

Written Assignment #3

Due: Feb 28, Wed, 11:30AM in canvas.

Problem 1. [80pts]

Consider the following schemas, that are available from our database server in the cs5530 database (cs5530db).

student(sid, sname, sex, age, year, gpa)

dept(dname, numphds)

prof(pname, dname)

course(cno, dname, cname)

major(dname, sid)

section(dname, cno, sectno, pname)

enroll(sid, dname, cno, sectno, grade)

We showed how to connect to the database server in class. You can use either MySQL Workbench or connect via linux/unix mysql client by “mysql -u 5530user -p -h georgia.eng.utah.edu cs5530db”. (password was announced in canvass). Once connected, you can “show tables”, “describe tablename”, and type any SQL statements (must end with ;) etc.

Answer the followings using SQL queries. Submit **both your SQL queries and the query results from the database server**. Note that you can store all your SQL queries in a txt file (e.g., query.txt, end each SQL with a semicolon and separate each SQL query with an empty line), and execute them all in once by “source query.txt” (if you are using the mysql client to connect to the DB server). An example of query.txt with 3 queries is given below:

```
Select * from student;
```

```
Select dname  
from  
dept;
```

```
Select * from major;
```

To capture the query results from the database server in Unix/Linux, you can use the “script” command. In particular, when you are ready to execute all your queries from query.txt. Do the followings:

- 1) script output.txt
- 2) connect to the DB server and “source query.txt;”
- 3) quit the server by “quit” or “exit”
- 4) type ctrl+d to end the scripting process.
- 5) all screen printout will be captured in output.txt.

Questions:

1. What is the age of the oldest student.
2. Find the names and gpas of the students who have enrolled in course 302.
3. Find the names and majors of students who have taken an advanced course (i.e., the course title contains a keyword ”Advanced” somewhere).
4. Find the names of students who have enrolled in both a course offered by the “Computer Sciences” department and a course offered by the “Mathematics” department.
5. For each department, find the average gpa of the students majoring in that department along with the difference between the students with the highest and lowest gpa.
6. How many students have only one major?
7. Find the name(s) of the student(s) who have taken the least number of courses (same cno with

different sectno will be viewed as the same course for this purpose).

8. Find the name(s) of the oldest 3rd year student(s) (i.e., year = 3).
9. Print the ids, names, and gpas of the students who have taken all Computer Sciences courses.
10. For those departments that have no majors (i.e., students who major in that department) taking a “Computer Sciences” course, print the department name and the number of PhD students in the department.
11. Find the student names for each year with the maximum gpa.
12. Find the name(s) of the professor(s) who has (have) taught the least number of courses (multiple sections of the same course should be counted as multiple teaching assignments for this purpose).
13. For each department, find the student name (along with the departmentname) with the maximum average grade (average grade is calculated with respect to the grade column from the enroll table).
14. Find the sections (dname, cno, sectno) with the highest enrollment.
15. Find the department with more than 5 studnets.
16. For all departments that offer the same number of courses, print the department name that has the least number of professors; print the number of courses and the number of professors too for each output.

Problem 2. [20pts]

Consider the following relational database 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. Each department has exactly one manager.

Emp(eid int, ename varchar(30), age int, salary float)

Works(eid int, did int, pct_time float)

Dept(did int, budget float, managerid int foreign key references Emp(eid))

Write SQL to express the following integrity constraints (domain, key, foreign key, column or table constraint, and assertion, choose the one that you think is the most appropriate).

1. Employees should make no more than \$200,000.
2. Every manager must also be an employee.
3. A manager must always have a higher salary than any employee that he or she manages.
4. The total percentage of appointments for an employee must be 100%.
5. No employee can be a manager for more than 2 departments.