Deep Bodra
5801 1841

REPORT

GROUP MEMBER NAMES

Individual Project

RUNNING THE CODE

Run Test (test.cc)

Make sure that *.bin.meta files are there for each *.bin file

```
make test.out
./test.out [1-6]
```

Run gtest (gtest_relop.cc)

```
make gtest_relop.out
./gtest_relop.out
```

IMPLEMENTATION DETAILS

1. RelationalOpThreadMemberHolder
    a. Since functions that run as a thread accept only one argument, I have made use of instances of RelationalOpThreadMemberHolder which holds necessary arguments for a given relational operator.
    b. This avoids a lot of duplicate code as there are a lot of arguments that are the same for many functions

2. SelectPipe
    a. Operate
        i. It reads the records from the input pipe sequentially and compares them with the CNF.
        ii. If the record matches the CNF then it is added to the pipe

3. SelectFile
    a. Operate
        i. It reads the records from the DBFile instead of the pipe.
        ii. It hands over the task of finding the record with a given CNF to the DBFile
        iii. If the DBFile is a heap file then the records will be fetched sequentially
        iv. If the DBFile is a sorted file then the records will be fetched using binary search (if the sort order is compatible) and sequentially otherwise

4. Project
    a. It reads the records from inputPipe and uses the Project method to actually project and pushes to the output pipe

5. Join
    a. Operate
        i. It uses GetSortOrders function to see if the CNF is acceptable or not
        ii. If yes then SortMergeJoin is used otherwise NestedLoopJoin is used
    b. SortMergeJoin
        i. Two BigQ instances are used to get the records from the two input pipes in sorted order in the pipes sortedLeftPipe and sortedRightPipe respectively
        ii. The smaller of the two head records is removed repeatedly until the comparison of the two head records becomes 0
        iii. The head records from the sortedLeftPipe are removed until the comparison with the right head is 0

iv. These removed records are stored in a recordBuffer (an instance of FixedSizeRecordBuffer) which has the size equal to the runLength page of records
v. Each record in the buffer is merged with the head of the sortedRightPipe until the record in the buffer continues to match with the head of the pipe
vi. The mergedRecord is sent to the outputPipe
vii. This is done until there are records in both the pipes

c. NestedLoopJoin
i. It first dumps all the records in the rightPipe into a DBFile using PipeToFile function
ii. The records from the leftPipe are loaded into the leftBuffer (an instance of FixedSizeRecordBuffer)
iii. Each record from the leftPipe is compared with each record in the dumpFile.
iv. If they match then the records are merged and pushed into the outputPipe

d. FixedSizeRecordBuffer
i. It basically a wrapper on top of an array pointer that can accommodate only runLength pages of records only

6. DuplicateRemoval
a. Operate
i. It first uses BigQ to sort the records based on the CNF
ii. It then compares the adjacent records.
iii. If they are different then the currentRecord
iv. The idea is that if there are duplicates then they must appear adjacent when sorted

7. Sum
a. Operate
i. It calls a template function CalculateSum based on the function type (Int or Double)

b. CalculateSum
i. It reads the records from the pipe and applies the function to the record and keeps adds it to the current sum
ii. Once the final sum has been calculated it creates a new schema with one attribute of either Int or Double based on the type
iii. It uses this schema to create a new record with the calculated sum as the value of this new attribute using ComposeRecord

8. GroupBy
    a. Operate
        i. It calls a template function CalculateSum based on the function type (Int or Double)
    b. MakeGroups
        i. It is similar to DuplicateElimination but it keeps calculating the sum until the records match
        ii. If they don't then it uses AddGroup template function
    c. AddGroup
        i. It creates a new record with one attribute (just like in Sum) and merges that record with the current record (with unnecessary attributes stripped off using Project)

9. WriteOut
    a. Operate
        i. It calls Write method of Record for each record in the pipe
    b. Write (in Record.cc)
        i. It loops through all the attributes, casts each attribute value into pointer based on the type (Int, Double, String) and write it to the file

