Homework Assignment 2

Maximum earnable: 100 pt.
Due: 11:59PM April 7, 2023

- Read the assignment carefully. *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.
- 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.
- Posting any of the assignment problems on the Internet and asking the solutions is prohibited.
- When finished, submit your work to *LMS.*

Read Chapters 3 of Database System Concepts and answer the following questions.

1. (1 pt. per blank) Fill in the blanks.

(a) The ( ) provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
(b) The ( ) provides the ability to query information from the database and to insert tuples into, delete tuples from, and modify tuples in the database.
(c) The primary key attributes are required to be ( ) and ( ).
(d) The ( ) specifies that the values of attributes for any record in the relation must correspond to values of the primary key attributes of some tuple in another relation.
(e) Subqueries that return only one tuple containing a single attribute are called ( ).
(f) The ( ) clause causes the records in the result of a query to appear in sorted order.
(g) The ( ) clause provides a way of defining a temporary relation whose definition is available only to the query in which the clause occurs.

2. (4 pt.; Exercise problem 3.6) The SQL **LIKE** operator is case sensitive (in most systems), but the **LOWER**() function on strings can be used to perform case-insensitive matching. Show how to write a query that finds departments whose names contain the string "sci" as a substring, regardless of the case.

3. (4 pt.; Exercise problem 3.20) Show that, in SQL, **<> ALL** is identical to **NOT IN**.

4. (4 pt.; Exercise problem 3.19) List two reasons why null values might be introduced into the database.

5. (2 pt. each; based on Exercise problems 3.9, 3.10, 3.16, and 3.17) Consider the relational database of Figure 3.19, where the primary keys are underlined. Given an expression in SQL for each of the following queries.

a. Find the ID of each employee who does not work for "First Bank Corporation".
SQL Query:

b. Find the ID, name, and city of residence of each employee who works for "First Bank Corporation" and earns more than $10,000.
SQL Query:

c. Find the ID of each employee who earns more than every employee of "Small Bank Corporation".
SQL Query:

d. Assume that companies may be located in several cities. Find the name of each company that is located in every city in which "Small Bank Corporation" is located. Your query should run on MySQL.
SQL Query:

e. Find the name of the company that has the most employees (or companies, if there is a tie).
SQL Query:

f. Find the name of each company whose employees earn a higher salary on average, than the average salary at "First Bank Corporation".
SQL Query:

g. Modify the database so that the employee whose ID is '12345' now lives in a city called "Newtown".
SQL Query:

h. Find ID and name of employee who lives in the same city as the location of the company for which the employee works.
SQL Query:

i. Find ID and name of each employee who earns more than the average salary of all employees of her or his company.
SQL Query:

j. Find the company that has the smallest payroll (sum of all salary in a company).
SQL Query:

k. Given all employees of "First Bank Corporation" a 10 percent raise.
SQL Query:

l. Delete all tuples in the works relation for employees of "Small Bank Corporation".
SQL Query:

Launch and access the MySQL databases distributed with the class Docker image. Below uses the "**university**" database (NOT *university_small*), which shares the same schemas with the database used in the lectures but contains a larger set of data records collected within a different period of time.

6. (3 pt. each) Find the answers to the following questions and provide the SQL queries showing how you find them. All queries should be complete to obtain the listed answers solely by themselves.

a. Find the number of all courses offered in Fall and that of Spring, respectively.
Answer:

SQL Query to obtain your answer:

b. How many unique course names (titles) are among the courses offered by the *university*?
Answer:

SQL Query to obtain your answer:

c. What is the average monthly salary of the instructors in the *Cybernetics* department? Round the answers at the second decimal place, if necessary.
* ROUND(): https://dev.mysql.com/doc/refman/8.0/en/mathematical-functions.html#function_round
Answer:

SQL Query to obtain your answer:

d. Find the names of departments whose budget is higher than that of *Psychology*. List them in alphabetic order.
Answer:

SQL Query to obtain your answer:

e. List the names of the students in the *Geology* department whose name starting with 'C'.
Answer:

SQL Query to obtain your answer:

f. (Exercise problem 3.29) Find the ID and name of each *History* student whose name begins with the letter 'D' and who has not taken at least five *Music* courses.
Answer:

SQL Query to obtain your answer:

g. Find all *Physics* and *Comp. Sci.* students whose name is longer than 11 characters.
Answer:

SQL Query to obtain your answer:

h. Find the number of *Comp. Sci.* students total credits greater than that of AT LEAST ONE student in the *English* department.
Answer:

SQL Query to obtain your answer:

i. Which of the university buildings can accommodate more than 100 people? *Hint: see the sum of classroom capacities.*
Answer:

SQL Query to obtain your answer:

j. Find **all instructor ID**s who had *taught until 2003* but *had not taught after 2003*.
Hint: Attribute teaches.ID is the instructor ID.
Answer:
SQL Query to obtain your answer:

k. 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:*

dept_name	num_students
Civil Eng.	120
Languages	119
History	117
Pol. Sci.	109
Comp. Sci.	108

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

l. (Exercise problem 3.22) Rewrite the WHERE clause

WHERE UNIQUE (SELECT title FROM course)

without using the UNIQUE construct.

Answer:

7. (4 pt. each) Find the answers to the following questions and provide the SQL queries showing how you find them. All queries should be complete to obtain the listed answers solely by themselves.

a. 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.

course_id	sec_id	semester	year	num_students
105	1	Fall	2009	327
105	2	Fall	2002	307
137	1	Spring	2002	304
158	1	Fall	2008	291
158	2	Spring	2008	286
169	1	Spring	2007	300
169	2	Fall	2002	280
192	1	Fall	2002	338
200	1	Spring	2007	299
200	2	Fall	2002	292
237	1	Spring	2008	302

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

b. (Exercise problem 3.26) For each student who has retaken a course at least twice (*i.e.*, the student has taken the course at least three times), show the course ID and the student's ID. Please display your results in order of course ID and do not display duplicate rows.
Answer:

SQL Query to obtain your answer:

c. (Based on exercise problem 3.31) Find the ID and name of each instructor who has never given an A grade in any course s/he has taught. Order result by name.
Answer:

SQL Query to obtain your answer:

d. (Based on exercise problem 3.28) Find the names of the instructors who teach every course taught in his/her department. Order result in reverse alphabetical order.
Answer:

SQL Query to obtain your answer:

e. (Exercise problem 3.30) Consider the following SQL query on the university schema:

```
SELECT AVG(salary) - (SUM(salary)/COUNT(*))
FROM instructor;
```

We might expect that the result of this query is zero since the average of a set of numbers is defined to be the sum of the numbers divided by the number of numbers. Indeed, this is true for the example *instructor* relation in Figure 2.1. However, there are other possible instances of that relation for which the result would NOT be zero. Give one such instance, and explain why the results would not be zero.
Answer: