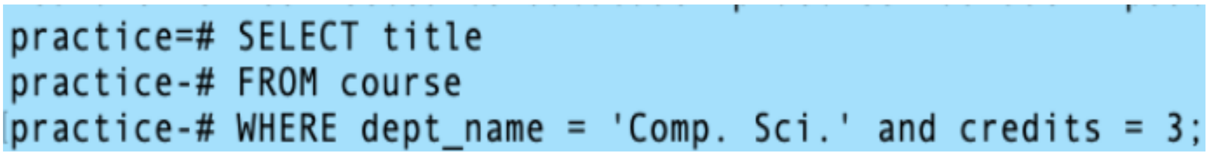
Assignment 3

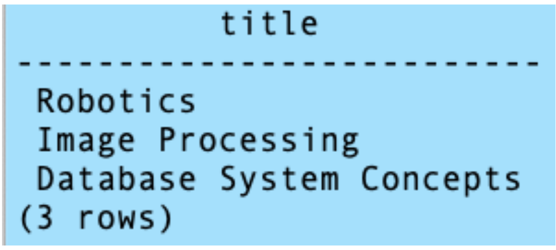
2013130874
영어영문학과 한석희

EXERCISE 1

a. Find the titles of courses in the Comp. Sci. department that have 3 credits

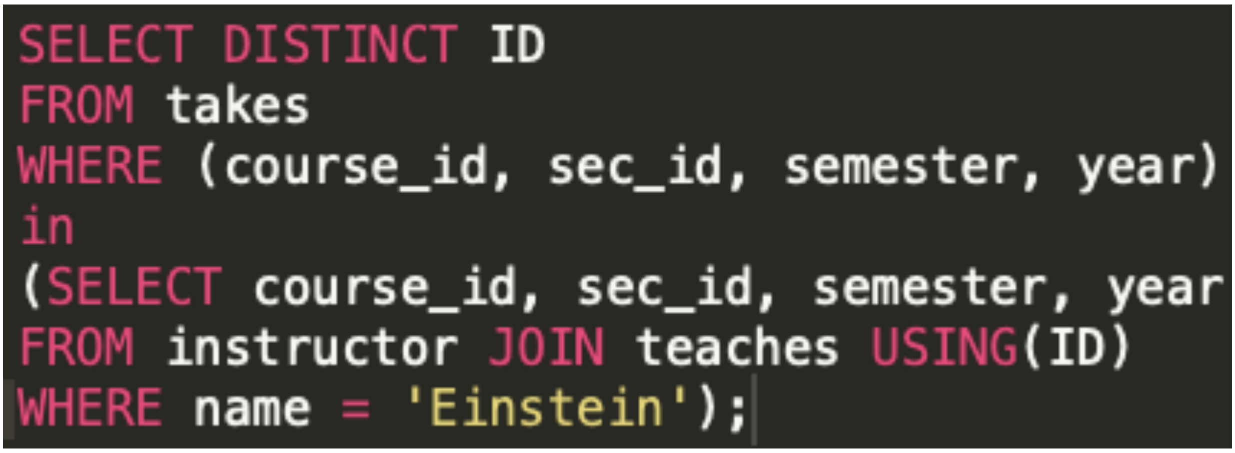


```
practice=# SELECT title
practice-# FROM course
practice-# WHERE dept_name = 'Comp. Sci.' and credits = 3;
```

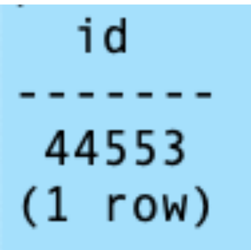


```
          title
--------------------------
 Robotics
 Image Processing
 Database System Concepts
(3 rows)
```

b. Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result

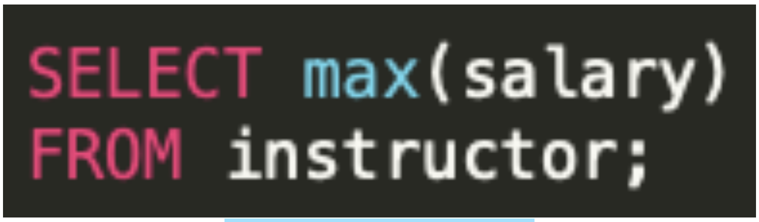


```
SELECT DISTINCT ID
FROM takes
WHERE (course_id, sec_id, semester, year)
in
(SELECT course_id, sec_id, semester, year
FROM instructor JOIN teaches USING(ID)
WHERE name = 'Einstein');
```

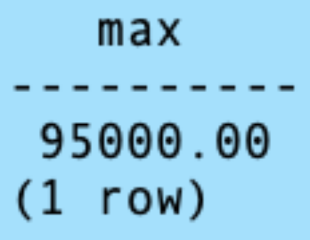


```
   id
--------
 44553
(1 row)
```

c. Find the highest salary of any instructor.

