Project Report

Group members:
Name: Rohit Karumuri
UFID: 90971158

How to run project:
make clean//to remove compiled files
make //to compile all files
./a4-1.out//to run the compiled binary for project 4.1
./a42.out//to run the compiled binary for project 4.2
runTestCases42.sh // to run all sql queries
make gtest //to compile google test cases
./gtest < tc6.sql to run test cases

QueryNode Class:

- This node class uses each part of the query string as single component which is used for parsing.
- Based on the operation, it has various child class node.
    - LeafNode – Node which is used to consist object names like table name, table attribute names etc.
    - UnaryNode – Node which consists of single table operation.
        - ProjectNode – For project based operation
        - RemdupliNode – For remove duplicate operation
        - SumNode – For sum operation
        - GroupByNode – For group by operation
        - WriteNode – For printing output operation
    - BinaryNode – Node which consists of two tables operation.
        - JoinNode – For join operation
- Query node has general functions which include constructor, print output methods. Also each child class has its own specialized member methods.

QueryPlanner Class:

- It consists of parsing, optimizing and compiling all nodes.
- It provides log of operations to output file.
- The planner object starts from root node. Then each component of query is converted to node and is linked to root node.

QueryPlanner Class Methods:

- Constructor, Destructor
- createLeafs – makes new leafnodes from query
- createJoins – makes joinnodes if join operations exist
- createSums – makes sumnodes, or remove duplicate nodes or group nodes based on the given query.
- createProjects – makes project nodes

- createDistinct – makes remove duplicate nodes
- createWrite – makes write output nodes
- orderJoins – orders elements of join operation
- evaluateOrder – gathers elements from two pipes of join and analyse the operation
- concateLists – connects two AndLists

Output

The output of all sql queries is stored in output42.txt.

The following screenshot of outputs of each sql query.

```
rohitk18@DESKTOP-JOQINBM:/mnt/d/Programming/C++/CppProjects/projects/db-sys-impl/project4b/p41$ ./a42.out < tc1.sql
-> WriteOut:
   * Output to 0x7fc978c1c760
   * Output pipe: 2
   * Input pipe: 1
   -> Project: 0
      * Output pipe: 1
      * Input pipe: 0
      -> Select from nation:
         * Output pipe: 0
```

```
rohitk18@DESKTOP-JOQINBM:/mnt/d/Programming/C++/CppProjects/projects/db-sys-impl/project4b/p41$ ./a42.out < tc2.sql
-> WriteOut:
   * Output to 0x7fdf1421c760
   * Output pipe: 4
   * Input pipe: 3
   -> Project: 0
      * Output pipe: 3
      * Input pipe: 2
      -> Join: ( Att 2 from left record = Att 4 from left record (Int))
         * Estimate = 8, Cost = 8
         * Output pipe: 2
         * Input pipe: 1, 0
         -> Select from nation:
            * Output pipe: 1
         -> Select from region:
            * Output pipe: 0
```

```
rohitk18@DESKTOP-JOQINBM:/mnt/d/Programming/C++/CppProjects/projects/db-sys-impl/project4b/p41$ ./a42.out < tc3.sql
-> WriteOut:
   * Output to 0x7f4b4d41c760
   * Output pipe: 4
   * Input pipe: 3
   -> Sum:       * Function:       * Output pipe: 3
      * Input pipe: 2
      -> Join: ( Att 2 from left record = Att 4 from left record (Int))
         * Estimate = 1, Cost = 1
         * Output pipe: 2
         * Input pipe: 1, 0
         -> Select from nation:
            * Output pipe: 1
         -> Select from region:
            * Output pipe: 0
```

```
rohitk18@DESKTOP-JOQINBM:/mnt/d/Programming/C++/CppProjects/projects/db-sys-impl/project4b/p41$ ./a42.out < tc4.sql
-> WriteOut:
   * Output to 0x7f5c3aa1c760
   * Output pipe: 4
   * Input pipe: 3
   -> Group by:
      * OrderMaker: NumAtts =     1
  0: -331611744 Int
      * Function:
         2 PushInt
      * Output pipe: 3
      * Input pipe: 2
      -> Join: ( Att 2 from left record = Att 4 from left record (Int))
         * Estimate = 1, Cost = 1
         * Output pipe: 2
         * Input pipe: 1, 0
         -> Select from nation:
            * Output pipe: 1
         -> Select from region:
            * Output pipe: 0
```

```
rohitk18@DESKTOP-JOQINBM:/mnt/d/Programming/C++/CppProjects/projects/db-sys-impl/project4b/p41$ ./a42.out < tc5.sql
-> WriteOut:
   * Output to 0x7fef35a1c760
   * Output pipe: 7
   * Input pipe: 6
   -> Group by:
      * OrderMaker: NumAtts =     1
  0: -1138602320 Int
      * Function:
         0 PushInt
         4 PushInt
         0 IntPlus
      * Output pipe: 6
      * Input pipe: 5
      -> RemoveDuplication:
         * Output pipe: 5
         * Input pipe: 4
         -> Join: ( Att 0 from left record = Att 10 from left record (Int))
            * Estimate = 48000, Cost = 48008
            * Output pipe: 4
            * Input pipe: 3, 0
            -> Join: ( Att 2 from left record = Att 4 from left record (Int))
               * Estimate = 8, Cost = 8
               * Output pipe: 3
               * Input pipe: 2, 1
               -> Select from nation:
                  * Output pipe: 2
               -> Select from region:
                  * Output pipe: 1
            -> Select from customer:
               * Output pipe: 0
```

Tests:

The test was based on tc6.sql file. Run command is ./gtest < tc6.sql

- Tables – Verifies the tables collected are correct
- GroupBy – Verifies the inputted group by clause is correct
- WhereClause – Verifies all andlists and checks all the left and right values are correct