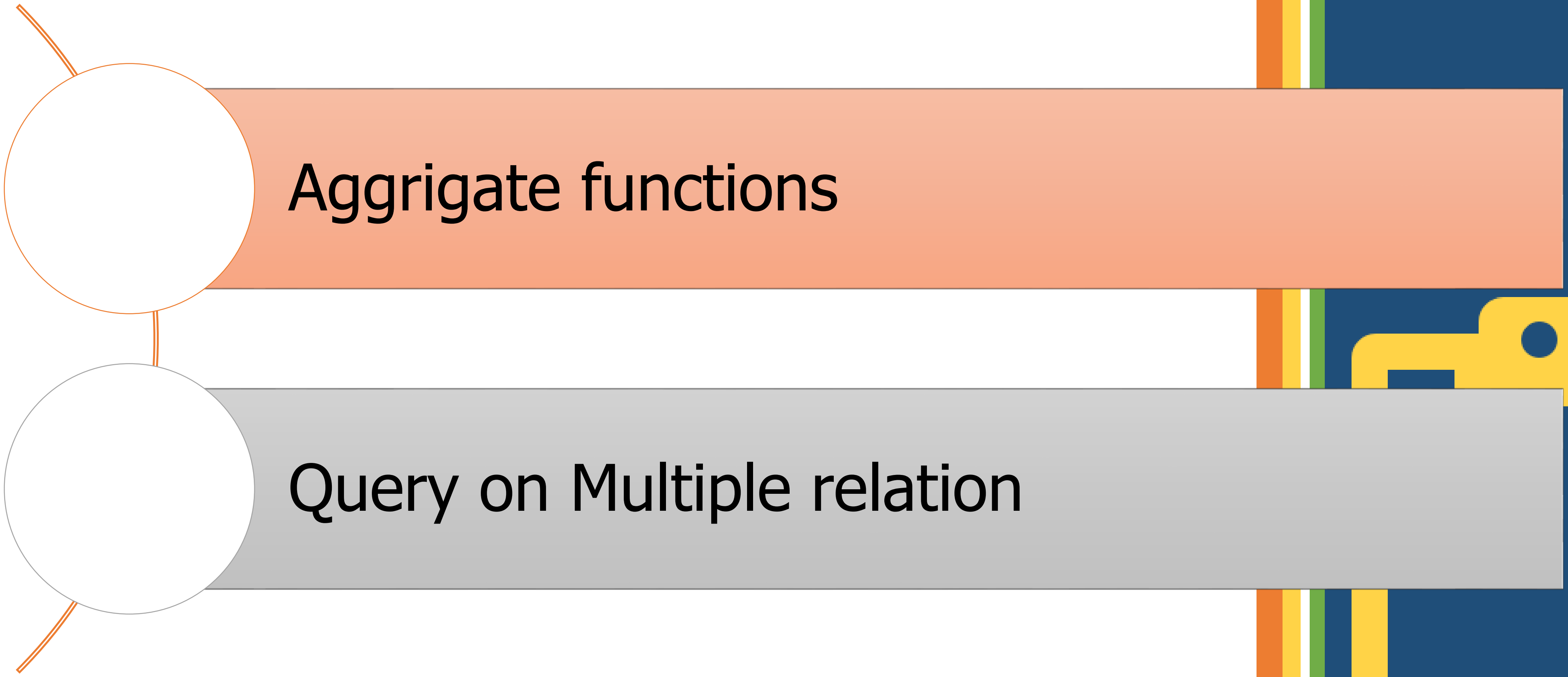
DBMS

Lab-5

Things we will complete today



Aggrigate functions

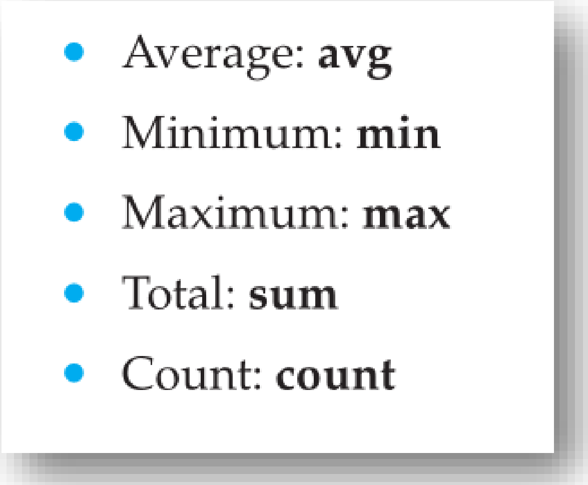
Query on Multiple relation

Aggregate Functions

Aggregate functions

Aggregate functions are functions that take a collection (a set or multiset) of values as input and **return a single value**.

SQL offers five built-in aggregate functions:

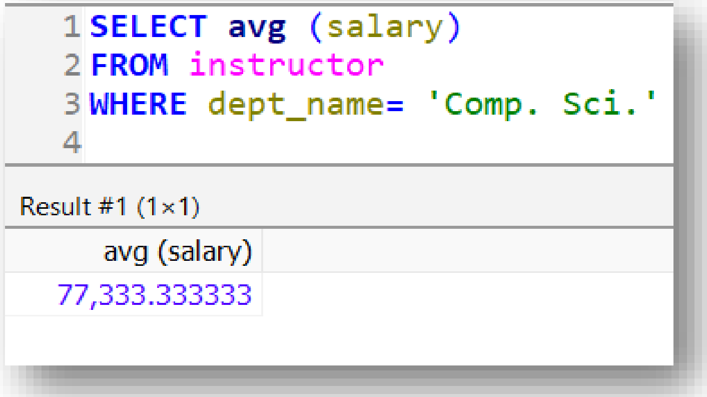


- Average: **avg**
- Minimum: **min**
- Maximum: **max**
- Total: **sum**
- Count: **count**

The input to **sum** and **avg** must be a collection of numbers, but the other operators can operate on collections of nonnumeric data types, such as strings, as well

Aggregate functions

- Find the average salary of instructors in the Computer Science department

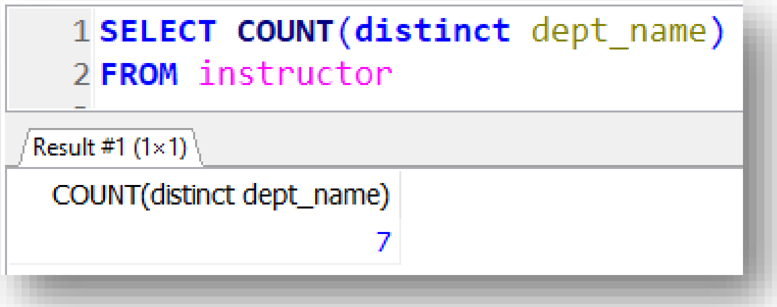


```
SELECT avg (salary)
FROM instructor
WHERE dept_name= 'Comp. Sci.'

```

```
Result #1 (1×1)
 avg (salary)
77,333.333333
```

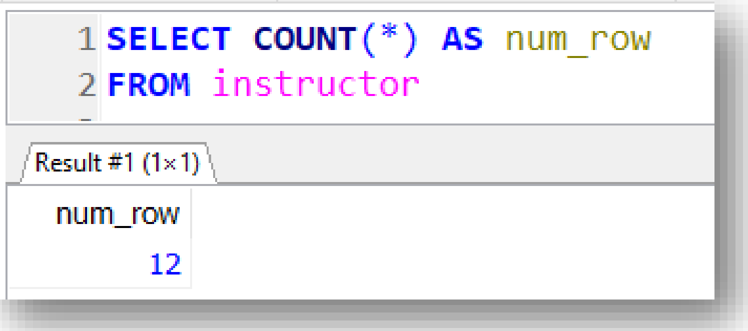
- Find the number of departments in *instructor* relation



```
SELECT COUNT(distinct dept_name)
FROM instructor
```

```
Result #1 (1×1)
COUNT(distinct dept_name)
                        7
```

- Find the number of tuples in the *instructor* relation

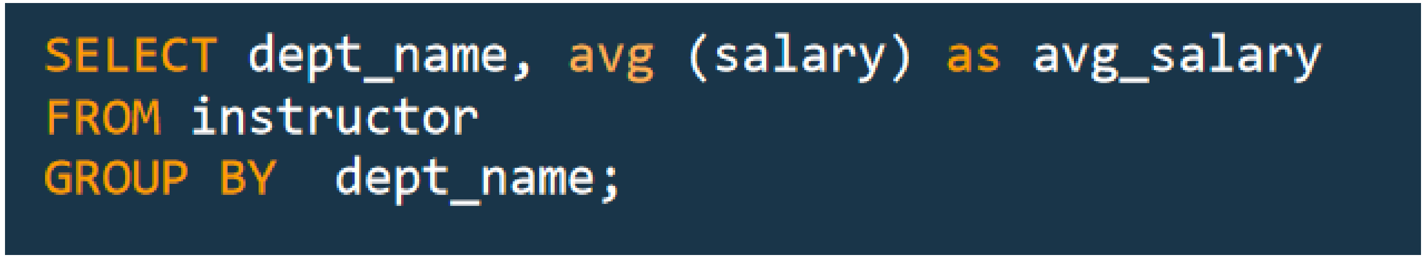


```
SELECT COUNT(*) AS num_row
FROM instructor
```

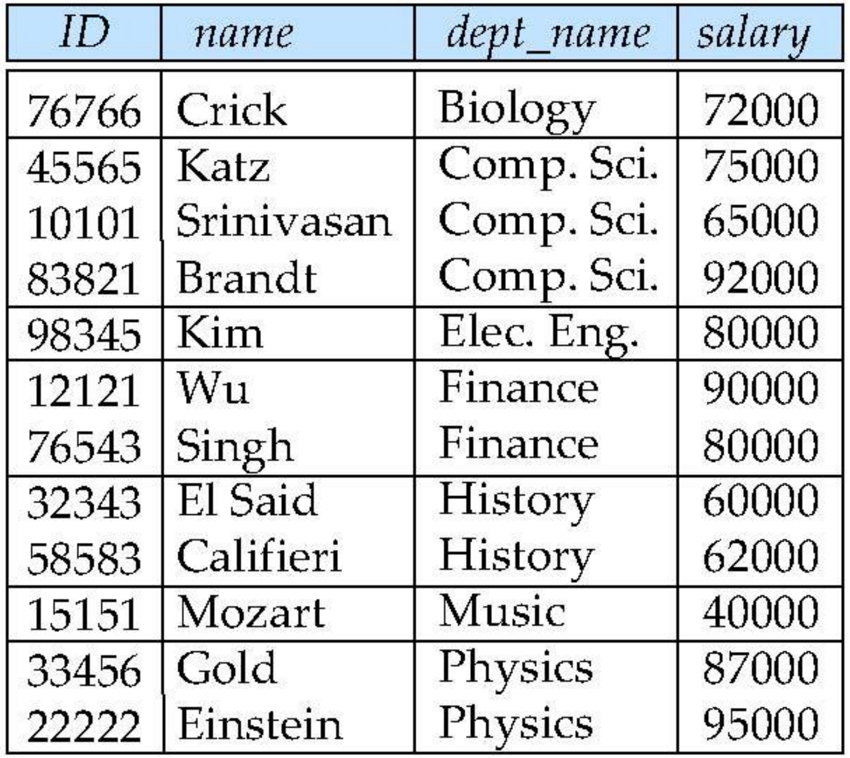
```
Result #1 (1×1)
num_row
     12
```

Aggregate functions - Group By

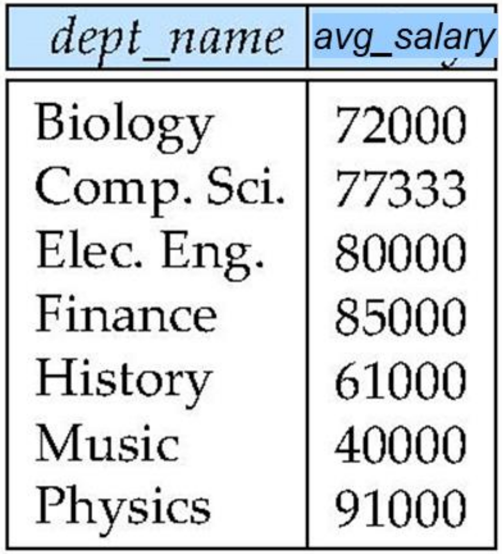
- The **GROUP BY** statement groups rows that have the same values into summary rows like "Find the average salary of instructors in each department"



```
SELECT dept_name, avg (salary) as avg_salary
FROM instructor
GROUP BY  dept_name;
```



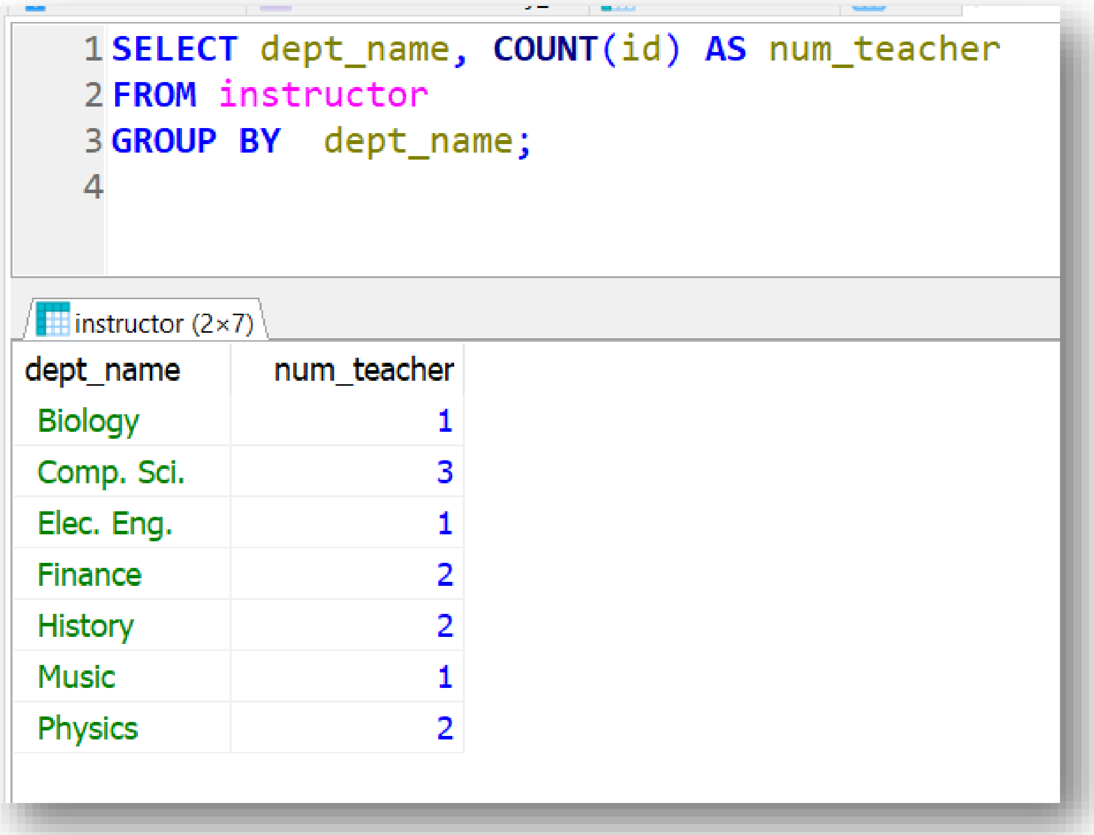
ID	*name*	*dept_name*	*salary*
76766	Crick	Biology	72000
45565	Katz	Comp. Sci.	75000
10101	Srinivasan	Comp. Sci.	65000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000
12121	Wu	Finance	90000
76543	Singh	Finance	80000
32343	El Said	History	60000
58583	Califieri	History	62000
15151	Mozart	Music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



dept_name	avg_salary
Biology	72000
Comp. Sci.	77333
Elec. Eng.	80000
Finance	85000
History	61000
Music	40000
Physics	91000