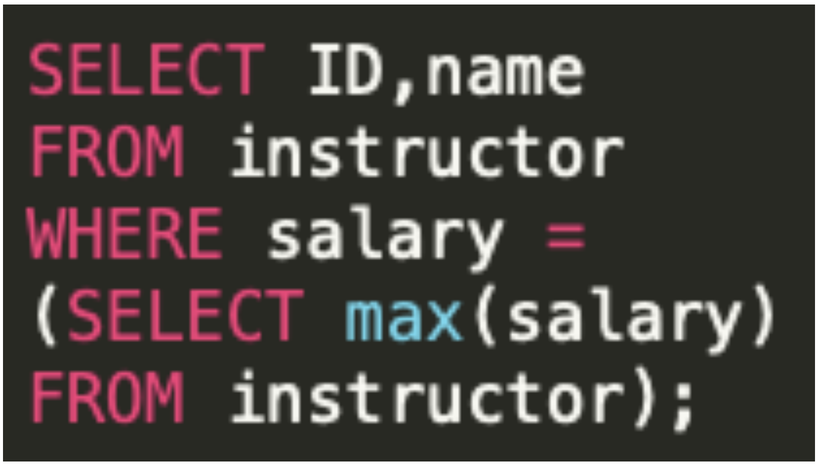


```
SELECT max(salary)
FROM instructor;
```

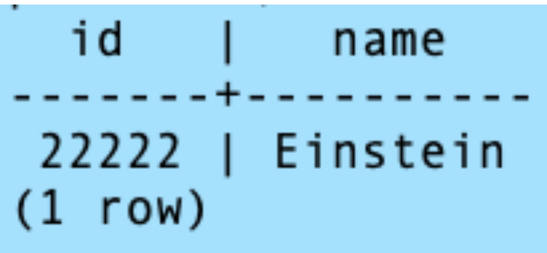


```
    max
-----------
 95000.00
(1 row)
```

d. Find all instructors earning the highest salary (there may be more than one with the same salary)

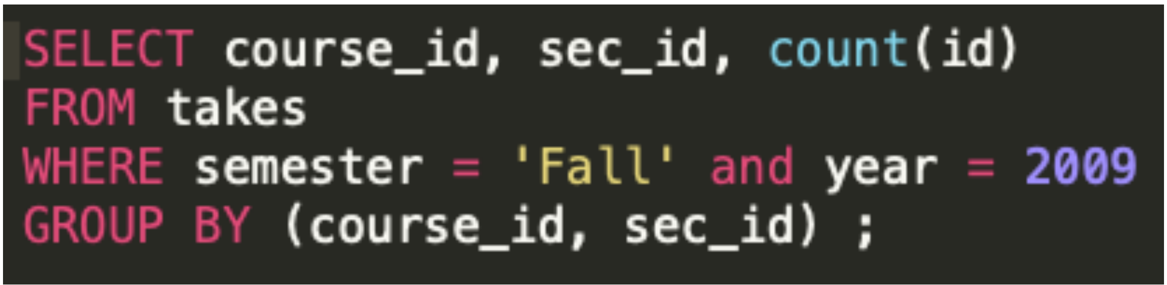


```
SELECT ID,name
FROM instructor
WHERE salary =
(SELECT max(salary)
FROM instructor);
```

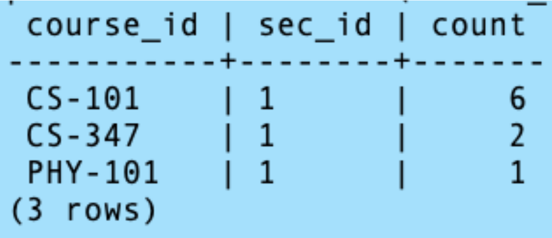


```
   id   |   name
-------+----------
 22222 | Einstein
(1 row)
```

