Written Assignment #2
CAS CS 460/660: Introduction to Database Systems
Fall 2017

Due: Wednesday, October 18, 2017, in class

Problem 1. [25pts]

Consider the following relational schema. An employee can work in more than one department; the *pct time* field of the Works relation shows the percentage of time that a given employee works in a given department; *managerid* is a foreign key references Emp on *eid* and is the *eid* of the manager of this department.

> Emp(<u>eid: integer</u>, ename: string, age: integer, salary: real)
> Works(<u>eid: integer</u>, <u>did: integer</u>, pct time: integer)
> Dept(<u>did: integer</u>, dname: string, budget: real, managerid: integer)

Write the following queries in SQL:

1. Find the managerids of managers who manage only departments with budgets greater than $1 million.
2. Find the enames of employees that work in more than one department.
3. Find the enames of managers who manage the department(s) with the smallest budget(s).
4. If a manager manages more than one department, he or she controls the sum of all the budgets for those departments. Find the managerids of managers who control more than $10 million.
5. For each department, print the department name and the minimum, maximum, and average age of the employees for each department.

Problem 2. [25pts]

Consider the following MovieStars database schema:

> MovieStar(<u>SNo</u>, sname, gender), Director(<u>DNo</u>, dname, ddob, dcity)
> Movie(<u>MNo</u>, DNo, title, genre, year, length, STNo, profit)
> Studio(<u>STNo</u>, sname)
> StarsIn(<u>SNo, MNo</u>, wage, firstrole)

The MovieStar relation stores information about movie stars. The Director relation stores information about directors and the (DNo) is the primary key. The Movie relation contains a list of movies. For each movie we store a movie number (MNo), director number (DNo), title, genre, year of production, length in minutes, the studio number where the movie were filmed and the profit. Each movie has a single genre (category). The Studio relation stores information about studios and the StarsIn relation stores information about who played in which movie. We store the star number (SNo), the movie number (MNo), the wages of the actor/actress for this movie and if he/she had a first role in the movie.

Write the following queries in SQL:

1. Find the names of stars who have appeared in at least one movie where the profits were greater than 100M.

2. Find the names of stars who have appeared in at least two movies where the profits were greater than 100M.
3. List the name of a star that has also directed a movie (was a director on a movie).
4. For each star, find the average profit of the movies he/she has appeared in and display it only if the sum of profits of the movies he/she has appeared in is more than 2M.
5. Find the names of stars who acted in at least one movie for each genre (have participated in movies for all genres).

Problem 3. (Based on Exercise 5.10) [25pts]

Consider again the following relational schema. An employee can work in more than one department; the *pct* time field of the Works relation shows the percentage of time that a given employee works in a given department.

Emp(<u>eid: integer</u>, ename: string, age: integer, salary: real)
Works(<u>eid: integer, did: integer</u>, pct time: integer)
Dept(<u>did: integer</u>, dname: string, budget: real, managerid: integer)

Write SQL integrity constraints (domain, key, foreign key, or CHECK constraints; or assertions) or SQL triggers to ensure each of the following requirements, considered independently. You have to give the CREATE TABLE statements for each relation and you can modify it to add constraints if needed.

1. Employees must make a minimum salary of $1000.
2. Every manager must be also be an employee.
3. The total percentage of all appointments for an employee must be under 100%.
4. Whenever an employee is given a raise, the manager's salary must be increased to be at least as much.

Problem 4. [25pts]

Consider the relation R=(A, B, C, D, E, F, G, H). Assume that the following set of FDs holds:

$\mathcal{F}=\{AB \rightarrow CEFGH,\ A \rightarrow D,\ F \rightarrow G,\ BF \rightarrow H,\ BCH \rightarrow ADEFG,\ BCF \rightarrow ADE\}$.

Find all the candidate keys of the relation R. Decompose the relation in the following normal forms (if it is not there already): a) 3NF, b) BCNF. For each decomposition that you make, state if it is (i) lossless join and (ii) dependency preserving.