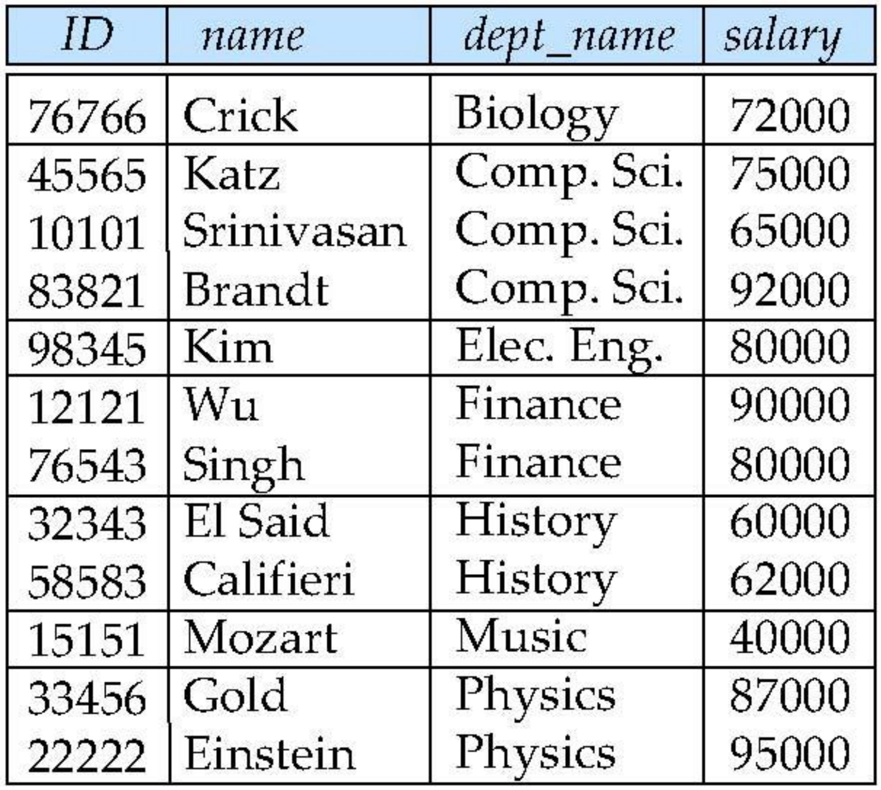
Aggregate functions - Group By

- "Find the number of of instructors in each department"



```sql
SELECT dept_name, COUNT(id) AS num_teacher
FROM instructor
GROUP BY  dept_name;
```

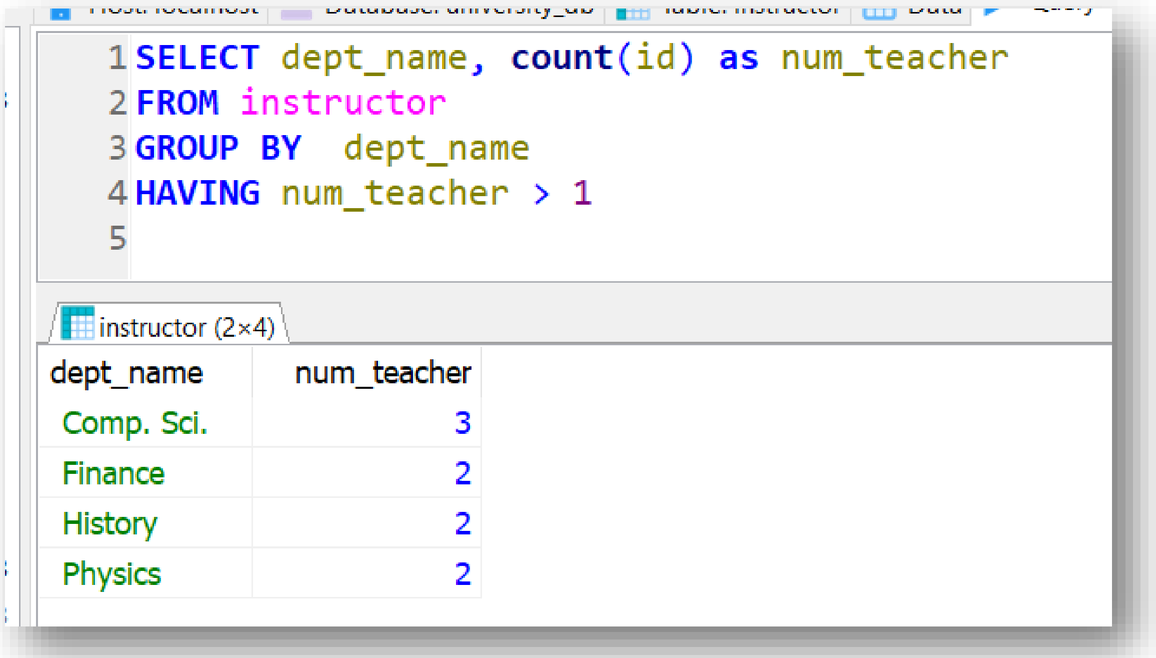
dept_name	num_teacher
Biology	1
Comp. Sci.	3
Elec. Eng.	1
Finance	2
History	2
Music	1
Physics	2



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

Aggregate Functions – Having Clause

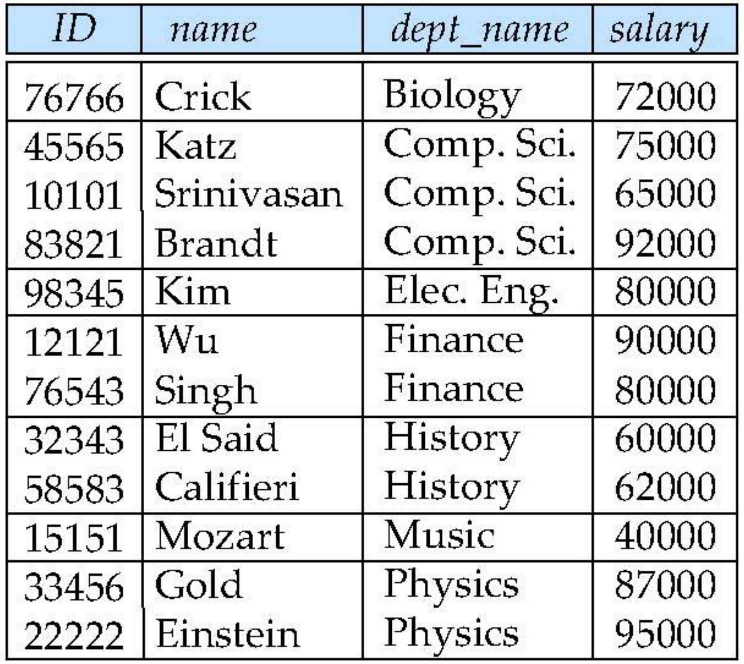
- “Find the number of of instructors in each department which **have more than one instructors**”



```sql
SELECT dept_name, count(id) as num_teacher
FROM instructor
GROUP BY  dept_name
HAVING num_teacher > 1

```

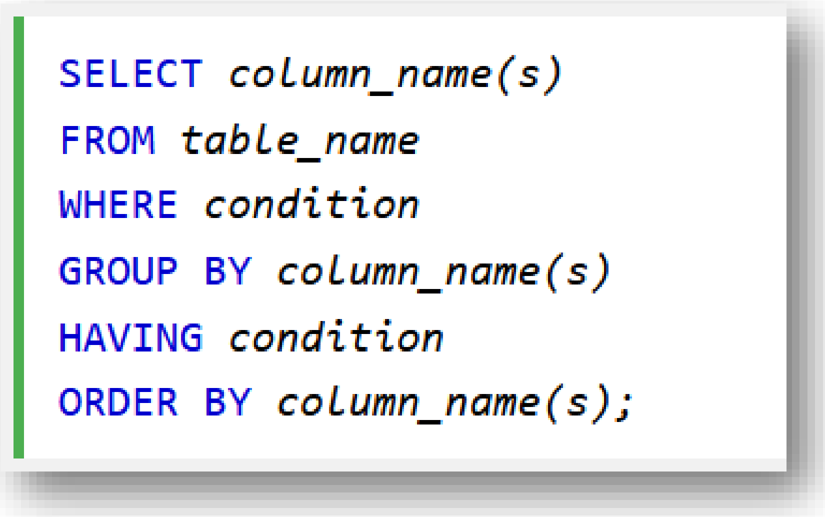
dept_name	num_teacher
Comp. Sci.	3
Finance	2
History	2
Physics	2



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

Aggregate Functions – Having Clause

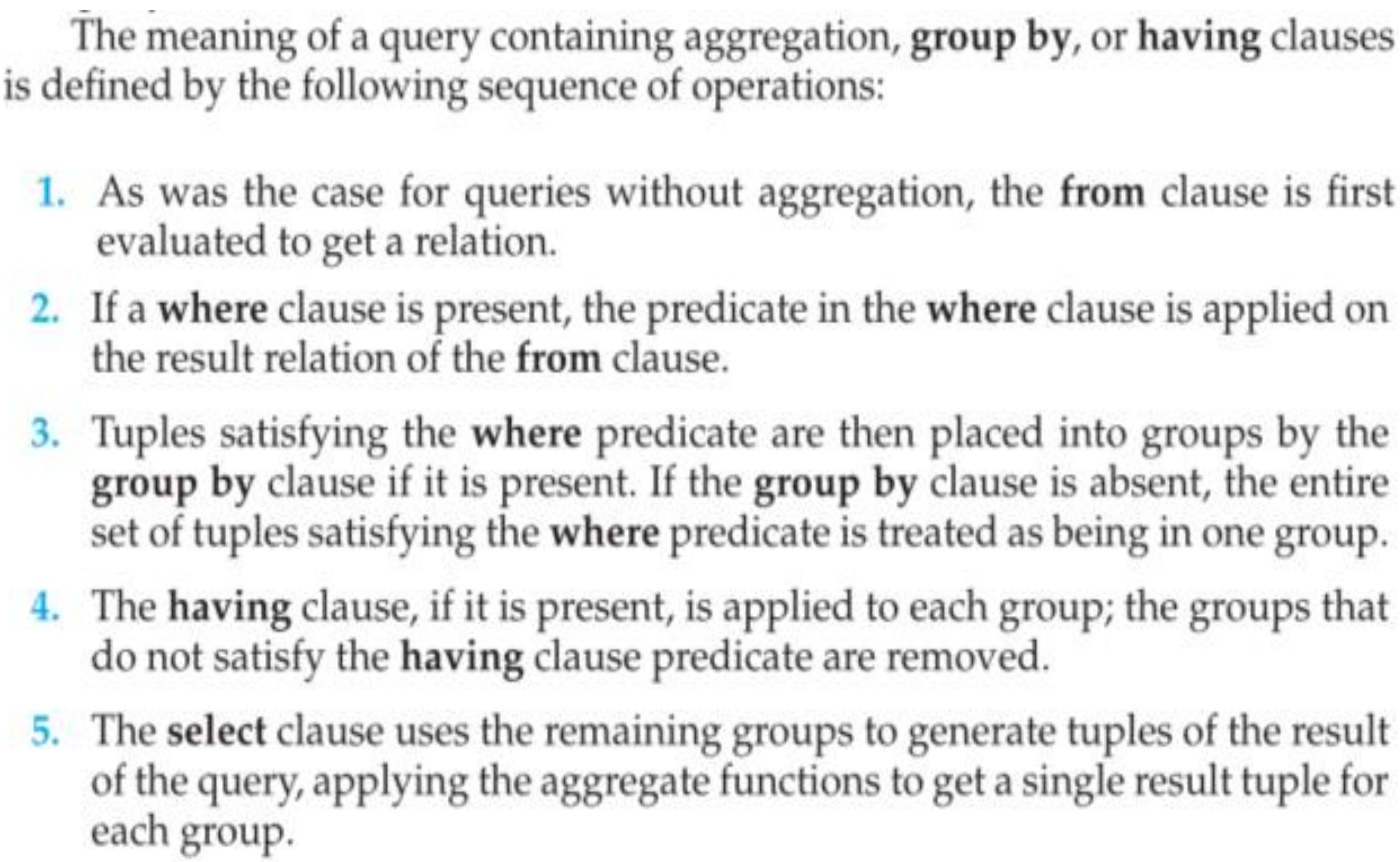
- *The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.*



```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```

Note: predicates in the **having** clause are applied after the formation of groups whereas predicates in the **where** clause are applied before forming groups

Aggregate Functions – Having Clause



The meaning of a query containing aggregation, **group by**, or **having** clauses is defined by the following sequence of operations:

1. As was the case for queries without aggregation, the **from** clause is first evaluated to get a relation.
2. If a **where** clause is present, the predicate in the **where** clause is applied on the result relation of the **from** clause.
3. Tuples satisfying the **where** predicate are then placed into groups by the **group by** clause if it is present. If the **group by** clause is absent, the entire set of tuples satisfying the **where** predicate is treated as being in one group.
4. The **having** clause, if it is present, is applied to each group; the groups that do not satisfy the **having** clause predicate are removed.
5. The **select** clause uses the remaining groups to generate tuples of the result of the query, applying the aggregate functions to get a single result tuple for each group.

Page 88

Null Values and Aggregates

- Total all salaries