e. Find the enrollment of each section that was offered in Autumn 2009

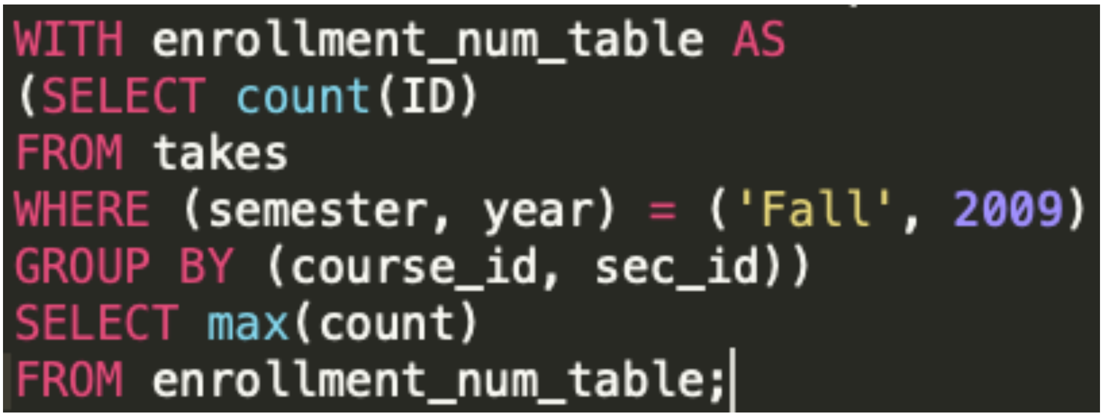


```
SELECT course_id, sec_id, count(id)
FROM takes
WHERE semester = 'Fall' and year = 2009
GROUP BY (course_id, sec_id) ;
```

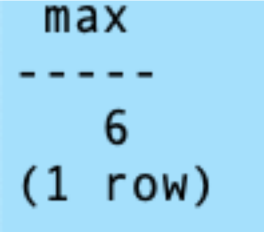


```
 course_id | sec_id | count
-----------+--------+-------
 CS-101    | 1      |     6
 CS-347    | 1      |     2
 PHY-101   | 1      |     1
(3 rows)
```

f. Find the maximum enrollment, across all sections, in Autumn 2009

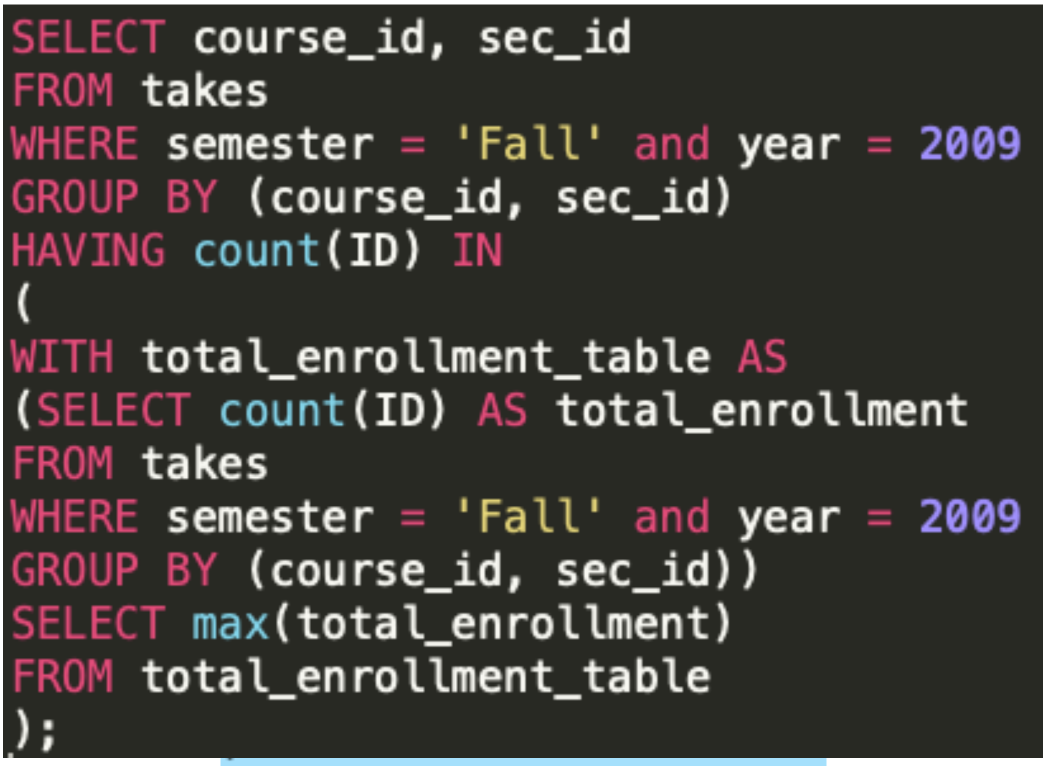


```
WITH enrollment_num_table AS
(SELECT count(ID)
FROM takes
WHERE (semester, year) = ('Fall', 2009)
GROUP BY (course_id, sec_id))
SELECT max(count)
FROM enrollment_num_table;
```

