Assignment 5
Putting it All Together

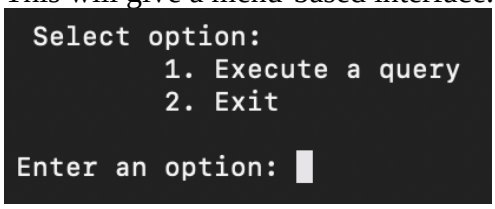
Group Members:

1. Dhiraj Mahesh Paryani, UFID: 1692 1261
2. Yogesh Laxman, UFID: 9451 2517

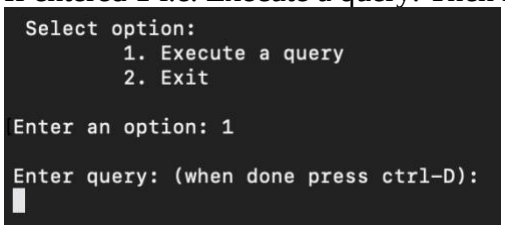
Instructions for compiling and running the code:

1. **Extracting folders:**
    a. Extract the contents of the folder “DhirajMaheshParyani_YogeshLaxman_p5.zip”
    b. Open the terminal and navigate to the “Project” folder which is inside the extracted folder.

2. **Running the tests:**
    a. Run “make a5.out”. This will compile the program files.
    b. Run “./a5.out” to run the program.
    c. This will give a menu-based interface.



    d. If entered 1 i.e. Execute a query. Then it will ask the user to enter a query.



    e. After the query is entered and ‘crtl-D’ is pressed it will execute the query.
    f. Enter 2 i.e Exit at any time to stop the database session.

Brief explanation of implementation:

This assignment puts together all of the pieces from previous assignments to have a working database system. Our database system supports CREATE, INSERT, DROP, SELECT, and SET query types. In this assignment, we have created 2 new classes i.e. Database and QueryRunner. Database will be the main entry point to do any database operations. QueryxRunner will be responsible for running query from the query plan.

Syntax of all query types is as follows:

1. **CREATE**
   a. CREATE TABLE mytable (att1 INTEGER, att2 DOUBLE, att3 STRING) AS HEAP;
   b. CREATE TABLE MYTABLE (att1 INTEGER, att2 DOUBLE, att3 STRING) AS SORTED ON att1, att2;

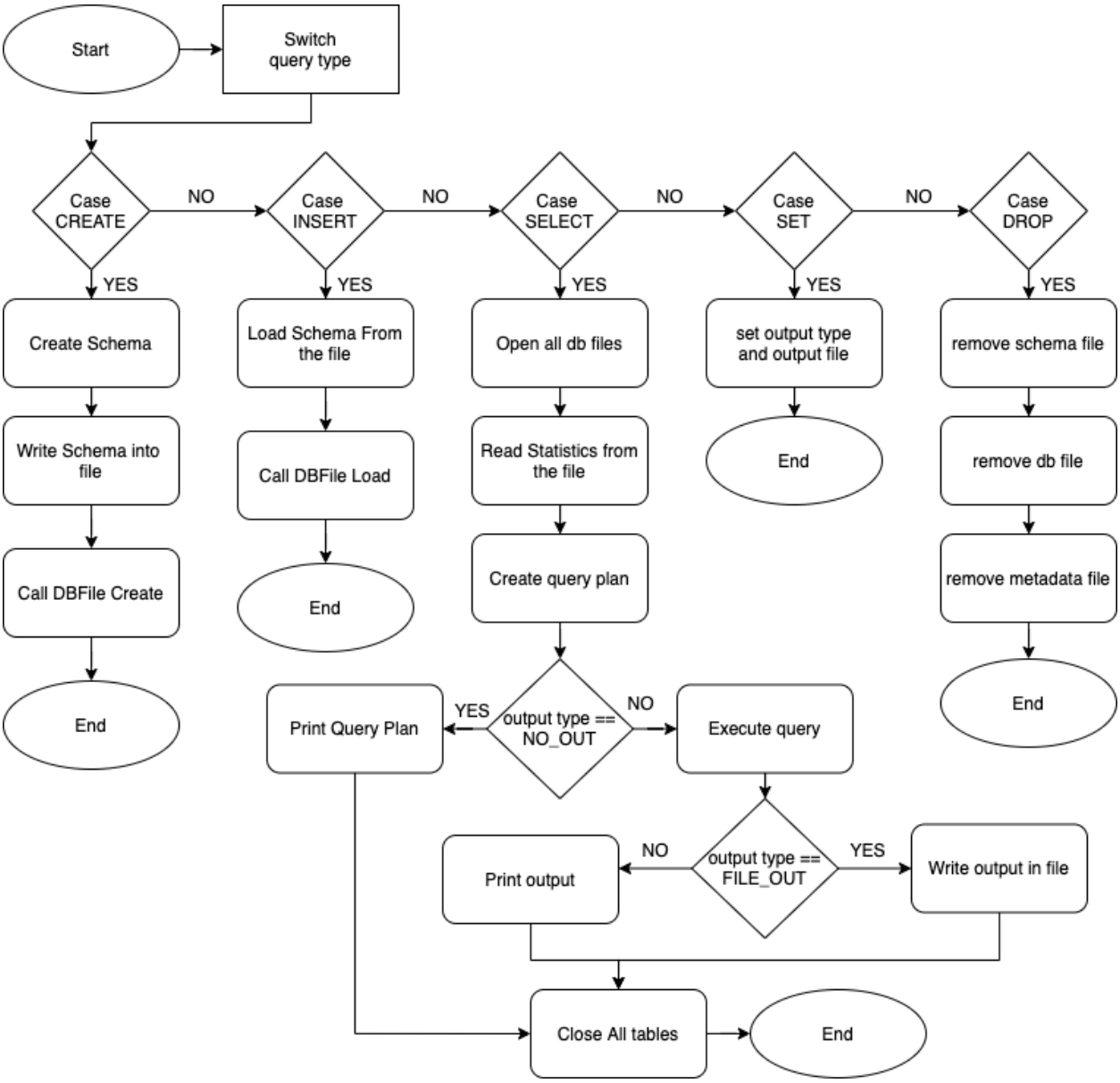
2. **INSERT**: INSERT ‘myfile’ INTO mytable