```
select sum (salary )
from instructor
```

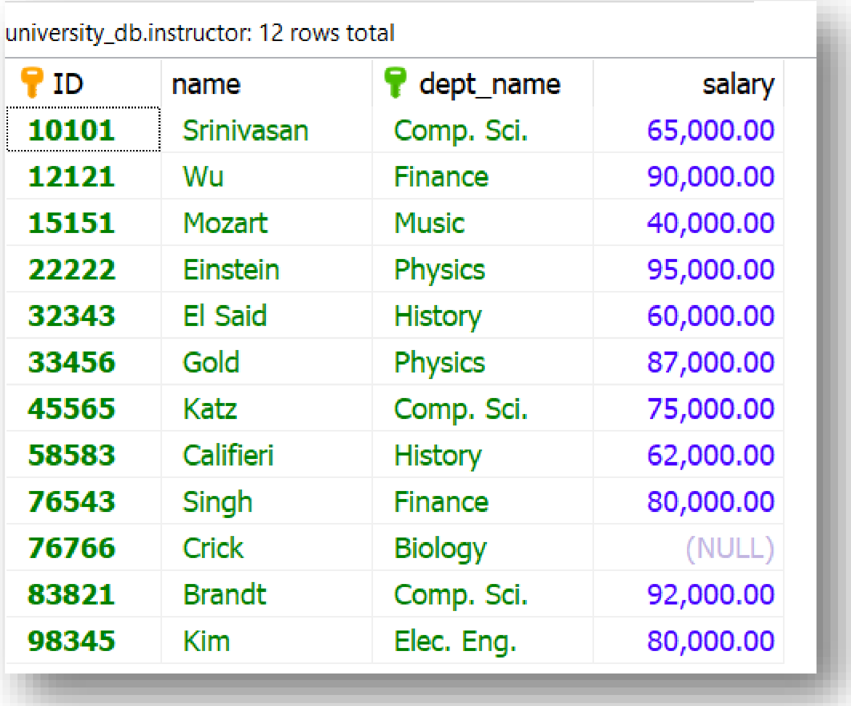
  - Above statement ignores null amounts
  - Result is *null* if there is no non-null amount

- All aggregate operations except **count(*)** ignore tuples with null values on the aggregated attributes

- What if collection has only null values?
  - count returns 0
  - all other aggregates return null

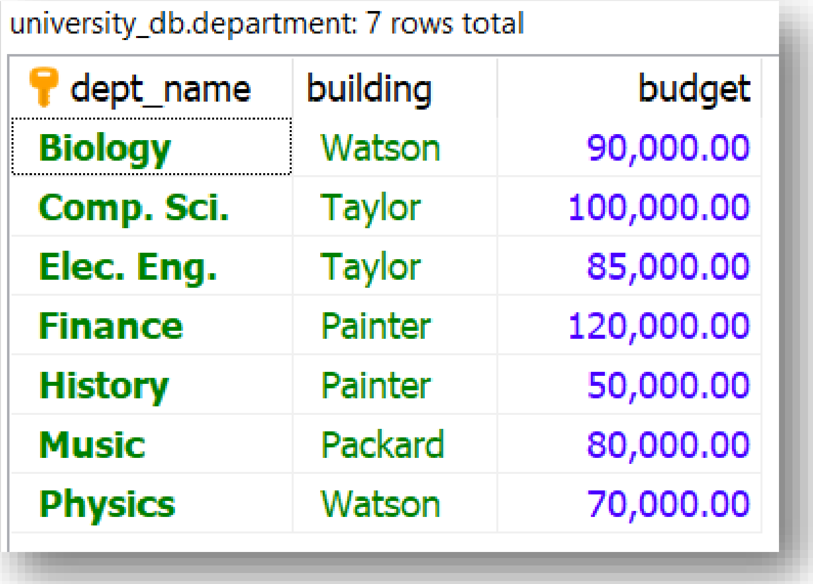
Query on Multiple relation

Show me a list containing all <u>teachers name</u>, their respective <u>department</u> and the <u>building</u> the department is situated.



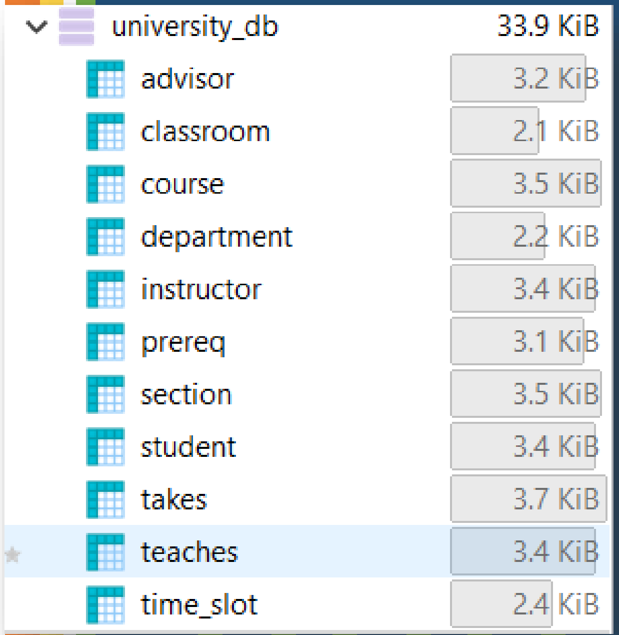
university_db.instructor: 12 rows total

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	(NULL)
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00



university_db.department: 7 rows total

dept_name	building	budget
Biology	Watson	90,000.00
Comp. Sci.	Taylor	100,000.00
Elec. Eng.	Taylor	85,000.00
Finance	Painter	120,000.00
History	Painter	50,000.00
Music	Packard	80,000.00
Physics	Watson	70,000.00



university_db	33.9 KiB
advisor	3.2 KiB
classroom	2.1 KiB
course	3.5 KiB
department	2.2 KiB
instructor	3.4 KiB
prereq	3.1 KiB
section	3.5 KiB
student	3.4 KiB
takes	3.7 KiB
teaches	3.4 KiB
time_slot	2.4 KiB

Cartesian Product

Cartesian Product

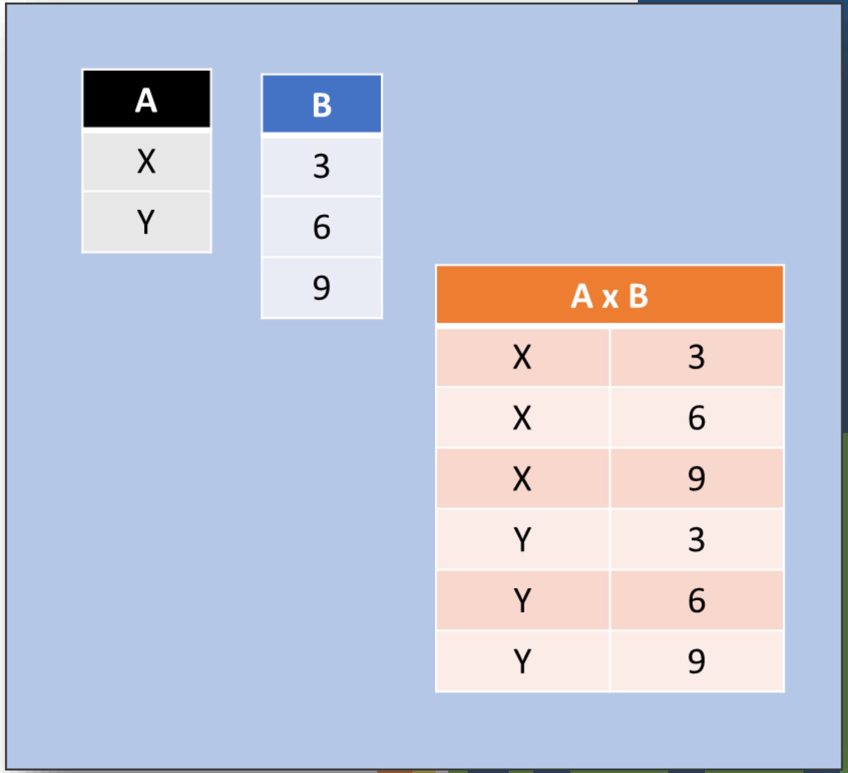
The Cartesian product of two sets A and B, denoted by A × B, is defined as the set consisting of all ordered pairs (a, b) for which

$$\mathbf{a \in A} \quad \text{and} \quad \mathbf{b \in B.}$$

For example, if

A = {x, y} and B = {3, 6, 9}, then

A × B = {(x, 3), (x, 6), (x, 9), (y, 3), (y, 6), (y, 9)}

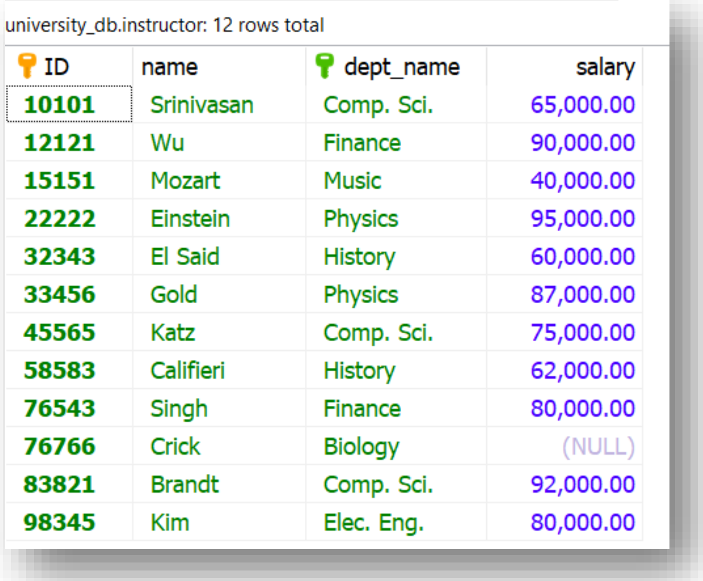


A
X
Y

B
3
6
9

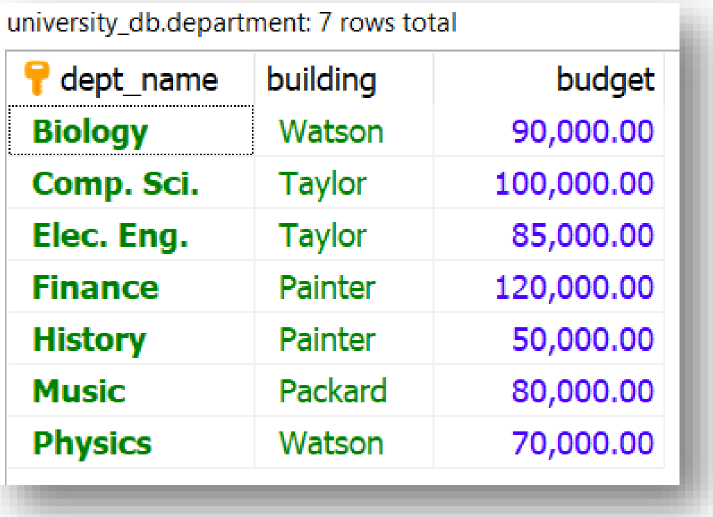
A x B	
X	3
X	6
X	9
Y	3
Y	6
Y	9

Cartesian Product



university_db.instructor: 12 rows total

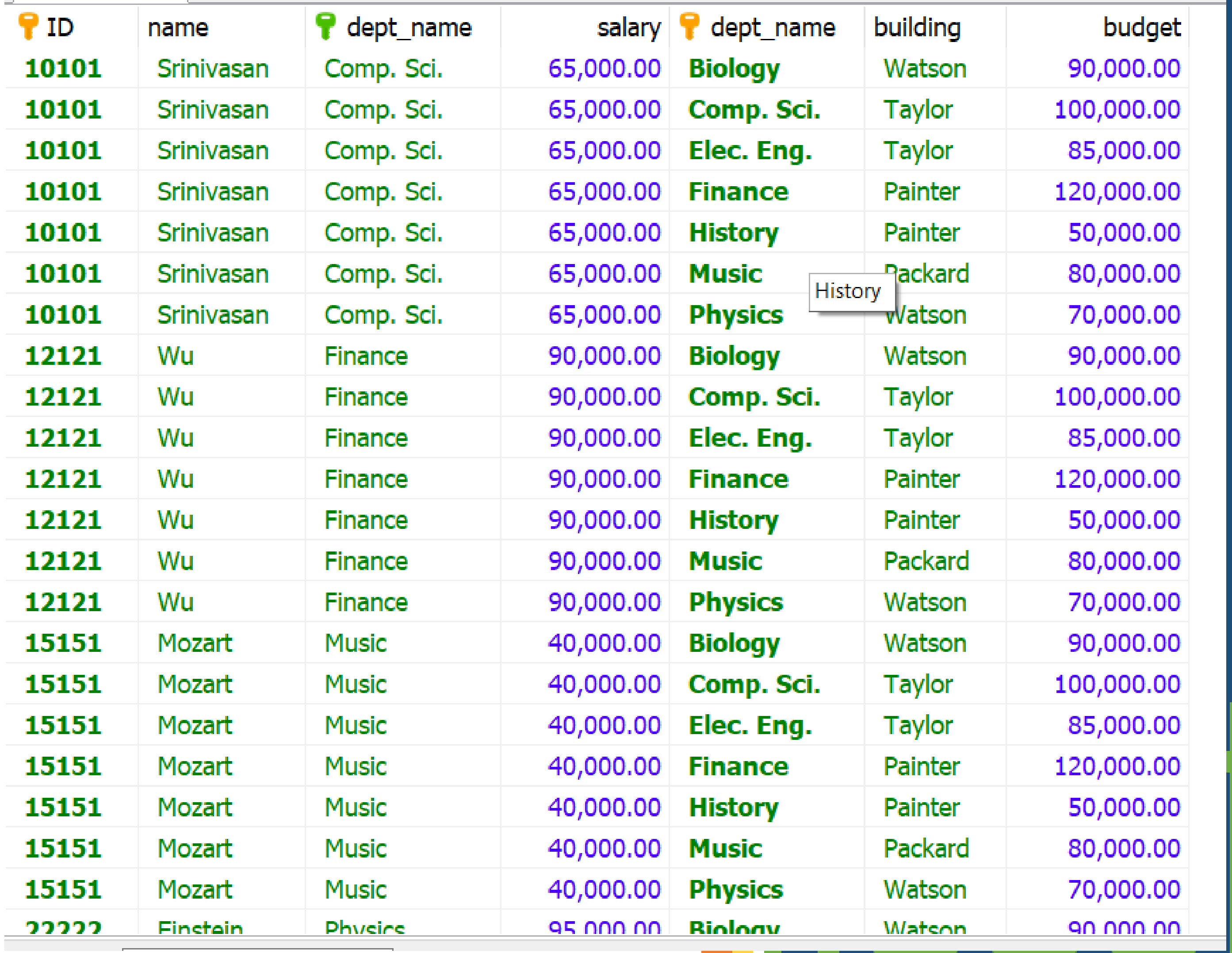
ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	(NULL)
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00



university_db.department: 7 rows total

dept_name	building	budget
Biology	Watson	90,000.00
Comp. Sci.	Taylor	100,000.00
Elec. Eng.	Taylor	85,000.00
Finance	Painter	120,000.00
History	Painter	50,000.00
Music	Packard	80,000.00
Physics	Watson	70,000.00

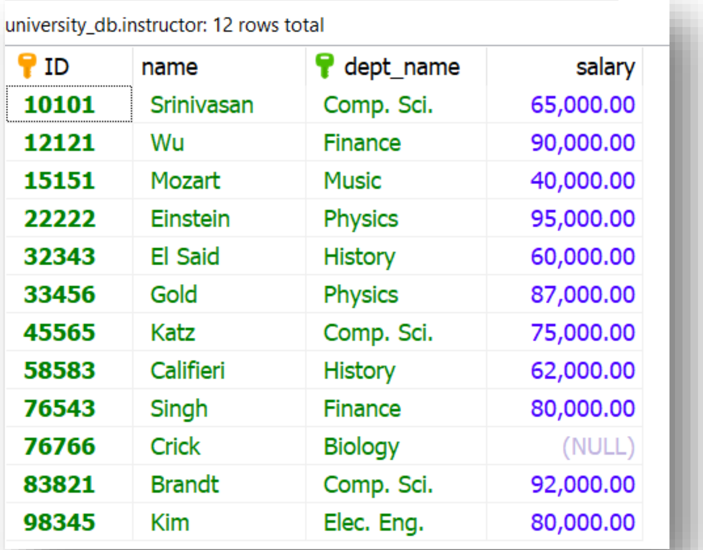
instructor X department



Result #1 (7×84)

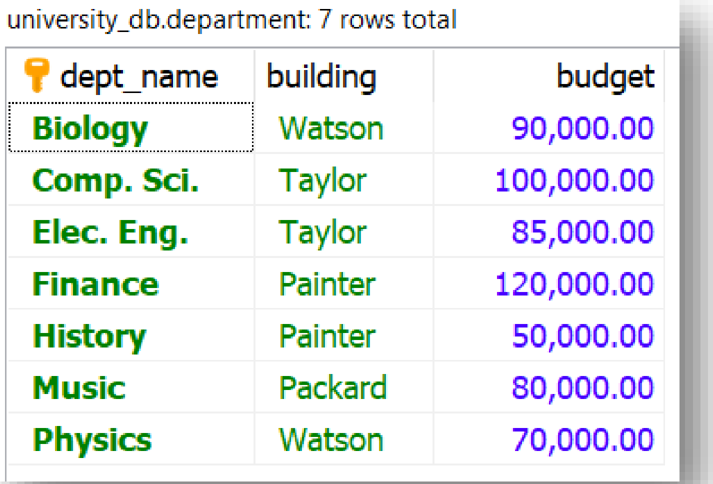
ID	name	dept_name	salary	dept_name	building	budget
10101	Srinivasan	Comp. Sci.	65,000.00	Biology	Watson	90,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	Comp. Sci.	Taylor	100,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	Elec. Eng.	Taylor	85,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	Finance	Painter	120,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	History	Painter	50,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	Music	Packard	80,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	Physics	Watson	70,000.00
12121	Wu	Finance	90,000.00	Biology	Watson	90,000.00
12121	Wu	Finance	90,000.00	Comp. Sci.	Taylor	100,000.00
12121	Wu	Finance	90,000.00	Elec. Eng.	Taylor	85,000.00
12121	Wu	Finance	90,000.00	Finance	Painter	120,000.00
12121	Wu	Finance	90,000.00	History	Painter	50,000.00
12121	Wu	Finance	90,000.00	Music	Packard	80,000.00
12121	Wu	Finance	90,000.00	Physics	Watson	70,000.00
15151	Mozart	Music	40,000.00	Biology	Watson	90,000.00
15151	Mozart	Music	40,000.00	Comp. Sci.	Taylor	100,000.00
15151	Mozart	Music	40,000.00	Elec. Eng.	Taylor	85,000.00
15151	Mozart	Music	40,000.00	Finance	Painter	120,000.00
15151	Mozart	Music	40,000.00	History	Painter	50,000.00
15151	Mozart	Music	40,000.00	Music	Packard	80,000.00
15151	Mozart	Music	40,000.00	Physics	Watson	70,000.00
22222	Einstein	Physics	95,000.00	Biology	Watson	90,000.00

Cartesian Product



university_db.instructor: 12 rows total

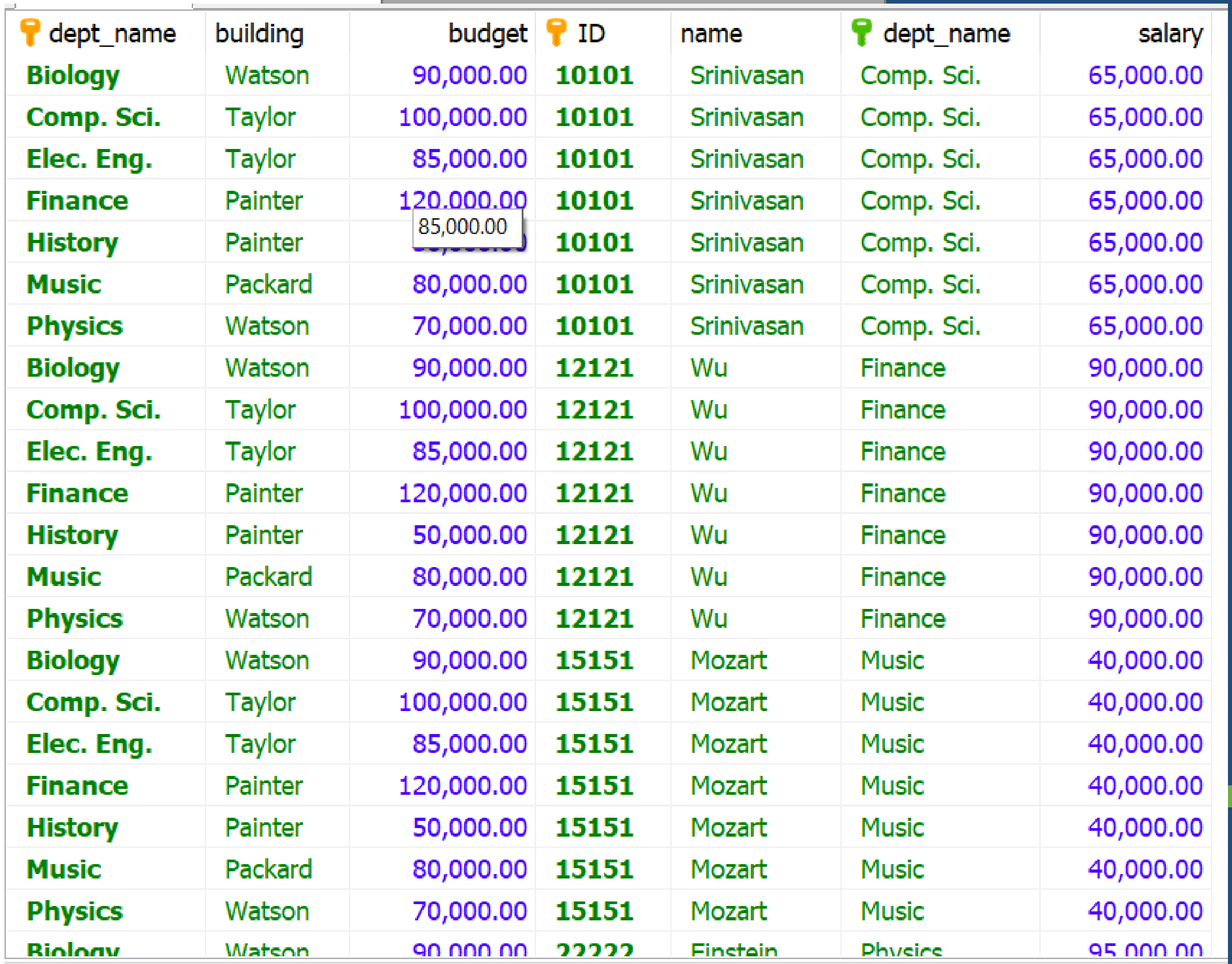
ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	(NULL)
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00



university_db.department: 7 rows total

dept_name	building	budget
Biology	Watson	90,000.00
Comp. Sci.	Taylor	100,000.00
Elec. Eng.	Taylor	85,000.00
Finance	Painter	120,000.00
History	Painter	50,000.00
Music	Packard	80,000.00
Physics	Watson	70,000.00

department X instructor

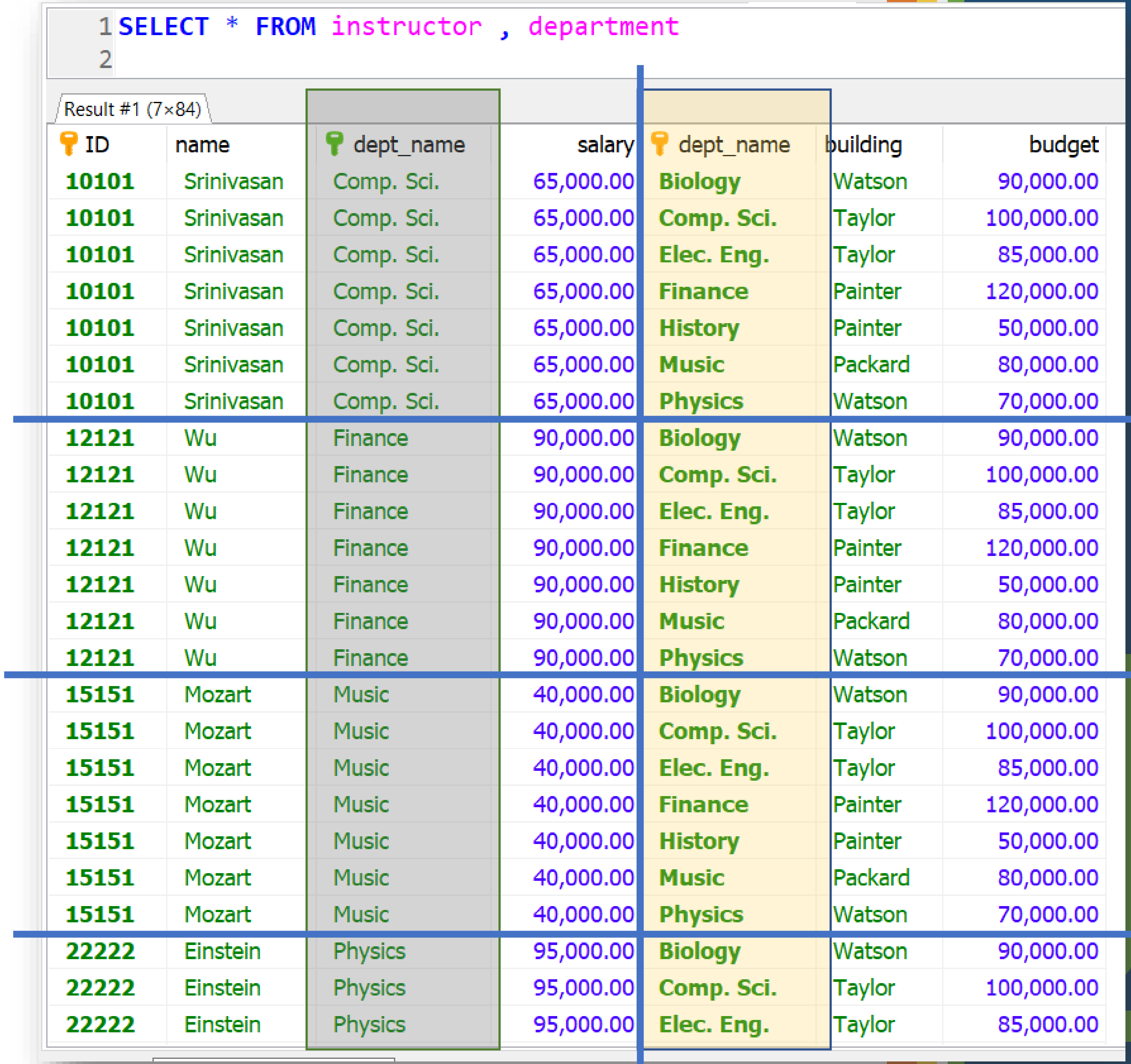


dept_name	building	budget	ID	name	dept_name	salary
Biology	Watson	90,000.00	10101	Srinivasan	Comp. Sci.	65,000.00
Comp. Sci.	Taylor	100,000.00	10101	Srinivasan	Comp. Sci.	65,000.00
Elec. Eng.	Taylor	85,000.00	10101	Srinivasan	Comp. Sci.	65,000.00
Finance	Painter	120,000.00	10101	Srinivasan	Comp. Sci.	65,000.00
History	Painter	85,000.00	10101	Srinivasan	Comp. Sci.	65,000.00
Music	Packard	80,000.00	10101	Srinivasan	Comp. Sci.	65,000.00
Physics	Watson	70,000.00	10101	Srinivasan	Comp. Sci.	65,000.00
Biology	Watson	90,000.00	12121	Wu	Finance	90,000.00
Comp. Sci.	Taylor	100,000.00	12121	Wu	Finance	90,000.00
Elec. Eng.	Taylor	85,000.00	12121	Wu	Finance	90,000.00
Finance	Painter	120,000.00	12121	Wu	Finance	90,000.00
History	Painter	50,000.00	12121	Wu	Finance	90,000.00
Music	Packard	80,000.00	12121	Wu	Finance	90,000.00
Physics	Watson	70,000.00	12121	Wu	Finance	90,000.00
Biology	Watson	90,000.00	15151	Mozart	Music	40,000.00
Comp. Sci.	Taylor	100,000.00	15151	Mozart	Music	40,000.00
Elec. Eng.	Taylor	85,000.00	15151	Mozart	Music	40,000.00
Finance	Painter	120,000.00	15151	Mozart	Music	40,000.00
History	Painter	50,000.00	15151	Mozart	Music	40,000.00
Music	Packard	80,000.00	15151	Mozart	Music	40,000.00
Physics	Watson	70,000.00	15151	Mozart	Music	40,000.00
Biology	Watson	90,000.00	22222	Einstein	Physics	95,000.00

Cartesian Product in SQL

This result is not very heplfull.

We need to filter out the records that make sense based on some **common fields** between two relations

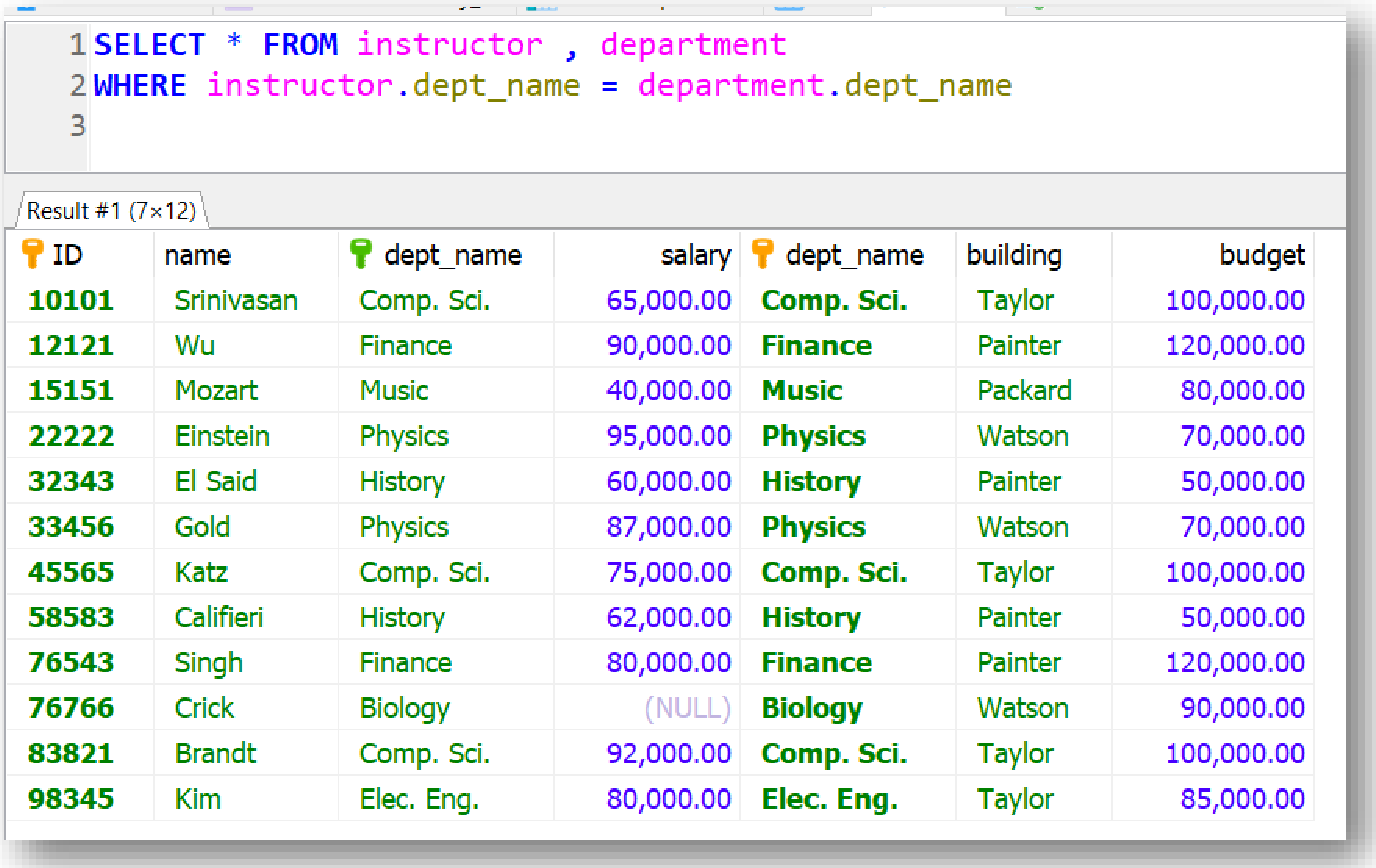


