dreamlegends / DB2

Branch: master ▾ DB2 / src / test / java / index / TestBPlusNode.java

Find file Copy path

dreamlegends init add all files

2c520b8 26 days ago

1 contributor

71 lines (58 sloc) 2.14 KB

Raw Blame History

```java
package index;

import static org.junit.Assert.assertEquals;

import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Optional;

import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.DisableOnDebug;
import org.junit.rules.TemporaryFolder;
import org.junit.rules.TestRule;
import org.junit.rules.Timeout;

import databox.DataBox;
import databox.IntDataBox;
import databox.Type;
import io.PageAllocator;
import table.RecordId;

public class TestBPlusNode {
    public static final String testFile = "TestBPlusNode";

    @Rule
    public TemporaryFolder tempFolder = new TemporaryFolder();

    // 1 seconds max per method tested.
    @Rule
    public TestRule globalTimeout = new DisableOnDebug(Timeout.seconds(1));

    private BPlusTreeMetadata getBPlusTreeMetadata(Type keySchema, int order)
        throws IOException {
      File file = tempFolder.newFile(testFile);
      PageAllocator allocator = new PageAllocator(file.getAbsolutePath(), false);
      return new BPlusTreeMetadata(allocator, keySchema, order);
    }

    @Test
    public void testFromBytes() throws IOException {
      int d = 5;
      BPlusTreeMetadata meta = getBPlusTreeMetadata(Type.intType(), d);
    // Leaf node.
    List<DataBox> leafKeys = new ArrayList<>();
    List<RecordId> leafRids = new ArrayList<>();
    for (int i = 0; i < 2 * d; ++i) {
      leafKeys.add(new IntDataBox(i));
      leafRids.add(new RecordId(i, (short) i));
    }
    LeafNode leaf = new LeafNode(meta, leafKeys, leafRids, Optional.of(42));

    // Inner node.
    List<DataBox> innerKeys = new ArrayList<>();
    List<Integer> innerChildren = new ArrayList<>();
    for (int i = 0; i < 2 * d; ++i) {
      innerKeys.add(new IntDataBox(i));
      innerChildren.add(i);
    }
    innerChildren.add(2*d);
    InnerNode inner = new InnerNode(meta, innerKeys, innerChildren);

    int leafPageNum = leaf.getPage().getPageNum();
    int innerPageNum = inner.getPage().getPageNum();
    assertEquals(leaf, BPlusNode.fromBytes(meta, leafPageNum));
    assertEquals(inner, BPlusNode.fromBytes(meta, innerPageNum));
  }
}
```

dreamlegends / DB2

Branch: master ▾ DB2 / src / test / java / query / TestINLJ.java

Find file Copy path

dreamlegends init add all files 2c520b8 26 days ago

1 contributor

165 lines (105 sloc) 4.38 KB

Raw Blame History

```
package query;

import database.Database;
import database.DatabaseException;
import databox.*;
import index.BPlusTree;
import index.BPlusTreeException;
import table.*;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import org.junit.Rule;

import org.junit.rules.TemporaryFolder;

import java.io.File;
import java.io.IOException;
import java.nio.charset.Charset;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.*;

import static org.junit.Assert.assertEquals;

public class TestINLJ {
    public static final String TestDir = "testDatabase";
    private Database db;
    private String filename;
    private File file;
    private String btree_filename = "TestBPlusTree";

    @Rule
    public TemporaryFolder tempFolder = new TemporaryFolder();

    @Before
    public void beforeEach() throws Exception {
        File testDir = tempFolder.newFolder(TestDir);
        this.filename = testDir.getAbsolutePath();
        this.db = new Database(filename);
        this.db.deleteAllTables();
        this.file = tempFolder.newFile(btree_filename);
    }

    @After
    public void afterEach() {
        this.db.deleteAllTables();
        this.db.close();
    }

    private BPlusTree getBPlusTree(Type keySchema, int order) throws BPlusTreeException {
        return new BPlusTree(file.getAbsolutePath(), keySchema, order);
    }


    @Test
    public void testINLJ_SJoinE() throws DatabaseException, BPlusTreeException, IOException, QueryPlanException {


        // create second table
        String table1Name = "student";
        String table2Name = "enrollment";

        Database.Transaction t1 = db.beginTransaction();

        BPlusTree rightBtree = loadStudent(t1);
        loadEnrollment(t1);

        // ******************* WRITE YOUR CODE BELOW ***********************
        // init INLJ Operator

        // loop and print result
        // ******************* WRITE YOUR CODE ABOVE ***********************

        t1.end();

        throw new UnsupportedOperationException("TODO: implement");


    }

    @Test
    public void testINLJ_SJoinEJoinC() throws DatabaseException, BPlusTreeException, IOException, QueryPlanException {

        // create second table
        String table1Name = "student";
        String table2Name = "enrollment";

        Database.Transaction t1 = db.beginTransaction();

        BPlusTree rightBtree = loadStudent(t1);
        loadEnrollment(t1);


        // ******************* WRITE YOUR CODE BELOW ***********************
        // init BtreeIndexScanOperator

        // init INLJ Operator

        // loop and print result
        // ******************* WRITE YOUR CODE ABOVE ***********************


        // ******************* WRITE YOUR CODE BELOW ***********************
        // use TestSourceOperator create a new DataSource that contains the join result
        // ******************* WRITE YOUR CODE ABOVE ***********************


        // ******************* WRITE YOUR CODE BELOW ***********************
        // init BtreeIndexScanOperator

        // init INLJ

        // loop and print result
        // ******************* WRITE YOUR CODE ABOVE ***********************

        t1.end();
        throw new UnsupportedOperationException("TODO: implement");

    }

    private BPlusTree loadStudent(Database.Transaction t1) throws  DatabaseException, BPlusTreeException, IOException{
        // Create student table/Schema

        // create b+ tree on id

        // create table


        // read from csv file

        // add each line to record and create a b+tree

        throw new UnsupportedOperationException("TODO: implement");
    }

    private void loadEnrollment(Database.Transaction t1) throws  DatabaseException, BPlusTreeException, IOException{
        // Create student table/Schema

        // create b+ tree on id

        // create table


        // read from csv file

        // add each line to record (you can create a tree here, change the return type)

        throw new UnsupportedOperationException("TODO: implement");    }

    private BPlusTree loadCourse(Database.Transaction t1) throws  DatabaseException, BPlusTreeException, IOException{
        // Create student table/Schema

        // create b+ tree on id

        // create table


        // read from csv file

        // add each line to record and create a b+tree

        throw new UnsupportedOperationException("TODO: implement");    }
}
```