

## 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/foldernam>`
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()**.

- 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

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

Opening Sorted DBFiles ps_partkey: [34494], ps_suppkey: [9501], ps_availability: [7438], ps_supplycost: [1.02], ps_comment: [regular, excuses, final, regular deposits wake, pinto beans according to th]
ps_partkey: [43764], ps_suppkey: [1277], ps_availability: [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_availability: [3399], ps_supplycost: [1.02], ps_comment: [Awfully, blithely bold requests haggle furiously. slyly final requests sleep, final, final theodolites
cajole, accounts play about the slyly unusual requests, bold courts haggle, hold]
ps_partkey: [68953], ps_suppkey: [954], ps_availability: [8611], ps_supplycost: [1.01], ps_comment: [lully, even dolphins wake carefully about the slyly final pinto beans]
ps_partkey: [193981], ps_suppkey: [3982], ps_availability: [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 hll]
ps_partkey: [76994], ps_suppkey: [9502], ps_availability: [9712], ps_supplycost: [1], ps_comment: [, carefully ironic platelets cajole furiously among the furiously regular asymptotes, furiously express
asymptotes wake careful]
ps_partkey: [74375], ps_suppkey: [6883], ps_availability: [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: [28468], ps_suppkey: [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 l]
ps_partkey: [27115], ps_suppkey: [9618], ps_availability: [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: [71984], ps_suppkey: [6999], ps_availability: [6816], ps_supplycost: [1.01], ps_comment: [rodolites are blithely across the special requests, quickly regular excuses are furiously against the
slyly final account]
ps_partkey: [43172], ps_suppkey: [685], ps_availability: [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: [155112], ps_suppkey: [5113], ps_availability: [7635], ps_supplycost: [1], ps_comment: [refully, bold packages, special somas cajole according to the foxes, furiously even account]
ps_partkey: [6333], ps_suppkey: [6334], ps_availability: [3711], ps_supplycost: [1.01], ps_comment: [s use slyly, fluffily express requests wake carefully ironic packages]
ps_partkey: [182497], ps_suppkey: [2490], 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: [61707], ps_suppkey: [1708], ps_availability: [3178], ps_supplycost: [1.02], ps_comment: [ide of the unusual, regular excuses, unusual, special packages are carefully across the even
theodolites, furil]
ps_partkey: [9897], ps_suppkey: [4098], ps_availability: [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: [122543], ps_suppkey: [5950], ps_availability: [5753], ps_supplycost: [1], ps_comment: [e the quickly ironic dependencies, slyly ironic accounts]
ps_partkey: [158093], ps_suppkey: [5639], ps_availability: [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_availability: [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: [139711], ps_suppkey: [9712], ps_availability: [426], ps_supplycost: [1.01], ps_comment: [lully unusual escapades sleep along the special instructions, final, bold ideas across the slyly ironic
ideas sleep dependent]
ps_partkey: [93653], ps_suppkey: [3654], ps_availability: [4473], ps_supplycost: [1.02], ps_comment: [ of the carefully final requests, bold deposits are slyly, instructions nod furiously instructions,
careful]

query1 returned 21 records
```

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:

```
(base) Yashs-Air:a3test yhpatt10$ ./gtest.out
[=====] Running 4 tests from 4 test suites.
[-----] Global test environment set-up.
[-----] 1 test from SelectFileTest
[ RUN      ] SelectFileTest.t1
Opening Sorted DBFile[      OK ] SelectFileTest.t1 (463 ms)
[-----] 1 test from SelectFileTest (463 ms total)

[-----] 1 test from SelectPipeTest
[ RUN      ] SelectPipeTest.t2
Opening Sorted DBFile[      OK ] SelectPipeTest.t2 (474 ms)
[-----] 1 test from SelectPipeTest (474 ms total)

[-----] 1 test from ProjectTest
[ RUN      ] ProjectTest.t3
Opening Sorted DBFile[      OK ] ProjectTest.t3 (130 ms)
[-----] 1 test from ProjectTest (130 ms total)

[-----] 1 test from SumTest
[ RUN      ] SumTest.t4
Opening Sorted DBFile[      OK ] SumTest.t4 (17 ms)
[-----] 1 test from SumTest (17 ms total)

[-----] Global test environment tear-down
[=====] 4 tests from 4 test suites ran. (1084 ms total)
[ PASSED  ] 4 tests.
(base) Yashs-Air:a3test yhpatt10$
```