Project 3: Relational Operations

Group Members:

1. Smrati Pandey, UFID: 4323-9459
2. Wins Goyal, UFID: 7357-1559

Instructions for Execution / Code compilation and to run tests:

I. To extract folders

a) Extract the contents of the folder, named *SmratiPandey_WinsGoyal_p3.zip*.
b) Open the terminal.
c) Navigate to the extracted folder.

II. To update "*tests.cat*"

a) In *test.cat* file, update: *catalog_path, dbfile_dir, & tpch_dir paths*.
b) Write or update these paths in the same line order as mentioned above.

III. To generate different types of *DBfiles*

Using last assignments' driver codes, create *.bin* files of *DBfiles* (can be *heap DBfile* or *sorted DBfile*) that will be needed to run this assignment.

S.No.	For Heap DBFile	For Sorted DBFile
1.	In *a1-test.cc* file, update: *catalog_path, dbfile_dir, & tpch_dir paths*	In *a2-test.h* file, update: *catalog_path, dbfile_dir, & tpch_dir paths*
2.	Run *"make a1test.out"*	Run *"make a2test.out"*
3.	Run *"./a1test.out"* Give appropriate inputs to generate *Heap DBfile*	Run *"./a2test.out"* Give appropriate inputs to generate *Sorted DBfile*

IV. To run the tests

Perform following these commands:

```
$ make clean
$ make test.out
$ ./test.out {x}  ## To run any test case b/w 1 to 6 replacing "x"
$ ./runTestCases.sh  ## To run test cases from 1 to 5
```

Last command will produce "*output1.txt*" file

Summarized Documentation of the implemented methods and functions:

This project demanded us to implement 8 relational operations, the source codes for which are written in *RelOp.h* and *RelOp.cc* These operations are encapsulated within classes, that are derived from the virtual baae class named "*RelationalOp*"

I. *RelationalOp*:

- Two of the data members of the classes of Relational Operations that are derived from this base class are common in all of them: (a) *pthread_t thread* (b) *int runLength*

- *thread* is used by the relational operations' classes to operate data asynchronously and *runLength* allows the amount of internal memory that can be used by these operations.
- Following are two public methods in this class which get executed when are called by the corresponding instances of the derived classes:
    1. *void WaitUntilDone():* This function blocks the next caller (relational operator instance), i.e. will make it wait, until the current relational operation (caller) is done executing and the thread inside the operation is destroyed.
    2. *void Use_n_Pages(int n):* This function assigns a value to '*runLength*' of the relational operator.

II. Derived Classes:

- A '*Run*' method is implemented with all derived relational operation class that enables to use the corresponding relational operation and sets it up. Common functionality of the 'Run' methods is:
    - It creates & inputs the internal data structure, used by the relational operator's *thread*, with inputs.
    - It spawns a *thread* that works from inside the relational operation, uses the created internal structure and calls the corresponding functions.
    - *Run* returns while the *thread* keeps performing until the tuples in the input pipe have been processed.
    - As the operation finishes, the output pipe is shut down.
- Following 'Relational Operators' with their corresponding functions have been implemented:

1. ***SelectPipe****:* Takes an *input pipe*, *output pipe* and a *CNF* value as the input parameters.

   It's *SetupThreadSelectPipe* method initialize the instance that will feed the records from the input pipe to a *While* loop which pushes the tuples that satisfy the CNF value, using the *CompareEngine's Compare* method, to the output pipe.

2. ***SelectFile:*** Takes a *DBFile* (that was generated earlier), a pipe and a *CNF* value as input parameters.

   It's *SetupThreadSelectFile* method calls the *initializeOperation* that gets the records from the input file scanning it and forwards those tuples to the output pipe that satisfy the *CNF* calling the input *DBFile's GetNext* method.

3. ***Project:*** Takes input pipe, output pipe, array of integers, number of attributes from input pipe and the number of attributes to keep as the input parameters.

   It's *initializeProject* method first gets the tuples from the input pipe, then passes these tuples to Record's Project method with current attribute value, input & output attribute values. This alters the records to keep only the attributes mentioned in the same order as in array of integers. These altered projected tuples are fed to the output pipe eventually.

4. ***Join:*** Takes two input pipes, one output pipe and a *CNF* value as the input parameters. The instance executes calling *join1*, *join2* and *joinFinal* methods.

   It's *initializeJoin* method calls the *GetSortOrders* function written in the *BigQ* class to create two separate *OrderMakers* for tuples of the left and right input pipes, using the *CNF*. The outputs of both of the BigQs are transferred to joinFinal (which uses a Sort-merge Join algorithm) to join the tuples from both input pipes according to the CNF value. If any of the OrderMakers is empty or not appropriate, the operation uses the block-nested loops join method.

5. ***DuplicateRemoval***: Takes input pipe, output pipe and input tuple's Schema as the input parameters.

   It's *initializeDR* method calls BigQ instance that uses OrderMaker to make an order maker from the schema passed in the input and BigQ will sort the tuples from the input pipe as per the newly constructed order maker. Then, it uses the CompareEngine's Compare method to check if every tuple is unique or not. Only the unique records are pushed into the output pipe. To eliminate duplicates, the last record is compared with the current obtained record. If same, the last record won't be pushed and rejected. Else, the last record is passed to the output pipe.

6. ***Sum***: It is initialized by the *initializeSum* method that in turn calls *SumHelper* method.

   Function is applied on each record obtained from the input pipe to update the sum everytime. After all the records are done processing, the final sum record is given to the output pipe. The sum record is created using '*InsertNewRec*' method of the *Record* class instance.