```
SELECT * FROM instructor , department
```

Result #1 (7×84)

ID	name	dept_name	salary	dept_name	building	budget
10101	Srinivasan	Comp. Sci.	65,000.00	Biology	Watson	90,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	Comp. Sci.	Taylor	100,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	Elec. Eng.	Taylor	85,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	Finance	Painter	120,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	History	Painter	50,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	Music	Packard	80,000.00
10101	Srinivasan	Comp. Sci.	65,000.00	Physics	Watson	70,000.00
12121	Wu	Finance	90,000.00	Biology	Watson	90,000.00
12121	Wu	Finance	90,000.00	Comp. Sci.	Taylor	100,000.00
12121	Wu	Finance	90,000.00	Elec. Eng.	Taylor	85,000.00
12121	Wu	Finance	90,000.00	Finance	Painter	120,000.00
12121	Wu	Finance	90,000.00	History	Painter	50,000.00
12121	Wu	Finance	90,000.00	Music	Packard	80,000.00
12121	Wu	Finance	90,000.00	Physics	Watson	70,000.00
15151	Mozart	Music	40,000.00	Biology	Watson	90,000.00
15151	Mozart	Music	40,000.00	Comp. Sci.	Taylor	100,000.00
15151	Mozart	Music	40,000.00	Elec. Eng.	Taylor	85,000.00
15151	Mozart	Music	40,000.00	Finance	Painter	120,000.00
15151	Mozart	Music	40,000.00	History	Painter	50,000.00
15151	Mozart	Music	40,000.00	Music	Packard	80,000.00
15151	Mozart	Music	40,000.00	Physics	Watson	70,000.00
22222	Einstein	Physics	95,000.00	Biology	Watson	90,000.00
22222	Einstein	Physics	95,000.00	Comp. Sci.	Taylor	100,000.00
22222	Einstein	Physics	95,000.00	Elec. Eng.	Taylor	85,000.00

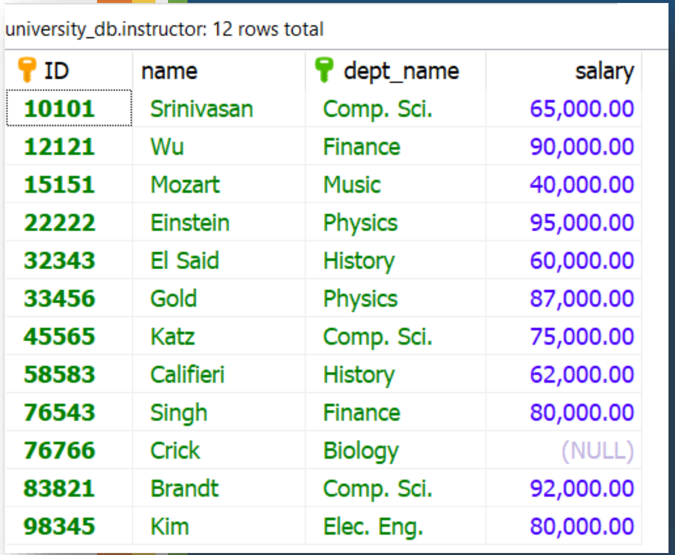
Cartesian Product in SQL



