

Lab 1.2 Writeup

Initial Decision:

Our team chose to meet the requirements in the lab by using a Java environment. The purpose of using this language was due to the fact that all of our group members were comfortable using the language and the fact that the language has easy to use (for a non SQL language) built in data structures.

Internal Architecture:

As data structures were one of the main reasons why we chose Java, it was quickly apparent to each of us that all students have the same attributes (i.e. each one has a name, bus stop, teacher, etc). Because of this we decided to make a Java class to store students. The Java class contains class variables for each of these different attributes that needed to be stored. Lastly, each instance of a student class was a single record. These records were linked using a Java Linked List.

Task Log:

Task	Assignee	Time
Java Class Implementation	Alex Martin, Chandler Warne	3 hr
Unit Testing (Design & Testing)	Garrett Chan	2 hr
Writeup	Chandler Warne	1 hr

Notes on Testing:

After Garrett prepared the unit tests, Alex tested the program. We found no bugs by performing unit tests. This is due to the fact that we actively tested our program as we developed it.

Final Notes:

"As a SQL user, this made me appreciate how easy and quick it is to use SQL and relational databases." ~Chandler

PART II:

Adaptations from Part I:

In order to accommodate the new teacher data, we created a new class (similar to our Part 1 student class) with the appropriate class variables or attributes that we needed to store the teacher data. Additionally, we subtracted the teacher's name out from our original class, as we no longer need the teacher's name stored along name the student.

In order to reference students and teachers jointly, we needed to use the class numbers as a sort of "foreign keys" in order to link between the two classes: teacher and student.

Lastly, not only did we add the new requirements but we also had to adapt the old requirements code in order to keep it working with the new data structure.

Syntax and Semantics Additions:

We discovered that two different types of queries were needed with this one addition:

1. Interaction with ONE table
2. Interaction with BOTH tables

In order to accomplish this, we made two functions in order to either look up student(s) given a classroom number or lookup a teacher(s) given a classroom number. Since the only foreign key is the classroom number, you can accomplish queries asking for information needing both tables by first getting the needed information out of one table and then using one of these two functions to "JOIN" the data to the second table.