# PART I: Writing SQL Queries

Consider the following relation schemas:

Student(*sid:* integer, *sname:* string, *major:* string, *standing:* string, *age:* integer)
Class(*name:* string, *meets at:* string, *room:* string, *fid:* integer)
Faculty(*fid*: integer, *fname:* string, *deptid:* integer)
Enrolled(*sid:* integer, *cname:* string)

The meaning of these relations is straightforward; for example, *Enrolled* records student-class pairs such that the student is enrolled in the class. A student's standing refers to the student's year, and can take on the values *FR* (Freshman), *SO* (Sophomore), *JR* (Junior), and *SR* (Senior).

Write SQL queries to achieve the following requirements. Note that no duplicates should be produced in any of the answers.

1. Find all Juniors who are enrolled in a class taught by any faculty whose surname begins with the letter T. Print the students and faculty names.

2. For all the standings except *JR*, print the standing and the average age of students in that group.

3. Find the names of all students who have a conflict i.e. they are enrolled in two classes that meet at the same time.

4. Find the *super hero* student(s) i.e. the one(s) enrolled in the maximum number of classes. Print the student(s) name(s).

5. Find the *super popular* faculty i.e. the one(s) with the maximum combined enrollment in the courses they teach. Print the faculty name(s).

6. Find the faculty members that have taught classes only in room R128. Print the faculty's name and the total number of classes she or he has taught.