PROJECT 3 REPORT: Relational Operations

GROUP MEMBERS:

1. SHREYA DASGUPTA (UFID - 47016900)
2. YASH PATTARKINE (UFID - 28616005)

INSTRUCTIONS TO COMPILE THE CODE:

1. Unzip “shreyaDasgupta_yashPattarkine_p1.zip”. Open terminal on the machine and go to the folder where the unzipped files are. Use command -- cd<pathname/foldername>
2. Command -- make clean.
3. Compile test.cc. Use command -- make test.out
4. Run test.out. For each of the queries, give the desired number. Use command -- ./test.out [1-5]
5. For gtests, compile it using -- make gtest.out. Make sure test.cc is already is compiled and you have the output files- RelOp.o and Function.o
6. Run gtest using -- ./gtest.out

BRIEF OVERVIEW OF ALL METHODS AND ATTRIBUTES OF RelOp CLASS:

Classes:

1. **SelectPipe**: The main run method of this class is taking 4 attributes which include 2 pipes, a CNF and a record literal.
    a. **Attributes**: in this class we have used the following attributes
        i. In and out pointers of the pipe class
        ii. a pointer to CNF class
        iii. A record pointer
        iv. An integer which gives us the run length
        v. A thread
    b. **Methods**: Except for the constructor and destructor, we have 4 methods in this class, one of which is run and other 3 are:
        i. WaitUntilDone(): It just joins the current thread. It is a void returning function.
        ii. Use_n_Pages(): It takes an integer as a parameter and sets the run length equal to that integer. It does not return anything.

iii. Operation_SelectPipe(): It is the function which is called by run. It compares each of the records coming from the input pipe with the CNF and inserts it into the output pipe accordingly.

2. **SelectFile**: This has the following specifications.
    a. **Attributes**: SelectFile utilizes the following attributes in order to perform its job.
        i. a pointer to DBFile instance
        ii. A pointer to Pipe-instance
        iii. A record-pointer
        iv. Run-length which is of integer type
        v. A pointer to CNF instance
    b. **Methods**:
        i. WaitUntilDone(): It just joins the current thread. This function returns void.
        ii. Use_n_Pages(): It takes an integer as a parameter and simply sets the run length equal to that integer.
        iii. Run(): This function takes in 4 parameters - the DBFile instance pointer, the pipe, the CNF specifying the operation and the record. This is responsible for populating the SelectFile parameters with the respective function parameters and starts a thread calling the Operation_SelectFile() method.
        iv. Operation_SelectFile(): This function is the heart of this class. It simply starts scanning from the first record of the DBFile and if there is a matching record, the pipe is populated with that one. We use a while loop to perform the same.

3. **Project**: The main run method of this class is taking 5 attributes which include 2 pipes, an array of integers and 2 integers which gives us records coming through the input pipe and to keep from those input records.
    a. **Attributes**: in this class we have used the following attributes
        i. In and out pointers of the pipe class
        ii. An array of integers
        iii. 2 integers, giving the number of attributes in and out.
        iv. An integer which gives the run-length
        v. A thread

b. **Methods**: Except for the constructor and destructor, we have 4 methods in this class, one of which is run and other 3 are:
    i. WaitUntilDone(): It just joins the current thread. It is a void returning function.
    ii. Use_n_Pages(): It takes an integer as a parameter and sets the run length equal to that integer. It does not return anything.
    iii. Operation_Project(): It is the function which is called by run. It takes the order from the array and decides which attributes are to be kept from the input records and in which order they are supposed to be put.

Similarly, all the other classes have similar structure with similar attributes and methods. The only difference is the function which is called by Run (i.e the function that essentially holds the logic). So, below, we are stating the logic of only that method of each class to avoid cumbersome organization and repeatedness..

4. **Join**: The method called by Run is **Operation_Join()** which performs the key responsibility.
    a. It is just taking 2 input pipes and joining all the records from those in accordance with the CNF that is given as the parameter. We are using 2 different BigQ classes for the tuples coming from the 2 different pipes.

