

COP 6726:DataBase System Implementation

Project 3:Relational Operations

Venkateswarlu Tanneru
vtanneru@ufl.edu

Rohit Yerramsetty
rohityerramsetty@ufl.edu

Monday 4 April, 2022

1 Instructions to run the Project:

This program was written, run, and tested on the Linux operating system. Please change all of the paths in the "test.cat" file before proceeding with the following steps. The first line of test.cat should provide the path to the catalogue file (catalog path), the path to heap files (dbfile dir), and the path to the tpch-dbgen data tables (tpch-dbgen path) (tpch dir).

- Run **a2test** file to generate the bin files which we use for this Project
 - make clean
 - make a2test.out
 - ./a2test.out
- Run **test.out** to make the output and run the queries for this Project
 - make
 - ./test.out (Query Number)
here the query no should be from 1-5 so that the output displayed is for that particular query
- Run the **runtestcases.sh** file to run all the queries at a time and make their results appeared in **output1.txt** file.
 - make clean
 - make
 - ./runtestcases.out
- Run the **gtest** files to generate gtest results
 - make gtest
 - ./gtest.out

2 Files Location:

- **MakeFile**: Make File runs the file structure in making the interpreted files in project.
- **tpch/**: Contains .tbl files which were generated using tpch data generator.
- **gtest/**: Google Test Unit testing library.
- **catalog**: Catalog file for the schema of .tbl files
- **test/**: Contains .bin files.

3 DBFiles extended function classes:

3.1 RelationOP Classes

In this project, the RelationOp class is utilized as a foundation class to define all of the relational operators. All operator classes inherit the RelationOp base class. This class inherits two methods from all relational operators that derive from it. These are the procedures:

3.1.1 Wait_Function

This function, as the name says, makes the caller wait until the current working thread of the relational operator completes the execution. The caller is then the only one who has the authority to start the execution. This solution assures thread safety for a resource shared by many threads.

3.1.2 Use_Pages_n

Several relational operators developed in this research make use of BigQ objects with run lengths. This function aids in the calculation of the run length for various BigQ objects.

3.2 SelectFile Class

This class is derived from the RelationOp class. The selectFile class selects a DBFile and reads the records contained within it. It also includes a CNF predicate that is applied to the DBFile's returned records. Once the predicate is applied, the records are placed in the output stream. It follows the processes outlined below.

3.3 SelectPipe Class

Like the SelectFile class, the SelectPipe class reads a DBFile, applies the predicate to the obtained items, and then writes the data to the output pipe. Rather of reading data from a DBFile, the SelectPipe class reads records from an input Pipe and then applies a CNF predicate to them before pushing them into the output pipe.

3.4 Project Class

The Class project modifies the number of characteristics available to records in a relation. The RelationOp class is the Project class's foundation.

3.5 Join Class

The Join class is built on the RelationOp class. It implements two join algorithms in addition to all of the inherited methods from the base class. When the OrderMaker object is not properly constructed after utilizing a certain CNF, the Sorted-Merge Join Algorithm is utilized as the starting algorithm. This algorithm is utilized in this situation. In all circumstances other than the previous one, the Block-Nested Join approach is used, allowing the CNF predicate to be implemented without the use of an OrderMaker object.

3.6 Duplicate removal Class

The DISTINCT command in SQL data manipulation language is used to get only distinct records or values from a table's fetched records. This Class implements the DISTINCT keyword functionality, sorting the resulting data and deleting duplicates in a defined schema.

3.7 Sum Class

In SQL data manipulation language, the SUM aggregate function is used to compute the sum of all or distinct values in an expression. This class implements the sum aggregation algorithm. A function object is used in this implementation. When the rows from the relation are chosen, this function is applied to the appropriately chosen rows, and the total is calculated. The output is then sent to the output stream as a record.

