

Conor Reisman

4/29/2013

CSE444

Lab 2

Design Decisions

I used a least recently used policy for buffer pool eviction. It isn't the most efficient policy, but since there's only 50 pages, time complexity isn't really a problem. It's also very easy to implement.

I am currently using a nested loop join. This isn't very good though, as it means the second and third queries never complete (query 2 ran for over 48 hours). I would prefer to use a hash algorithm. I'd read in a reasonable-sized number of tuples from one child, put them in a hash table, then read in the second child and look for matches. Then I'd repeat until I'd finished the entire first child. This is much faster since I'd have a roughly $O(1)$ lookup for each tuple instead of $O(N)$.

Since both aggregators use count, and it's unrelated to content, I abstracted it out, so `StringAggregator` is basically a wrapper over the abstract. Otherwise I used a number of different iterators to handle all of the special cases in `IntegerAggregator`.

All of my operators find the next tuple immediately after returning the previous one. This is to keep to the pattern that `hasNext` is not a mutator.

API Changes

The aggregator interface didn't make sense. We had to build and process the tuples in the aggregator, but the aggregator didn't know what to call the columns. So we had to reset the `TupleDesc` for each `Tuple`. I added additional constructors that took strings for column names. This worked much better.

I added an `addPage` method to `BufferPool` for when you need to insert a `Tuple` and there are no free slots.

Incomplete/Missing Implementation

Nothing that was required for lab 2 was left unimplemented.

Other

- I'm not a fan of how much things move back and forth between `HeapFile`, `BufferPool`, and `HeapPage`. It's not so much the interaction I hate, as the constant casting. It seems unnecessary.
- You don't have a way to add a page in the `DbFile` interface. This means I have to assume it's a `HeapFile` and cast again.
- For a full list, see `notes.txt`. For my time spent, check `log.txt` (around 13 hours). For my times, see `times.txt`.