
12345,Shankar,Comp. Sci.,32,CS-190,2,Spring,2009,A
12345,Shankar,Comp. Sci.,32,CS-315,1,Spring,2010,A
12345,Shankar,Comp. Sci.,32,CS-347,1,Fall,2009,A
19991,Brandt,History,80,HIS-351,1,Spring,2010,B
23121,Chavez,Finance,110,FIN-201,1,Spring,2010,C+
…
```

Exercise

student

ID	name	dept_name	tot_cred
128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46

takes

ID	course_id	sec_id	semester	year	grade
128	CS-101	1	Fall	2009	A
128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B

```
sqlite> select * from student natural join takes;
00128,Zhang,Comp. Sci.,102,CS-101,1,Fall,2009,A
00128,Zhang,Comp. Sci.,102,CS-347,1,Fall,2009,A-
12345,Shankar,Comp. Sci.,32,CS-101,1,Fall,2009,C
12345,Shankar,Comp. Sci.,32,CS-190,2,Spring,2009,A
12345,Shankar,Comp. Sci.,32,CS-315,1,Spring,2010,A
12345,Shankar,Comp. Sci.,32,CS-347,1,Fall,2009,A
19991,Brandt,History,80,HIS-351,1,Spring,2010,B
23121,Chavez,Finance,110,FIN-201,1,Spring,2010,C+
…
```

True or False?
All the students will appear in the query output.

Solution

student

ID	name	dept_name	tot_cred
128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

takes

ID	course_id	sec_id	semester	year	grade
128	CS-101	1	Fall	2009	A
128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B
23121	FIN-201	1	Spring	2010	C+
44553	PHY-101	1	Fall	2009	B-
45678	CS-101	1	Fall	2009	F
45678	CS-101	1	Spring	2010	B+
45678	CS-319	1	Spring	2010	B
54321	CS-101	1	Fall	2009	A-
54321	CS-190	2	Spring	2009	B+
55739	MU-199	1	Spring	2010	A-
76543	CS-101	1	Fall	2009	A
76543	CS-319	2	Spring	2010	A
76653	EE-181	1	Spring	2009	C
98765	CS-101	1	Fall	2009	C-
98765	CS-315	1	Spring	2010	B
98988	BIO-101	1	Summer	2009	A
98988	BIO-301	1	Summer	2010	

Snow does not appear in natural join output.

Natural "left outer" join

student

ID	name	dept_name	tot_cred
128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46

takes

ID	course_id	sec_id	semester	year	grade
128	CS-101	1	Fall	2009	A
128	CS-347	1	Fall	2009	A-
12345	CS-101	1	Fall	2009	C
12345	CS-190	2	Spring	2009	A
12345	CS-315	1	Spring	2010	A
12345	CS-347	1	Fall	2009	A
19991	HIS-351	1	Spring	2010	B

```
sqlite> select * from student left natural join takes;
00128,Zhang,Comp. Sci.,102,CS-101,1,Fall,2009,A
00128,Zhang,Comp. Sci.,102,CS-347,1,Fall,2009,A-
12345,Shankar,Comp. Sci.,32,CS-101,1,Fall,2009,C
…
45678,Levy,Physics,46,CS-101,1,Spring,2010,B+
45678,Levy,Physics,46,CS-319,1,Spring,2010,B
54321,Williams,Comp. Sci.,54,CS-101,1,Fall,2009,A-
54321,Williams,Comp. Sci.,54,CS-190,2,Spring,2009,B+
55739,Sanchez,Music,38,MU-199,1,Spring,2010,A-
70557,Snow,Physics,0,,,,,
76543,Brown,Comp. Sci.,58,CS-101,1,Fall,2009,A
…
```

left outer

To result of natural join,

- add a row for every row on left not matched by a row on the right
- each such row will have 'null' for columns on the right

Exercise: natural left outer join

student

id	name
1	Rivka
2	Cesar
3	Arturo

takes

course	id	time
DB	1	10
OS	3	4