3.8 GroupBy Class:

The GroupBy statement is used with aggregate functions in SQL data manipulation language; when used with the GroupBy clause, the aggregate functions are applied to the specified groups formed by the GroupBy statement. This class implements the GroupBy feature, which allows aggregate methods to be applied to groups of objects in the same order.

3.9 WriteOut Class

This class performs the same job as the print function. The documents display on the screen when you use the Print feature. The WriteOut class, on the other hand, operates in a somewhat different manner it accepts records from the input stream and writes them to a text file as raw records.

4 Results:

4.1 Test Case 1

```
www.Camero@ubuntu:~/my/hg/s/063/Projects/PS - ./test.out 1
** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location: catalog
tpch files dir: ./
heap files dir: ./

./partsup.bin
1
ps_partkey: [6333], ps_supkey: [6334], ps_avallqy: [3711], ps_supplycost: [1.01], ps_comment: [s use slyly. fluffily express requests wake carefully ironic packages]
ps_partkey: [9697], ps_supkey: [4098], ps_avallqy: [3012], ps_supplycost: [1.01], ps_comment: [s the bold pinto beans cajole carefully after the slyly unusual instructions. slyly special packages above the unusual, bold packages cajole blithely even Tiresias.
theodolites among the foxes are]
ps_partkey: [20488], ps_supkey: [469], ps_avallqy: [6884], ps_supplycost: [1], ps_comment: [furiously among the slyly ironic instructions. final, unusual packages wake slyly. final accounts cajole. deposits above the l]
ps_partkey: [27115], ps_supkey: [9618], ps_avallqy: [7968], ps_supplycost: [1.02], ps_comment: [e regular, ironic duponts. slyly special requests cajole quickly across the blithely express requests. deposits unwind carefully pending theodolites. pinto beans a
bout the even, regular theodolite]
ps_partkey: [34494], ps_supkey: [3921], ps_avallqy: [7438], ps_supplycost: [1.02], ps_comment: [regular excuses. final, regular deposits wake. pinto beans according to th]
ps_partkey: [43172], ps_supkey: [685], ps_avallqy: [6088], ps_supplycost: [1.01], ps_comment: [ites integrate blithely above the slyly regular instructions. asymptotes besides the regular, even accounts haggle carefully slyly bold requests. even pinto beans ]
ps_partkey: [43764], ps_supkey: [1277], ps_avallqy: [2244], ps_supplycost: [1.02], ps_comment: [; furious, ironic requests nag furiously against the silent packages-- furiously pending pinto beans use blithely careful]
ps_partkey: [51671], ps_supkey: [4777], ps_avallqy: [1399], ps_supplycost: [1.02], ps_comment: [lously. blithely bold requests haggle furiously. slyly final requests sleep. final, final theodolites cajole. accounts play about the slyly unusual requests. bold
courts haggle. bol]
ps_partkey: [60953], ps_supkey: [954], ps_avallqy: [8611], ps_supplycost: [1.01], ps_comment: [illy even dolphins wake carefully about the slyly final pinto beans]
ps_partkey: [61707], ps_supkey: [1788], ps_avallqy: [1178], ps_supplycost: [1.02], ps_comment: [ide of the unusual, regular excuses. unusual, special packages are carefully across the even theodolites: fur]
ps_partkey: [71984], ps_supkey: [6999], ps_avallqy: [6816], ps_supplycost: [1.01], ps_comment: [eodolites are blithely across the special requests. quickly regular excuses are furiously against the slyly final accou]
ps_partkey: [74375], ps_supkey: [6883], ps_avallqy: [864], ps_supplycost: [1.02], ps_comment: [lithely express asymptotes nag regular packages. special, ruthless instructions against the furiously ruthless packages boost around the packages. slyly bold accoun
ts use. furiously ironic pa]
ps_partkey: [76994], ps_supkey: [3982], ps_avallqy: [3712], ps_supplycost: [1], ps_comment: [; carefully ironic platelets cajole furiously among the furiously regular asymptotes. furiously express asymptotes wake caref]
ps_partkey: [93653], ps_supkey: [3654], ps_avallqy: [4473], ps_supplycost: [1.02], ps_comment: [f of the carefully final requests. bold deposits are slyly. instructions nod furiously instructions. careful]
ps_partkey: [102497], ps_supkey: [2488], ps_avallqy: [6481], ps_supplycost: [1], ps_comment: [fully final accounts. even accounts after the carefully final accounts haggle according to the blithely special requests. carefully unusual]
ps_partkey: [122545], ps_supkey: [5856], ps_avallqy: [5753], ps_supplycost: [1], ps_comment: [e the quickly ironic dependencies. slyly ironic accounts]
ps_partkey: [139711], ps_supkey: [9712], ps_avallqy: [4286], ps_supplycost: [1.01], ps_comment: [ully unusual escapades sleep along the special instructions. final, bold ideas across the slyly ironic ideas sleep dependc]
ps_partkey: [155112], ps_supkey: [5113], ps_avallqy: [7635], ps_supplycost: [1], ps_comment: [refully bold packages. special sonas cajole according to the foxes. furiously even accou]
ps_partkey: [158893], ps_supkey: [5639], ps_avallqy: [3751], ps_supplycost: [1], ps_comment: [lously unusual gifts maintain quickly according to the slyly pending deposits. quickly ]
ps_partkey: [193659], ps_supkey: [6179], ps_avallqy: [6686], ps_supplycost: [1.01], ps_comment: [can haggle. quickly express packages are blithely. even requests against the silent accounts sleep special packages. ironic ideas according to the furiously regul
ar dolphins use quickly plate]
ps_partkey: [193981], ps_supkey: [3982], ps_avallqy: [619], ps_supplycost: [1.01], ps_comment: [ic accounts after the unusual, regular instructions grow carefully around the blithely unusual dependencies. pending accounts along the bl]

query! returned 21 records
```

Figure 1: Output Test case 1

4.2 Test Case 2

```
venkytanneru@ubuntu:/mnt/hgfs/DBSI/Projects/P3$ ./test.out 2

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location:      catalog
tpch files dir:        ./
heap files dir:        ./

int: [31], string: [slate seashell steel medium moccasin], double: [931.03]
int: [1030], string: [orange floral olive ivory lace], double: [931.03]
int: [2029], string: [midnight brown dim violet almond], double: [931.02]
int: [3028], string: [puff slate tomato moccasin azure], double: [931.02]
int: [4027], string: [white ivory moccasin coral puff], double: [931.02]
int: [5026], string: [blanched blush pink light wheat], double: [931.02]
int: [6025], string: [purple medium light aquamarine dark], double: [931.02]
int: [7024], string: [forest rosy peach antique midnight], double: [931.02]
int: [8023], string: [mint salmon moccasin blanched beige], double: [931.02]
int: [9022], string: [peru misty sandy dark drab], double: [931.02]
int: [10021], string: [blush steel green sienna snow], double: [931.02]
int: [11020], string: [plum khaki powder beige peru], double: [931.02]

query2 returned 12 records
```

Figure 2: Output Test case 2

4.3 Test Case 3

```
venkytanneru@ubuntu:/mnt/hgfs/DBSI/Projects/P3$ ./test.out 3

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location:      catalog
tpch files dir:        ./
heap files dir:        ./

double: [9.24623e+07]

query3 returned 1 records
```

Figure 3: Output Test case 3

4.4 Test Case 4

•

```
venkytanneru@ubuntu:/mnt/hgfs/DBSI/Projects/P3$ ./test.out 4

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location:      catalog
tpch files dir:        ./
heap files dir:        ./

query4
./partsupp.bin
Enter sort merge

Finish read from sorted pipe
double: [4.00421e+08]
query4 returned 1 recs
```