```sql
SELECT * FROM instructor , department
WHERE instructor.dept_name = department.dept_name
```

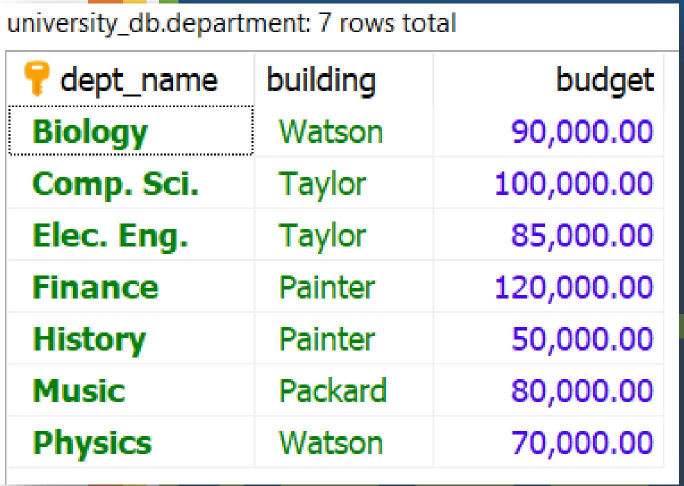
Result #1 (7×12)

ID	name	dept_name	salary	dept_name	building	budget
10101	Srinivasan	Comp. Sci.	65,000.00	Comp. Sci.	Taylor	100,000.00
12121	Wu	Finance	90,000.00	Finance	Painter	120,000.00
15151	Mozart	Music	40,000.00	Music	Packard	80,000.00
22222	Einstein	Physics	95,000.00	Physics	Watson	70,000.00
32343	El Said	History	60,000.00	History	Painter	50,000.00
33456	Gold	Physics	87,000.00	Physics	Watson	70,000.00
45565	Katz	Comp. Sci.	75,000.00	Comp. Sci.	Taylor	100,000.00
58583	Califieri	History	62,000.00	History	Painter	50,000.00
76543	Singh	Finance	80,000.00	Finance	Painter	120,000.00
76766	Crick	Biology	(NULL)	Biology	Watson	90,000.00
83821	Brandt	Comp. Sci.	92,000.00	Comp. Sci.	Taylor	100,000.00
98345	Kim	Elec. Eng.	80,000.00	Elec. Eng.	Taylor	85,000.00



university_db.instructor: 12 rows total

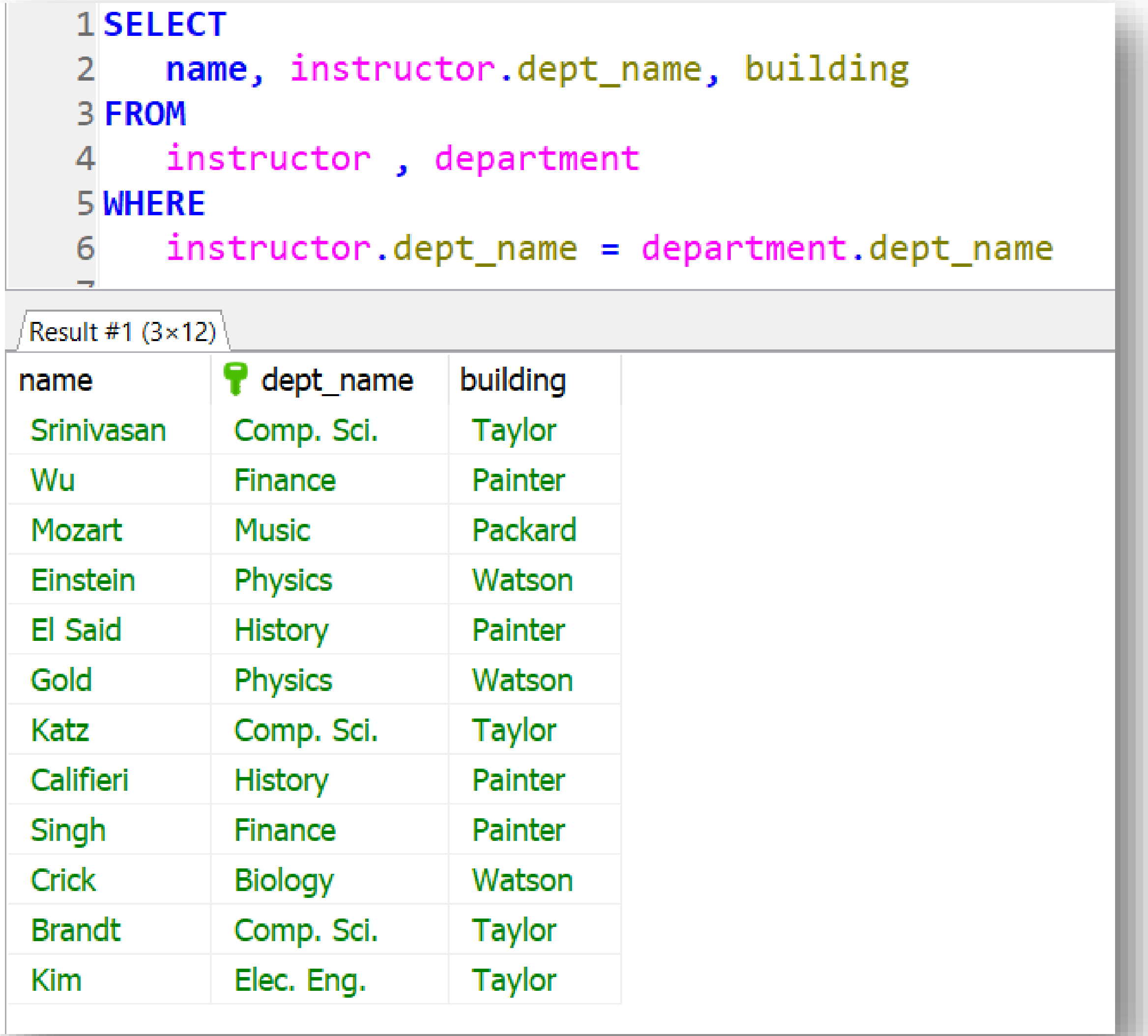
ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	(NULL)
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00



university_db.department: 7 rows total

dept_name	building	budget
Biology	Watson	90,000.00
Comp. Sci.	Taylor	100,000.00
Elec. Eng.	Taylor	85,000.00
Finance	Painter	120,000.00
History	Painter	50,000.00
Music	Packard	80,000.00
Physics	Watson	70,000.00

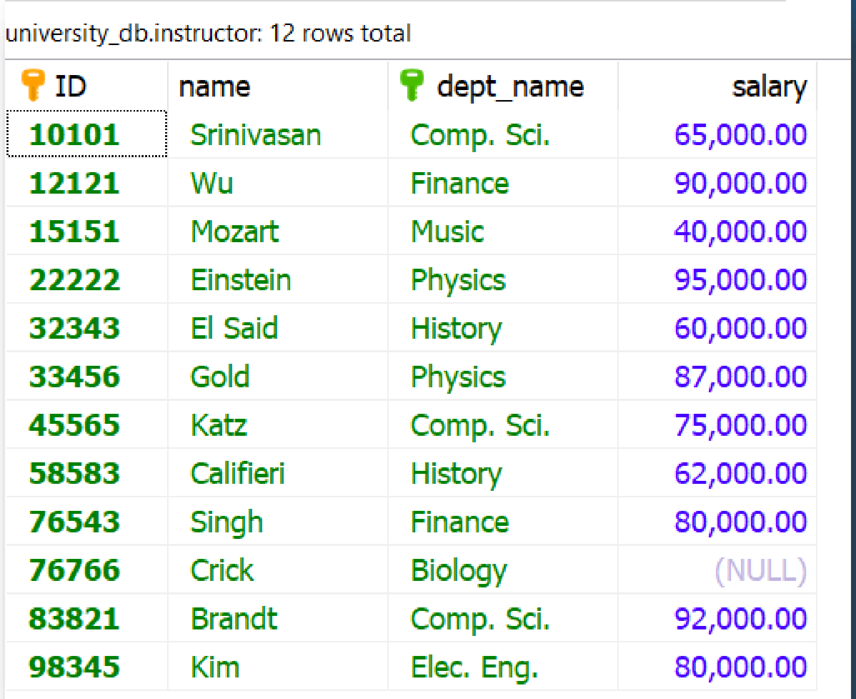
Show me a list containing all teachers name, their respective department and the building the department is situated.



```sql
SELECT
    name, instructor.dept_name, building
FROM
    instructor , department
WHERE
    instructor.dept_name = department.dept_name
```

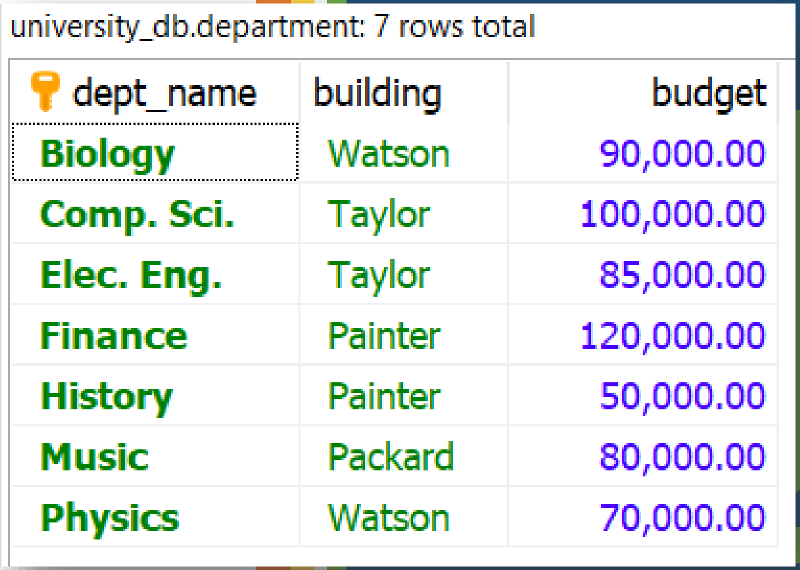
Result #1 (3×12)

name	dept_name	building
Srinivasan	Comp. Sci.	Taylor
Wu	Finance	Painter
Mozart	Music	Packard
Einstein	Physics	Watson
El Said	History	Painter
Gold	Physics	Watson
Katz	Comp. Sci.	Taylor
Califieri	History	Painter
Singh	Finance	Painter
Crick	Biology	Watson
Brandt	Comp. Sci.	Taylor
Kim	Elec. Eng.	Taylor



university_db.instructor: 12 rows total

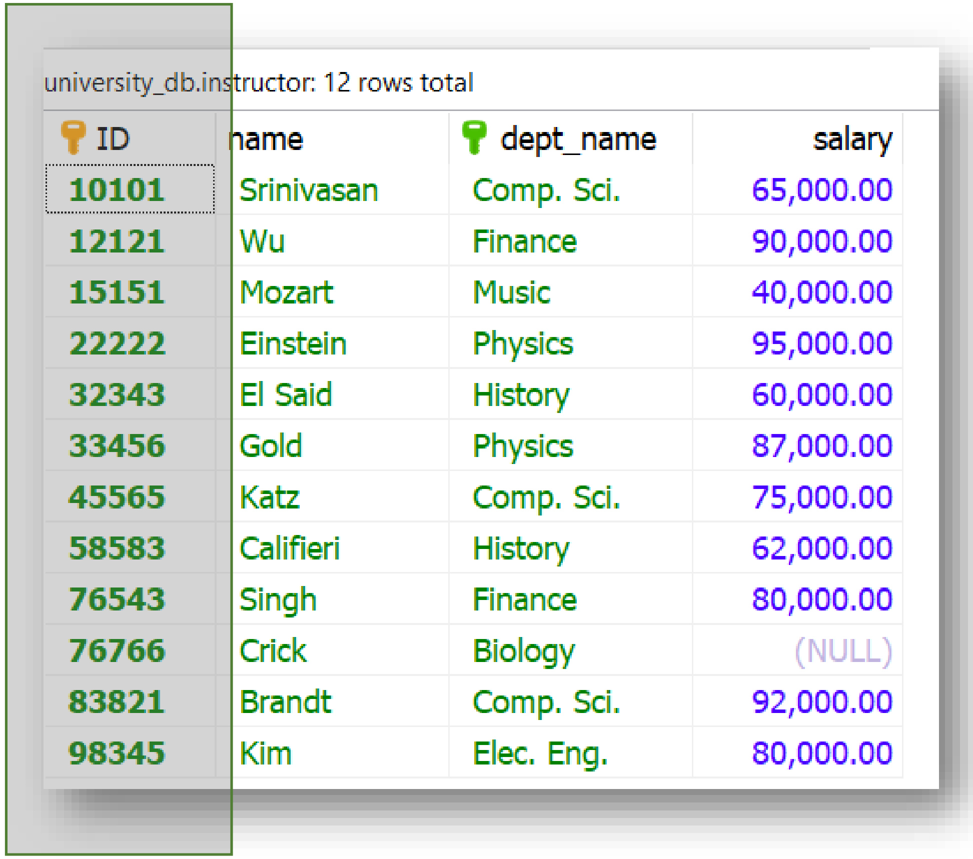
ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	(NULL)
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00



university_db.department: 7 rows total

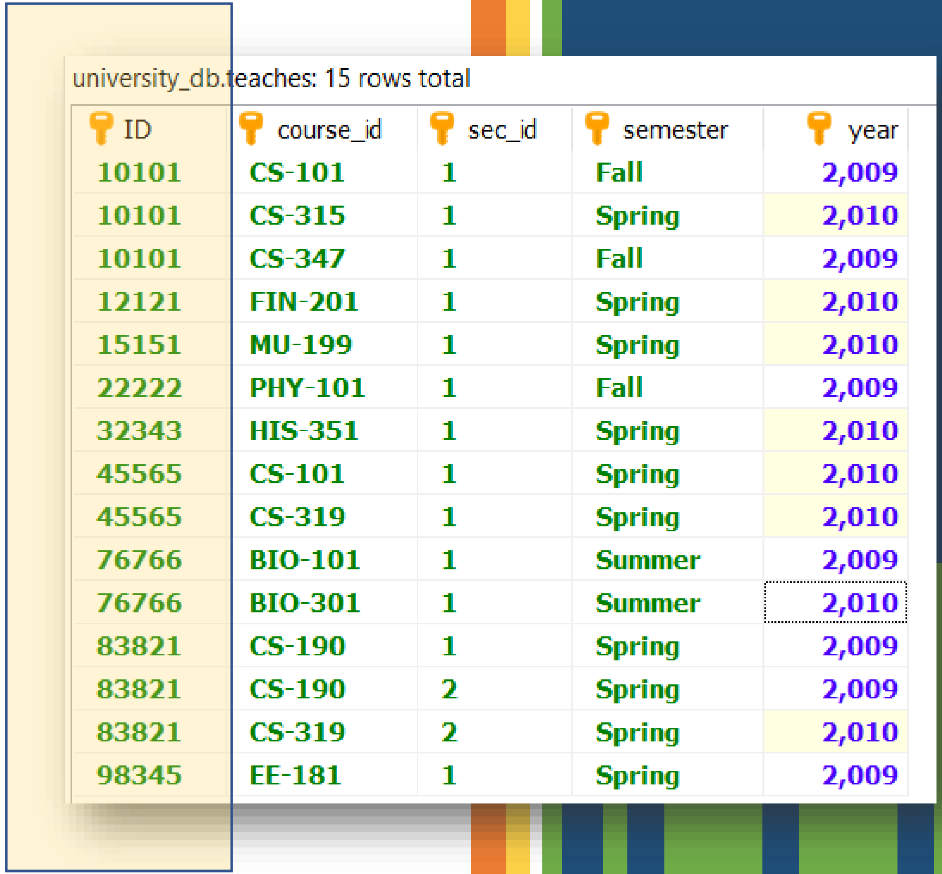
dept_name	building	budget
Biology	Watson	90,000.00
Comp. Sci.	Taylor	100,000.00
Elec. Eng.	Taylor	85,000.00
Finance	Painter	120,000.00
History	Painter	50,000.00
Music	Packard	80,000.00
Physics	Watson	70,000.00

Find the names of all instructors who have taught some course and the course_id of those courses, and year



university_db.instructor: 12 rows total

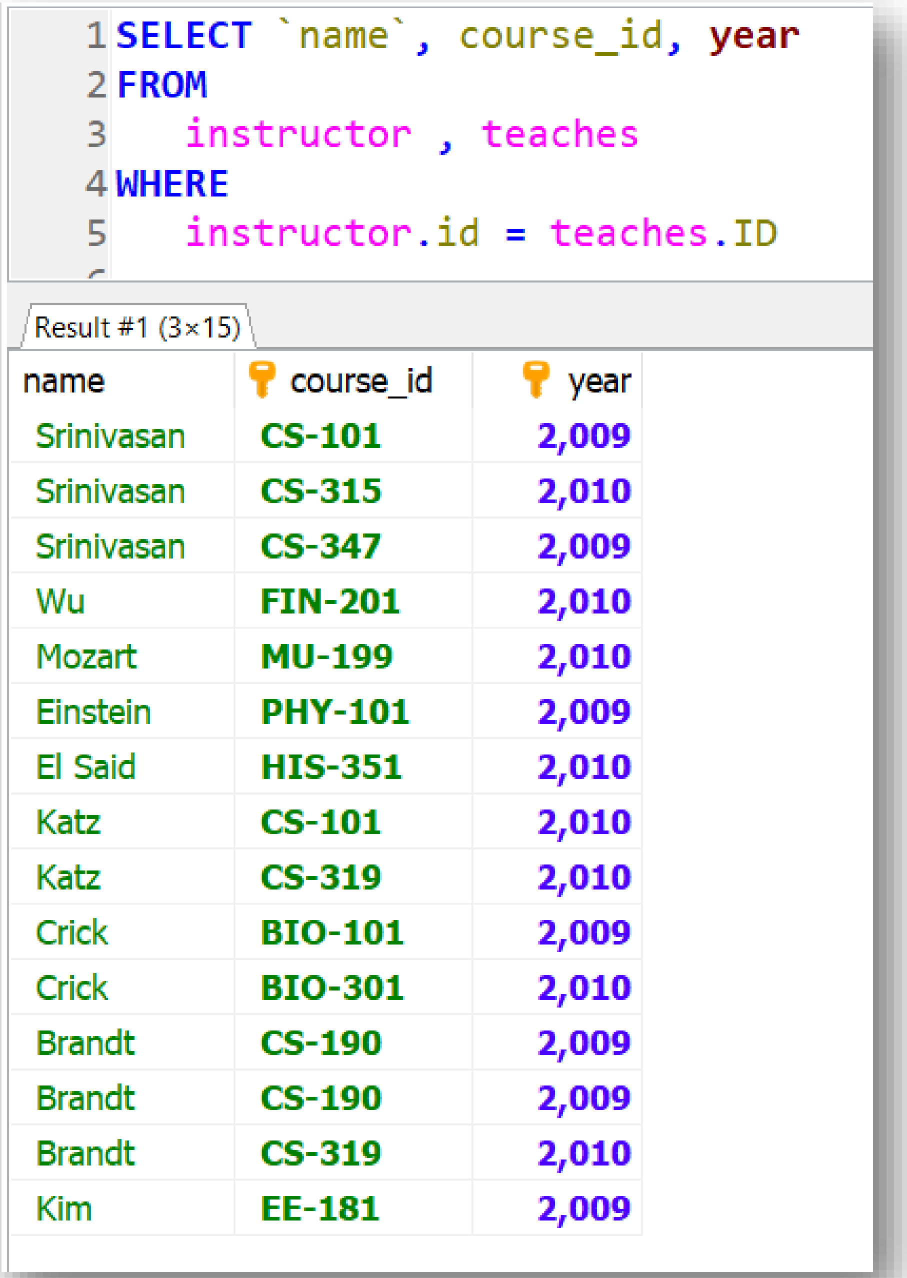
ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	(NULL)
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00



university_db.teaches: 15 rows total

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2,009
10101	CS-315	1	Spring	2,010
10101	CS-347	1	Fall	2,009
12121	FIN-201	1	Spring	2,010
15151	MU-199	1	Spring	2,010
22222	PHY-101	1	Fall	2,009
32343	HIS-351	1	Spring	2,010
45565	CS-101	1	Spring	2,010
45565	CS-319	1	Spring	2,010
76766	BIO-101	1	Summer	2,009
76766	BIO-301	1	Summer	2,010
83821	CS-190	1	Spring	2,009
83821	CS-190	2	Spring	2,009
83821	CS-319	2	Spring	2,010
98345	EE-181	1	Spring	2,009

04-Dec-21 10:43 AM