```
select *
  from student natural left outer join takes ;
```

What is the result?

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

Exercise: natural right outer join

takes

course	id	time
DB	1	10
OS	3	4

student

id	name
1	Rivka
2	Cesar
3	Arturo

```
select *
  from takes natural right outer join student;
```

What is the result?

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

What does the natural full outer join do?

T1

a	b
A	1
B	3
C	4

T2

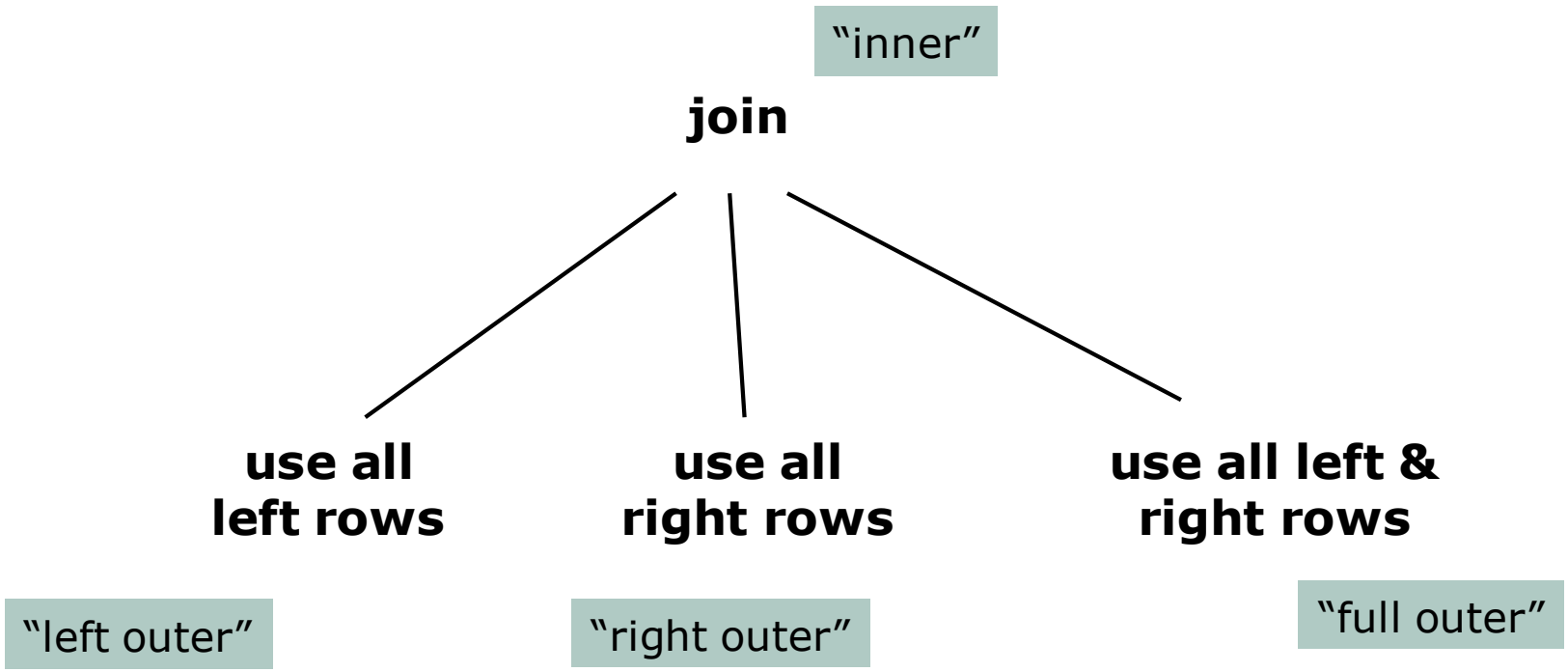
b	c
1	foo
2	bar
3	baz

```
select *
  from T1 natural full outer join T2;
```

What is the result?

a	b	c
A	1	foo
B	3	baz
C	4	null
null	2	bar

All the joins



Plus, every kind of join can be "natural"