README file

The Project was developed using Eclipse IDE for Java.

Instructions for running the application:

From eclipse:

1) Extract the compressed folder downloaded from eLearning.
2) Import the folder TeamEDavisBase into the Eclipse IDE.
3) Run the DavisBase.java to start the application.

(or)

From local terminal:

4) Extract the compressed folder TeamEDavisBase
5) Open terminal or command prompt
6) Navigate to the path where the folder has been extracted.
7) Change directory to the TeamEDavisBase\src\ in terminal and compile all files using the below command:

```
javac *.java
```

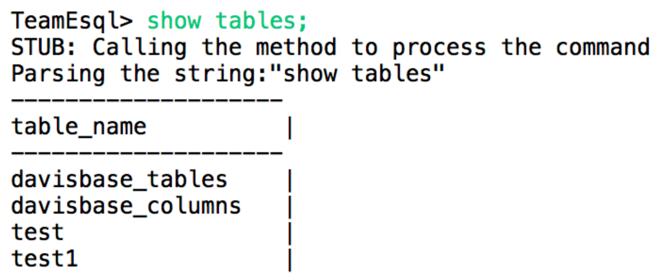
8) Change directory to parent directory (TeamEDavisBase\src)
9) Run DavisBase.class file using the below command:

```
Java DavisBase
```

The following commands are supported:

(a) Displays a list of all tables in DavisBase

Syntax: *SHOW TABLES;*



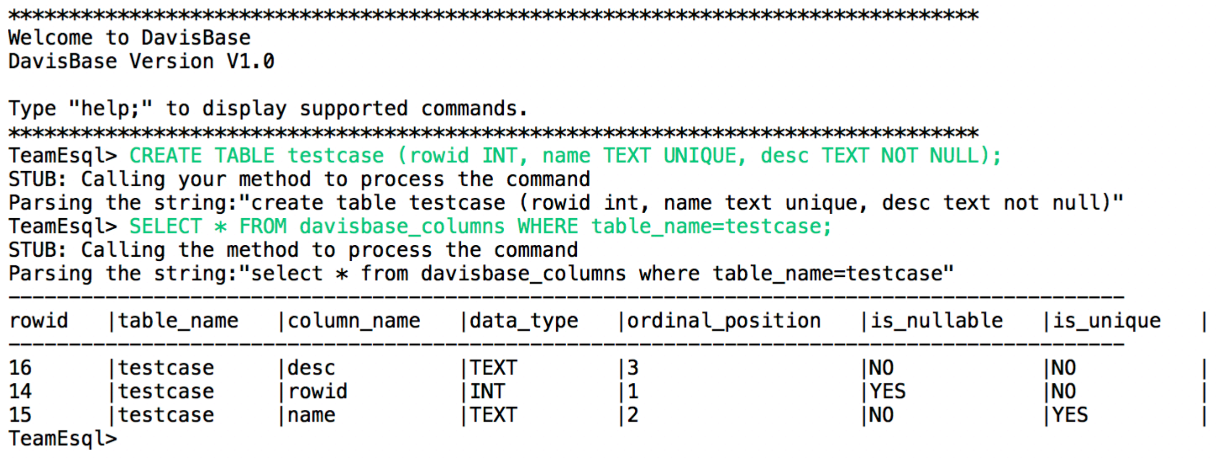
```
TeamEsql> show tables;
STUB: Calling the method to process the command
Parsing the string:"show tables"
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
test                |
test1               |
```

README file

(b) Create a new table

Syntax: CREATE TABLE <table_name> (<column name> <constraint>, ..) ;

- The first column is primary key, so it has to be of data type Integer.
- There is an inbuilt AUTO INCREMENT function implemented for the primary key which will increment maximum key when no value is passed.
- The columns followed by primary key can be defined with constraints UNIQUE/NOT NULL. The constraint will be stored in davisbase_columns against the respective parameter as shown below.

