Homework #2
Due: February 20, 2020

1. Practice the 20+ SQL examples discussed in class.

2. Consider the following relational schema:
   
   Employee(eid: integer, ename: string, age: integer, salary: real)
   Department(did: integer, dname: string, budget: real, mgrid: integer)
   Works(eid: integer, did: integer, pct_time: integer)
   
   An employee can work in more than one department; the \textit{pct\_time} field of the Works relation shows the percentage of time that a given employee works in a given department.
   
   a) (25 pts.) Write an SQL query to find the IDs of managers who control the largest amounts. \textbf{(Hint: Create a table in the WHERE clause to compute the total budget for each manager)}
   
   b) (30 pts.) Write an SQL trigger to express the following constraint: “Whenever an employee is given a raise, the manager’s salary must be increased to be at least as much.” (increasing a manager’s salary to be equal to the employee who received the raise, if the manager’s salary is less than the employee’s new salary)

3. (25 pts.) Consider the following relational schema:
   
   Faculty(fid: integer, fname: string, deptid: integer)
   Student(snum: integer, sname: string, major: string, level: string, age: integer)
   Class(name: string, meets\_at: time, room: string, fid: integer)
   Enrolled(snum: integer, cname: string)
   
   Enrolled has one record per student-class pair such that the student is enrolled in the class. Write an SQL assertion for the following integrity constraint: “Every faculty member must teach at least two courses.”
4. (20 pts.) We discussed in class a simple join algorithm as illustrated below:

This scheme is very computationally expensive because each tuple of the relation $R$ must be compared with each tuple of the relation $S$. This strategy incurs substantial disk access. Design a more efficient join algorithm, in which each tuple of relation $R$ needs to be compared with only a small subset of the $S$ tuples.