TEST CASES

TPCH data size=1GB

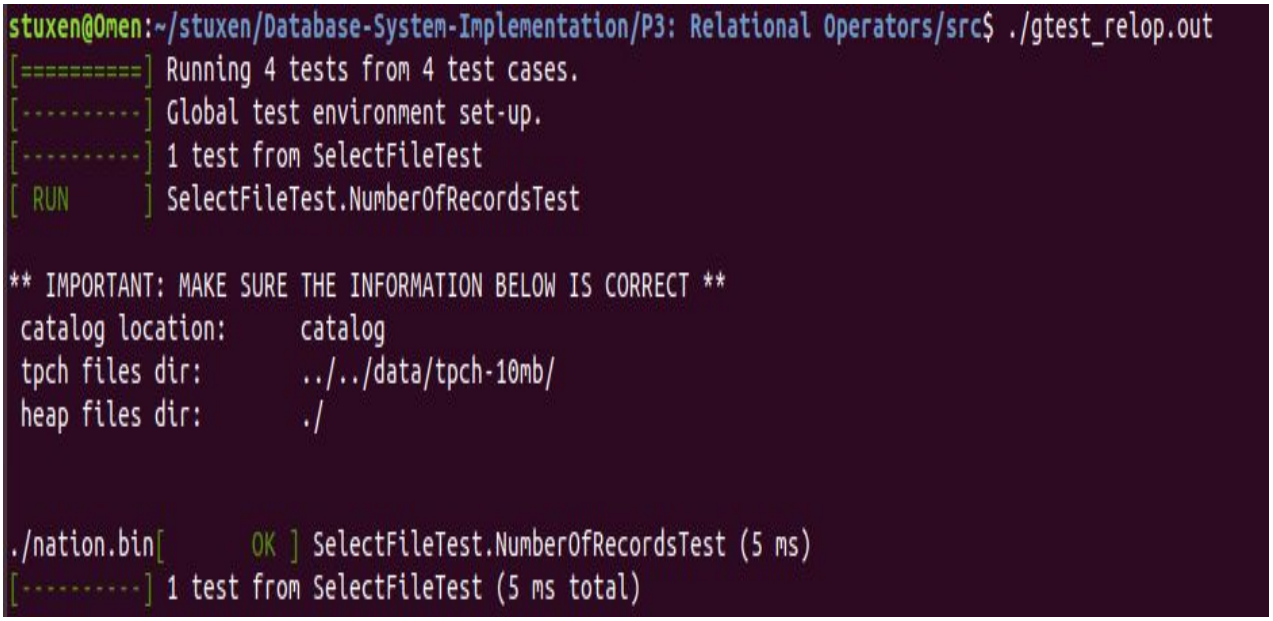
```
 tpch files dir:    ../../data/tpch-10mb/
 heap files dir:    ./


ps_partkey: [6333], ps_suppkey: [6334], ps_availqty: [3711], ps_supplycost: [1.01], ps_comment: [s use slyly. fluffily express requests wake carefully ironic packages]
ps_partkey: [9097], ps_suppkey: [4098], ps_availqty: [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: [20468], ps_suppkey: [469], ps_availqty: [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: [27115], ps_suppkey: [9618], ps_availqty: [7966], ps_supplycost: [1.02], ps_comment: [e regular, ironic dugouts. slyly special requests cajole quickly across the blithely express requests. deposits unwind
carefully pending theodolites. pinto beans about the even, regular theodolite]
ps_partkey: [34494], ps_suppkey: [9501], ps_availqty: [7438], ps_supplycost: [1.02], ps_comment: [egular excuses. final, regular deposits wake. pinto beans according to th]
ps_partkey: [43172], ps_suppkey: [685], ps_availqty: [6600], 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_suppkey: [1277], ps_availqty: [2344], 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_suppkey: [4177], ps_availqty: [3399], ps_supplycost: [1.02], ps_comment: [iously. 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_suppkey: [954], ps_availqty: [8611], ps_supplycost: [1.01], ps_comment: [ully even dolphins wake carefully about the slyly final pinto beans]
ps_partkey: [61707], ps_suppkey: [1708], ps_availqty: [3178], ps_supplycost: [1.02], ps_comment: [ide of the unusual, regular excuses. unusual, special packages are carefully across the even theodolites: furi]
ps_partkey: [71984], ps_suppkey: [6999], ps_availqty: [6016], 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_suppkey: [6883], ps_availqty: [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 accounts use. furiously ironic pa]
ps_partkey: [76994], ps_suppkey: [9502], ps_availqty: [9712], ps_supplycost: [1], ps_comment: [. carefully ironic platelets cajole furiously among the furiously regular asymptotes. furiously express asymptotes wake caref]
ps_partkey: [93653], ps_suppkey: [3654], ps_availqty: [4473], ps_supplycost: [1.02], ps_comment: [ of the carefully final requests. bold deposits are slyly. instructions nod furiously instructions. careful]
ps_partkey: [102497], ps_suppkey: [2498], ps_availqty: [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: [122543], ps_suppkey: [5056], ps_availqty: [5753], ps_supplycost: [1], ps_comment: [e the quickly ironic dependencies. slyly ironic accounts]
ps_partkey: [139711], ps_suppkey: [9712], ps_availqty: [4286], ps_supplycost: [1.01], ps_comment: [ully unusual escapades sleep along the special instructions. final, bold ideas across the slyly ironic ideas sleep dependenc]
ps_partkey: [155112], ps_suppkey: [5113], ps_availqty: [7635], ps_supplycost: [1], ps_comment: [refully bold packages. special somas cajole according to the foxes. furiously even accou]
ps_partkey: [158093], ps_suppkey: [5639], ps_availqty: [3751], ps_supplycost: [1], ps_comment: [iously unusual gifts maintain quickly according to the slyly pending deposits. quickly ]
ps_partkey: [193659], ps_suppkey: [6179], ps_availqty: [6606], 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 regular dolphins use quickly plate]
ps_partkey: [193981], ps_suppkey: [3982], ps_availqty: [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]


 query1 returned 21 records
./partsupp.bin

********************************************************************************************************************************************************************************

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:  catalog
 tpch files dir:    ../../data/tpch-10mb/
 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
./part.bin
```

```
********************************************************************************************************************************************************************************

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:  catalog
 tpch files dir:    ../../data/tpch-10mb/
 heap files dir:    ./


double: [9.24623e+07]


 query3 returned 1 records
./supplier.bin

********************************************************************************************************************************************************************************

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:  catalog
 tpch files dir:    ../../data/tpch-10mb/
 heap files dir:    ./


 query4
double: [4.00421e+08]
 query4 returned 1 recs


********************************************************************************************************************************************************************************

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:  catalog
 tpch files dir:    ../../data/tpch-10mb/
 heap files dir:    ./


 query5 finished..output written to file ps.w.tmp


********************************************************************************************************************************************************************************
```

