COMP 3005: Database Management Systems **(Due: Monday Oct. 7, 2019 (11:59 PM))**

Assignment #2

Instructor: Ahmed El-Roby *Name:* , *ID:*

Instructions: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- The accepted formats for your submission are: pdf, docx, txt, and java. More details below.
- You can either write your solutions in the tex file (then build to pdf) or by writing your solution by hand or using your preferred editor (then convert to pdf or docx). However, you are encouraged to write your solutions in the tex file (5% bonus). If you decide not to write your answer in tex, it is your responsibility to make sure you write your name and ID on the submission file.
- If you use the tex file, make sure you edit line 28 to add your name and ID. Only write your solution and do not change anything else in the tex file. If you do, you will be penalized.
- All questions in this assignment use the university schema discussed in class (on culearn), unless otherwise stated.
- For SQL questions, upload a text file with your queries in the format shown in the file "template.txt" uploaded on culearn. An example submission is in the file "sample.txt". You will be penalized if the format is incorrect or there is no text file submission.
- For programming questions, upload your .java file.
- Bonus questions are clearly marked by the keyword "BONUS" in the question title.
- Late submissions are allowed for 24 hours after the deadline above with a penalty of 10% of the total grade of the assignment. Submissions after more than 24 are not allowed.

Q 1: (2 points)

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).

Q 2: (2 points)

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;
```

Q 3 (BONUS): (2 points)

Consider the following relation definition:

```
create table manager(
emp_id         char(20),
manager_id     char(20),
primary key emp_id,
foreign key (manager_id) references manager(emp_id) on delete cascade)
```

The foreign key constraint means that every manager has to be an employee. Explain what is going to happen when a manager is deleted.

Q 4: (5 points)

Consider the following schema:
employee(ID, person_name, street, city
works(ID, company_name, salary)
company(company_name, city)
manages(ID, manager_id)
Write a query to find the ID of each employee with no manager:
(a) Using outer join. (2 marks)

(b) Without using outer join. (3 marks)

Q 5: (12 points)

Write a Java program that finds all prerequisites for a given course using JDBC. The program should:

- Takes a course id value as input using keyboard.
- Finds the prerequisites of this course through a SQL query.
- For each course returned, repeats the previous step until no new prerequisites can be found.
- Prints the results.

Don't forget to handle the case for cyclic prerequisites. For example, if course A is prerequisite to course B, course B is prerequisite to course C, and course C is prerequisite to course A, do not infinite loop.

Q 6: (5 points)

Consider the following schema:
employee(emp_name, street, city)
works(emp_name, company_name, salary)
Write a function *avg_sal* that takes a company name as input and finds the average salary of employees in the company. Then, write a SQL query that uses this function to find companies whose employees earn (on average) higher salary than the company "Losers Inc.".

Q 7 (BONUS): (3 points)

Consider the following schema:
branch(branch_name, branch_city, assets)
customer (customer_name, customer_street, customer_city)
loan (loan_number, branch_name, amount)
borrower (customer_name, loan_number)
account (account_number, branch_name, balance)
depositor (customer_name, account_number)
Write a trigger that enforces the following: On delete of an account, for each owner of the account, check if the owner has any remaining accounts, and if she does not, delete her from the depositor relation.