Homework Assignment 4

Maximum earnable: 83 pt.
Due: 11:59PM, May 8, 2025

- 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.
- **Use of ChatGPT or similar AI tools**: Students are prohibited from using ChatGPT or similar AI platforms to directly obtain solutions for this assignment. The intent of the assignment is to exercise your understanding and application of the course material. Leveraging AI tools to bypass this learning process is considered a breach of academic integrity. Any evidence of such behavior will result in penalties.
- When finished, submit your work to *LMS.*

1. (4 pt. each) Join operations.

(a) (Exercise 4.1) Consider the following SQL query that seeks to find a list of titles of all courses taught in Spring 2017 along with the name of the instructor.

```
SELECT name, title
FROM instructor NATURAL JOIN teaches NATURAL JOIN section NATURAL JOIN course
WHERE semester = 'Spring' AND year = 2017;
```

What is wrong with this query?

(b) (Exercise 4.16) Write an SQL query using the university schema to find the ID of each student who has never taken a course at the university. Do this using no subqueries and no set operations (use an outer join).

(c) (Exercise 4.17) Express the following query in SQL using no subqueries and no set operations.

```
SELECT ID
FROM student
EXCEPT
SELECT s_id
FROM advisor
WHERE i_ID IS NOT NULL;
```

(d) (Exercise 4.20) Show how to define a view *tot_credits*(*year, num_credits*), giving the total number of credits taken in each year.

(e) (Exercise 4.21) For the view that you have defined in the previous problem (Problem 4(d)), explain why the database system would not allow a tuple to be inserted into the database through this view.

2. (3 pt. each) Consider the following database table schemas expressed in SQL queries.

```
CREATE TABLE city (
      ID           INT           NOT NULL,           # unique city ID
      Name         CHAR(35)      NOT NULL,           # name of city
      CountryCode  CHAR(3)       NOT NULL,           # country code
      Population   INT           NOT NULL,           # population of city
      PRIMARY KEY (ID),
      FOREIGN KEY (CountryCode) REFERENCES country (Code)
);

CREATE TABLE country (
      Code         CHAR(3)       NOT NULL,           # unique country code
      Name         CHAR(52)      NOT NULL,           # name of country
      Continent    VARCHAR(20)   NOT NULL,           # name of continent
      Population   INT           NOT NULL,           # Population of country
      LifeExpect   DECIMAL(3,1) DEFAULT NULL,        # Life expectancy of country
      PRIMARY KEY (Code),
      CHECK (Continent IN ('Asia','Europe','North America',
                           'Africa','Oceania','Antarctica',
                           'South America'))
);

CREATE TABLE countryLanguage(
      CountryCode  CHAR(3)       NOT NULL,           # country code
      Language     CHAR(30)      NOT NULL,           # language used in the country
      Percentage   DECIMAL(4,1) NOT NULL,            # percentage language used
      PRIMARY KEY (CountryCode, Language),
      FOREIGN KEY (CountryCode) REFERENCES country (Code)
);
```

(a) Write an SQL query that returns the number of distinct *countries* per *continent*.

Answer

(b) Write an SQL query that returns the number of distinct *countries* that use `Language = 'English'`.

Answer

(c) Write an SQL query that returns the number of distinct *countries* that use `Language = 'English'` per *continent*.

Answer

(d) Write an SQL query that lists distinct country names, within which one of the languages is used by more than 99 percent of time.

Answer

(e) Write an SQL query that increase *populations* of *countries* (`country.Population`) whose *population* is over 100,000,000 by 1% and all others by 2%.

Answer

(f) Write an SQL query that modifies the table schema of *city* and adds a new attribute named *Remark* whose type is VARCHAR(512).

Answer

3. Launch and access the MySQL databases distributed with the class virtual machine. 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 p eriod of time.
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) (2 pt.) Find the number of all courses offered in Fall and that of Spring, respectively.
Answer:

SQL Query to obtain your answer:

(b) (2 pt.) How many unique course names (titles) are among the courses offered by the *university*?
Answer:

SQL Query to obtain your answer:

(c) (2 pt.) 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) (3 pt.) 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) (4 pt.) Find the ID and name of each *History* student whose name begins with the letter 'D' and who has not taken at least three *Psychology* courses.
Answer:

SQL Query to obtain your answer:

(f) (3 pt.) Find all *Physics* and *Comp. Sci.* students whose name is longer than 11 characters.
Answer:

SQL Query to obtain your answer:

(g) (4 pt.) Find the number of *Comp. Sci.* student total credits greater than that of AT LEAST ONE student in the *English* department.
Answer:

SQL Query to obtain your answer:

(h) (4 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:

SQL Query to obtain your answer:

4. (3 pt. each) Given the following tables, evaluate the result of the following queries. Remember that the output of a query is a table.

* Table `AGENT`

AGENT_CODE	**AGENT_NAME**	**WORKING_AREA**	**COMMISSION**
A001	Subbarao	Bangalore	0.14
A002	Mukesh	Mumbai	0.11
A003	Alex	London	0.13
A004	Ivan	San Jose	0.15
A005	Anderson	Brisban	0.13
A006	McDen	London	0.15
A007	Ramasundar	Bangalore	0.15
A008	Alford	New York	0.12
A009	Benjamin	Hampshair	0.11
A010	Santakumar	Chennai	0.14
A011	Kumar	Bangalore	0.15
A012	Lucida	San Jose	0.12

* Table `AGENT_PHONE`

AGENT_CODE	**PHONE_NO**
A001	077-1
A001	077-2
A003	075-4
A003	075-5
A004	008-2
A004	008-3
A004	008-4
A010	007-2
A013	044-1

(a)
```
SELECT AGENT_NAME FROM AGENTS
WHERE COMMISSION > SOME(
    SELECT COMMISSION FROM AGENTS
    WHERE WORKING_AREA = 'Bangalore'
);
```

Answer

(b)
```
SELECT AGENT_NAME FROM AGENTS
WHERE COMMISSION IN(
    SELECT COMMISSION FROM AGENTS
    WHERE WORKING_AREA = 'Brisban'
);
```

Answer

(c)
```
SELECT AGENT_NAME FROM AGENTS
WHERE COMMISSION < ALL(
    SELECT COMMISSION FROM AGENTS
    WHERE WORKING_AREA = 'London'
);
```

Answer

(d)
```
SELECT AGENT_NAME
FROM AGENTS AS A
WHERE WORKING_AREA = 'San Jose' AND
    EXISTS (SELECT * FROM AGENTS AS B
            WHERE COMMISSION < 0.14 AND A.AGENT_CODE = B.AGENT_CODE)
```

Answer

(e)
```
SELECT COUNT(*)
FROM AGENTS
NATURAL JOIN AGENTS_PHONE;
```

Answer

(f)
```
SELECT COUNT(*)
FROM AGENTS a
LEFT JOIN AGENTS_PHONE p
ON a.AGENT_CODE = p.AGENT_CODE;
```

Answer

(g)
```
SELECT COUNT(*)
FROM AGENTS a
RIGHT JOIN AGENTS_PHONE p
ON a.AGENT_CODE = p.AGENT_CODE;
```

Answer