EE477 Database and Big Data Systems, Spring 2021
HW2*

Due date: 23/04/2021 (11:59pm)

Submission instructions: Use KAIST KLMS to submit your homeworks. Your submission should be one gzipped tar file whose name is `YourStudentID_hw2.tar.gz`. For example, if your student ID is 20210000, and it is for homework #2, please name the file as `20210000_hw2.tar.gz`. You can also use these extensions: tar, gz, zip, tar.zip. Do not use other options not mentioned here.

Your zip file should contain **two** things; one PDF file (`hw2.pdf`) and the Ethics Oath pdf file. Do not include Korean letters in any file name or directory name when you submit. **If you violate any of the file name format or extension format, we will deduct 1 point of total score per mistake.** *Submitting writeup*: Prepare answers to the homework questions into a single PDF file. You can use the following template. Please write as succinctly as possible.

Ethics Oath: For every homework submission, please fill out and submit the **PDF** version of this document that pledges your honor that you did not violate any ethics rules required by this course and KAIST. You can either scan a printed version into a PDF file or make the Word document into a PDF file after filling it out. Please sign on the document and submit it along with your other files.

Discussions with other people are permitted and encouraged. However, when the time comes to write your solution, such discussions (except with course staff members) are no longer appropriate: you must write down your own solutions independently. If you received any help, you must specify on the top of your written homework any individuals from whom you received help, and the nature of the help that you received. *Do not, under any circumstances, copy another person's solution.* We check all submissions for plagiarism and take any violations seriously.

*Material adapted from Google Cloud Manual and Simon Fraser University CMPT354.

1 Relational Algebra (40 points)

In this problem, we will use a bank database that is similar to HW1, but has more tables and columns. Unlike HW1, no one has the same firstName and lastName at the same time. The database consists of 7 tables with the following schemas.

- Customer = {customerID, firstName, lastName, income, birthData}
- Account = {accNumber, type, balance, branchNumber$^{\text{FK-Branch}}$}
- Owns = {customerID$^{\text{FK-Customer}}$, accNumber$^{\text{FK-Account}}$}
- Transactions = {transNumber, accNumber$^{\text{FK-Account}}$, amount, date}
- Employee = {sin, firstName, lastName, salary, startDate, branchNumber$^{\text{FK-Branch}}$}
- PersonalBanker = {customerID$^{\text{FK-Customer}}$, sin$^{\text{FK-Employee}}$}
- Branch = {branchNumber, branchName, street, numberEmployess, managerSIN$^{\text{FK-Employee}}$, budget}

Write relational algebra queries learned in the class to answer questions (1) to (10). Your answer to each question should consist of a single relational algebra query. You may use input relation names to differentiate between attributes with the same name in the results of a join or Cartesian product (e.g., referring to Employee.firstName or Customer.firstName in the results of the Cartesian product of Employee and Customer).

Note **In this task, you should type the queries by a computer, not taking pictures of your hand-written queries when you submit the answers.** Because some hand-written queries are hard to recognize or understand the meaning. If you submit the answers with hand-written, your score would be deducted.

[Task] Write relational algebra queries (4 points each).

(1) You are to find the SINs, and first and last names of employees who own an account in the branch in which they work.

(2) The customer IDs of customers who have personal bankers in either the Vancouver or Metrotown branches (note that the personal bankers must be distinct employees as an employee only works at one branch).

(3) The SINs, first and last names and salaries of employees who are both personal bankers and managers

(4) The SINs and salaries of employees who earn more than the manager of their branch.

(5) The customer IDs, first names, last names and incomes of customers who have an account at a branch with a budget no more than $3,200,000.

(6) The branch names of branches that employ at least one employee whose last name is Martin, and at least one employee whose last name is Jackson.

(7) The customer IDs of customers who own a joint account (an account that is owned by more than one customer)

(8) The first names, last names and birth dates of customers who own an account in the London branch, and the first names, last names and start dates of employees who work in the London branch (i.e. one query that returns one list of first and last names and dates of these 2 groups of people).

(9) The first and last names of customers whose first name is Steve and income is less than $40,000.

(10) The customer IDs of customers whose accounts have no transactions with amounts of which the absolute value is less than $3,000 (i.e. all their transactions are either greater than or equal to $3,000 or less than or equal to -$3,000).

2 Database Design (60 points)

2.1 Entity-Relationship Model

In this problem, we will design a university database that models KAIST using an E/R diagram. The database should include information about students, departments, professors, courses, and buildings. The details are as follows.

1. The information of a student includes a student ID, a name, a department, and any courses they take. Here, the student ID is unique for each student. Each student can have multiple majors (departments).
   If a student lives in a dormitory, the student information should include the name of the building he or she lives in. Each student has at most one dormitory.
   Each student must be a graduate student or undergraduate student. If a student is in graduate school, the student information must include his or her research field. For undergraduate students, the research field should have a NULL value.

2. The information of a department includes a department name, the head of the department, professors, and buildings. The department name is unique, and each department has one or more buildings. Also, a building can be shared by multiple departments.

3. The information about a professor includes an employee ID, a name, affiliated departments, and the names of the courses the professor teaches. The employee ID is unique for each professor. A professor can be affiliated with one or more departments.

4. The information of a course includes a course name, a course code, and the professors who teach it. The code of each course is unique, and each course can be taught by one or more professors.

5. The information of a building contains the building's name (e.g., IT building) and the building code (e.g., $N1$). The code of each building is unique.

Solve the following two tasks (10 points each).

Note **We highly recommend writing answers with the computer. If you want to write it by hand, please make sure the answers are clear and neat.** We will deduct the points if the answers are hard to understand.

[Task 1] Draw an E/R diagram based on the above requirements.

When generating the diagram, try to follow the best practices covered in class. Since there can be multiple correct solutions, briefly explain why you made any design choices.

[Task 2] Convert the E/R diagram into a schema.

Again, try to follow the best practices covered in class. Since there can be multiple correct solutions, briefly explain why you made any design choices.

2.2 Normal Forms

In this problem, we will use a real dataset (called Black Friday) to identify functional dependencies (FDs) in a table and normalize its schema.

You can download the Black Friday dataset on Kaggle using this link (CSV file). Alternatively, you can find the same dataset in the Kaggle website:
https://www.kaggle.com/llopesolivei/blackfriday
The CSV file is one table where the first row contains the attribute names.

Solve the following four tasks (10 points each).

Note **We highly recommend writing answers with the computer. If you want to write it by hand, please make sure the answers are clear and neat.** We will deduct the points if the answers are hard to understand.

[Task 1] Find all the non-trivial FDs in the table based on its schema.

[Task 2] For each FD in the previous task, provide an SQL query that shows that the FD indeed holds on the table.

[Task 3] Does the table contain redundant information? Are there potential anomalies? Briefly explain with examples.

[Task 4] Normalize the table to remove redundant information and prevent anomalies. Write down the resulting schema and briefly explain whether the schema is in 3NF, BCNF, and/or 4NF.