Jackson Goyette, Spencer Schurk
CSC 365
Lab 1-2

Lab 1-2 (Why Databases?) Writeup

Team Members: Spencer Schurk and Jackson Goyette

Initial Decisions:

We created this program in Java, since that was one of the parameters of the lab. We decided to use GitHub as a collaborative platform, so we could both work on the lab outside of the lab itself and independently and still receive feedback on the changes we made.

Internal Architecture:

The main structure we used for storage was an ArrayList of Rows, a class we defined in the program. Every Row contains all of the data that would be found on a row of the text file input (Student First/Last name, grade, GPA, etc.). To populate this ArrayList, we used a buffered reader to parse through the file and create Rows in our ArrayList that contained the student information. For the inputs, we created separate functions to handle each, that an inputHandler function called.

For Lab 1-2, we wanted to keep our Row class the same, and modify how we took in the text files to accommodate that. We made this decision because all of our methods already assumed our data would be found in the Row class, and worked with them accordingly. Since we wanted to keep using Rows, we spliced together the information we got from students.txt and teachers.txt.

Additions to Query Architecture:

NR1: CS: or ClassroomStudents: <number>
    Example: CS: 101
NR2: CT: or ClassroomTeachers: <number>
    Example: CT: 101
NR3: GT: or GradeTeachers: <number>
    Example: GT: 1
NR4: E[nrollment]
NR5: An[alytics]: G[rade] or T[eacher] or B[us]
    Example: An: G
    This will list each grade, with the average GPA next to it
    T will list each teacher (last name), with average GPA next to it
    B will list Bus routes, with average GPA next to it

Task Log:

- ArrayList and Row implementation: Spencer/Jackson, Fri. 9/21, approx. 1 hr.
- inputHandler and main calls: Spencer, Sat. 9/22 - Mon. 9/24, approx. 2.5 hrs.
- initRows and structure: Spencer/Jackson, Fri. 9/21 and Mon. 9/24, approx. 2 hrs.
- Test cases: Spencer/Jackson, Mon 9/24 - Tues. 9/25, approx. 1 hr.
- Four methods in Part 2: Jackson, Thurs. 9/27, approx. 1.5 hrs.
- Changing the input method to the file types: Spencer, Thurs. 9/27, approx. 2 hrs.
- NR5 for part 2: Spencer, Sat. 9/28, approx 1 hr.
- Testing for part 2: Spencer, Sat 9/28, approx 1 hr.

Testing:

Running through the test cases, Spencer found only one bug: when attempting to find the average of a grade that didn't exist, schoolsearch would output "grade: 0" as opposed to not printing anything. It took about 2 minutes to fix up this bug.