Homework Assignment #2
Database Systems – Fall 2022/2023

- This assignment includes four questions.
- Submission is through the course website and in pairs – only one of the two should submit and that student will also be the one to receive feedback. A submission for three people is not permitted, if you are unable to find a pair then you should submit by yourself (it is not necessary to ask for permissions to submit by yourself).
- Your solution should be submitted in a single PDF file containing all answers, named with your ID, such as "123456789.pdf". If multiple students are submitting, then separate the ID numbers with underscore, such as "123456789_987654321.pdf".
- Include your full name in the pdf file.

QUESTION 1

You have been hired to design a database for the scenario below.

A) Create an ER diagram describing all the required entity sets, the relationships between them and constraints on these entities and relationships (keys, referential integrity, etc.).

- Keep your design simple but faithful to the scenario.
- You may add titles to the connecting lines where you think it may be helpful for clarity.
- There may be some requirements the ER diagram is not able to model. Use the ERD tools you have learned to create entity sets, attributes and relationships diagram to best describe the scenario.
- Make sure each entity set has a unique identifier (key).
- You can use any tool you like to create the diagram (power point, visio, graphical software, a scanned page in clear handwriting, ...) but the symbols must be as learned in class.
  Do NOT submit an EER diagram created through MySQL Workbench!

B) Translate the diagram into a relational schema. Provide the table names, column names, the key(s) of each table and the foreign keys between tables in the following format:

University(<u>name</u>, ranking, address)

Student(<u>ID</u>, name, studies_at)

Student(studies_at) → University(name)

….

When translating "Is a" relationship follow the ER style conversion method.

The scenario:
You will design a schema for a new social network, *FriendBoost*.
A user that registers to FriendBoost provides his/her details - unique username, first name, last name and birth date.
The network supports three main operations: following friends, adding photos and browsing pages.

Friends - Each user can follow friends that are also users of FriendBoost.

Photos - Each user can upload many photos and FriendBoost records for each photo its uploader (the user who uploaded the photo) and a unique URL. Users can tag other users in photos. Each photo may have tags with the users that appear in it. For each tag, FriendBoost records the tagger, tagged user and the location in the photo (X and Y coordinates). Finally, each photo may appear in one or more photo albums. Photo albums will be described next.

Pages - In FriendBoost, users can create pages of two types: photo albums and calendar events.
Each page has a unique URL. In addition, FriendBoost wants to keep track of how many times each user has entered each page, therefore, a log is needed to record this information.

Photo Albums - Each photo album has a title and a description of its contents.

Calendar Events - Calendar events have start and end times, and a description. In addition, users can be invited to events by other users. For each user who has been invited as a guest to some event, FriendBoost records their RSVP status (attending/maybe/not attending). There is no need to record any information about the inviter (do not include the inviter in your diagram).

QUESTION 2

You are given a relation R(A, B, C, D, E, F, G, H, I, J) with the following functional dependencies.

A, D -> H

C, I -> A, J

E, H -> B, F

G -> D, E

J -> I

A) Find all the nontrivial FDs that follow from the initial set of FDs.

- Recall that a nontrivial FD is an FD derived by using the transitive rule, the left-side should be minimal and the right-side should be maximal.
    - For example,
        - Deriving `B, A, D -> H` from `A, D -> H` is not left-side minimal, and it doesn't use the transitive rule. Thus, this is not a nontrivial FD.
- Hints
    - There are exactly 10 nontrivial FD rules.

    - You shouldn't do more than 4 rounds of transitive jumps.
- Tips
    - Start by decomposing the given rules into smaller rules (i.e., rules that have only 1 item on the right side).
    - Number each rule and try to use it to keep track of your calculations.
        - For example,
            - (1) A, D -> H
            - (2) C, I -> A
            - (1+2) C, I, D -> H

B) Find all the minimal keys for R.

C) Decompose R into BCNF using the lossless decomposition algorithm we have learned in class. Show all the decomposition steps and provide (all) the minimal keys of the output relations.

D) Is it possible to decompose one of the output relations you have obtained in C and still have a lossless decomposition to BCNF? Answer with yes or no, and briefly explain your answer.

QUESTION 3

Given the following relation and set of FDs:

Customer(id, first_name, last_name, membership_level, membership_fee, discount)

Id -> first_name, last_name, membership_level, discount
membership_level -> membership_fee, discount

A) Are there any violations of the BCNF? Explain your answer.
B) What would we benefit from applying a BCNF normalization on this relation? Give 3 advantages with an example for each.

QUESTION 4

Alice got the relation R(A, B, C, D) with the FDs:
A -> C
C, D -> B

She decomposed it into R1(A, C) and R2(A, B, D).
A) Is the result of Alice's decomposition in 3NF? Explain briefly.
B) Is the result of Alice's decomposition lossless? If not, give a counter example. Explain briefly.
C) Is the result of Alice's decomposition dependency preserving? If not, give a counter example. Explain briefly.