**Write Up**

**CSC 173- Project 4**

**Collaborators:**

**Sandesh Paudel**

**Azmayeen Fayeque Rhythm**

The project is divided into three parts. We will describe each part in the following Write up:

**Part 1**

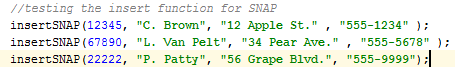
In the first part of the project we created 5 relational databases or tables: CSG, SNAP, CDH, CP CR. (Source: textbook: Fig 8.1 and 8.2; Section 8.2).

**For SNAP**:

We first define a structure type called SNAPTuple. Create SNAPLIST Hashtable of size SNAPSIZE that stores a Tuplelist. Using type SNAPTABLE we created SNAPTABLE1. Testing for the Insert, delete and lookup functions are done in the main method where is it written neatly with comments. There will be an example here as well of how we tested the methods:

**Hash function:** We create a hash function that takes a key, and returns key % SNAPSIZE. In SNAP – StudentId is the key.

**InsertSNAP**: Inserts the data based on each attribute: StudentId, Name, Address, Phone.



**PrintSNAPTuple**: After inserting the data, this method prints out the tuples in order as given in the book.

**DeleteSNAPTuple**: This deletes individual Tupple. The delete function is integrated into the LOOKUP function; it is called from there. It looks at a few cases:

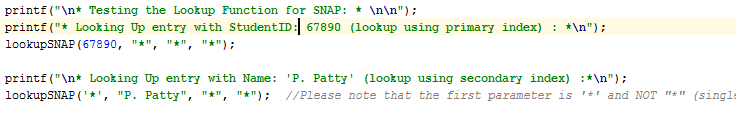
* Removing if there is only one item in the list.
* Removing middle item.
* Removing the last tuple that is not the first tuple.

**lookupSNAP:** It looks up the tuples based on some specification. Finding the right tuple with the key as Student ID. Look up index as SNAP function. It also looks at a few cases:

* All Values are given as the lookup parameters, returns true or false.
* One Value is given:

Such as: Student ID is given

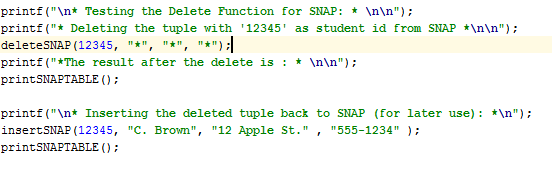
The variable booleanFound keeps track of any queries that has been found. If no student is found given the specific attribute value it prints out: Student not found.



**printSNAPTABLE:** This method printsSNAPTABLE. We iterate through the SNAPLIST and add the tuples to the table while there is a tuple in the list.



**deleteSNAP**: This method deletes tuples based on their specification.



The same functions are also implemented for the other relations:

CSG:

**insertCSG**

**lookupCSG**

**deleteCSG**

**printCSG**

CDH:

**insertCDH**

**lookupCDH**

**deleteCDH**

**printCDH**

CR:

**insertCR**

**lookupCR**

**deleteCR**

**printCR**

CP:

**insertCP**

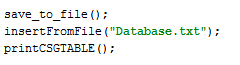
**lookupCP**

**deleteCP**

**printCP**

**Reading from the file and Saving to the file:**

**save\_to\_file method**: Saves all the tables and data inside in a txt. file called Database.txt. It prints all the tables like in order. In the main method just call **save\_to\_file();** and a txt file will be created.

**InsertFromFile:** It takes a filename to take in the data and insert it. To check if this method works: call it in the main method, and give **Database.txt** as its parameter. If you print any table after calling insertFromFile then it will print the data in that table twice. For example, in the main method: we test this method by printing the CSG table which prints each tuple twice. 

**For Part 2:**

**Question 1:**

We wrote two functions to answer the query and helper functions to support those. In the **Question2one** method we have studentName and courseName as parameters.

It uses lookupSNAP to find the name of the student. While the id of the student is not null it uses method **Question21a** that uses lookupCSG to find the grade a student received given a course and id. Ultimately, **Question2one** returns the result of the Part 2 – Question 1. In the main method, we test this Question using these two test examples.

*For: What grade did StudentName get in Course-Name?*

*Question2one ("L. Van Pelt", "CS101");*

*Question2one ("C. Brown", "CS101");*

**Question 2:**

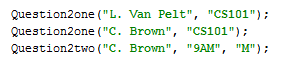
The method Question2two is used to test this question. This method takes in studentId, hour and day and gives the result of lookupCR(course(given), find Room).   
First, it uses lookupSNAP given a name to find the id. If ID exists then it calls Question22a where it uses the ID to find Course and Grade for a specific student. It uses Question22b and call lookupCDH. If it finds a match in CDH then it uses the course from the tuple that match as a parameter for Question22c which takes course as a parameter and returns Room. In other words, where the student is at the given time and day.

In the main method, we test it by the following example:

For: Where is StudentName at Time on Day?”

**Question2two ("C. Brown", "9AM", "M");**

Below is the screenshot from the main methodthat test the two questions in part 2.



**Part 3:**

In this part, we implemented the Relational Algebra operations and used them to solve three examples.

**For the first Question:**

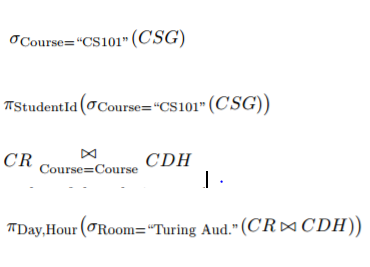
The implementation of Union, Intersection, Difference, Selection, Projection and Join.

**Union:** It takes twoSNAPTABLE as parameter and adds them to create a new TABLE.

**Intersection:** It takes two SNAPTABLE as parameter, iterates through one of them for each tuple and if there is a match between the tuples of the other SNAPTABLE then add them to the new list.

**Difference:** It takes two SNAPTABLE as parameter, adds all the tuples of r to a new table, for all tuples in s if there is a match between the tuples then delete it from the new TABLE and then return it. **To implement -> Join: Selection: Projection,** we use the four examples:

That are given in the instructions.



We test the four examples in the main method:

