Theodore Kim
JinZhao Su
CS-UY 3083: Introduction to Databases
Homework #3

1. What’s the most episodes any tv series had?

```
SELECT MAX(max_eps) FROM tvseries
```

2. Find the TV series that has the most episodes. Results should show the TV series and the number of episodes

```
SELECT title, max_eps
FROM tvseries
WHERE max_eps = (SELECT MAX(max_eps)
                       FROM tvseries)
```

3. Find the number of TV series made before the year 2000

```sql
SELECT COUNT(*)
FROM tvseries
WHERE year < 2000
```

4. Find the first and last names of persons who favorite TV series is an old TV series. (series made before 2000)

```
SELECT first_name, last_name
FROM favorite
WHERE year < 2000
```

5. Find the first name and last name of each person who watches more than 5 TV series.

```
SELECT first_name, last_name
FROM watches
GROUP BY first_name, last_name
HAVING COUNT(*) > 5
```

6. Find the number of students who’ve taken CS-101

```
SELECT COUNT(DISTINCT ID, course_id)
FROM takes
WHERE course_id = “CS-101”
```

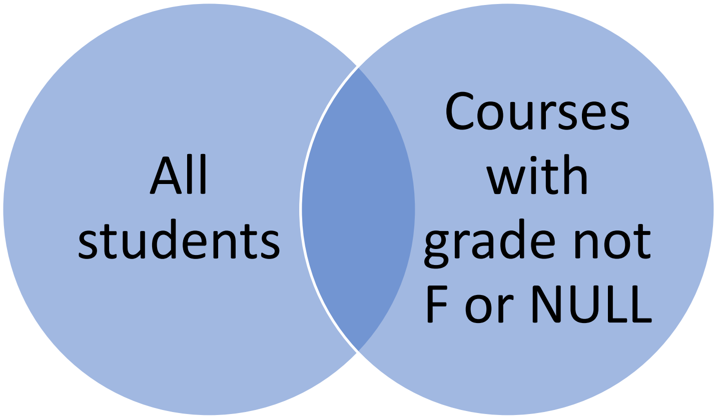
7. Find the number of students who’ve gotten each grade in CS-101. The result should be a relation with attributes named grade and num.

```
SELECT grade, COUNT(*) AS num
FROM takes
WHERE course_id=“CS-101”
GROUP BY grade
```

8. Find the number of courses each students has taken. The result should be a relation with attributes named ID and num.

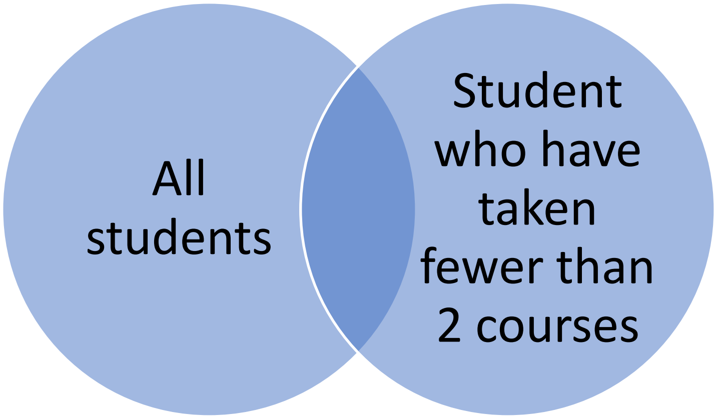
```
SELECT ID, COUNT(DISTINCT ID, course_id) AS num
FROM takes
GROUP BY ID
```

9. Find the total number of credits each student has passed (grades other than F or NULL), based on the information in the takes table (not on the student.tot cred attribute)



```
SELECT t.ID, SUM(c.credits) AS passed_cred
FROM takes AS t
JOIN course AS c ON c.course_id = t.course_id
WHERE t.grade IS NOT NULL AND t.grade != “F”
GROUP BY t.ID
```

10. Find IDs and names of students in the Comp Sci department who have taken fewer than two courses



```
SELECT DISTINCT s.ID, s.name
FROM student AS s
JOIN (SELECT ID, COUNT(*) AS num_courses FROM takes GROUP BY ID) AS t
ON t.ID = s.ID
WHERE s.dept_name = “Comp. Sci.” AND t.num_courses < 2
```

