Database Systems Implementation

Programming assignment - 2

Name: R. P. Nihal
RUID: 172002406
Net ID: pr377

Design Decisions

- JoinPredicate.java - Just a simple code to compare two fields using a predicate during join. Implemented the JoinPredicate() and filter() method.

- Join.java

  1) Implemented the join constructor which accepts the children to join, and the predicate to join them on.

  2) Implemented the open(), close(), and rewind() methods which opens, closes and rewinds the Iterators respectively

  3) Implemented the SNL_readNext() method by using the simple SNL logic.

  4) For the SMJ_readNext() method, I implemented an additional method called goToRecord() in the HeapFileIterator class. This method takes the record ID as the parameter and jumps to the corresponding tuple location in the DB file using the page number and slot ID.

  5) For the PNL_readNext() method, I created a new wrapper class called pagewiseDBIterator over the DBIterator class. This class helps in reading one page at a time, but we need a way to store the data on these pages temporarily.For this, I implemented the addTuple() method in the HeapPage class which helps to use the HeapPage instance as a temporary buffer for storing data during the iteration. (I got the idea of creating a new wrapper class from a source on GitHub)

Feedback on the Assignment

This assignment gave me immense insight on the implementation of Databases. It also increased my understanding of the OOPS concepts and Java in general. In respect to this assignment, SNL was easy to implement. I found implementing the SMJ algorithm little tough in the part where we have to iterate the inner relation back to the first match, for every tuple of the outer relation. In the case of PNL, I spent a lot of time thinking over the implementation of page wise iteration. I spent around 25 hours for doing this assignment.