

## Assignment #1

Instructor: Ahmed El-Roby

Name: Ryan Lo, ID: 101117765

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

- The accepted format for your submission is **pdf** only. 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). However, you are encouraged to write your solutions in the tex file. 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 (available on Brightspace under Resources → University\_Toy\_Database), 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 (50% of the question) if the format is incorrect or there is no text file submission.**

**Q 1:**

(7 points)

Answer the following questions using the university schema discussed in class:

(a) The primary key for the *advisor* relation is *s\_id*. Suppose a student can have more than one supervisors. Would *s\_id* still be a primary key in *advisor*? If yes, why? If not, what would be a suitable primary key? (3 marks)

If a student could have more than one supervisor then *s\_id* would not be a primary key in *advisor*. It wouldn't be enough to uniquely identify an advisor, in this case a more suitable primary id would be both *s\_id* and *i\_id*.

(b) The primary key for *prereq* is both attributes *course\_id* and *prereq\_id*. Why wouldn't only *course\_id* work as primary key? (2 marks)

If only *course\_id* was the primary key, and a course requires you to have multiple courses as a prerequisite, *course\_id* would not be unique anymore and there would be multiple instances of *course\_id*. Therefore, having just *course\_id* as a primary would not work.

(c) Given the existing schema of *teaches*, two or more instructors can teach the same section. How can the primary key be changed to restrict a section to one instructor only? (2 marks)

Currently we have the instructor ID as the primary key which allows for multiple sections per instructor, changing the primary key in the *teaches* table to the section id, *sec\_id* instead would make the section unique and will only allow one section.

**Q 2:**

(12 points)

Consider the following bank database schema:

*branch*(*branch\_name*, *branch\_city*, *assets*)  
*customer*(*ID*, *customer\_name*, *customer\_street*, *customer\_city*)  
*loan*(*loan\_number*, *branch\_name*, *amount*)  
*borrower*(*ID*, *loan\_number*)

*account(account\_number, branch\_name, balance)*

*depositor(ID, account\_number)*

Write an expression in relational algebra to find the following:

- (a) Find the cities that host branches that have a loan that is greater than \$50000. (3 marks)

$$\pi_{branch\_city}(\sigma_{amount > 50000}(loan \bowtie_{loan.branch\_name=branch.branch\_name} (branch)))$$

- (b) Find the ID of each depositor who has an account with a balance greater than \$50000 at the “Ottawa” branch. (3 marks)

$$\pi_{ID}(\sigma_{branch\_name="Ottawa" \wedge balance > 50000}(depositor \bowtie_{depositor.account\_number=account.account\_number} (account)))$$

- (c) Find the names of customers who have at least one loan amount that is greater than at least one of their account balance. (6 marks)

$$\pi_{customer\_name}(\pi_{loan.amount > account.balance}(borrower \bowtie_{loan.loan\_number=borrower.loan\_number} \bowtie_{depositor.ID=borrower.ID} depositor \bowtie_{account\_number=depositor.account\_number} account \bowtie_{customer.ID=borrower.ID} customer)))$$

**Q 3:**

(33 points)

Using the university database schema discussed in class, write the SQL statements that do:

- (a) Create a new course (“Aces of Databases”) with ID (“COMP5118”) in the Computer Science department (“Comp. Sci.”) with 0 credit hours. (3 marks)

- (b) Create a section ‘A’ for this course in the Winter of 2020 with no known location or time, yet. (4 marks)

- (c) Enroll all students in the department into this course. (5 marks)

- (d) One student with ID 12345 cannot take this course because of violating the prerequisite requirements (didn’t pass COMP3005). Unregister this student from the new section. (3 marks)

- (e) For each student who took a course at least twice, show the course\_ID and the student ID. (5 marks)

- (f) Find the ID and name of instructors who never gave a grade ‘A’ in the courses they taught (note that instructors who never taught a course satisfy this condition). (5 marks)

- (g) Rewrite the previous query so that you make sure that the instructor taught at least one course. (5 marks)

- (h) Find the lowest, across all departments, of the per-department maximum salary. (3 marks)

**Q 4**

(5 points)

Consider the following car insurance schema:

*person(driver\_id, name, address)*

*car(licence, model, year)*

*accident(report\_number, date, location)*

*owns(driver\_id, licence\_plate)*

*participated(report\_number, licence\_plate, driver\_id, damage\_amount)*

Write SQL queries to:

- (a) Find the number of accidents involving a car belonging to a person named “Ahmed El-Roby”. (3 marks)
- (b) Update the damage amount for the car with licence plate “DB007” in the accident with report number “AR2020” to \$3000. (2 marks)