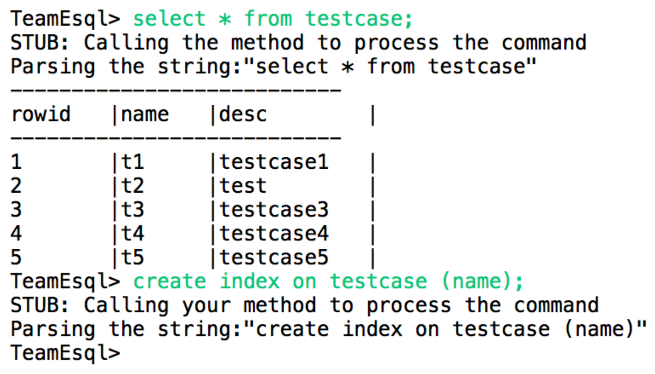


```
********************************************************************************
Welcome to DavisBase
DavisBase Version V1.0

Type "help;" to display supported commands.
********************************************************************************
TeamEsql> CREATE TABLE testcase (rowid INT, name TEXT UNIQUE, desc TEXT NOT NULL);
STUB: Calling your method to process the command
Parsing the string:"create table testcase (rowid int, name text unique, desc text not null)"
TeamEsql> SELECT * FROM davisbase_columns WHERE table_name=testcase;
STUB: Calling the method to process the command
Parsing the string:"select * from davisbase_columns where table_name=testcase"
-------------------------------------------------------------------------------------------
rowid    |table_name    |column_name    |data_type    |ordinal_position    |is_nullable    |is_unique    |
-------------------------------------------------------------------------------------------
16       |testcase      |desc           |TEXT         |3                   |NO             |NO           |
14       |testcase      |rowid          |INT          |1                   |YES            |NO           |
15       |testcase      |name           |TEXT         |2                   |NO             |YES          |
TeamEsql>
```

(c) Create Index

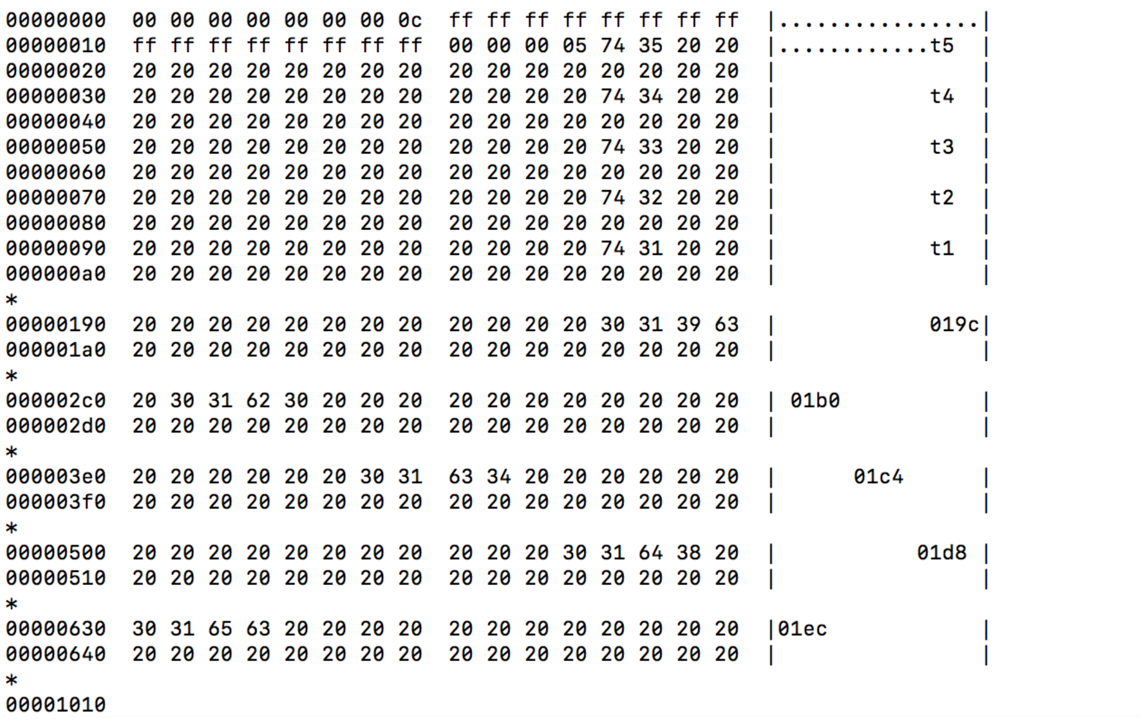
Syntax: CREATE INDEX ON <table_name> (<column name>);



```
TeamEsql> select * from testcase;
STUB: Calling the method to process the command
Parsing the string:"select * from testcase"
---------------------------
rowid    |name    |desc        |
---------------------------
1        |t1      |testcase1   |
2        |t2      |test        |
3        |t3      |testcase3   |
4        |t4      |testcase4   |
5        |t5      |testcase5   |
TeamEsql> create index on testcase (name);
STUB: Calling your method to process the command
Parsing the string:"create index on testcase (name)"
TeamEsql>
```

Below is the index file stored in the backend.

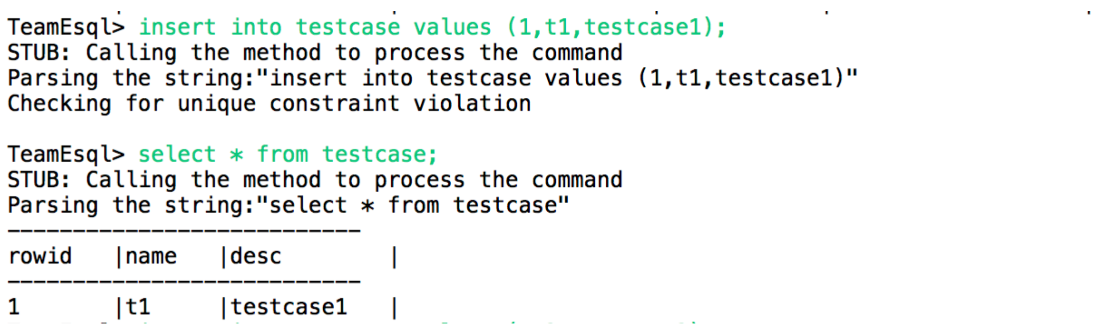
README file



```
00000000  00 00 00 00 00 00 00 0c  ff ff ff ff ff ff ff ff  |................|
00000010  ff ff ff ff ff ff ff ff  00 00 00 05 74 35 20 20  |............t5  |
00000020  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
00000030  20 20 20 20 20 20 20 20  20 20 20 20 74 34 20 20  |            t4  |
00000040  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
00000050  20 20 20 20 20 20 20 20  20 20 20 20 74 33 20 20  |            t3  |
00000060  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
00000070  20 20 20 20 20 20 20 20  20 20 20 20 74 32 20 20  |            t2  |
00000080  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
00000090  20 20 20 20 20 20 20 20  20 20 20 20 74 31 20 20  |            t1  |
000000a0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
*
00000190  20 20 20 20 20 20 20 20  20 20 20 20 30 31 39 63  |            019c|
000001a0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
*
000002c0  20 30 31 62 30 20 20 20  20 20 20 20 20 20 20 20  | 01b0           |
000002d0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
*
000003e0  20 20 20 20 20 20 30 31  63 34 20 20 20 20 20 20  |      01c4      |
000003f0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
*
00000500  20 20 20 20 20 20 20 20  20 20 20 30 31 64 38 20  |           01d8 |
00000510  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
*
00000630  30 31 65 63 20 20 20 20  20 20 20 20 20 20 20 20  |01ec            |
00000640  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  |                |
*
00001010
```

(d) Insert record into table

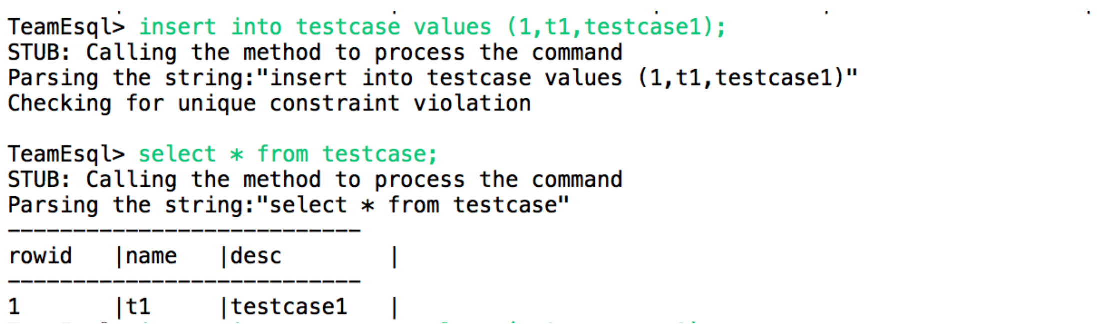
Syntax: INSERT INTO table_name <column_list> VALUES <value_list>;



```
TeamEsql> insert into testcase values (1,t1,testcase1);
STUB: Calling the method to process the command
Parsing the string:"insert into testcase values (1,t1,testcase1)"
Checking for unique constraint violation

TeamEsql> select * from testcase;
STUB: Calling the method to process the command
Parsing the string:"select * from testcase"
----------------------------
rowid   |name   |desc       |
----------------------------
1       |t1     |testcase1  |
```

