James Solum

Student Interests and Careers

24th March 2018

OVERVIEW

The project I hope to produce is a database that keeps track of college student data such as classes taken, hobbies, majors/minors, jobs and ideal job. The purpose of this database would be to primarily gather data that could possibly be used to draw connections between student interests and possible careers or vice versa.

Some questions this database may help answer include:

- What careers do students who study certain subjects typically go into?
- What careers do students who study certain subjects want?
- What subjects do students with certain hobbies generally study?
- What kind of jobs do students with a certain amount of units usually have?
- Are there jobs that typically hire students with a certain GPA?
- Do students with higher GPAs strive for different kinds of careers than students with lower GPAs?
- And more!

These connections not only are interesting in and of themselves, but also are extremely relevant to students interested more about their field of study, career, and hobbies and how they connect with their wider student community.

THE RELATIONS

The relations found in this database are as follows:

1. student(firstName, lastName, sex, email, gpa) key: [email] & [firstName, Lastname]
    a. Provide the first and last name so that if someone ever wants to contact a student then they know the name of the person they are contacting.
    b. Provide the sex of the student so that we can see if there are associations between hobbies, careers, and academic interests among different sexes.
    c. Provide the email as a key to the relation, as well as a way to contact the student.
    d. And provide the gpa for analysis on the effect of GPA on careers, hobbies, and academic interests.

e. [1]FD's: [firstName, lastName -> sex, email, gpa], [email -> firstName, lastName, sex, gpa].

2. job(id, role, salary, industry) key: [id]
   a. Since a key consisted of all the attributes, an artificial id has been introduced to limit repetitive data.
   b. Provide the role of the job for information on what the job actually is. An example of a role would be a secretary or software developer.
   c. Include the salary for analysis on what kind of students get what kinds of salaries.
   d. Include the industry to provide a more general context of student careers. Some examples of an industry would be technology, retail, or manufacturing.
   e. FD's: since role and industry don't necessarily point to salary, this relation has no nontrivial FD's.
3. hobby(hobbyName) key: [hobbyName]
   a. The name of the hobby should be enough information to describe what it is. Some examples include soccer, filmmaking, or snorkeling.[2]
4. subject(subjectName) key: [subjectName]
   a. The subjectName describes an academic subject. Some examples include: psychology, computer science, or mathematics.
5. class(className, units, subjectName) key: [className]
   a. The class name is the official ID the college has for the class. An example would be CS-105, or PSY-025.
   b. Units are provided so we can get information about how many units a student has earned.
   c. We are going to assume that there are no cross-listed classes, since generally a cross-listed class is still named after one specific subject (e.g. my codes and encryption class was a cross-listed math and cs class, but was still listed as CS-***).
   d. FD's: [className -> units]
6. employedBy(email, jobId) key: [email, jobId]
   a. This relation connects a student with a job he is currently employed by. A student can be employed with multiple jobs at a time, and many students can have the same job. *many-many
   b. FD's: since a student can have multiple jobs and email does not necessarily determine the students job. Therefore, there are no nontrivial FD's in this relation.
7. participates(email, hobbyName) key: [email, hobbyName]
   a. This relation connects a student to a hobby. A student can have many hobbies, and many students can have the same hobby. *many-many

[1] FD's will be surrounded by brackets for the sake of readability.
[2] Since this relation only has one attribute it will not have any nontrivial FD's and therefore none will be listed. Every relation with only one attribute will not have a list of FD's.

    b. FD's: since a student can have multiple hobbies and multiple students can have the same hobby, there are no nontrivial FD's in this relation.
8. studies(email, subName) key: [email, subName]
    a. This relation connects a student with an academic subject. An example of this would be a major or a minor. A student can multiple majors, minors, or a combination of both. Furthermore a many students can study the same subjects. *many-many
    b. FD's: Since a student can study multiple subjects and a subject can be studied by multiple students there are no nontrivial FD's in this relation.
9. takes(email, className) key: [email, className]
    a. This relation connects students with classes that the student has taken. A student can take many classes, and a class can be taken by many students. *many-many

DESIGN CHOICES

Combining

Based on the relations and the entity-relation diagram it could seem reasonable, at least at a surface level, to combine some of these relations. One of the pairs of relations that could be considered combining would be the class and subject relations. These should not be combined however, because there is also a many-many relationship between students and subject. Therefore, we will keep the subject as a separate entity. A similar situation would be between a student and a dream job. Likewise, these two relations should not be combined because there is also the many-many relationship between students and current jobs.

The teaches relation was absorbed into the class relation because teaches was a many-one relationship. Having this relationship introduced redundancy which is easily avoided by combining the relationship into class.

BCNF

Since the functional dependencies of the relations are not too complex, it is clear that all the relations are BCNF. Therefore, there is no need for decomposing.

Flaws

One flaw could be that this database does not take into account cross-listed classes. I don't think this will be an issue however, since these are not very common. Another flaw, could be that it does not take into account past jobs. I do not think this is necessary, because this database will primarily be used by students to learn about current students.