

Database System Implementation Assignment 2

Files Modified

1. Join.java
2. JoinPredicate.java

Number of Hours Spent

17 Hours.

Part A

Initialisation of variables was done in JoinPredicate's Constructor and the JoinPredicate.filter() has been implemented. The filter function will return true, if during the comparison of two tuples, if the predicate on specified field values returns true.

Part B

- **Join Constructor:** Initialisation of variables
- **Join.getTupleDesc():** returns combined Tuple Descriptor of two tuple relations using TupleDesc.combine(TupleDesc, TupleDesc)
- **Join.open():** Opens the iterator.
- **Join.close():** Closes the Iterator.
- **Join.rewind():** Resets the iterator to start.
- **Join.joinTuple():** Simple two while loops are used to join tuples t1 and t2 passed as an input according to new tuple description received from the function getTupleDesc().
- **Join.SNL_readNext():** Implementation iterates over all the tuples of the outer relation and inner relation using nested while loop.
- **Join.PNL_readNext():** Implementation iterates over all the tuples of the inner relation for all the tuples of the outer page before moving to the next page of outer relation. The following variables have been used to supplement the Page Nested Loop Join:
 - (HeapFileIterator) _outerHeapFile is used to for iterating over the Pages of the Outer Relation and store it in (Page) _outerHeapFilePage using the (DbIterator) _outerPage to iterate over the page. (boolean) isOuterTupleInSamePage is used to check if the tuple is in the same page.
 - (boolean) rewindInnerRelationFlag is used to check if the innerRelation needs to be reset after it has iterated over all the tuples of the outer page.
- **Join.SMJ_readNext():** Implementation implements the sort merge join taking care of the number of pages fetched and the border cases where one or both relations have run out of tuples to iterate over. The following variables have been used to supplement the Sort Merge Join:
 - (Boolean) borderFlag is used for border case where both of the two relations have run out of Tuples to iterate over because of hasNext() function.
 - (Boolean) pendingFlag is used is case of multiple matches to return tuples along with a (ArrayList<Tuples>) pendingList containing all the tuples to be returned once an initial tuple has been returned.

Observations & Issues

The issues I faced were mostly in Sort Merge Join Implementation. The case that was most hard to detect and implement was the one either or both of the relations ran out of tuples to iterate over and there were still some joins that could have happened. Another case I faced some issues were in the case of multiple matches. I was able to solve all the issues and run the tests successfully though I observed that the data in the two data sets used by MoreTestJoin were corrupted. Another thing to remember while testing for PNL, setJoinAlgo needs to be updated to PNL from SNL in joinTest.java. Lastly, I have used the code from previous assignment with suggested changes.