BA in Computer Science
BA in Mathematics and Computer Science
MSc in Computer Science

Michaelmas Term 2019

DATABASES

SQL Practical Assignment

Introduction

The Databases practical uses the database containing data about a US university for a particular semester (US semester = Oxford term), used in the practice sheet. So that you can experiment with adding data to the database, you will be asked to create your own private copy.

If your username is a01xx, you can connect to your private database by running:

```
tw03> psql -h tr01
```

This will automatically connect user a01xx to database a01xx

The schema of the private database will be the same as before:

- `student(`**sid**`, sname, sex, age, year, gpa)`
  Here `sid` is a unique student identifier (a number), and `gpa` is the student's grade point average.
- `dept(`**dname**`, numphds)`
  A department is identified by its name, and we also record the number of PhD students in the department.
- `major(`**dname**`, `**sid**`)`
  A student takes a degree in one or more departments, and this is recorded in the major relation.
- `prof(`**pname**`, dname)`
  Each professor is identified (perhaps unrealistically) by a name, and the professor is associated with a department.
- `course(`**cno**`, cname, `**dname**`)`
  A course is identified by its unique course number and the department where it is given. The course also has a name.
- `section(`**dname**`, `**cno**`, `**sectno**`, pname)`
  A course is split up into sections. Each section is taught by a specific professor.
- `enroll(`**sid**`, grade, `**dname**`, `**cno**`, `**sectno**`)`
  The enroll relation records which students are taking a section, and we also record their grade for this section. The grade is a number between 0 and 4.

To initialise the database, or reset to the original state after you have done some modifications, you can execute the command:

```
a01xx=> \i /usr/local/practicals/databases/init-database.sql
```

Questions

The purpose of this practical will be to practice SQL. You will need to write SQL queries for each question below (one query per question) and run each on the database.

The query answers must not contain duplicates, but you should use the SQL keyword `distinct` only when necessary.

1. Print the names of students who have taken Thermodynamics (any section), along with their grades.
2. Print the names of the courses that the student with studentid 16 has taken, along with the name of the professor (or professors) who taught them in that course.
3. Print the names of departments that have one or more students majoring in the department who are under (strictly less than) 19 years old.
4. Print the course and section number of all sections that have less than 12 students in them.

Note: your answers should work on *any data*. I.e. if the question asks for a query returning the total number of students, and your data set happened to have 10 students in it, you could not just have a webpage that always returned the answer 10!

1. Write an SQL insert command that inserts "Benedikt, M." into the "Computer Sciences" department.

Optional

Write a query asking for the ids of students that took only sections that had more than 10 students in them.

Assignment Submission

Turn in your queries along with sample output. Please use only SQL features covered in lectures.