DBMS LAB 9

NAME: SACHIN HANSDA
ROLL NO.: 111601019

LAB REPORT

3. Compile entire server side code, and start the server with a database studentdb . Go to ~/SimpleDB_2.10, and give following command
a. javac simpledb/server/Startup.java
b. java simpledb.server.Startup studentdb

```
dbms@dbms-VirtualBox:~/SimpleDB_2.10/SimpleDB_2.10$ javac simpledb/server/Startup
.java
dbms@dbms-VirtualBox:~/SimpleDB_2.10/SimpleDB_2.10$ java simpledb.server.Startup
studentdb
new transaction: 1
creating new database
transaction 1 committed
database server ready
```

4. There is a helper file CreateStudentDB.java in studentClient/simpledb to create the database studentdb. Compile the code, and execute to create the database. Go to ~/SimpleDB_2.10, and give following command
a. javac studentClient/simpledb/CreateStudentDB.java
b. java -cp .:studentClient/simpledb CreateStudentDB

```
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ javac studentClient/simpledb/CreateStudentDB.java
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ java -cp .:studentClient/simpledb CreateStudentDB
Table STUDENT created.
STUDENT records inserted.
Table DEPT created.
DEPT records inserted.
Table COURSE created.
COURSE records inserted.
Table SECTION created.
SECTION records inserted.
Table ENROLL created.
ENROLL records inserted.
```

5. There is a helper that is giving an example query. Compile and execute to understand.
a. javac studentClient/simpledb/StudentMajor.java
b. java -cp .:studentClient/simpledb:simpledb/remote/ StudentMajor

```
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ javac studentClient/simpledb/StudentMajor.java
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ java -cp .:studentClient/simpledb:simpledb/remote
/ StudentMajor
Name    Major
joe     compsci
max     compsci
lee     compsci
amy     math
sue     math
kim     math
pat     math
bob     drama
art     drama
```

6. There is another helper that is giving an example query. Compile and execute to understand.

a. javac studentClient/simpledb/FindMajors.java

b. java -cp .:studentClient/simpledb:simpledb/remote/ FindMajors math

```
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ javac studentClient/simpledb/FindMajors.java
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ java -cp .:studentClient/simpledb:simpledb/remote
/ FindMajors math
Here are the math majors
Name    GradYear
amy     2004
sue     2005
kim     2001
pat     2001
```

7. Now, we will start the SQL client.

a. javac studentClient/simpledb/SQLInterpreter.java

b. java -cp .:studentClient/simpledb:simpledb/remote/ SQLInterpreter

```
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ javac studentClient/simpledb/SQLInterpreter.java
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ java -cp .:studentClient/simpledb:simpledb/remote
/ SQLInterpreter

SQL>
```

8. Give a simple query and test. Look into the CreateStudentDB.java to know about the tables and fields.

a. select SName from student where gradyear=2004

```
SQL> select SName from student where gradyear=2004

     sname
----------
       joe
       amy
       art
       lee
```

9. Change the prompt from SQL> to IITPKD>

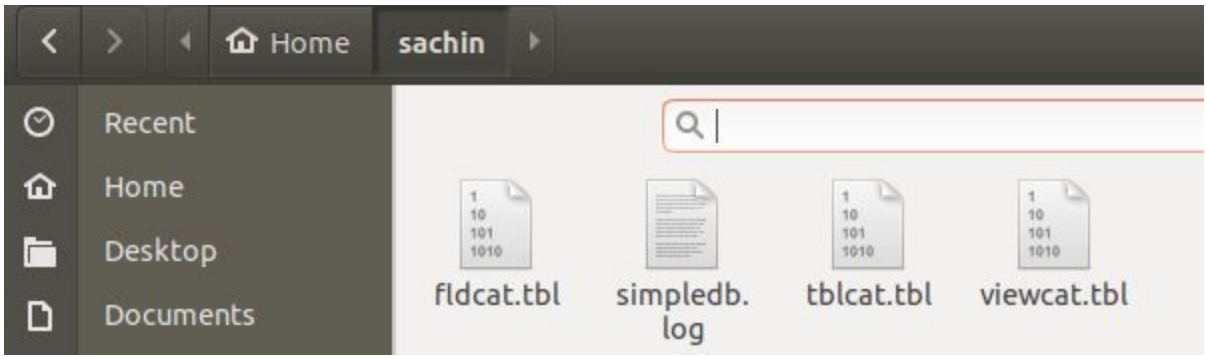
```
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ javac studentClient/simpledb/SQLInterpreter.java
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ java -cp .:studentClient/simpledb:simpledb/remote
/ SQLInterpreter

IITPKD>
```