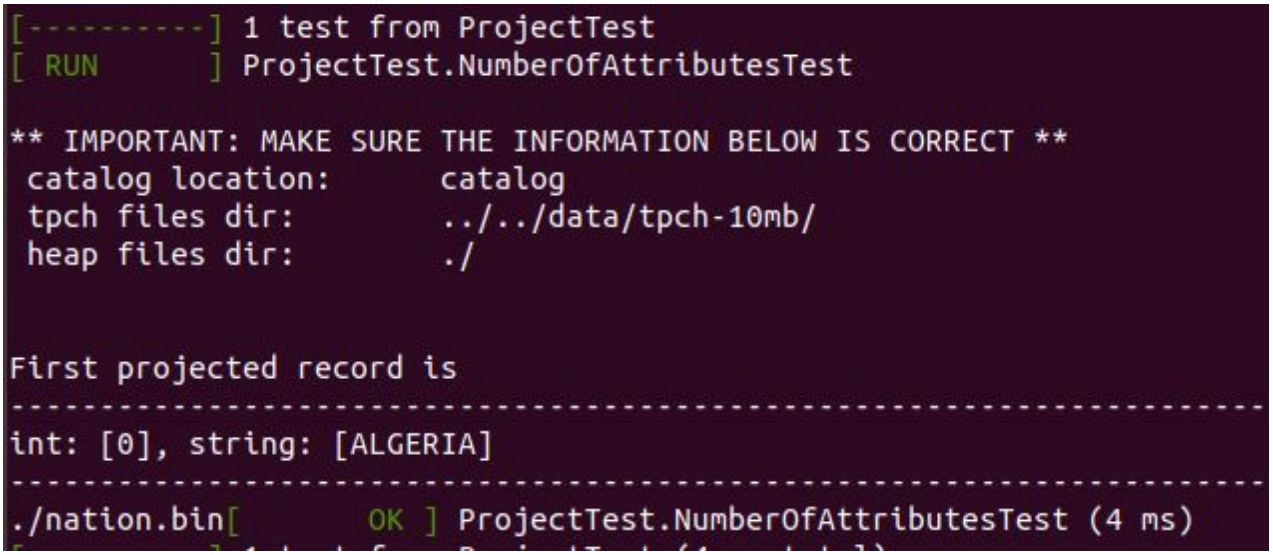
```
**********************************************************************************************************************************************************************

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:  catalog
 tpch files dir:    ../../data/tpch-10mb/
 heap files dir:    ./


 query6
double: [1.68363e+07]
double: [1.65363e+07]
double: [1.59263e+07]
double: [1.6502e+07]
double: [1.65789e+07]
double: [1.52194e+07]
double: [1.61179e+07]
double: [1.57786e+07]
double: [1.65973e+07]
double: [1.62349e+07]
double: [1.56964e+07]
double: [1.75655e+07]
double: [1.51408e+07]
double: [1.45051e+07]
double: [1.50595e+07]
double: [1.4906e+07]
double: [1.62842e+07]
double: [1.68236e+07]
double: [1.63006e+07]
double: [1.59569e+07]
double: [1.64843e+07]
double: [1.59514e+07]
double: [1.60954e+07]
double: [1.55616e+07]
double: [1.57613e+07]
 query6 returned sum for 25 groups (expected 25 groups)
```