```sql
SELECT `name`, course_id, year
FROM
    instructor , teaches
WHERE
    instructor.id = teaches.ID
```

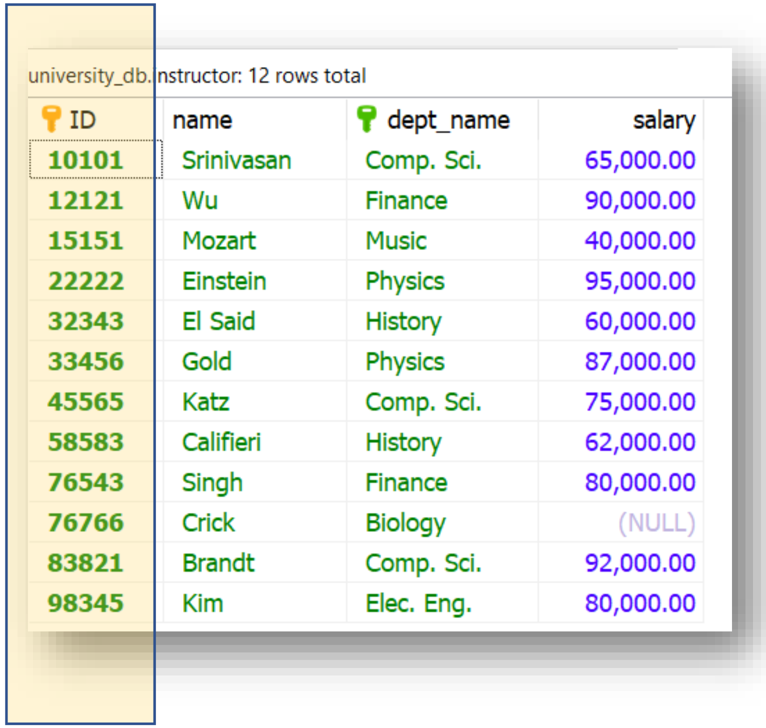
Result #1 (3×15)

name	course_id	year
Srinivasan	CS-101	2,009
Srinivasan	CS-315	2,010
Srinivasan	CS-347	2,009
Wu	FIN-201	2,010
Mozart	MU-199	2,010
Einstein	PHY-101	2,009
El Said	HIS-351	2,010
Katz	CS-101	2,010
Katz	CS-319	2,010
Crick	BIO-101	2,009
Crick	BIO-301	2,010
Brandt	CS-190	2,009
Brandt	CS-190	2,009
Brandt	CS-319	2,010
Kim	EE-181	2,009

Find the names of all instructors who have taught some course and the course_id of those courses

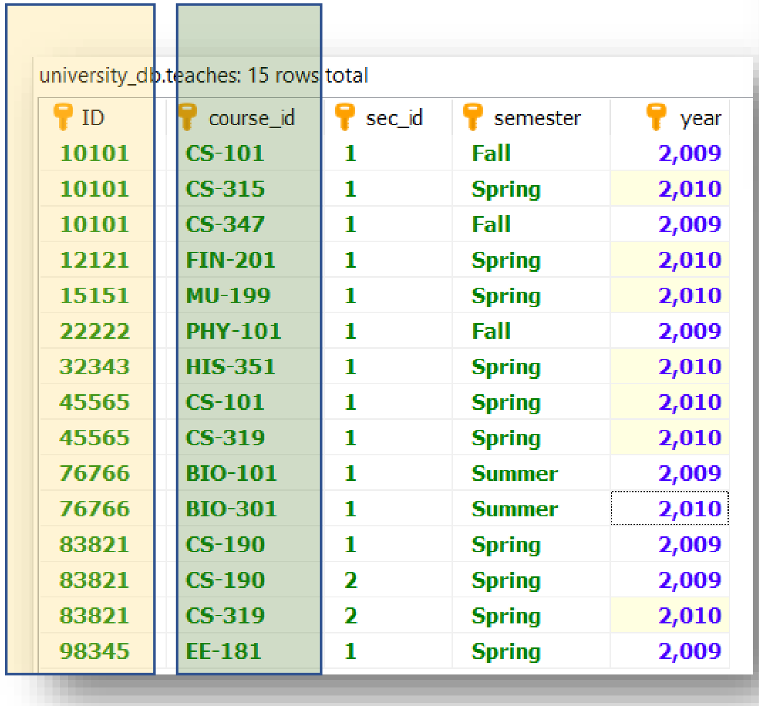
What if I also need course name?

Find the names of all instructors who have taught some course and the course_id, course name of those courses, and year



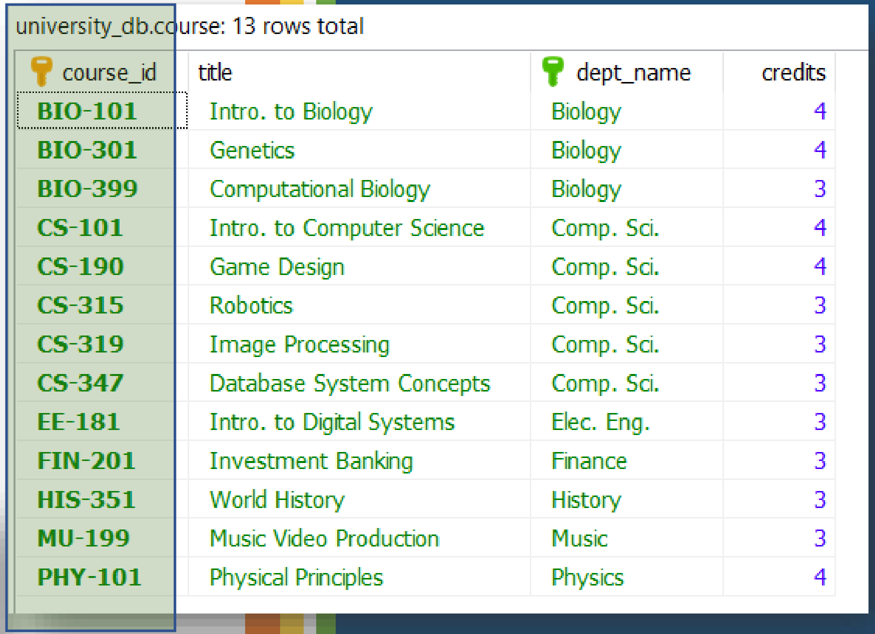
university_db.instructor: 12 rows total

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	(NULL)
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00



university_db.teaches: 15 rows total

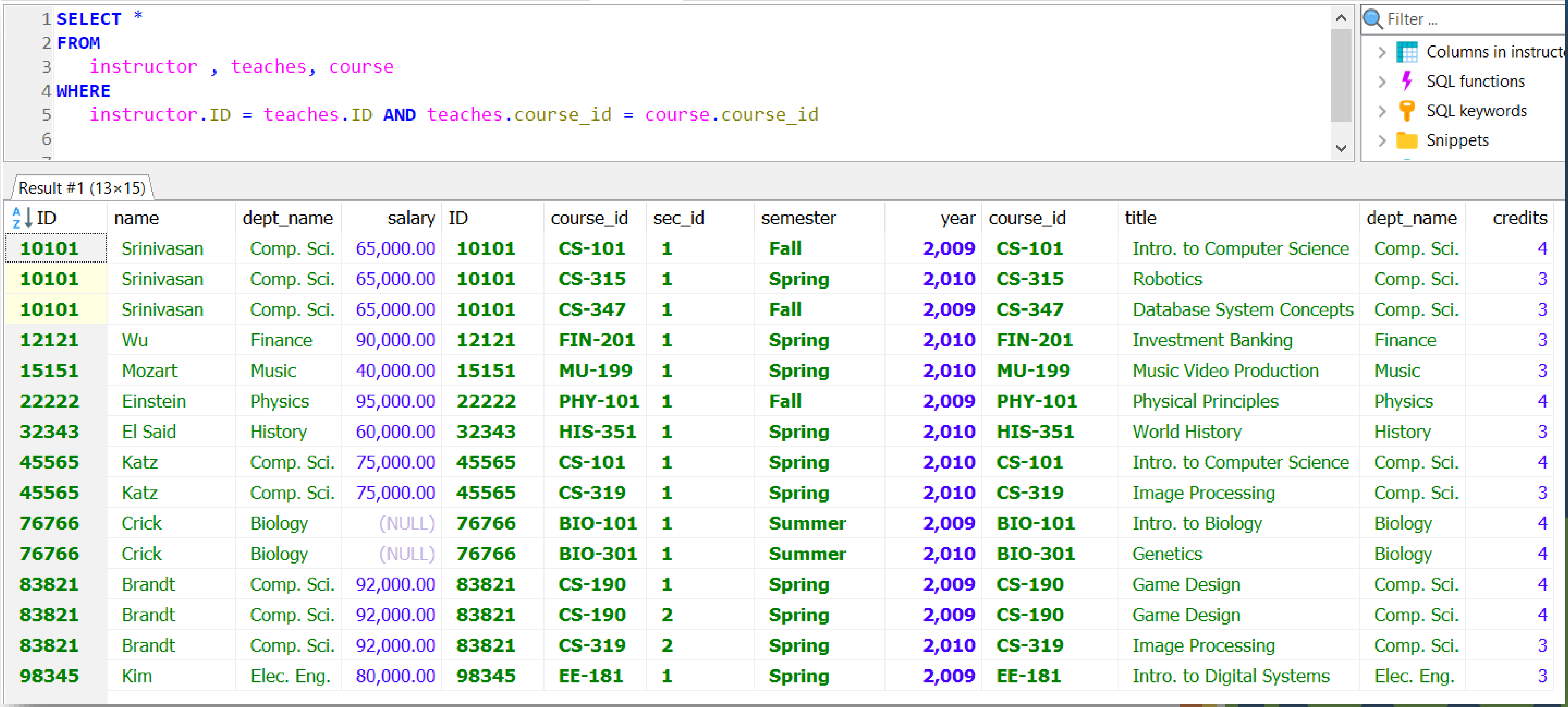
ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2,009
10101	CS-315	1	Spring	2,010
10101	CS-347	1	Fall	2,009
12121	FIN-201	1	Spring	2,010
15151	MU-199	1	Spring	2,010
22222	PHY-101	1	Fall	2,009
32343	HIS-351	1	Spring	2,010
45565	CS-101	1	Spring	2,010
45565	CS-319	1	Spring	2,010
76766	BIO-101	1	Summer	2,009
76766	BIO-301	1	Summer	2,010
83821	CS-190	1	Spring	2,009
83821	CS-190	2	Spring	2,009
83821	CS-319	2	Spring	2,010
98345	EE-181	1	Spring	2,009



university_db.course: 13 rows total

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 Principles	Physics	4

04-Dec-21 10:43 AM



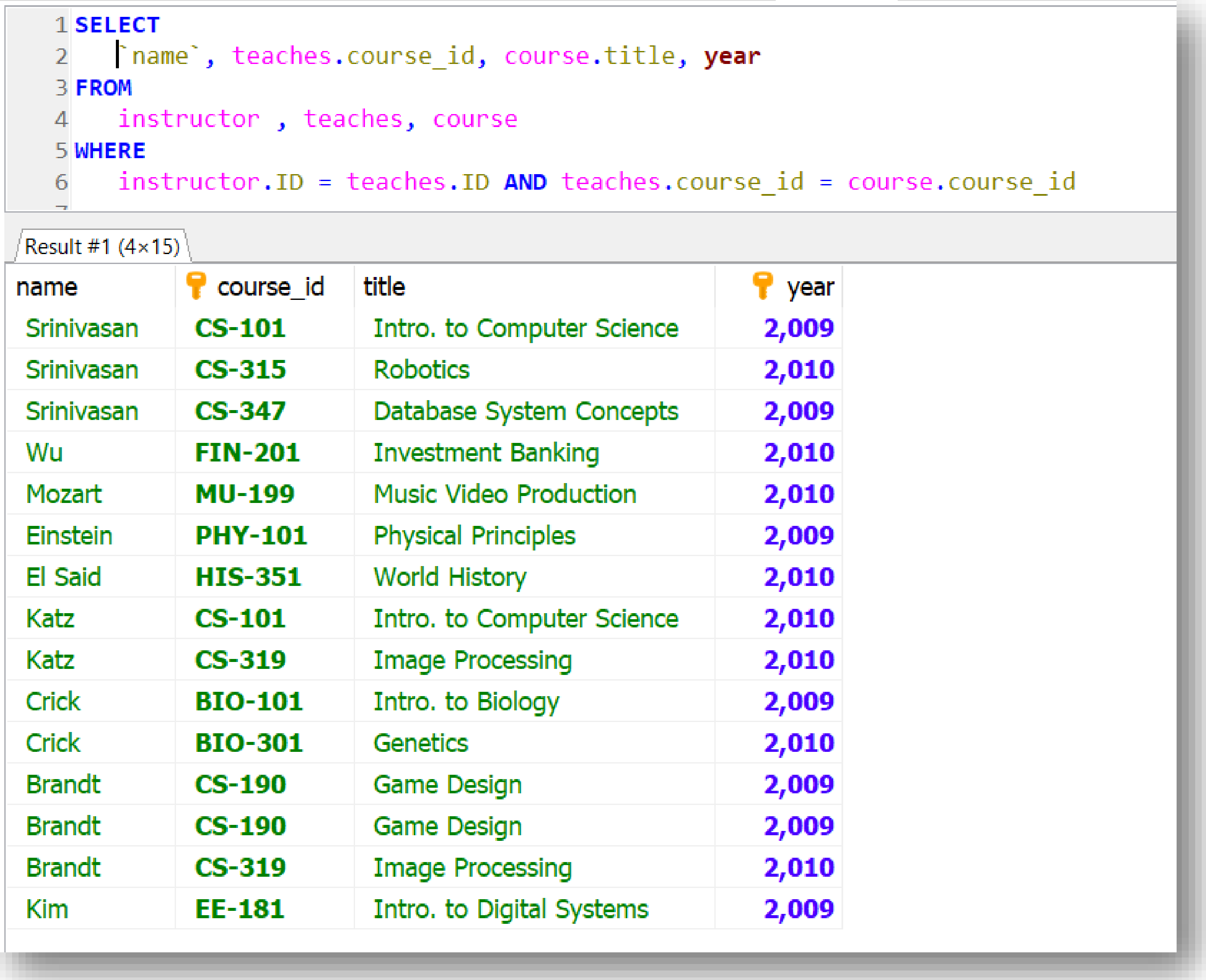
```sql
SELECT *
FROM
   instructor , teaches, course
WHERE
   instructor.ID = teaches.ID AND teaches.course_id = course.course_id

```

Result #1 (13×15)

ID	name	dept_name	salary	ID	course_id	sec_id	semester	year	course_id	title	dept_name	credits
10101	Srinivasan	Comp. Sci.	65,000.00	10101	CS-101	1	Fall	2,009	CS-101	Intro. to Computer Science	Comp. Sci.	4
10101	Srinivasan	Comp. Sci.	65,000.00	10101	CS-315	1	Spring	2,010	CS-315	Robotics	Comp. Sci.	3
10101	Srinivasan	Comp. Sci.	65,000.00	10101	CS-347	1	Fall	2,009	CS-347	Database System Concepts	Comp. Sci.	3
12121	Wu	Finance	90,000.00	12121	FIN-201	1	Spring	2,010	FIN-201	Investment Banking	Finance	3
15151	Mozart	Music	40,000.00	15151	MU-199	1	Spring	2,010	MU-199	Music Video Production	Music	3
22222	Einstein	Physics	95,000.00	22222	PHY-101	1	Fall	2,009	PHY-101	Physical Principles	Physics	4
32343	El Said	History	60,000.00	32343	HIS-351	1	Spring	2,010	HIS-351	World History	History	3
45565	Katz	Comp. Sci.	75,000.00	45565	CS-101	1	Spring	2,010	CS-101	Intro. to Computer Science	Comp. Sci.	4
45565	Katz	Comp. Sci.	75,000.00	45565	CS-319	1	Spring	2,010	CS-319	Image Processing	Comp. Sci.	3
76766	Crick	Biology	(NULL)	76766	BIO-101	1	Summer	2,009	BIO-101	Intro. to Biology	Biology	4
76766	Crick	Biology	(NULL)	76766	BIO-301	1	Summer	2,010	BIO-301	Genetics	Biology	4
83821	Brandt	Comp. Sci.	92,000.00	83821	CS-190	1	Spring	2,009	CS-190	Game Design	Comp. Sci.	4
83821	Brandt	Comp. Sci.	92,000.00	83821	CS-190	2	Spring	2,009	CS-190	Game Design	Comp. Sci.	4
83821	Brandt	Comp. Sci.	92,000.00	83821	CS-319	2	Spring	2,010	CS-319	Image Processing	Comp. Sci.	3
98345	Kim	Elec. Eng.	80,000.00	98345	EE-181	1	Spring	2,009	EE-181	Intro. to Digital Systems	Elec. Eng.	3

Find the names of all instructors who have taught some course and the course_id, course name of those courses, and year



