CS 143 Lab 2 Writeup

Team member information:
Andrew Yoon 903912647
Kenneth Shi 304063313

We implemented Lab 2 by following the guide given in the spec. Exercise 1 was relatively straightforward, and didn't give us any trouble. We chose to implement join with simple nested loops since we felt this was the easiest method to wrap our heads around. We ran into a little bit of trouble in passing a specific test before we realized that our variables for the nested loops should be global variables so they'd be able to save their state from loop to loop.

For the Aggregator classes, we use a hashmap to map group-by fields to their aggregate values efficiently. We also needed to keep track of the number of tuples to average in a second hashmap in IntegerAggregator. mergeTupleIntoGroup computes the aggregate value, and iterator() actually creates the output tuples.

Exercises 3 and 4 were also straightforward, as we were basically implementing insert and delete methods three times. For reading and writing the files, in the previous lab we used a FileChannel, but for this lab a FileOutputStream seemed to be more suited for our purposes. The only difficult part of this lab was part of Exercise 3, when figuring out how to generate the correct bitmask in markSlotUsed().

The implementation of Exercise 5 followed very simply from the spec. We use a random eviction policy for our BufferPool. To choose the next page to evict, we simply get the first page in the set returned by cache.keySet().

No changes were made to the API. There are no known issues or missing/incomplete sections of our source code. The code passes all given unit tests and system tests. We spent a few hours or more per night for about 5 nights on this project. Overall, this lab was easier than the first lab in terms of implementation, mainly because it took us a lot of time to understand file structures and the organization of the code during the first lab, so we were already comfortable with the second lab.