Yuechen Zhao

1126125

CSE 444 Lab 1 Write-up

For the implementation of Catalog.java, I used Map<Integer, Table> as a private field, which has file.getId (the identifier of this file) as key and a helper class Table as value. The helper class Table stores all information about a table, including the contents of the table, the table's name, and the name of table's primary key. I used a map because a lot of methods in Catalog.java have tableid as parameter, so it would be efficient to make the tableid the key.

For the implementation of BufferPool.java, I used Map<PageId, Page> as a private field. I mapped from PageId to Page because it allows quick lookup of a PageId to determine if it is already in the BufferPool, and it allows quick access to the corresponding Page.

For the implementation of HeapPage's iterator method, I created a helper class called HPTupleIterator, which takes the list of used slots as parameter, and returns an iterator over these used tuples. I implemented this way because calling remove on this iterator should throw an UnsupportedOperationException.

For the implementation of HeapFile.java's iterator, I created a helper class called HFTupleIterator, which takes the list of tuples in the pages in this HeapFile as parameter. I implemented this way because calling next() on the iterator should throw a NoSuchElementException if the iterator is uninitialized or hasNext() == false, instead of returning null.

I did not make changes to the API.

In BufferPool.java, if there is insufficient space in the buffer pool, a page should be evicted and the new page should be added in its place. However, for lab 1, I just throw a DbException instead. Also, all the methods and classes that has a TransactionId are also incomplete.

I spend approximately 2 days on this lab, and I think keeping track of all the useful methods for each class is the difficult point, ie. knowing where to find the information I need.