**Homework Assignment 1-3**

Due: 11:59PM Thursday, March 26, 2020

|  |
| --- |
| * This is an individual work; Please be clear with HGU CSEE Standard:   + Submitting assignments or program codes written by others or acquired from the internet without explicit approval of the professor is regarded as cheating.   + Showing or lending one’s own homework to other student is also considered cheating that disturbs fair evaluation and hinders the academic achievement of the other student.   + It is regarded as cheating if two or more students conduct their homework together and submit it individually when the homework is not a group assignment. * Read the assignment carefully. *In this assignment, you will need to* ***write and execute several SQL queries****; and* ***submit the results of your queries***. * You are **allowed to re-use any of the queries from the lecture slides** while developing solutions to the problems. * When finished, submit your work to *LMS.* |

In this assignment **we use the "*university*" database** (NOT *university\_small*), which shares the same schemas with the database we use during the lectures but contains a larger set of data records collected within a different period of time.

All queries should be complete to obtain the listed answers solely by themselves.

1. (1 pt.) Find the number of all courses offered in Fall and Spring, respectively.

Answer:

Spring : 45

Fall : 46

Query to obtain your answer:

SELECT *count*(DISTINCT course\_id) FROM section WHERE semester = 'Spring'  
UNION  
SELECT *count*(DISTINCT course\_id) FROM section WHERE semester = 'Fall';

시계, 검은색, 측정기, 하얀색이(가) 표시된 사진

자동 생성된 설명

2. (1 pt.) Find all students who have “**db**” as a substring in their name.

Answer:

Goldbu, Sandberg, and Sandberg(person with the same name, different ID)

Query to obtain your answer:

SELECT ID, name FROM student WHERE name like '%db%';

텍스트, 실외, 점수판, 거리이(가) 표시된 사진

자동 생성된 설명

3. (4 pt.) How many **instructors** are in the *university*? How many **instructors** are in the *Statistics* department? How many **students** are in the *Geology* department?

Answer:

**50 instructors** are in the university.

**6 instructors** are in the *Statistics* department.

**92 students** are in the *Geology* department.

Query to obtain your answer:

SELECT *count*(DISTINCT ID) FROM instructor  
UNION  
SELECT *count*(DISTINCT ID) FROM instructor WHERE dept\_name = 'Statistics'  
union  
SELECT *count*(DISTINCT ID) FROM student WHERE dept\_name = 'Geology';

실외, 거리, 검은색, 표지판이(가) 표시된 사진

자동 생성된 설명

4. (4 pt.) How many **student names** are in the *Marketing* department? How many **unique student names** are in the *Astronomy* department? How many **unique course names (titles)** are among the courses offered by the *university*?

Answer:

**85 student names** are in the *Marketing* department.

**104 unique student names** are in the *Astronomy* department.

**133 unique course names (titles)** are among the courses offered by the *university.*

Query to obtain your answer:

SELECT *COUNT*(name) FROM student WHERE dept\_name = 'Marketing'  
union  
SELECT *COUNT*(DISTINCT name) FROM student WHERE dept\_name = 'Astronomy'  
union  
SELECT *COUNT*(DISTINCT title) FROM course;

텍스트, 거리, 검은색, 표지판이(가) 표시된 사진

자동 생성된 설명

5. (3 pt.) Write a query that counts the **number of students for each department** and sort the results in **descending order of the student counts**. *Hint: the head of the query result looks like the following:*

A screenshot of a cell phone

Description automatically generated

Query (you do not need to submit your query result):

SELECT dept\_name, *COUNT*(DISTINCT ID) AS num\_students  
FROM student GROUP BY dept\_name ORDER BY *COUNT*(DISTINCT ID) DESC;

6. (5 pt.) What is the **average monthly salary** of the instructors in the *Cybernetics* department? What is the **highest average monthly salary measured at the department level**? For both questions, please round the answers at the second decimal place, if necessary.

