6.172 Project 1 beta writeup

Katie Siegel

1 Design Decisions

There were a few overarching trends to the design decisions I made. First, I frequently chose to use ConcurrentHashMaps when storing pools of data, such in the BufferPool class. I reasoned that ConcurrentHashMaps would be more accommodating to implementing locking and concurrency in future labs. Additionally, a HashMap would allow for the quickest access to the information desired, as often the implementation called for a key/value style structure of data.

I also made a notable design decision when implementing the Catalog class; I chose to map file IDs to a new data structure that would hold a file's name, the file itself, and the primary key field of that file. I chose to map the file ID to the rest of a file's information because the most common access method for a file's information in the Catalog class is through its file ID. Only one method requires a reverse lookup; this method has a comparatively slower runtime to the other lookup-by-file-ID methods, as I have to iterate through the entire pool of files and look for one that matches the lookup name. However, the general expected time among all methods in the Catalog class is much faster through this HashMap structure.

Additionally, with regards to the hash codes generated in each class, I leaned towards always generating the hash code in the final step from a string generated from some combination of the fields in a class.

When I implemented class-specific iterators, I chose to create a new internal iterator class. I felt that this made the iterator object easier to work with and made my code structure more modular.

Most other steps in the lab involved implementation of getters and setters, as well as other basic methods that did not require significant design decisions.

2 Changes to API

I did not make any changes to the API. The most I changed classes was to add a custom Iterator class to certain classes, which did not change the external-facing methods of those classes at all.

3 Missing or Incomplete Code

My code for the BufferPool does not take into account evicting pages from the buffer pool of pages (implemented as a ConcurrentHashMap). This is because lab1 does not require handling evicting pages.

4 General Comments

I spent around 10 hours on this lab. The lab was generally straightforward, but I did get stuck for a while trying to understand how to implement the page structure. Reading the Catalog and BufferPool class documentation helped with this. I also got stuck at the very end, where my code passed all of the individual tests, but didn't pass the system test at the end. I found that the error was a few major steps back, when I rounded wrong on one calculation.