COP 6726 – Database System Implementation

Project 5: Putting It All Together

Group Members:

Sanjay Reddy Banda, UF ID: 5878-2239

Suprith Reddy Gurudu, UF ID: 9961-2134

Video Demo Link:

https://uflorida-my.sharepoint.com/:v:/g/personal/sbanda_ufl_edu/ESAZ9S188Y9BsGJ1-1M4fToBqnPdsbwDnqDuAE6HM0yd4w?e=8CM16K

Compilation and Execution:

To compile the code, run the following command:

```
>> make
```

To execute the main.cc code, change the directory to the specific folder (a5) and run the following command:

```
>> ./a5.out

>> <Enter Query>

And press CTRL + D

Or

>> ./a5.out < testcase.sql
```

To compile the gTest (gtests.cc) code, run the following command:

```
>> make gtest.out
```

To execute the gTest (gtests.cc) code, run the following command:

```
>> ./gtest.out
```

Code Explanation (modified methods):

Filename: main.cc

Classname: QueryNode

Methods:

Virtual void Wait():

This function waits until all threads finishes its execution.

virtual void Execute(unordered_map<int, Pipe *> &pMap):

This virtual function will execute specified database function.

Classname: JoinNode <- QueryNode

Methods:

void Execute(unordered_map<int, Pipe *> &pMap):

This function executes join database method with DB File.

Classname: ProjectNode <- QueryNode

Methods:

void Execute(unordered_map<int, Pipe *> &pMap):

This function executes project database method with DB File.

Classname: SelectFileNode <- QueryNode

Methods:

void Execute(unordered_map<int, Pipe *> &pMap):

This function executes select file database method with DB File.

Classname: SelectPipeNode <- QueryNode

Methods:

void Execute(unordered_map<int, Pipe *> &pMap):

This function executes select pipe database method with DB File.

Classname: SumNode <- QueryNode

Methods:

void Execute(unordered_map<int, Pipe *> &pMap):

This function executes sum aggregate database method with DB File.

Classname: DistinctNode <- QueryNode

Methods:

void Execute(unordered_map<int, Pipe *> &pMap):

This function executes distinct database method with DB File.

Classname: GroupByNode <- QueryNode

Methods:

void Execute(unordered_map<int, Pipe *> &pMap):

This function executes group by database method with DB File.

void initSchemaMap(SchemaMap &map):

Creates the schema object for all the tables and inserts the objects into the map.

void initStatistics (Statistics &s):

Initializes the Statistics objects by adding all the relations and appropriate attributes.

void CopyTablesNamesAndAliases (TableList *tableList, Statistics &s, vector<char *> &tableNames, AliaseMap &map):

This function copies the table names and aliases.

void CopyNameList(NameList *nameList, vector<string> &names):

This function copies the name list.

Filename: gtest.cc

TEST(DATABASEENGINE, Query1) –

Google test for validating the test case for “SELECT n.n_nationkey FROM nation AS n WHERE (n.n_name = 'UNITED STATES')” scenario. It verifies by total number of records returned in this scenario i.e., for this query it returns 1 record.

TEST(DATABASEENGINE, Query2) –

Google test for validating the test case for "SELECT n.n_name FROM nation AS n, region AS r WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_nationkey > 5)" scenario. It verifies by total number of records returned in this scenario i.e., for this query it returns 19 records.

TEST(DATABASEENGINE, Query3) –

Google test for validating the test case for "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')" scenario. It verifies by total number of records returned in this scenario i.e., for this query it returns 1 record.

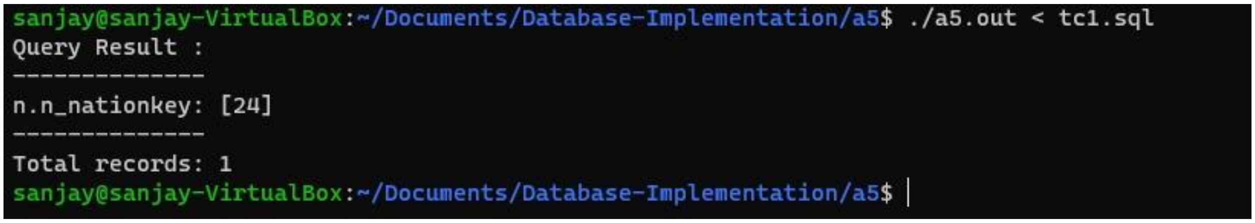
TEST(DATABASEENGINE, Query4) –
Google test for validating the test case for "SELECT SUM (n.n_regionkey) FROM nation AS n, region AS r WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_name = 'UNITED STATES') GROUP BY n.n_regionkey" scenario. It verifies by total number of records returned in this scenario i.e., for this query it returns 1 record.

TEST(DATABASEENGINE, Query5) –
Google test for validating the test case for "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" scenario. It verifies by total number of records returned in this scenario i.e., for this query it returns 5 records.

Results for the Test Cases:

Test Case 1:

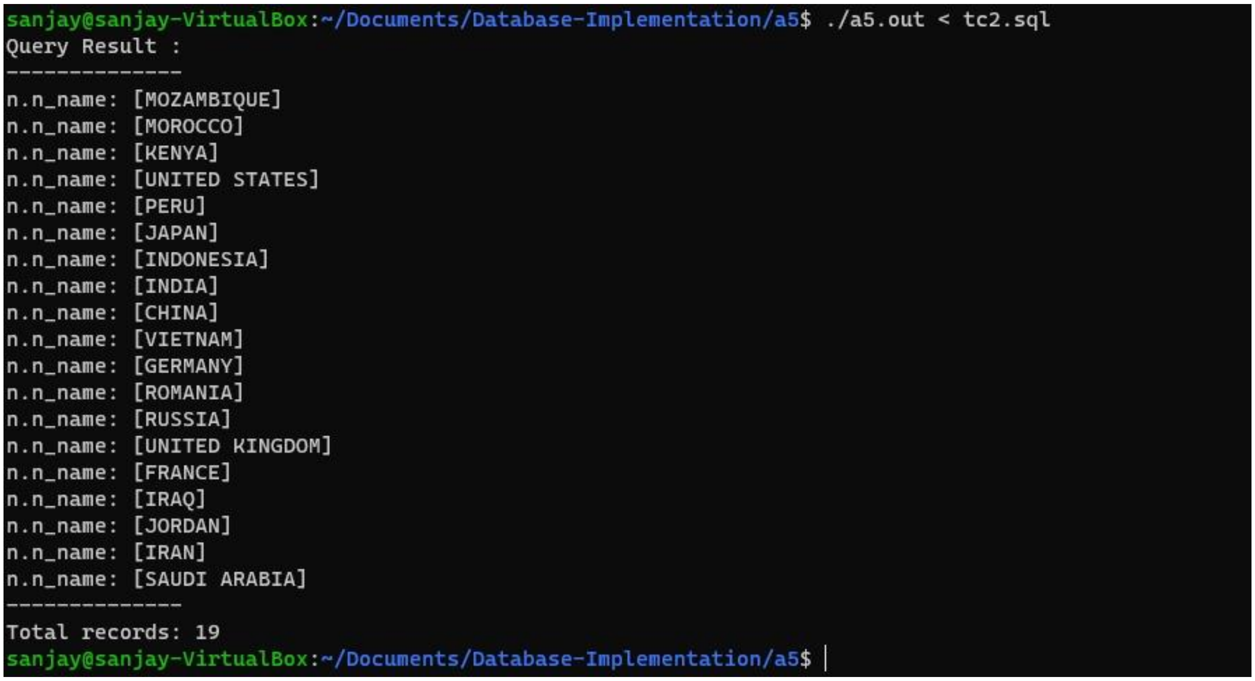
Input - "SELECT n.n_nationkey FROM nation AS n WHERE (n.n_name = 'UNITED STATES')"



```
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$ ./a5.out < tc1.sql
Query Result :
--------------
n.n_nationkey: [24]
--------------
Total records: 1
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$ |
```

Test Case 2:

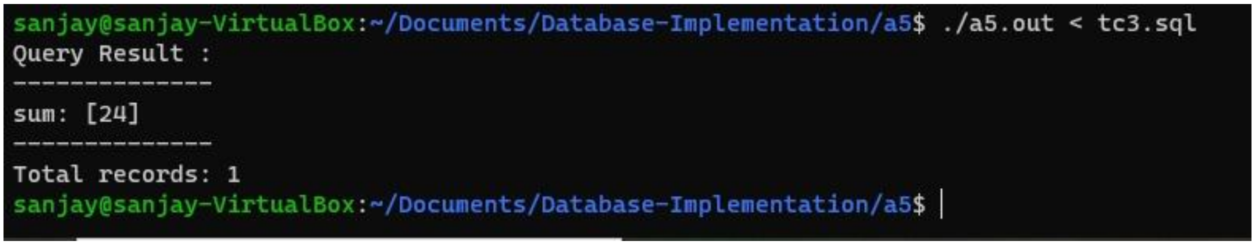
Input - "SELECT n.n_name FROM nation AS n, region AS r WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_nationkey > 5)"



```
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$ ./a5.out < tc2.sql
Query Result :
--------------
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]
--------------
Total records: 19
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$
```

Test Case 3:

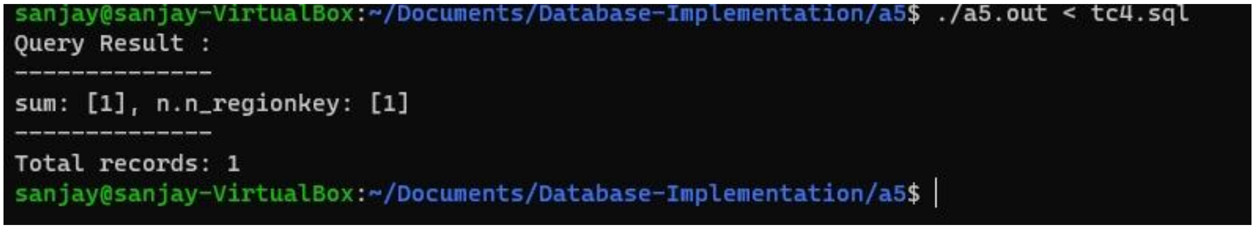
Input - "SELECT SUM (n.n_nationkey) FROM nation AS n, region AS r WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_name = 'UNITED STATES')"



```
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$ ./a5.out < tc3.sql
Query Result :
--------------
sum: [24]
--------------
Total records: 1
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$
```

Test Case 4:

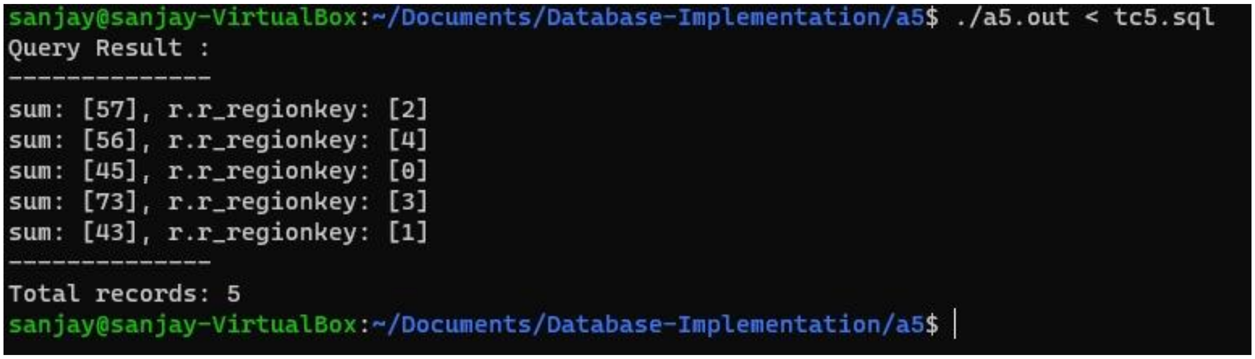
Input - "SELECT SUM (n.n_regionkey) FROM nation AS n, region AS r WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_name = 'UNITED STATES') GROUP BY n.n_regionkey"



```
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$ ./a5.out < tc4.sql
Query Result :
--------------
sum: [1], n.n_regionkey: [1]
--------------
Total records: 1
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$ |
```

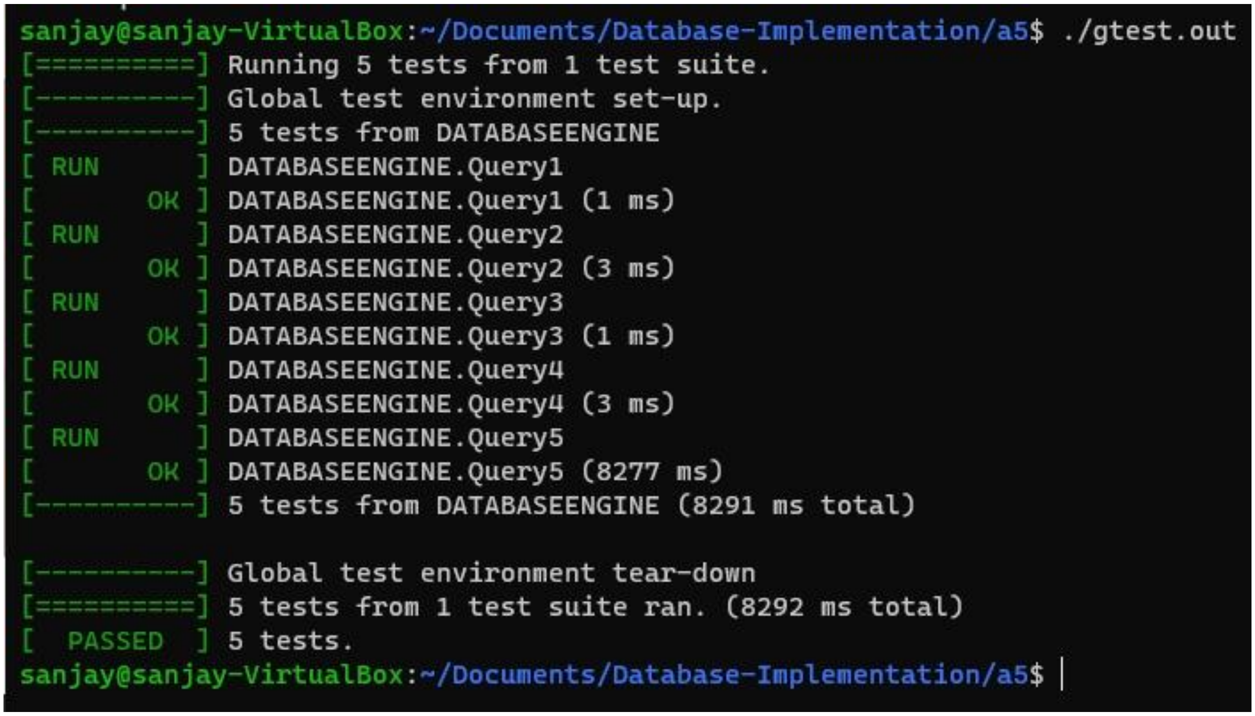
Test Case 5:

Input - "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"



```
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$ ./a5.out < tc5.sql
Query Result :
--------------
sum: [57], r.r_regionkey: [2]
sum: [56], r.r_regionkey: [4]
sum: [45], r.r_regionkey: [0]
sum: [73], r.r_regionkey: [3]
sum: [43], r.r_regionkey: [1]
--------------
Total records: 5
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$ |
```

Results for gTests:



```
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$ ./gtest.out
[==========] Running 5 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 5 tests from DATABASEENGINE
[ RUN      ] DATABASEENGINE.Query1
[       OK ] DATABASEENGINE.Query1 (1 ms)
[ RUN      ] DATABASEENGINE.Query2
[       OK ] DATABASEENGINE.Query2 (3 ms)
[ RUN      ] DATABASEENGINE.Query3
[       OK ] DATABASEENGINE.Query3 (1 ms)
[ RUN      ] DATABASEENGINE.Query4
[       OK ] DATABASEENGINE.Query4 (3 ms)
[ RUN      ] DATABASEENGINE.Query5
[       OK ] DATABASEENGINE.Query5 (8277 ms)
[----------] 5 tests from DATABASEENGINE (8291 ms total)

[----------] Global test environment tear-down
[==========] 5 tests from 1 test suite ran. (8292 ms total)
[  PASSED  ] 5 tests.
sanjay@sanjay-VirtualBox:~/Documents/Database-Implementation/a5$
```