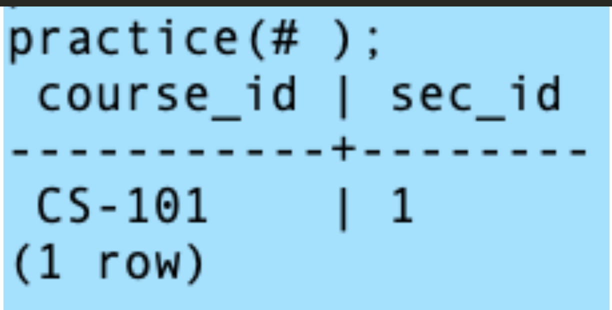


```
  max
------
    6
(1 row)
```

g. Find the sections that had the maximum enrollment in Autumn 2009



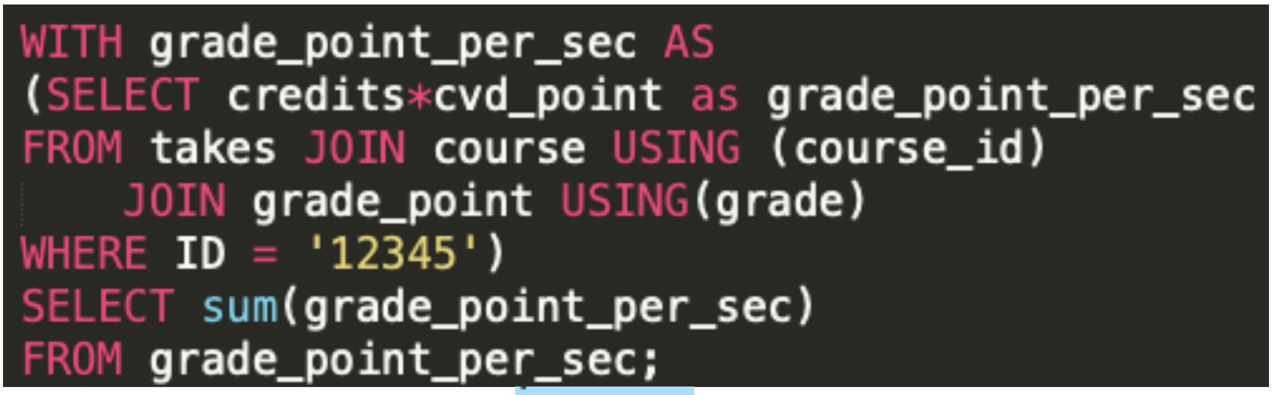
```
SELECT course_id, sec_id
FROM takes
WHERE semester = 'Fall' and year = 2009
GROUP BY (course_id, sec_id)
HAVING count(ID) IN
(
WITH total_enrollment_table AS
(SELECT count(ID) AS total_enrollment
FROM takes
WHERE semester = 'Fall' and year = 2009
GROUP BY (course_id, sec_id))
SELECT max(total_enrollment)
FROM total_enrollment_table
);
```



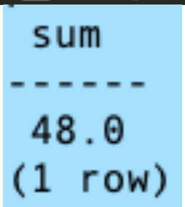
```
practice(# );
 course_id | sec_id
-----------+--------
 CS-101    | 1
(1 row)
```

Practice Exercise 2

a. Find the total grade-points earned by the student with ID 12345, across all courses taken by the student