Answer:

**average monthly salary** of the instructors in the *Cybernetics* department : 8028.9

**highest average monthly salary measured at the department level : 6466.7**

Query to obtain your answer:

SELECT *AVG*(salary/12) AS average\_monthly\_salary  
FROM instructor where dept\_name = 'Cybernetics'  
union   
SELECT *AVG*(salary/12) AS highest\_average\_monthly\_salary  
FROM instructor;

표지판, 검은색, 하얀색, 거리이(가) 표시된 사진

자동 생성된 설명

7. (3 pt.) Write a query that **lists up all classes** that have been open in the university, together **with the number of students** who were in each class. More specifically, enumerate all the *course IDs, section IDs, years, and semesters*, along with the *number of students who took each of the classes*. *Hint: you may want to come up with a result that starts as below.*

A close up of a white background

Description automatically generated

Query (you do not need to submit your query result):

SELECT course\_id, sec\_id, semester, year, *COUNT*(\*)  
FROM takes GROUP BY course\_id, sec\_id, semester, year;

8. (4 pt.) Find all *Physics* and *Comp. Sci.* students whose **name is longer than 11 characters**.

Answer:

*Physics* student : Krishnakumar / *Comp. Sci.* students : 0

Query to obtain your answer:

select name AS Physics FROM student  
where dept\_name = 'Physics' AND name Like '\_\_\_\_\_\_\_\_\_\_\_\_%'  
UNION  
select name AS Com FROM student  
where dept\_name = 'Comp. Sci.' AND name Like '\_\_\_\_\_\_\_\_\_\_\_\_%';

거리, 검은색, 표지판, 앉아있는이(가) 표시된 사진

자동 생성된 설명

9. (4 pt.) Which of the university **buildings** can accommodate **more than 100 people**? *Hint: see the sum of classroom capacities.*

Answer:

Saucon, Stabler, Taylor, Whitman can accommodate **more than 100 people.**

Query to obtain your answer:

select building, *SUM*(capacity) FROM classroom  
GROUP BY building HAVING *SUM*(capacity) > 100;

거리, 검은색, 표지판, 시계이(가) 표시된 사진

자동 생성된 설명

10. (4 pt.) Find the number of *Comp. Sci.* student total credits greater than that of **SOME** (at least one) students in the *English* department.

Answer:

99 *Comp. Sci.* student’s total credits are greater than that of **SOME** (at least one) students in the *English* department.

Query to obtain your answer:

SELECT *COUNT*(DISTINCT ID)  
FROM student  
WHERE dept\_name = 'Comp. Sci.' AND tot\_cred > SOME(SELECT tot\_cred  
 FROM student  
 WHERE dept\_name = 'English');

스크린샷, 검은색, 표지판, 하얀색이(가) 표시된 사진

자동 생성된 설명

11. (6 pt.) Which of the university **buildings** has the **highest number of classrooms**? **How many classrooms** in that building? **Write a single query** that returns both information. *Hint: use the classroom table.*

Answer:

Power **buildings** has the **highest number of classrooms.**

1689 **classrooms** in that building.

Query to obtain your answer:

SELECT building, *MAX*(sum)  
FROM (SELECT building AS build, *SUM*(room\_number) AS sum  
 FROM classroom  
 GROUP BY building) AS tmp  
 , classroom  
WHERE tmp.build = classroom.building  
GROUP BY building ORDER BY sum DESC;

점수판, 텍스트, 검은색, 앉아있는이(가) 표시된 사진

자동 생성된 설명

12. (6 pt.) Write a query that lists the students who have earned the **highest total credits** (tot\_cred) **in each department**. If there are ties, include all the tied students in the result. Your query should return a relation with the department name, student name, and the total credits they earned; and the result should be sorted by the department name and student name in ascending order. *Hint: the result may start as below.*

A screenshot of a cell phone

Description automatically generated

Query (you do not need to submit your query result):

SELECT dept\_name, name, top.B  
FROM (SELECT dept\_name AS A, *MAX*(tot\_cred) AS B  
 FROM student  
 GROUP BY dept\_name  
 ) AS top  
 ,student  
WHERE top.A = dept\_name AND top.B = tot\_cred  
GROUP BY dept\_name, name ORDER BY dept\_name ASC;

13. (5 pt.) Find all **instructor IDs** who had *taught until 2003* but *had not taught after 2003*. *Hint: Attribute teaches.ID is the instructor ID.*

Answer:

Instructor ID : 25946, 15347, 90643, 80759, 73623, 42782 had taught until 2003 but not had taught after 2003.

Query to obtain your answer:

SELECT DISTINCT ID  
FROM teaches  
WHERE year <= 2003  
AND ID NOT IN (SELECT DISTINCT ID  
 FROM teaches  
 WHERE year > 2003);

스크린샷이(가) 표시된 사진

자동 생성된 설명