Wednesday, May 22, 2019

# Lab3 Report

---

Jiangbei Li

## - Design Decision

- Predicate, JoinPredicate, Filter
  - Write codes as the doc says.
- Join
  - In function fetchNext(), I used nested loops join. I create a member variable tmp1 to remember the tuple that the 1st iterator child1 in.
  - If tmp1 != null, I will keeping visiting the next of the 2ed iterator child2 until I find a correct relation (tmp1, tmp2)
  - Else I will visit the next of child1. For each tmp1 in child1, I will visit all the elements in child2.
- HashEquiJoin
  - I create a hashmap to save the elements in child1. The key of hashmap is Field and the value is an ArrayList of Tuple.
  - I create a Iterator listIt for the ArrayList I am visiting now and a member variable tmp2 to remember the tuple that child2 is visiting.
  - In fetchNext(), I firstly visit all the elements in listIt.
  - Then visit the next element of child2, and find the corresponding arrayList in hashmap.
- IntegerAggregator
  - I create 4 maps: maxMap, minMap, sumMap, countMap to remember the value of the group with key of Field and value of Integer.
  - I also create a groupMap to save all answer tuples.
  - In mergeTupleIntoGroup(), I will change value in the corresponding map and calculate the needed value. Then create a new tuple and set its field as the calculated value. Finally put it in the groupMap.
- StringAggregator, Aggregate
  - Like the count operation in the IntegerAggregator.

- Heappage
  - In markSlotUsed(), calculate the Byte and Bit position and mark the header to 1.
  - In deleteTuple(), call the function markSlotUsed.
  - In insertTuple(), find an empty slot and mark the slot. And set a new recordId for t.
- HeapFile
  - In deleteTuple(), use BufferPool.getPage and delete the tuple. Return the page.
  - In insertPage(), Traverse all the pages in the file. If there is an empty slot, then insert the Tuple. Otherwise, create a new page, insert the tuple and write it on the Disk.
- Insert, Delete
  - Create a bool variable to remember whether calling the function fetchNext.
  - In fetchNext(), make sure that fetch is false. Traverse the child, insert/delete the tuples and count the number. Mark the fetch true and create a new tuple to save the number.

## - Difficulty and Time

- I spent 3 days on lab3, and creating new pages on Disk disturbs me for a long time. Then I found the answer in Function getEmptyPage of BTreeFile.java.

## - Weakness

- I didn’t write codes for function getChildren and setChildren. Obviously, the test does not call these functions.