7. ***GroupBy***: It is initialized by the *initializeGB* method that in turn calls *groupByHelper* and *newHelper* methods. This has both the characteristics of the *DuplicateRemoval* & the *Sum* operators.

   First, the records from the input pipe are sorted according to the GroupBy OrderMaker of BigQ instance, i.e. grouping is done for the attributes given to the OrderMaker. This groups together the sorted tuples while the Function class updates the sum for each of these groups. The new record to the output pipe has Sum as the first attribute and the grouping attributes as the rest of the attributes.

8. ***WriteOut***: It is initialized by the *initializeWriteOut* method.

   The records from the input pipe are converted to string, so as to write them to the output file. *OutputRecFile* performs this conversion of the tuples to the strings.

Screenshots of Running All Test Case Results:

I. Results of all the queries.

```
** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
catalog location:  catalog
tpch files dir:    /home/smrati/Downloads/a3test/tpch-dbgen/
heap files dir:    /home/smrati/Downloads/a3test/bin/


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 eve
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. pi
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 pi
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 requ
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
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 furi
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
```

```
** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:  catalog
 tpch files dir:    /home/smrati/Downloads/a3test/tpch-dbgen/
 heap files dir:    /home/smrati/Downloads/a3test/bin/


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


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

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:  catalog
 tpch files dir:    /home/smrati/Downloads/a3test/tpch-dbgen/
 heap files dir:    /home/smrati/Downloads/a3test/bin/


double: [9.24623e+07]


 query3 returned 1 records


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

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:  catalog
 tpch files dir:    /home/smrati/Downloads/a3test/tpch-dbgen/
 heap files dir:    /home/smrati/Downloads/a3test/bin/


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

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

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:  catalog
 tpch files dir:    /home/smrati/Downloads/a3test/tpch-dbgen/
 heap files dir:    /home/smrati/Downloads/a3test/bin/


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

ps.w.tmp File

```
9958    int:[9962]
9959    int:[9963]
9960    int:[9964]
9961    int:[9965]
9962    int:[9966]
9963    int:[9967]
9964    int:[9968]
9965    int:[9969]
9966    int:[9970]
9967    int:[9971]
9968    int:[9972]
9969    int:[9973]
9970    int:[9974]
9971    int:[9975]
9972    int:[9976]
9973    int:[9977]
9974    int:[9978]
9975    int:[9979]
9976    int:[9980]
9977    int:[9981]
9978    int:[9982]
9979    int:[9983]
9980    int:[9984]
9981    int:[9985]
9982    int:[9986]
9983    int:[9987]
9984    int:[9988]
9985    int:[9989]
9986    int:[9990]
9987    int:[9991]
9988    int:[9992]
9989    int:[9993]
9990    int:[9994]
9991    int:[9995]
9992    int:[9996]
9993    int:[9997]
9994    int:[9998]
9995    int:[9999]
9996    int:[10000]
9997
```

Query 6

```
smrati@smrati-XPS-15-7590:~/Downloads/a3test$ ./test.out 6

** IMPORTANT: MAKE SURE THE INFORMATION BELOW IS CORRECT **
 catalog location:      catalog
 tpch files dir:        /home/smrati/Downloads/a3test/tpch-dbgen/
 heap files dir:        /home/smrati/Downloads/a3test/bin/


 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)
```

GTests:

- Perform following commands to execute the GTests:

```
$ make clean
$ make myGtest.out
$ ./myGtest.out
```

- Note: Before executing the above commands, generate *tpch* files (1 GB) using the *dbgen* program. If needed, also update *the catalog, the dbfile output* and *tpch directories*, in the *myGtest.cc* file.

a) *Test for Project:*
This tests the *Project* instance of the relational operator of the same name by passing 55 tuples to the input pipe1 and pipe2. Only the the attributes that to be projected are taken into account are used for records from pipe1 and pipe2 and corresponding output number of tuples that should be there are checked.

b) *Test for Select File*:
This tests the *SelectFile* instance of the relational operator of the same name by passing 55 tuples to the input pipe and check the output number of tuples that should be there as per the matching *CNF* value.

c) *Test for Select Pipe*:
This tests the *SelectPipe* instance of the relational operator of the same name by passing 55 tuples to the input pipe and check the output number of tuples in the output file that should be there as per the matching *CNF* value.

d) *Test for Write Out*:
This tests the *WriteOut* instance of the relational operator of the same name by passing 15 tuples to the input pipe and checks the output file.

Following is the screenshot of all these four test cases in execution (terminal):

```
smrati@smrati-XPS-15-7590:~/Downloads/a3test$ ./myGtest.out
[==========] Running 4 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 4 tests from SuccessTestForClearData
[ RUN      ] SuccessTestForClearData.ProjectClearData
[       OK ] SuccessTestForClearData.ProjectClearData (0 ms)
[ RUN      ] SuccessTestForClearData.SelectFileClearData
[       OK ] SuccessTestForClearData.SelectFileClearData (0 ms)
[ RUN      ] SuccessTestForClearData.SelectPipeClearData
[       OK ] SuccessTestForClearData.SelectPipeClearData (0 ms)
[ RUN      ] SuccessTestForClearData.WriteOutClearData
[       OK ] SuccessTestForClearData.WriteOutClearData (0 ms)
[----------] 4 tests from SuccessTestForClearData (0 ms total)

[----------] Global test environment tear-down
[==========] 4 tests from 1 test case ran. (1 ms total)
[  PASSED  ] 4 tests.
```