Figure 4: Output Test case 4

4.5 Test Case 5

```
venkytanneru@ubuntu:/mnt/hgfs/DBSI/Projects/P3$ ./test.out 5

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location:      catalog
tpch files dir:        ./
heap files dir:        ./

./partsupp.bin
query5 finished..output written to file ps.w.tmp
```

Figure 5: Output Test case 5

5 Gtest:

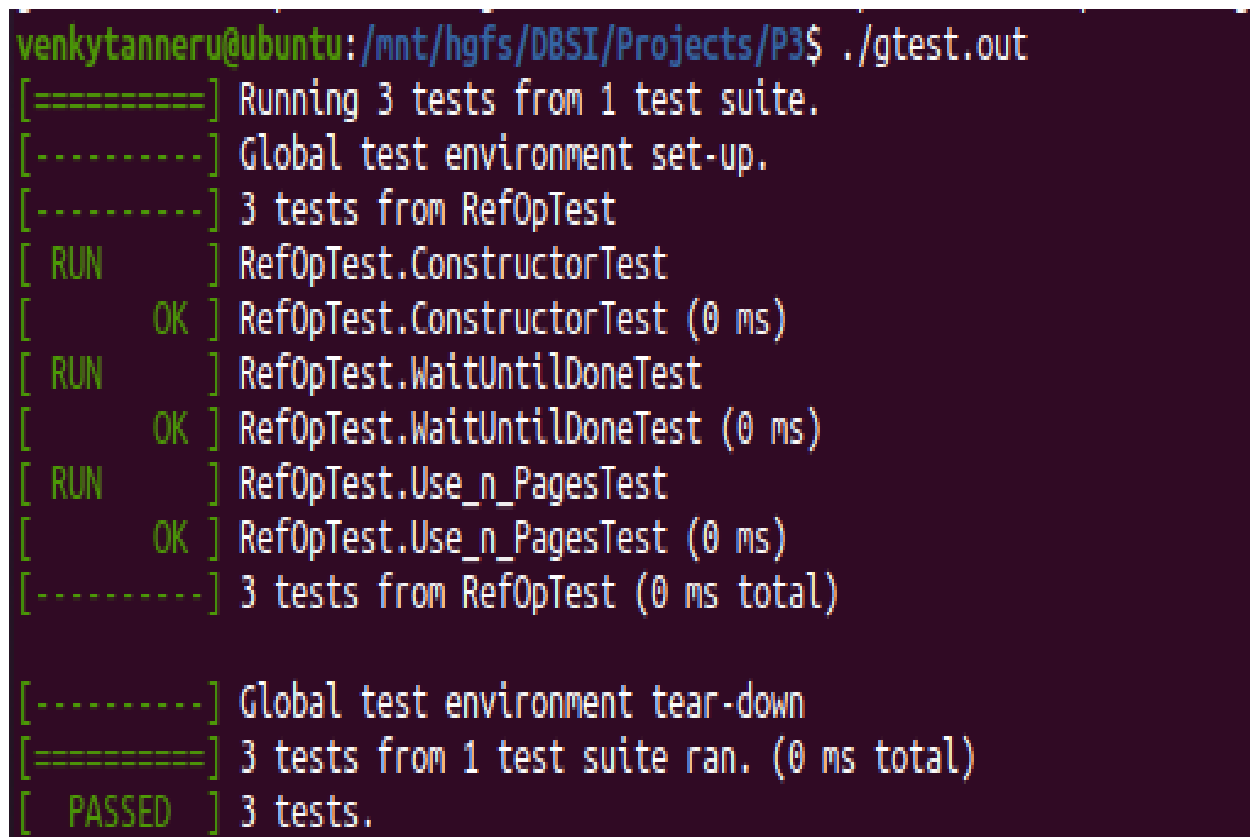
To begin running the gtest, go to the project's root directory, which includes the gtest.cpp file.

5.1 Commands To Run the Gtest Program:

- Type **make clean** and then **make gtest** to compile Google Test.
- Type **./gtest.out** to run the unit tests.
- The unit tests should run and the gtest results should be shown.

5.2 Gtest results:

we ran three tests on gtests. The testing is done on paper records. Create is being tested after successfully comparing records. On successfully comparing runs testing non-class method records, Open and Close. Check to see if the records are successfully placed in the Created file. Below is a snapshot of the gtest.



```
venkytanneru@ubuntu:/mnt/hgfs/DBSI/Projects/P3$ ./gtest.out
[=====] Running 3 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 3 tests from RefOpTest
[ RUN    ] RefOpTest.ConstructorTest
[       OK ] RefOpTest.ConstructorTest (0 ms)
[ RUN    ] RefOpTest.WaitUntilDoneTest
[       OK ] RefOpTest.WaitUntilDoneTest (0 ms)
[ RUN    ] RefOpTest.Use_n_PagesTest
[       OK ] RefOpTest.Use_n_PagesTest (0 ms)
[-----] 3 tests from RefOpTest (0 ms total)

[-----] Global test environment tear-down
[=====] 3 tests from 1 test suite ran. (0 ms total)
[ PASSED ] 3 tests.
```

Figure 6: Gtest Results

5.3 Runtestcases file

If we run the runttestcases.sh file using the command **./runttestcases.out** then the results of all the queries were printed at **Output1.txt**

```

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location: catalog
tpch files dir: ./
heap files dir: ./

./partsupp.bin
1
ps_partkey: [5388], ps_supply: [5388], ps_availability: [3711], ps_supplycost: [1.81], ps_comment: [s use slyly, fluffily express requests wake carefully ironic packages]
ps_partkey: [5897], ps_supply: [4898], ps_availability: [3812], ps_supplycost: [1.81], ps_comment: [s the bold pinto beans cajole carefully after the slyly unusual instructions. slyly special packages above the unusual, bold packages cajole blithely even Tiresias. theodolites among the foxes are]
ps_partkey: [28468], ps_supply: [469], ps_availability: [6884], ps_supplycost: [1], ps_comment: [furiously among the slyly ironic instructions. final, unusual packages wake slyly. final accounts cajole. deposits above the i]
ps_partkey: [27515], ps_supply: [9618], ps_availability: [7966], ps_supplycost: [1.82], ps_comment: [e regular, ironic duponts. slyly special requests cajole quickly across the blithely express requests. deposits unwind carefully pending theodolites. pinto beans about the even, regular theodolite]
ps_partkey: [34868], ps_supply: [9981], ps_availability: [7438], ps_supplycost: [1.82], ps_comment: [regular excuses. final, regular deposits wake. pinto beans according to th]
ps_partkey: [43375], ps_supply: [6881], ps_availability: [6688], ps_supplycost: [1.81], ps_comment: [liss integrate blithely above the slyly regular instructions. asymptotes besides the regular, even accounts haggle carefully slyly bold requests. even pinto beans ]
ps_partkey: [43764], ps_supply: [1277], ps_availability: [2344], ps_supplycost: [1.82], ps_comment: [furious, ironic requests nag furiously against the silent packages-- furiously pending pinto beans use blithely careful]
ps_partkey: [51671], ps_supply: [4177], ps_availability: [3399], ps_supplycost: [1.82], ps_comment: [lously, blithely bold requests haggle furiously. slyly final requests sleep. final, final theodolites cajole. accounts play about the slyly unusual requests. bold courts haggle. bol]
ps_partkey: [58951], ps_supply: [984], ps_availability: [8414], ps_supplycost: [1.81], ps_comment: [lily even dolphins wake carefully about the slyly final pinto beans]
ps_partkey: [61797], ps_supply: [3788], ps_availability: [3378], ps_supplycost: [1.82], ps_comment: [liss of the unusual, regular excuses. unusual, special packages are carefully across the even theodolites: fur]
ps_partkey: [71984], ps_supply: [6999], ps_availability: [6816], ps_supplycost: [1.81], ps_comment: [eodolites are blithely across the special requests. quickly regular excuses are furiously against the slyly final accou]