11. Find the average number of credits of courses in each department. The result should be a relation with attributes dept_name and avg_credits

```
SELECT dept_name, AVG(credits) AS avg_credits
FROM course
GROUP BY dept_name
```

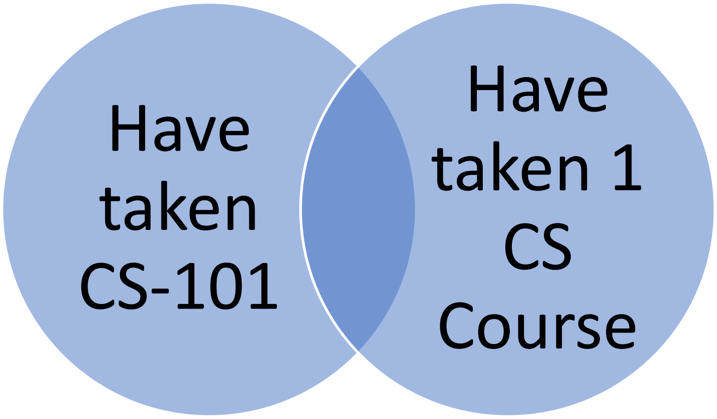
12. Find the number of sections taught in each building, each year.

```
SELECT building, year, COUNT(*) AS sections_taught
FROM section
GROUP BY building, year
```

13. Find the IDs of students who’ve taken ‘CS-315’ and any course with the ’Bio’ prefix (LIKE ’Bio%’)

```
SELECT DISTINCT ID
FROM takes
WHERE course_id=“CS-315” AND course_id LIKE “Bio%”
```

14. Find the IDs of students who've taken 'CS-101' but have not taken any other 'CS' courses (LIKE 'CS%')

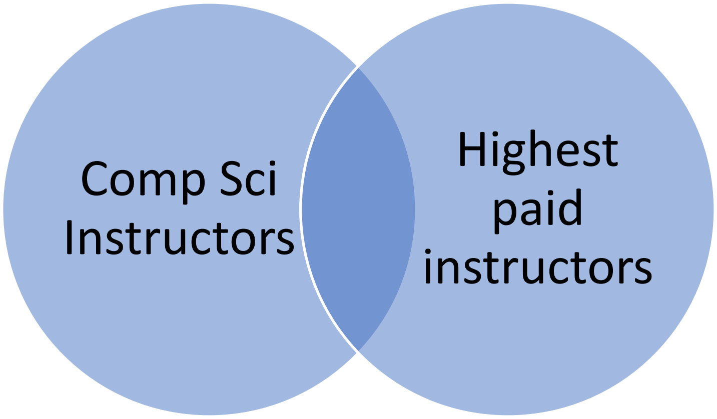


```
SELECT DISTINCT t1.ID
FROM takes AS t1
JOIN (SELECT ID, COUNT(*) AS cscount FROM takes WHERE course_id LIKE
“CS%” GROUP BY ID) AS t2
ON t1.ID = t2.ID
WHERE t1.course_id = “CS-101” AND t2.cscount = 1
```

15. Find the maximum salary of an instructor in the Physics department

```
SELECT MAX(salary) FROM instructor WHERE dept_name = “Physics”
```

16. Find the name of the highest paid instructor in the Comp Sci department. (If there is a tie, report all of them.)



```
SELECT name
FROM instructor AS t1
JOIN (SELECT dept_name, MAX(salary) AS max_salary
      FROM instructor
      GROUP BY dept_name) AS t2
ON t2.dept_name = t1.dept_name AND t2.max_salary = t1.salary
WHERE t1.dept_name = “Comp. Sci.”
```

17. Find the name of the student who got the most ’A’ grades

```
SELECT student.name
FROM student
JOIN takes ON takes.ID = student.ID
WHERE takes.grade = “A”
GROUP BY student.name
HAVING COUNT(*) = (SELECT MAX(num) AS max_a
                  FROM (SELECT ID, COUNT(*) AS num
                        FROM takes
                        WHERE grade = "A"
                        GROUP BY ID) AS t1)
```

18. Find the department with the highest average salary

```
SELECT dept_name
FROM instructor
GROUP BY dept_name
HAVING AVG(salary) = (SELECT MAX(avg_salary)
                      FROM (SELECT AVG(salary) AS avg_salary
                           FROM instructor
                           GROUP BY dept_name) AS t1)
```