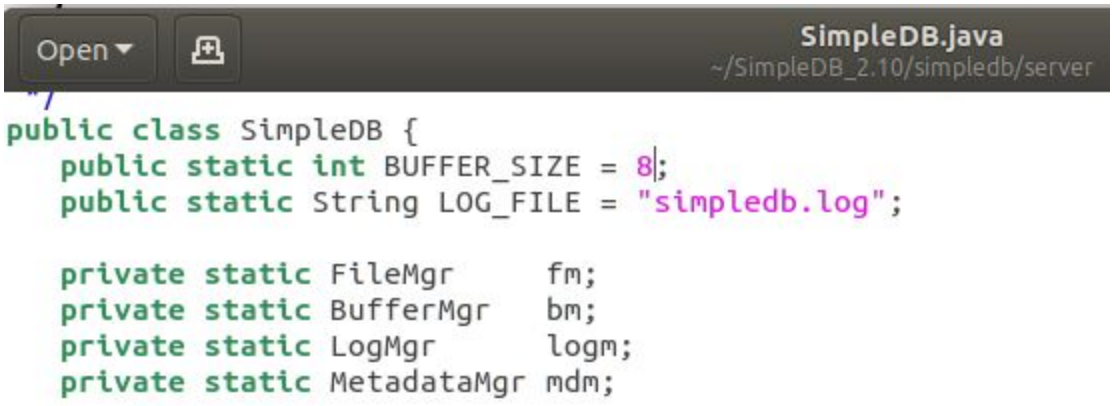
10. Find the complete path how a select query is getting executed.
Ans: Program starts from main()
It connects to jdbc:simpledb://localhost
On select query main() calls doQuery()
It enters doQuery
There, it gets a resultset using executeQuery()
Then it prints results from resultset till the the pointer in resultset reaches the end

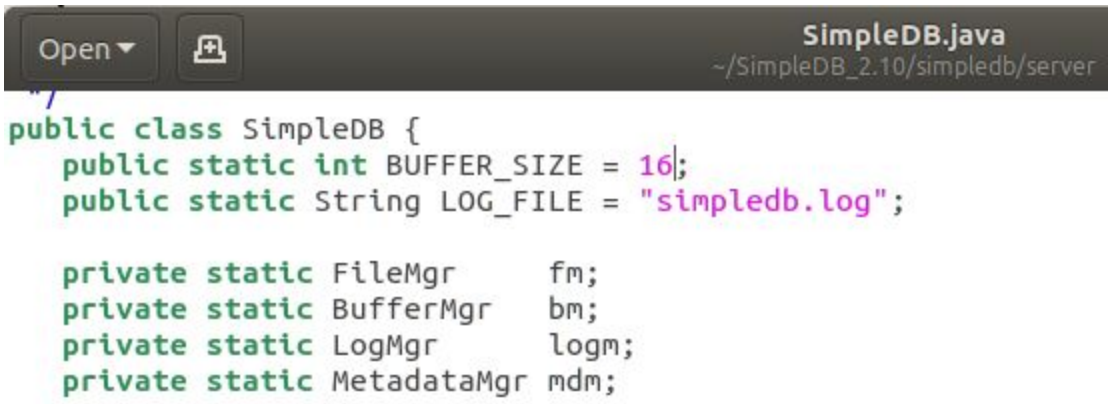
11. Change the location where tables are getting stored.

```
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ javac simpledb/server/Startup.java
dbms@dbms-VirtualBox:~/SimpleDB_2.10$ java simpledb.server.Startup sachin
new transaction: 1
creating new database
transaction 1 committed
database server ready
```



12. Increase the buffer size.





14. Allow select * from SQL cleint

```
public QueryData query() {
   lex.eatKeyword("select");
   Collection<String> fields = selectList();
   lex.eatKeyword("from");
   Collection<String> tables = tableList();
   Predicate pred = new Predicate();
   if (lex.matchKeyword("where")) {
      lex.eatKeyword("where");
      pred = predicate();
   }
   return new QueryData(fields, tables, pred);
}

private Collection<String> selectList() {
   Collection<String> L = new ArrayList<String>();
   L.add(field());
   if (lex.matchDelim(',')) {
      lex.eatDelim(',');
      L.addAll(selectList());
   }
```

```
public QueryData query() {
   lex.eatKeyword("select");
   Collection<String> fields = selectListHelper();
   lex.eatKeyword("from");
   Collection<String> tables = tableList();
   Predicate pred = new Predicate();
   if (lex.matchKeyword("where")) {
      lex.eatKeyword("where");
      pred = predicate();
   }
   return new QueryData(fields, tables, pred);
}

private Collection<String> selectListHelper() {
   if (lex.matchDelim('*')) {
      Collection<String> L = new ArrayList<String>();
      lex.eatDelim('*');
      L.add ("*");
      return L;
   } else {
      return selectList();
```