CSCE-608 Database Systems

Project #2: TinySQL Interpreter

This file contains screenshots for better understanding of the way our TinySQL Interpreter works.

1. Compiling the source code
    a. Open terminal (/command prompt) and go to "TinySQL_Interpreter" Project location.
    b. Run the command

    javac -target 1.8 -source 1.8 -d ./out/ src/parser/*.java src/storageManager/*.java src/interpreter/*.java -Xlint:unchecked

MINGW64:/d/workspace/TinySQL_Interpreter

```
LightSaber@Katana-PC MINGW64 /
$ cd /d/workspace/TinySQL_Interpreter/

LightSaber@Katana-PC MINGW64 /d/workspace/TinySQL_Interpreter (master)
$ javac -target 1.8 -source 1.8 -d ./out/ src/parser/*.java src/storageManager/*.java src/interpreter/*.java -Xlint:unchecked

LightSaber@Katana-PC MINGW64 /d/workspace/TinySQL_Interpreter (master)
$
```

2. Running the program
    a. Go to the “/out” directory. Run command: cd out
    b. Run the command

    java -cp . interpreter/Main

MINGW64:/d/workspace/TinySQL_Interpreter/out

```
LightSaber@Katana-PC MINGW64 /
$ cd /d/workspace/TinySQL_Interpreter/

LightSaber@Katana-PC MINGW64 /d/workspace/TinySQL_Interpreter (master)
$ javac -target 1.8 -source 1.8 -d ./out/ src/parser/*.java src/storageManager/*.java src/interpreter/*.java -Xlint:unchecked

LightSaber@Katana-PC MINGW64 /d/workspace/TinySQL_Interpreter (master)
$ cd out/

LightSaber@Katana-PC MINGW64 /d/workspace/TinySQL_Interpreter/out (master)
$ java -cp . interpreter/Main
--------------------------------------------------------------------------------
----------------------- Welcome to TinySQL Interpreter -----------------------
--------------------------------------------------------------------------------


 Input Options:
 1. Single Query
 2. File upload
 Press 0 to exit

 Choose (1/2)?
```

3. Running a single query

MINGW64:/d/workspace/TinySQL_Interpreter/out

```
LightSaber@Katana-PC MINGW64 /d/workspace/TinySQL_Interpreter/out (master)
$ java -cp . interpreter/Main
--------------------------------------------------------------------------------
----------------------- Welcome to TinySQL Interpreter -------------------------
--------------------------------------------------------------------------------


 Input Options:
 1. Single Query
 2. File upload
 Press 0 to exit

 Choose (1/2)?
1
 Enter a TinySQL query:
create table student (name STR20, rollNumber INT)
 Processing input ...

 Output is logged in file D:\workspace\TinySQL_Interpreter\out\Result.txt

 Input Options:
 1. Single Query
 2. File upload
 Press 0 to exit

 Choose (1/2)?
```

4. Uploading a query file

```
 Input Options:
 1. Single Query
 2. File upload
 Press 0 to exit

 Choose (1/2)?
2
 Enter full path to the input file:
D:\workspace\TinySQL_Interpreter\test\TinySQL-TextWin.txt
 Processing input ...

 Output is logged in file D:\workspace\TinySQL_Interpreter\out\Result.txt

 Input Options:
 1. Single Query
 2. File upload
 Press 0 to exit

 Choose (1/2)?
```

5. Output file path: $ProjectDir + “/out/Result.txt”

```
LightSaber@Katana-PC MINGW64 /d/workspace/TinySQL_Interpreter/out (master)
$ ls
interpreter/  parser/  production/  Result.txt  storageManager/  test/

LightSaber@Katana-PC MINGW64 /d/workspace/TinySQL_Interpreter/out (master)
$
```

6. Exiting the interface

```
 Input Options:
 1. Single Query
 2. File upload
 Press 0 to exit

 Choose (1/2)?
0
 Thanks for using TinySQL interpreter

LightSaber@Katana-PC MINGW64 /d/workspace/TinySQL_Interpreter/out (master)
$
```

7. Output of some sample queries:

- CREATE TABLE course (sid INT, homework INT, project INT, exam INT, grade STR20)

```
Create Statement :
tableName = 'course'
attributes = {sid=INT, homework=INT, project=INT, exam=INT, grade=STR20}

Successfully created relation course

System elapse time = 62 ms
Calculated Disk elapse time = 0.0 ms
Calculated Disk I/Os = 0
```