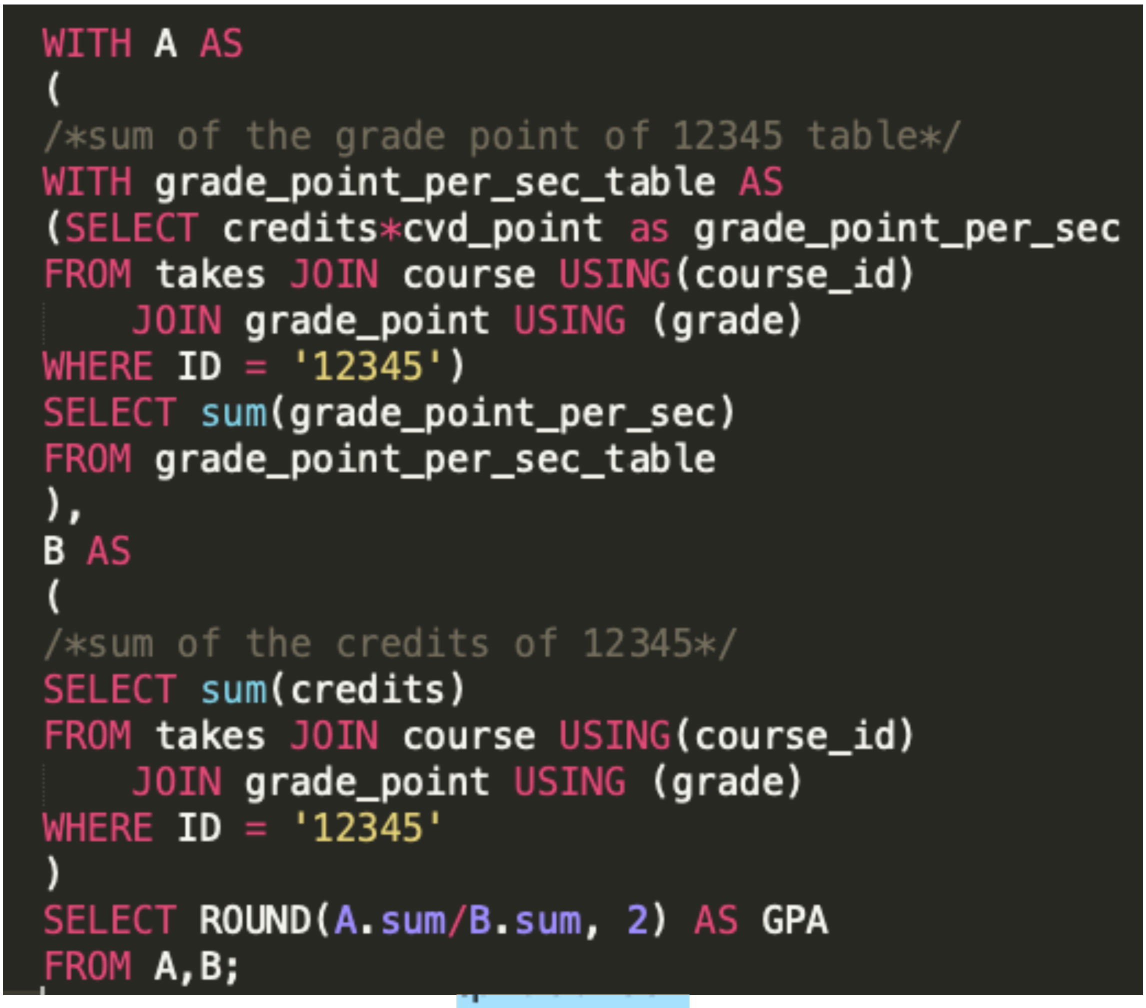


```
WITH grade_point_per_sec AS
(SELECT credits*cvd_point as grade_point_per_sec
FROM takes JOIN course USING (course_id)
    JOIN grade_point USING(grade)
WHERE ID = '12345')
SELECT sum(grade_point_per_sec)
FROM grade_point_per_sec;
```

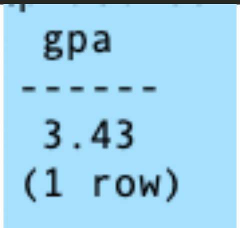


```
 sum
------
 48.0
(1 row)
```

b. Find the grade-point average (GPA) for the above student, that is, the total grade-points divided by the total credits for the associated courses.