```
SELECT
   `name`, teaches.course_id, course.title, year
FROM
   instructor , teaches, course
WHERE
   instructor.ID = teaches.ID AND teaches.course_id = course.course_id
```

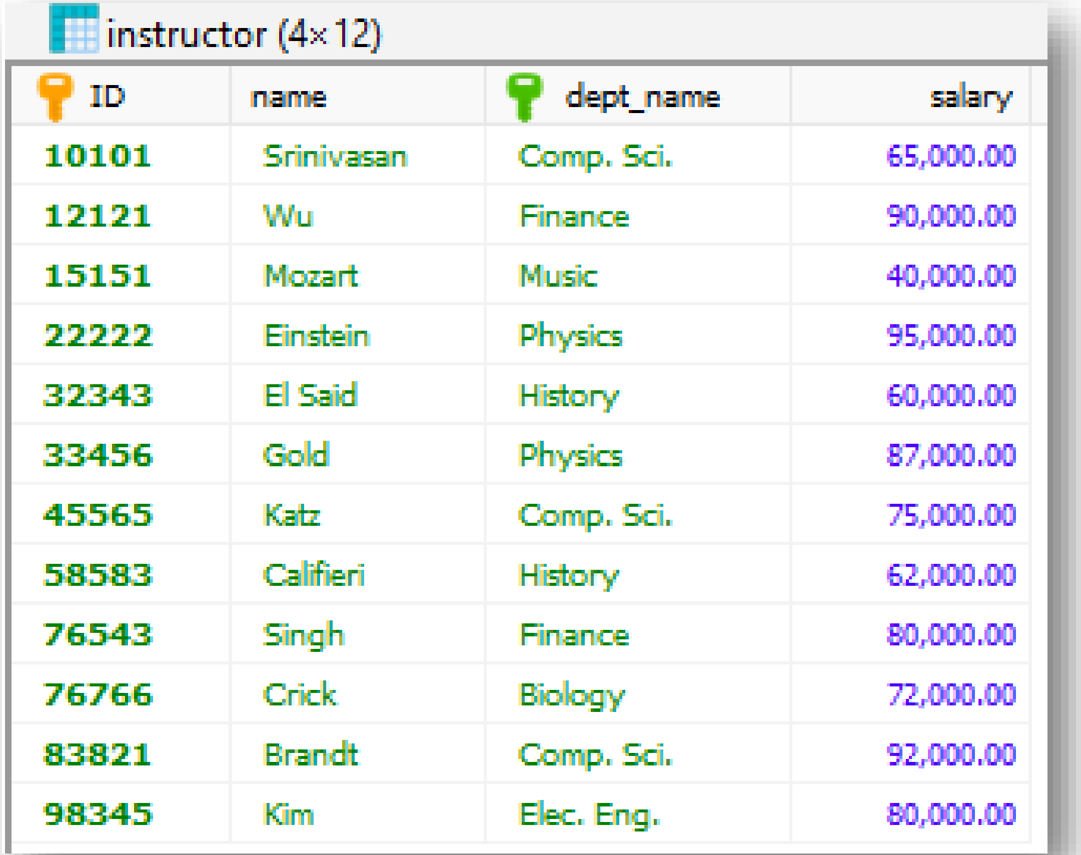
Result #1 (4×15)

name	course_id	title	year
Srinivasan	CS-101	Intro. to Computer Science	2,009
Srinivasan	CS-315	Robotics	2,010
Srinivasan	CS-347	Database System Concepts	2,009
Wu	FIN-201	Investment Banking	2,010
Mozart	MU-199	Music Video Production	2,010
Einstein	PHY-101	Physical Principles	2,009
El Said	HIS-351	World History	2,010
Katz	CS-101	Intro. to Computer Science	2,010
Katz	CS-319	Image Processing	2,010
Crick	BIO-101	Intro. to Biology	2,009
Crick	BIO-301	Genetics	2,010
Brandt	CS-190	Game Design	2,009
Brandt	CS-190	Game Design	2,009
Brandt	CS-319	Image Processing	2,010
Kim	EE-181	Intro. to Digital Systems	2,009

04-Dec-21 10:43 AM

Combining a relation with itself

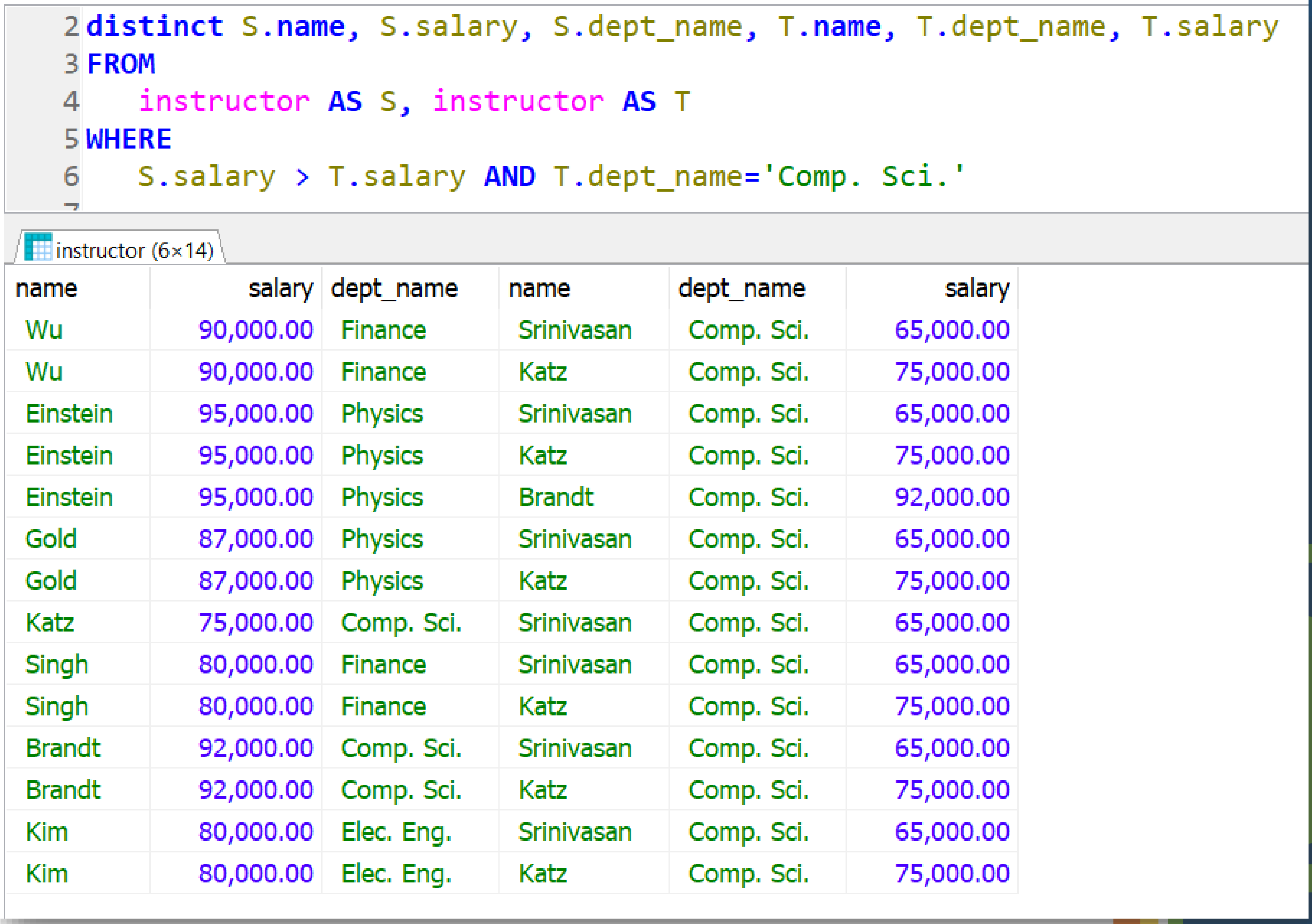
Find the names of all instructors who have a higher salary than some instructor in ‘Comp. Sci’.



instructor (4× 12)

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65,000.00
12121	Wu	Finance	90,000.00
15151	Mozart	Music	40,000.00
22222	Einstein	Physics	95,000.00
32343	El Said	History	60,000.00
33456	Gold	Physics	87,000.00
45565	Katz	Comp. Sci.	75,000.00
58583	Califieri	History	62,000.00
76543	Singh	Finance	80,000.00
76766	Crick	Biology	72,000.00
83821	Brandt	Comp. Sci.	92,000.00
98345	Kim	Elec. Eng.	80,000.00

Find the names of all instructors who have a higher salary than some instructor in ‘Comp. Sci’.



```
distinct S.name, S.salary, S.dept_name, T.name, T.dept_name, T.salary
FROM
   instructor AS S, instructor AS T
WHERE
   S.salary > T.salary AND T.dept_name='Comp. Sci.'
```

name	salary	dept_name	name	dept_name	salary
Wu	90,000.00	Finance	Srinivasan	Comp. Sci.	65,000.00
Wu	90,000.00	Finance	Katz	Comp. Sci.	75,000.00
Einstein	95,000.00	Physics	Srinivasan	Comp. Sci.	65,000.00
Einstein	95,000.00	Physics	Katz	Comp. Sci.	75,000.00
Einstein	95,000.00	Physics	Brandt	Comp. Sci.	92,000.00
Gold	87,000.00	Physics	Srinivasan	Comp. Sci.	65,000.00
Gold	87,000.00	Physics	Katz	Comp. Sci.	75,000.00
Katz	75,000.00	Comp. Sci.	Srinivasan	Comp. Sci.	65,000.00
Singh	80,000.00	Finance	Srinivasan	Comp. Sci.	65,000.00
Singh	80,000.00	Finance	Katz	Comp. Sci.	75,000.00
Brandt	92,000.00	Comp. Sci.	Srinivasan	Comp. Sci.	65,000.00
Brandt	92,000.00	Comp. Sci.	Katz	Comp. Sci.	75,000.00
Kim	80,000.00	Elec. Eng.	Srinivasan	Comp. Sci.	65,000.00
Kim	80,000.00	Elec. Eng.	Katz	Comp. Sci.	75,000.00

The Rename Operation

- The SQL allows renaming relations and attributes using the as clause:

  old-name **as** new-name

- Find the names of all instructors who have a higher salary than some instructor in ‘Comp. Sci’.

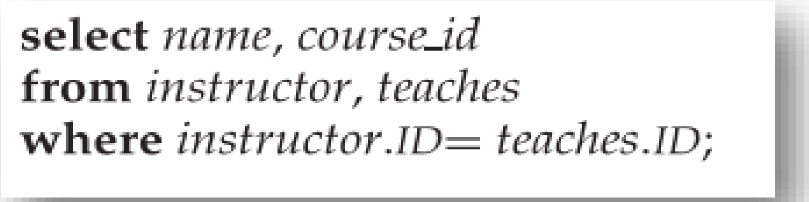
```
select
        distinct  S.name
from
        instructor as S, instructor as T
where
        S.salary > T.salary and T.dept_name = ‘Comp. Sci.’
```

- Keyword as is optional and may be omitted

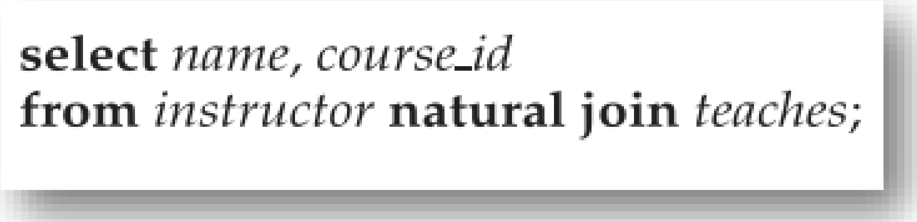
  instructor **as** T ≡ instructor T

The Natural Join

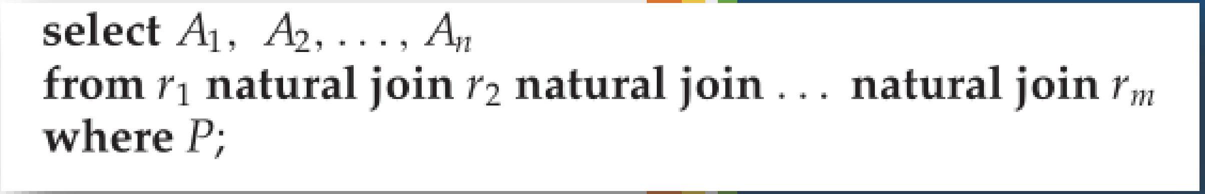
- To make the life of an SQL programmer easier for this common case, SQL supports an operation called the **natural join**



select *name*, *course_id*
from *instructor*, *teaches*
where *instructor*.*ID*= *teaches*.*ID*;



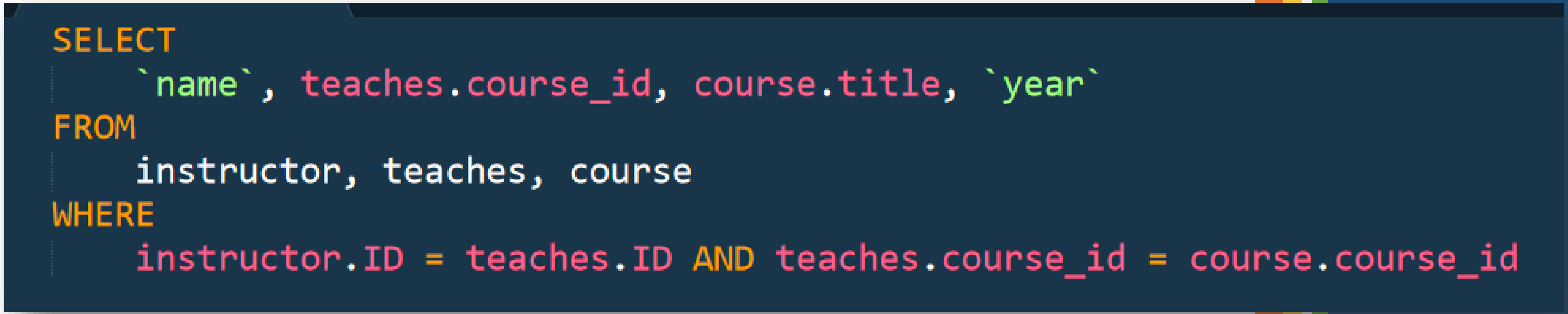
select *name*, *course_id*
from *instructor* **natural join** *teaches*;



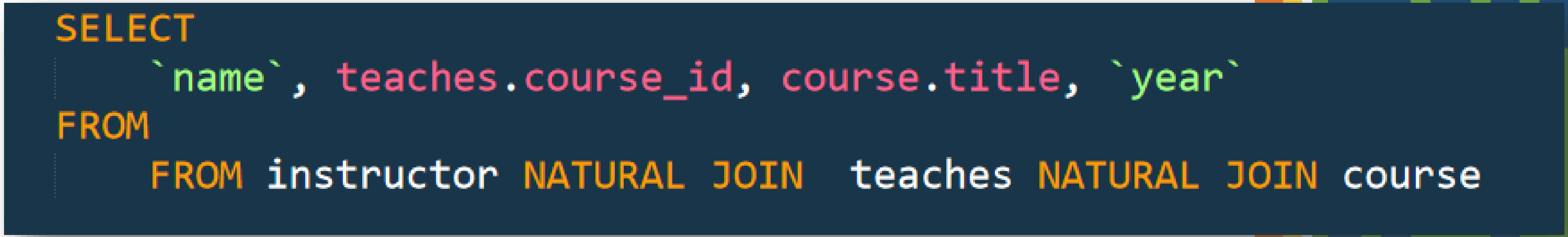
select A_1, A_2, . . . , A_n
from r_1 **natural join** r_2 **natural join** . . . **natural join** r_m
where P;

The Natural Join

- To make the life of an SQL programmer easier for this common case, SQL supports an operation called the **natural join**



```
SELECT
    `name`, teaches.course_id, course.title, `year`
FROM
    instructor, teaches, course
WHERE
    instructor.ID = teaches.ID AND teaches.course_id = course.course_id
```



```
SELECT
    `name`, teaches.course_id, course.title, `year`
FROM
    FROM instructor NATURAL JOIN  teaches NATURAL JOIN course
```

Demo.

End.