

Join GitHub today

Dismiss

GitHub is home to over 36 million developers working together to host and review code, manage projects, and build software together.

Sign up

Branch: master ▾ cs186 / hw4 / src / main / java / edu / berkeley / cs186 / database / query / BNLJOperator.java

Find file Copy path

 siyk Added in homework hw4

25c1d7f on Oct 19, 2017

1 contributor

127 lines (107 sloc) 4.18 KB

Raw Blame History   

```
1 package edu.berkeley.cs186.database.query;
2
3 import java.nio.ByteBuffer;
4 import java.util.ArrayList;
5 import java.util.Iterator;
6 import java.util.List;
7 import java.util.NoSuchElementException;
8
9 import edu.berkeley.cs186.database.Database;
10 import edu.berkeley.cs186.database.DatabaseException;
11 import edu.berkeley.cs186.database.common.BacktrackingIterator;
12 import edu.berkeley.cs186.database.databox.DataBox;
13 import edu.berkeley.cs186.database.io.Page;
14 import edu.berkeley.cs186.database.table.Record;
15 import edu.berkeley.cs186.database.table.Schema;
16
17 public class BNLJOperator extends JoinOperator {
18
19     private int numBuffers;
20
21     public BNLJOperator(QueryOperator leftSource,
22                        QueryOperator rightSource,
23                        String leftColumnName,
24                        String rightColumnName,
25                        Database.Transaction transaction) throws QueryPlanException, DatabaseException {
26         super(leftSource, rightSource, leftColumnName, rightColumnName, transaction, JoinType.BNLJ);
27
28         this.numBuffers = transaction.getNumMemoryPages();
29         this.stats = this.estimateStats();
30         this.cost = this.estimateIOCost();
31     }
32
33     public Iterator<Record> iterator() throws QueryPlanException, DatabaseException {
34         return new BNLJIterator();
35     }
36
37     public int estimateIOCost() throws QueryPlanException {
38         //This method implements the the IO cost estimation of the Block Nested Loop Join
39
40         int usableBuffers = numBuffers - 2; //Common mistake have to first calculate the number of usable buffers
41
42         int numLeftPages = getLeftSource().getStats().getNumPages();
```

```

43     int numRightPages = getRightSource().getStats().getNumPages();
44
45     return ((int) Math.ceil((double) numLeftPages / (double) usableBuffers)) * numRightPages + numLeftPages;
46 }
47
48
49
50
51 /**
52  * An implementation of Iterator that provides an iterator interface for this operator.
53  */
54 private class BNLIIterator implements Iterator<Record> {
55     private Iterator<Record> leftIterator;
56     private Iterator<Record> rightIterator;
57     private Record leftRecord;
58     private Record nextRecord;
59
60     public BNLIIterator() throws QueryPlanException, DatabaseException {
61         this.leftIterator = BNLIOperator.this.getLeftSource().iterator();
62         this.rightIterator = null;
63         this.leftRecord = null;
64         this.nextRecord = null;
65     }
66
67     /**
68      * Checks if there are more record(s) to yield
69      *
70      * @return true if this iterator has another record to yield, otherwise false
71      */
72     public boolean hasNext() {
73         if (this.nextRecord != null) {
74             return true;
75         }
76         while (true) {
77             if (this.leftRecord == null) {
78                 if (this.leftIterator.hasNext()) {
79                     this.leftRecord = this.leftIterator.next();
80                     try {
81                         this.rightIterator = BNLIOperator.this.getRightSource().iterator();
82                     } catch (QueryPlanException q) {
83                         return false;
84                     } catch (DatabaseException e) {
85                         return false;
86                     }
87                 } else {
88                     return false;
89                 }
90             }
91             while (this.rightIterator.hasNext()) {
92                 Record rightRecord = this.rightIterator.next();
93                 DataBox leftJoinValue = this.leftRecord.getValues().get(BNLIOperator.this.getLeftColumnIndex());
94                 DataBox rightJoinValue = rightRecord.getValues().get(BNLIOperator.this.getRightColumnIndex());
95                 if (leftJoinValue.equals(rightJoinValue)) {
96                     List<DataBox> leftValues = new ArrayList<DataBox>(this.leftRecord.getValues());
97                     List<DataBox> rightValues = new ArrayList<DataBox>(rightRecord.getValues());
98                     leftValues.addAll(rightValues);
99                     this.nextRecord = new Record(leftValues);
100                     return true;
101                 }
102             }
103             this.leftRecord = null;
104         }
105     }
106
107     /**
108      * Yields the next record of this iterator.
109      *
110      * @return the next Record

```

```

    * @throws NoSuchElementException if there are no more Records to yield
    */
    public Record next() {
        if (this.hasNext()) {
            Record r = this.nextRecord;
            this.nextRecord = null;
            return r;
        }
        throw new NoSuchElementException();
    }

    public void remove() {
        throw new UnsupportedOperationException();
    }
}

```