ps_partkey: [74379], ps_supply: [6883], ps_availability: [864], ps_supplycost: [1.82], ps_comment: [lithely express asymptotes nag regular packages. special, ruthless instructions against the furiously ruthless packages boont around the packages. slyly bold accounts use. furiously ironic ps]
ps_partkey: [76994], ps_supply: [9981], ps_availability: [9712], ps_supplycost: [1], ps_comment: [e carefully ironic platelets cajole furiously among the furiously regular asymptotes. furiously express asymptotes wake caref]
ps_partkey: [88851], ps_supply: [3854], ps_availability: [4471], ps_supplycost: [1.82], ps_comment: [f of the carefully final requests. bold deposits are slyly. instructions not furiously instructions. careful]
ps_partkey: [188497], ps_supply: [2488], ps_availability: [6491], ps_supplycost: [1], ps_comment: [fully final accounts. even accounts after the carefully final accounts haggle according to the blithely special requests. carefully unusual]
ps_partkey: [112543], ps_supply: [5856], ps_availability: [5751], ps_supplycost: [1], ps_comment: [e the quickly ironic dependencies. slyly ironic accounts]
ps_partkey: [139718], ps_supply: [9712], ps_availability: [4286], ps_supplycost: [1.81], ps_comment: [lily unusual escapades sleep along the special instructions. final, bold ideas across the slyly ironic ideas sleep dependec]
ps_partkey: [155112], ps_supply: [5113], ps_availability: [2035], ps_supplycost: [1], ps_comment: [refally bold packages. special times cajole according to the foxes. furiously even accou]
ps_partkey: [158893], ps_supply: [5639], ps_availability: [3751], ps_supplycost: [1], ps_comment: [lously unusual gifts maintain quickly according to the slyly pending deposits. quickly ]
ps_partkey: [193593], ps_supply: [6179], ps_availability: [6686], ps_supplycost: [1.81], ps_comment: [can haggle. quickly express packages are blithely. even requests against the silent accounts sleep special packages. ironic ideas according to the furiously regular dolphins use quickly plate]
ps_partkey: [193981], ps_supply: [3982], ps_availability: [619], ps_supplycost: [1.81], ps_comment: [lc accounts after the unusual, regular instructions grow carefully around the blithely unusual dependencies. pending accounts along the bl]

query3 returned 21 records

*****

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location: catalog
tpch files dir: ./
heap files dir: ./

int: [31], string: [slate seashell steel medium moccasin], double: [931.83]
int: [1838], string: [orange floral olive ivory lace], double: [931.83]
int: [1829], string: [midnight brown dia violet almond], double: [931.82]
int: [1828], string: [puff slate tomato moccasin sure], double: [931.82]
int: [4827], string: [white ivory moccasin coral puff], double: [931.82]
int: [5826], string: [blanched blush pink light wheat], double: [931.82]
int: [5825], string: [purple medium light aquamarine dark], double: [931.82]
int: [7824], string: [forest rusy peach antique midnight], double: [931.82]
int: [8823], string: [mint salmon moccasin blanched beige], double: [931.82]
int: [9822], string: [peru misty sandy dark drab], double: [931.82]
int: [10821], string: [blush steel green sienna snow], double: [931.82]
int: [11820], string: [plum khaki powder beige peru], double: [931.82]

query2 returned 12 records

*****

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location: catalog
tpch files dir: ./
heap files dir: ./

double: [9.24623e+07]

```

Figure 7: Runttest cases Results

```

*****

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location: catalog
tpch files dir: ./
heap files dir: ./

query4
./partsupp.bin
Enter sort merge
Finish read from sorted pipe
double: [4.88421e+08]
query4 returned 1 recs

*****

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location: catalog
tpch files dir: ./
heap files dir: ./

./partsupp.bin
query5 finished..output written to file ps.w.tmp

```

Figure 8: Runttest cases Results