3. **DROP**: DROP TABLE mytable;

4. **SET**:
   a. SET OUTPUT STDOUT;
   b. SET OUTPUT ‘myfile’;
   c. SET OUTPUT NONE;

5. **SELECT**…
   SELECT can contain projection, join, group by, selection, distinct, sum aggregate function, sum aggregate function with distinct.

Following flow chart gives the information about query execution.



Results (1GB):

Query - 1:

```
Enter query: (when done press ctrl-D):
SELECT n.n_nationkey
FROM nation AS n
WHERE (n.n_name = 'UNITED STATES')

n.n_nationkey: [24]

Query ran successfully.
```

Query - 2:

```
Enter query: (when done press ctrl-D):
SELECT n.n_name
FROM nation AS n, region AS r
WHERE (n.n_regionkey = r.r_regionkey) AND (n.n_nationkey > 5)

n.n_name: [KENYA]
n.n_name: [MOROCCO]
n.n_name: [MOZAMBIQUE]
n.n_name: [PERU]
n.n_name: [UNITED STATES]
n.n_name: [JAPAN]
n.n_name: [CHINA]
n.n_name: [INDIA]
n.n_name: [INDONESIA]
n.n_name: [VIETNAM]
n.n_name: [FRANCE]
n.n_name: [ROMANIA]
n.n_name: [RUSSIA]
n.n_name: [GERMANY]
n.n_name: [UNITED KINGDOM]
n.n_name: [JORDAN]
n.n_name: [IRAQ]
n.n_name: [IRAN]
n.n_name: [SAUDI ARABIA]

Query ran successfully.
```

Query - 3:

```
Enter query: (when done press ctrl-D):
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')

SUM: [24]

Query ran successfully.
```

Query - 4:

```
Enter query: (when done press ctrl-D):
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

SUM: [1]

Query ran successfully.
```

Query - 5:

```
Enter query: (when done press ctrl-D):
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

SUM: [45]
SUM: [43]
SUM: [57]
SUM: [73]
SUM: [56]

Query ran successfully.
```

GTests:

Instructions to run:

1. Optional: Update catalog_dir, dbfile_dir paths in test.cat file.
2. Run “make gTestDatabase.out”.
3. Run “./gTestDatabase.out”.

Details:

Following are the details of 2 written GTests:

1. SETQuery: This test checks the SET query of the database. This executes the SET query and then accesses the database object to see that changes are correctly reflected or not.

2. CREATEQuery: This test verifies CREATE query of the database. This executes the CREATE query and then checks whether the binary file, metadata file and schema file is present or not. After that It executes the DROP query. Then again checks previous files should be absent.

Output:

```
[==========] Running 2 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 2 tests from Database
[ RUN      ] Database.SETQuery

Output mode changed successfully.
[       OK ] Database.SETQuery (0 ms)
[ RUN      ] Database.CreateQuery

Table created successfully.

Table dropped successfully.
[       OK ] Database.CreateQuery (3 ms)
[----------] 2 tests from Database (3 ms total)

[----------] Global test environment tear-down
[==========] 2 tests from 1 test case ran. (3 ms total)
[  PASSED  ] 2 tests.
dhirajmaheshparyani@dhiraj Project %
```