GTEST

1. NumberOfRecordsTest
    a. It tests `SelectFile` operator based on the number of records in the output for this query
    SELECT * FROM nation WHERE n_nationkey<10
    b. The number of records in the output should be 10



2. NumberOfAttributesTest
    a. It tests `Project` operator based on the number of attributes in the output for this query
    SELECT n_nationkey, n_name FROM nation WHERE n_nationkey=0
    b. The number of attributes in the output record should be 2



3. SumCalculationTest
    a. It tests `SUM` operator based on the sum for this query
    SELECT SUM(n_nationkey) FROM nation
    b. The sum for this query should be 300. Why 300? n_nationkey= 0, 1, .. 24. Therefore, n*(n+1)/2 = 24*25/2=300

```
[----------] 1 test from ProjectTest (4 ms total)

[----------] 1 test from SumTest
[ RUN      ] SumTest.SumCalculationTest

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        ../../data/tpch-10mb/
 heap files dir:        ./


Sum record is
------------------------------------------------------------
int: [300]
------------------------------------------------------------
./nation.bin[       OK ] SumTest.SumCalculationTest (6 ms)
```

4. NumberOfAttributesTest
    a. It tests `JOIN` operator based on the number of attributes for this query
    SELECT * FROM supplier, partsupp WHERE s_suppkey = ps_suppkey
    b. The number of attributes in the output record should be 7+5=12

```
[----------] 1 test from JoinTest
[ RUN      ] JoinTest.NumberOfAttributesTest

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        ../../data/tpch-10mb/
 heap files dir:        ./


First joined record is
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
int: [1], string: [Supplier#000000001], string: [ N kD4on9OM Ipw3,gf0JBoQDd7tgrzrddZ], int: [17], string: [27-918-335-1736], double: [5755.94], string: [each slyly above the careful], int: [190000], int:
[1], int: [4988], ps_supplycost: [209.46], string: [theodolites until the regular accounts are carefully at the final courts. furiously final sentiments use ruthless, even asymptotes. carefully express ]
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
[       OK ] JoinTest.NumberOfAttributesTest (3520 ms)
[----------] 1 test from JoinTest (3520 ms total)

[----------] Global test environment tear-down
[==========] 4 tests from 4 test cases ran. (3535 ms total)
[  PASSED  ] 4 tests.
stuxen@Omen:~/stuxen/Database-System-Implementation/P3: Relational Operators/src$
```

CONCLUSION

We now have added relational operators, an integral part of the database systems.