5. **DuplicateRemoval**: The method called by Run is **Operation_DuplicateRemoval()**
    a. It is just removing the duplicates coming from the input pipes. So basically, the output pipe will be free of redundant records. For this, we are comparing each of the records coming from the input pipe with the next record and inserting them into the output pipe only if they are different.

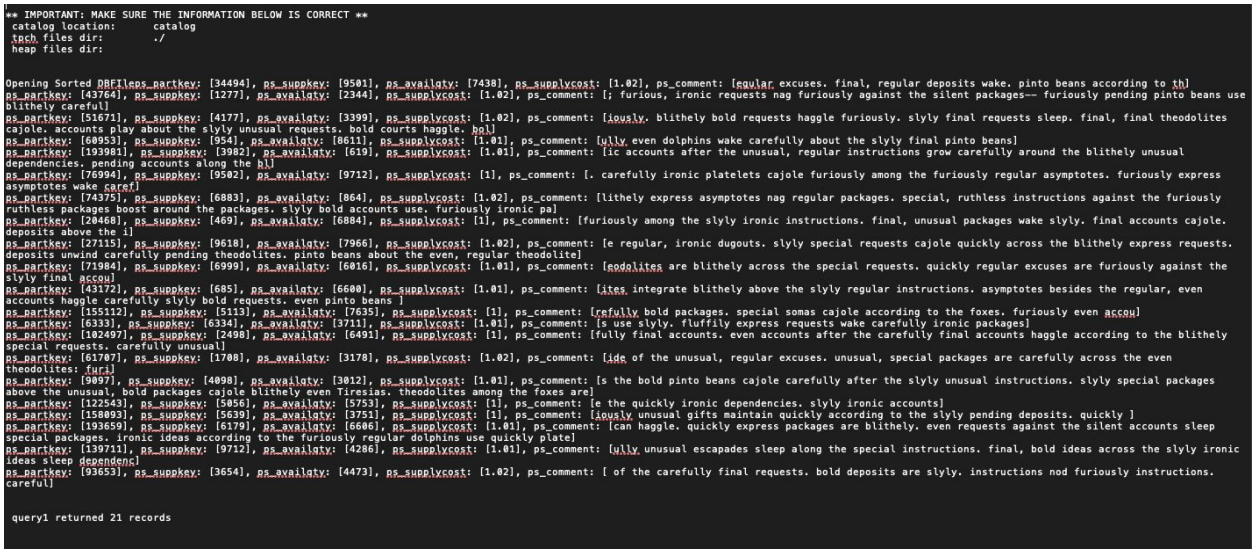
6. **Sum**: Run method initializes all the necessary parameters and calls **Operation_Sum()**
    a. This function takes the aggregate of the records from the input pipe and sums them over and pushes out the final sum as a singleton in the output pipe.
7. **GroupBy**: The method called by Run is **Operation_GroupBy()**.
    a. This works very similarly to the Sum-operation, the only difference being that it creates groups based on the specified attributes and displays the

sum of each such group on the pipe as output. The way it does this is to first sort the records and group them based on the instance from OrderMaker Class. Then we perform the summation repeatedly for each group, the sum to be computed is mentioned in the Function Class (already given).

8. **WriteOut**: The method called by Run is **Operation_WriteOut()**.
    a. In this, for each record coming out of the input pipe, we are simply writing that record to the file.

Output1.txt screen shots:

q1



q2

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


Opening Sorted DBFIleint: [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
```

q3

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


Opening Sorted DBFIledouble: [9.24623e+07]


 query3 returned 1 records
```

q4

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


 query4
Opening Sorted DBFileOpening Sorted DBFiledivi
Input pipe is shutting down
Input pipe is shutting down
double: [nan]
 query4 returned 1 recs
```

q5

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


Opening Sorted DBFileInput pipe is shutting down
 query5 finished..output written to file ps.w.tmp
```

Gtests Results: