COP 6726 – Database System Implementation
Project 5 Putting it all together
Project Members
Desikan Sundararajan : 56159991
Aditya Karlekar - 8888 – 9598

1) Setting configuration files:

a) Open the file location present in the default source code directory
b) Set the appropriate filepaths for the first four lines.
   i) Line 1 : Schema file path
   ii) Line 2 : DB File directory
   iii) Line 3 : TPCH File directory
   iv) Line 4 : Run length in case of a sorted file

2) Instructions to run the code:

c) make clean
d) make a5.out
e) ./a5.out
f) Follow the instructions on screen

3) Running Gtests

a) make clean
b) make mygtest,out
c) ./mygtest.out

4) Generating DB *.tbl files (not necessary as 10mb files already included)

a) mkdir ~/git;
b) cd git;
c) git clone https://github.com/electrum/tpch-dbgen.git
d) make
e) ./dbgen -s {filesize}

5) Creating the .bin files

a) make clean
b) Make a2test.out
c) Follow the instructions on the screen. Make sure the correct file path is provided in the a2-test.h file.

Link to video demonstration : https://youtu.be/jeRBXh8nzeM

Project Overview:

In this project, we brought together all the functionalities developed in previous projects and completed the database system. The completed system has the ability to create a database, run queries like selecting files, performing join operations, supports aggregate functions and boasts filtering capabilities like Group by. The user creates the table using the CREATE TABLE command. While creating, the user also needs to mention the type of file it wants to create namely HEAP or SORTED. In the case of a SORTED file, the user needs to provide the sort order too. Once the table is created, the user can input files into the database using the INSERT command. After that, the user can run the desired queries.

Some of the functions written for this project's purposes are given below:

void ExecuteCreateTable:
This function is used to create a table when the user runs the CREATE TABLE query. The function gets its input from the parser. Based on the type of file HEAP or SORTED, it invokes the corresponding DBFile instance and creates the necessary table.

int ExecuteInsertFile:
This function is used to insert the file into the previously created database when the user runs the INSERT command. It invokes the DBfile.open function and inputs the schema of the file into the database (.bin) file.

int ExecuteDropTable:
This function is used to remove the table from memory when DROP command is issued by the user. The function removes the file from memory using remove command.

int QueryPlan.cc:ExecuteNode:
This function performs the actual execution of each QueryPlan node depending on the kind of operation it is.

Program Flow :

The program execution starts through the main function. In the main function, we have defined certain variables such as createTable, dropTable, insertFile etc as extern variables. These variables have been defined in our parser and get loaded up everytime the user inputs a query. Based on the query input by the user, we create a queryplan

instance and call the approporiate function such as CreateTable, LoadFile, ExecuteQuery etc.

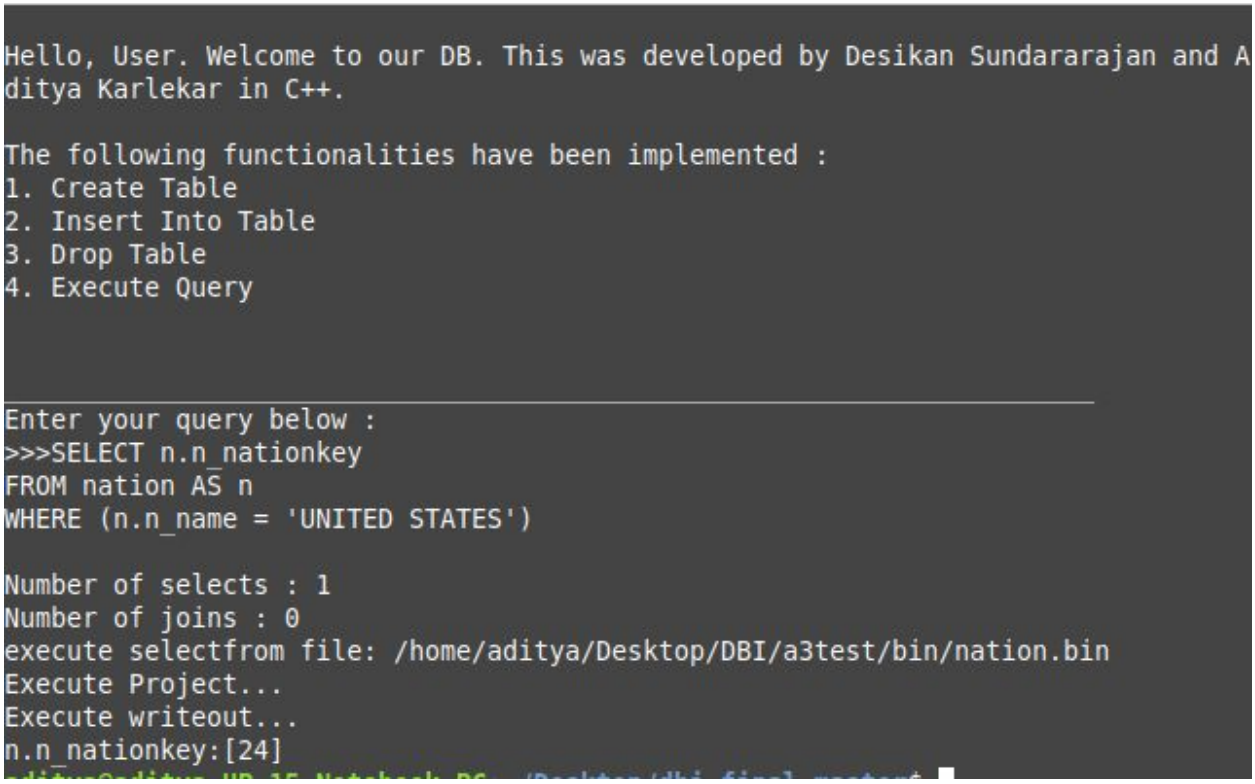
We defined a QueryPlan object instance as a tree of QueryPlan nodes. A QueryPlan node is essentially a particular operation type such as SUM, GROUP BY, SELECT PIPE etc. Whenever we are given a query, we first determine the optimized query plan, the details of which are defined in Optimizer.cc. After this we perform the actual execution of this query plan obtained in the function ExecuteNode(). We do a traversal of this tree object in a manner that is similar to the query plan execution (which we formulated in the previous project). Based on the operation type of the node, we run the particular operation's Run() function defined in RelOp.cc (Project 3). We feed input and output pipes and the resultant output pipe is in a schema that is in accordance with the optimized queryplan. As the queryplan moves from one operation to the next, the output schema of one operation becomes the input schema for the next operation in the QueryPlan node traversal. The final result is then printed out onto the screen.

We also have implemented the functionality of setting the output parameters in the file output_path. In order to set the output, the user needs to run the command : ***SET OUTPUT NONE/STDOUT/FILEPATH.*** The NONE option just prints out the query plan without performing the execution of the query, the STDOUT prints the output to the screen and specifying a filepath writes the output to the file present in the location of *filepath.* These variables are set in a file called output_path.

Output Screenshots

We ran the queries for both 1GB data 10MB data. Given below are screenshots of the results we obtained for 1GB data. For 10MB data, the bin and header files have been included in the project submission for review.

Query 1



Query 2

```
Enter your query below :
>>>SELECT n.n_name
FROM nation AS n, region AS r
WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_nationkey > 5)

Number of selects : 1
Number of joins : 1
execute selectfrom file: /home/aditya/Desktop/DBI/a3test/bin/nation.bin
execute selectfrom file: /home/aditya/Desktop/DBI/a3test/bin/region.bin
Execute Join...
Execute Project...
Execute writeout...
n.n_name:[MOZAMBIQUE]
n.n_name:[MOROCCO]
n.n_name:[KENYA]
n.n_name:[UNITED STATES]
n.n_name:[PERU]
n.n_name:[JAPAN]
n.n_name:[INDONESIA]
n.n_name:[INDIA]
n.n_name:[CHINA]
n.n_name:[VIETNAM]
n.n_name:[GERMANY]
n.n_name:[ROMANIA]
n.n_name:[RUSSIA]
n.n_name:[UNITED KINGDOM]
n.n_name:[FRANCE]
n.n_name:[IRAQ]
n.n_name:[JORDAN]
n.n_name:[IRAN]
n.n_name:[SAUDI ARABIA]
```

Query 3

```
Hello, User. Welcome to our DB. This was developed by Desikan Sundararajan and Aditya Karlekar in C++.

The following functionalities have been implemented :
1. Create Table
2. Insert Into Table
3. Drop Table
4. Execute Query


Enter your query below :
>>>SELECT SUM (n.n_nationkey)
FROM nation AS n, region AS r
WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_name = 'UNITED STATES')

Number of selects : 1
Number of joins : 1
execute selectfrom file: /home/aditya/Desktop/DBI/a3test/bin/nation.bin
execute selectfrom file: /home/aditya/Desktop/DBI/a3test/bin/region.bin
Execute Join...
Execute Sum...
Execute Project...
Execute writeout...
sum:[24.000000]
```

Query 4

```
aditya@aditya-HP-15-Notebook-PC:~/Desktop/dbi-final-master$ ./a5.out

Hello, User. Welcome to our DB. This was developed by Desikan Sundararajan and Aditya Karlekar in C++.

The following functionalities have been implemented :
1. Create Table
2. Insert Into Table
3. Drop Table
4. Execute Query



_____________________________________________________
Enter your query below :
>>>SELECT SUM (n.n_regionkey)
FROM nation AS n, region AS r
WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_name = 'UNITED STATES')
GROUP BY n.n_regionkey

Number of selects : 1
Number of joins : 1
execute selectfrom file: /home/aditya/Desktop/DBI/a3test/bin/nation.bin
execute selectfrom file: /home/aditya/Desktop/DBI/a3test/bin/region.bin
Execute Join...
Execute Group BY...
Execute Project...
Execute writeout...
sum:[1.000000]
```

Query 5

```
_____________________________________________________
Enter your query below :
>>>SELECT SUM DISTINCT (n.n_nationkey + r.r_regionkey)
FROM nation AS n, region AS r, customer AS c
WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_nationkey = c.c_nationkey) AND (n.n_nationkey > 10)
GROUP BY r.r_regionkey

Number of selects : 1
Number of joins : 2
execute selectfrom file: /home/aditya/Desktop/DBI/a3test/bin/nation.bin
execute selectfrom file: /home/aditya/Desktop/DBI/a3test/bin/region.bin
Execute Join...
execute selectfrom file: /home/aditya/Desktop/DBI/a3test/bin/customer.bin
Execute Join...
Execute Group BY...
Execute Project...
Execute writeout...
sum:[268287.000000]
sum:[257125.000000]
sum:[341936.000000]
sum:[442436.000000]
sum:[333702.000000]
aditya@aditya-HP-15-Notebook-PC:~/Desktop/dbi-final-master$
```

GTests

We have written Gtests to test the functionality of the creation, deletion and loading of tables. Given below are a few screenshots of its working :