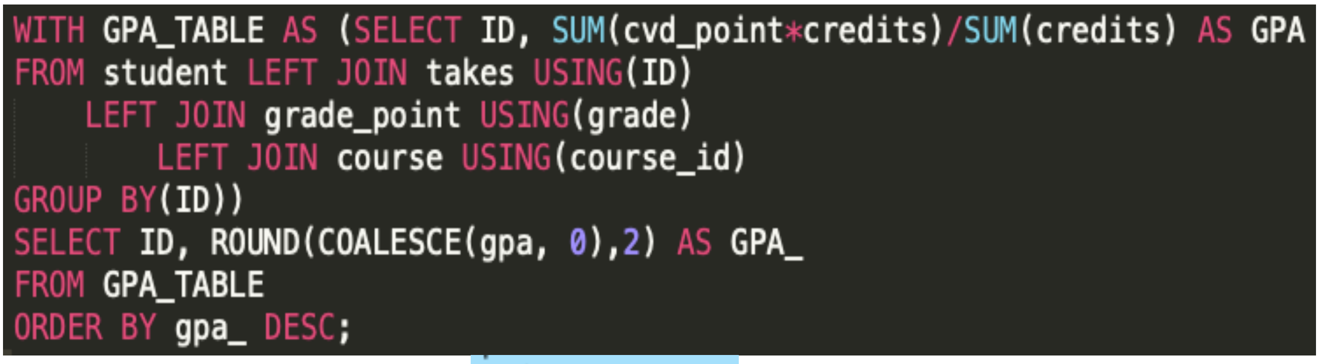


```
WITH A AS
(
/*sum of the grade point of 12345 table*/
WITH grade_point_per_sec_table AS
(SELECT credits*cvd_point as grade_point_per_sec
FROM takes JOIN course USING(course_id)
    JOIN grade_point USING (grade)
WHERE ID = '12345')
SELECT sum(grade_point_per_sec)
FROM grade_point_per_sec_table
),
B AS
(
/*sum of the credits of 12345*/
SELECT sum(credits)
FROM takes JOIN course USING(course_id)
    JOIN grade_point USING (grade)
WHERE ID = '12345'
)
SELECT ROUND(A.sum/B.sum, 2) AS GPA
FROM A,B;
```

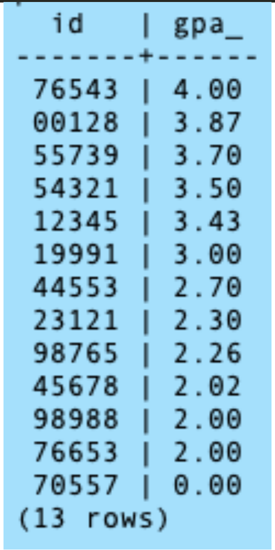


```
 gpa
------
 3.43
(1 row)
```

c. Find the ID and the grade-point average of every student?



```
WITH GPA_TABLE AS (SELECT ID, SUM(cvd_point*credits)/SUM(credits) AS GPA
FROM student LEFT JOIN takes USING(ID)
    LEFT JOIN grade_point USING(grade)
        LEFT JOIN course USING(course_id)
GROUP BY(ID))
SELECT ID, ROUND(COALESCE(gpa, 0),2) AS GPA_
FROM GPA_TABLE
ORDER BY gpa_ DESC;
```

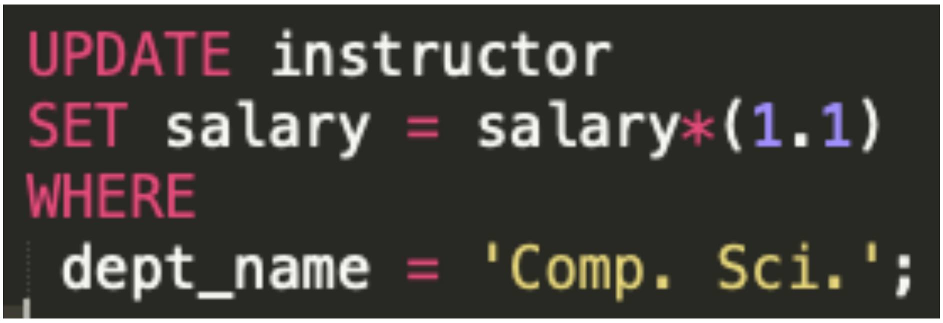


```
  id    | gpa_
-------+------
 76543 | 4.00
 00128 | 3.87
 55739 | 3.70
 54321 | 3.50
 12345 | 3.43
 19991 | 3.00
 44553 | 2.70
 23121 | 2.30
 98765 | 2.26
 45678 | 2.02
 98988 | 2.00
 76653 | 2.00
 70557 | 0.00
(13 rows)
```

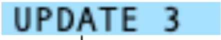
Practice Exercise 3

Write the following inserts, deletes or updates in SQL, using the university schema.

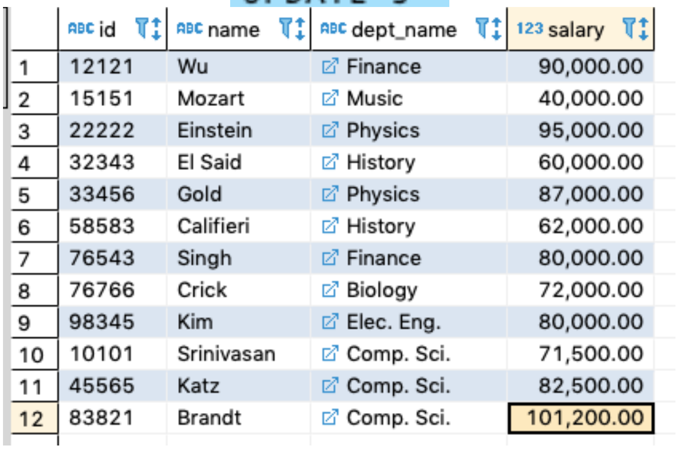
a. Increase the salary of each instructor in the Comp. Sci. department by 10%



```
UPDATE instructor
SET salary = salary*(1.1)
WHERE
 dept_name = 'Comp. Sci.';
```

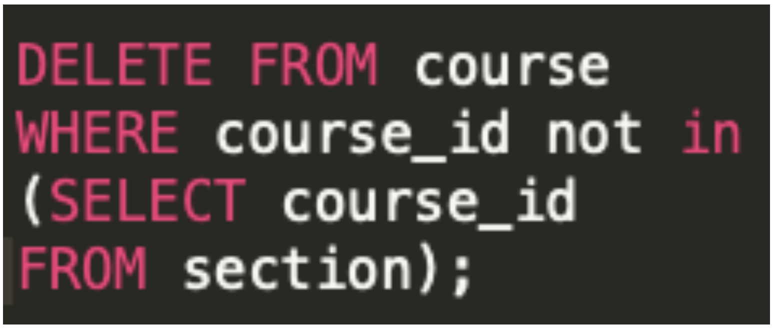


```
UPDATE 3
```

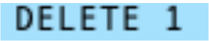


	id	name	dept_name	salary
1	12121	Wu	Finance	90,000.00
2	15151	Mozart	Music	40,000.00
3	22222	Einstein	Physics	95,000.00
4	32343	El Said	History	60,000.00
5	33456	Gold	Physics	87,000.00
6	58583	Califieri	History	62,000.00
7	76543	Singh	Finance	80,000.00
8	76766	Crick	Biology	72,000.00
9	98345	Kim	Elec. Eng.	80,000.00
10	10101	Srinivasan	Comp. Sci.	71,500.00
11	45565	Katz	Comp. Sci.	82,500.00
12	83821	Brandt	Comp. Sci.	101,200.00

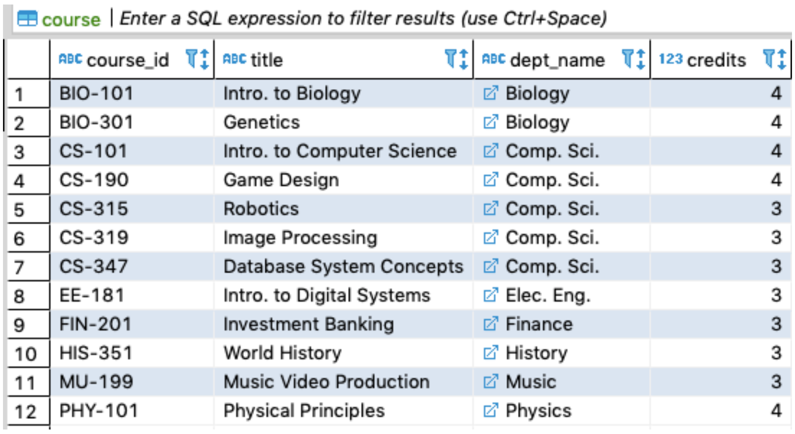
b. Delete all courses that have never been offered (that is, do not occur in the section relation)



```
DELETE FROM course
WHERE course_id not in
(SELECT course_id
FROM section);
```



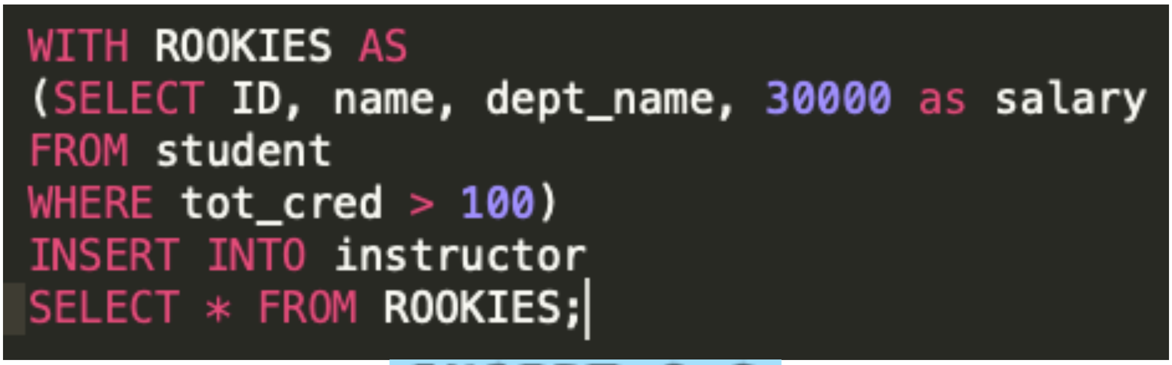
```
DELETE 1
```



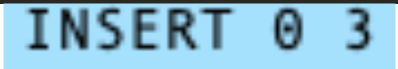
course | *Enter a SQL expression to filter results (use Ctrl+Space)*

	course_id	title	dept_name	credits
1	BIO-101	Intro. to Biology	Biology	4
2	BIO-301	Genetics	Biology	4
3	CS-101	Intro. to Computer Science	Comp. Sci.	4
4	CS-190	Game Design	Comp. Sci.	4
5	CS-315	Robotics	Comp. Sci.	3
6	CS-319	Image Processing	Comp. Sci.	3
7	CS-347	Database System Concepts	Comp. Sci.	3
8	EE-181	Intro. to Digital Systems	Elec. Eng.	3
9	FIN-201	Investment Banking	Finance	3
10	HIS-351	World History	History	3
11	MU-199	Music Video Production	Music	3
12	PHY-101	Physical Principles	Physics	4

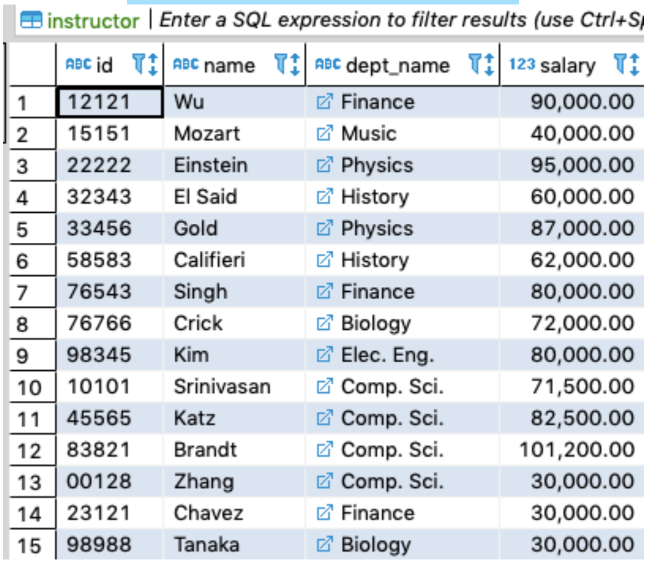
c. Insert every student whose tot_cred attribute is greater than 100 as an instructor in the same department, with a salary of $30,000



```
WITH ROOKIES AS
(SELECT ID, name, dept_name, 30000 as salary
FROM student
WHERE tot_cred > 100)
INSERT INTO instructor
SELECT * FROM ROOKIES;
```



```
INSERT 0 3
```



instructor | *Enter a SQL expression to filter results (use Ctrl+S*

	id	name	dept_name	salary
1	12121	Wu	Finance	90,000.00
2	15151	Mozart	Music	40,000.00
3	22222	Einstein	Physics	95,000.00
4	32343	El Said	History	60,000.00
5	33456	Gold	Physics	87,000.00
6	58583	Califieri	History	62,000.00
7	76543	Singh	Finance	80,000.00
8	76766	Crick	Biology	72,000.00
9	98345	Kim	Elec. Eng.	80,000.00
10	10101	Srinivasan	Comp. Sci.	71,500.00
11	45565	Katz	Comp. Sci.	82,500.00
12	83821	Brandt	Comp. Sci.	101,200.00
13	00128	Zhang	Comp. Sci.	30,000.00
14	23121	Chavez	Finance	30,000.00
15	98988	Tanaka	Biology	30,000.00