

Jacquelyn Moreno and Thomas Kaizer: CSC173: Project 4 Writeup

Part 1

1. Our database implementation follows the model shown in the textbook. That is, each tuple is represented by a struct whose name is the schema of the relation. The struct contains a variable for each attribute in the relation. The relations themselves are hash tables.
2. For all relations in our database, insert, lookup and delete are supported. For CSG, lookup is supported with all parameters included, or with the grade parameter missing. For example, *lookup* ((*"CS101"*, *12345*, *" "*), *Course-StudentId-Grade*) is acceptable as well as *lookup* ((*"CS101"*, *12345*, *"A"*), *Course-StudentId-Grade*). For course-prerequisite and CDH, lookup and delete are only supported with all parameters included. For SNAP, lookup is supported with all parameters included, and with only the student ID parameter included. For course-room, lookup is supported with all parameters included, and with only the course parameter specified.
3. As you run our project, it will go through each individual relation and insert each of its tuples, and then test the lookup and delete methods. For a given relation, if a lookup is supported with parameters missing our program will demonstrate that. It will then prompt you to press enter to continue to the next relation. Our project tests each of the operations in example 8.2 of the textbook, as well as our own operations to demonstrate the functionality of our database.
- 4.

Part 2

1. We demonstrate the "What grade did *studentName* get in *courseName*?" function twice with students who are enrolled in a course and once with a student who is not enrolled in a course.
2. We demonstrate the "Where is *studentName* at *time* on *day*?" function twice with students who are at a location at that time, and once with a student who is not.

Part 3

For example 8.12 we made a new CSG hashTable to store the resulting tuples, and then used our primary index on course to find all the "CS101" tuples. We then added them to the result and returned it.

For example 8.13 we took in the result from the previous example and added only the studentID component to the result.

For example 8.14 we made a new hash table with schema CRDH, and then iterated through the CDH relation. For each tuple in CDH, We used our primary index on course in the CR relation to quickly find any tuples with a matching course attribute. When we found a matching attribute, we added it to the result.

For example 8.15, we used the same method for selection, projection and join as described.