- INSERT INTO course (sid, homework, project, exam, grade) VALUES (1, 99, 100, 100, "A")

```
Insert Statement :
tableName = 'course'
attributes = [sid, homework, project, exam, grade]
values = [1, 99, 100, 100, A]
selectStatement = null

Tuple(s) Inserted Successfully
******RELATION DUMP BEGIN******
sid    homework    project    exam    grade
0: 1    99 100    100    A
******RELATION DUMP END******

System elapse time = 78 ms
Calculated Disk elapse time = 74.63 ms
Calculated Disk I/Os = 1
```

- SELECT * FROM course

```
Select Statement :
hasDistinct = false
columns = [*]
tables = [course]
condition = null
orderColumn = 'null'

Selected Tuple Count : 1
Selected Tuples :
Columns [sid, homework, project, exam, grade]
1  99 100    100    A

System elapse time = 77 ms
Calculated Disk elapse time = 74.63 ms
Calculated Disk I/Os = 1
```

- INSERT INTO course (sid, homework, project, exam, grade) SELECT * FROM course

```
Insert Statement :
tableName = 'course'
attributes = [sid, homework, project, exam, grade]
values = null
selectStatement = Select Statement :
hasDistinct = false
columns = [*]
tables = [course]
condition = null
orderColumn = 'null'

Selected Tuple Count : 3
Selected Tuples :
Columns [sid, homework, project, exam, grade]
1  99 100    100    A
2  -2147483648    100    100    E
3  100    100    100    E

Tuple(s) Inserted Successfully
******RELATION DUMP BEGIN******
sid    homework   project    exam   grade
0: 1   99 100    100    A
1: 2   -2147483648    100    100    E
2: 3   100    100    100    E
3: 1   99 100    100    A
4: 2   -2147483648    100    100    E
5: 3   100    100    100    E
******RELATION DUMP END******

System elapse time = 482 ms
Calculated Disk elapse time = 479.89 ms
Calculated Disk I/Os = 7
```

- DELETE FROM course WHERE grade = "E"

```
Delete Statement :
tableName = 'course'
condition = [grade, =, E]

Relation course after Tuple Deletion
 ******RELATION DUMP BEGIN******
sid    homework   project    exam   grade
0: 1   99 100    100    A
1: 1   99 100    100    A
******RELATION DUMP END******

System elapse time = 535 ms
Calculated Disk elapse time = 533.26 ms
Calculated Disk I/Os = 8
```

- DROP TABLE course

```
Drop Statement :
tableName = 'course'

Successfully removed all tuples and Dropped Relation course

System elapse time = 0 ms
Calculated Disk elapse time = 0.0 ms
Calculated Disk I/Os = 0
```

- SELECT DISTINCT course.grade, course2.grade FROM course, course2 WHERE course.sid = course2.sid AND [course.exam > course2.exam OR course.grade = "A" AND course2.grade = "A"] ORDER BY course.exam

```
Select Statement :
hasDistinct = true
columns = [course.grade, course2.grade]
tables = [course, course2]
condition = [course.sid, =, course2.sid, AND , [, course.exam, >, course2.exam, OR ,
course.grade, =, A, AND , course2.grade, =, A, ]]
orderColumn = 'course.exam'

Selected Tuple Count : 1
Selected Tuples :
Columns [course.grade, course2.grade]
A  A

System elapse time = 12260 ms
Calculated Disk elapse time = 12116.599999999995 ms
Calculated Disk I/Os = 186
```

- SELECT * FROM r, s, t WHERE r.a=t.a AND r.b=s.b AND s.c=t.c

```
Select Statement :
hasDistinct = false
columns = [*]
tables = [r, s, t]
condition = [r.a, =, t.a, AND , r.b, =, s.b, AND , s.c, =, t.c]
orderColumn = 'null'

Selected Tuple Count : 46
Selected Tuples :
Columns [r.a, r.b, s.b, s.c, t.a, t.c]
0  0  0  0  0  0
0  0  0  0  0  0
0  0  0  0  0  0
0  0  0  0  0  0
0  0  0  0  0  0
0  0  0  0  0  0
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
2  2  2  2  2  2
2  2  2  2  2  2
2  2  2  2  2  2
2  2  2  2  2  2
2  2  2  2  2  2
2  2  2  2  2  2
3  3  3  3  3  3
3  3  3  3  3  3
3  3  3  3  3  3
3  3  3  3  3  3
0  0  0  0  0  0
0  0  0  0  0  0
0  0  0  0  0  0
0  0  0  0  0  0
0  0  0  0  0  0
0  0  0  0  0  0
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
2  2  2  2  2  2
2  2  2  2  2  2
2  2  2  2  2  2
2  2  2  2  2  2
2  2  2  2  2  2
2  2  2  2  2  2

System elapse time = 3063 ms
Calculated Disk elapse time = 2175.78 ms
Calculated Disk I/Os = 33
```