With the auto increment feature on primary, the value for the column can be skipped for it to be generated dynamic.

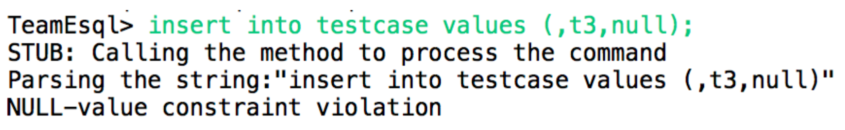


```
TeamEsql> insert into testcase values (1,t1,testcase1);
STUB: Calling the method to process the command
Parsing the string:"insert into testcase values (1,t1,testcase1)"
Checking for unique constraint violation

TeamEsql> select * from testcase;
STUB: Calling the method to process the command
Parsing the string:"select * from testcase"
----------------------------
rowid   |name   |desc       |
----------------------------
1       |t1     |testcase1  |
```

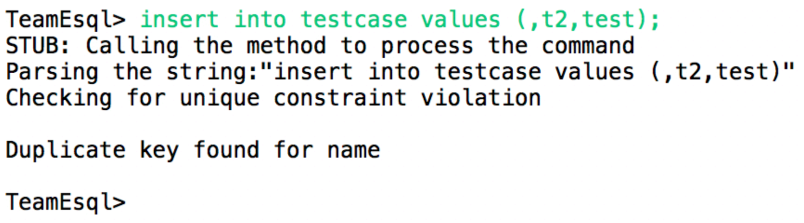
README file

When null is inserted into any not null column, then the database engine throws error.



```
TeamEsql> insert into testcase values (,t3,null);
STUB: Calling the method to process the command
Parsing the string:"insert into testcase values (,t3,null)"
NULL-value constraint violation
```

When duplicate value is passed for column with unique constraint, then the database engine throws error.



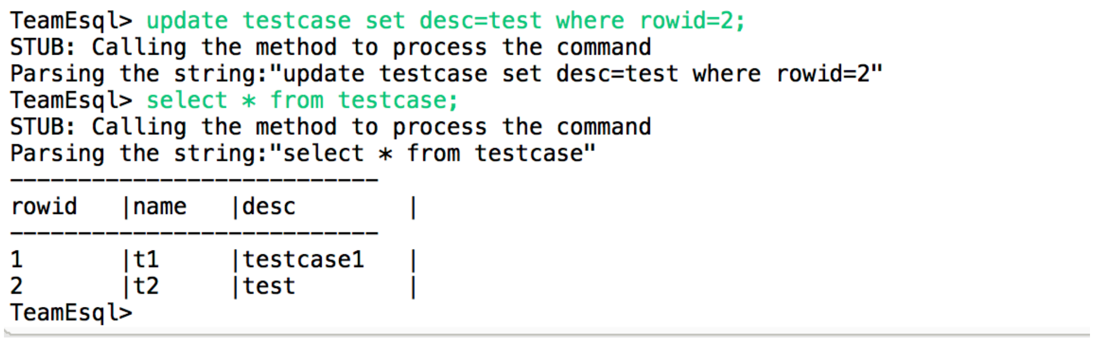
```
TeamEsql> insert into testcase values (,t2,test);
STUB: Calling the method to process the command
Parsing the string:"insert into testcase values (,t2,test)"
Checking for unique constraint violation

Duplicate key found for name

TeamEsql>
```

(e) Update one or more records in table

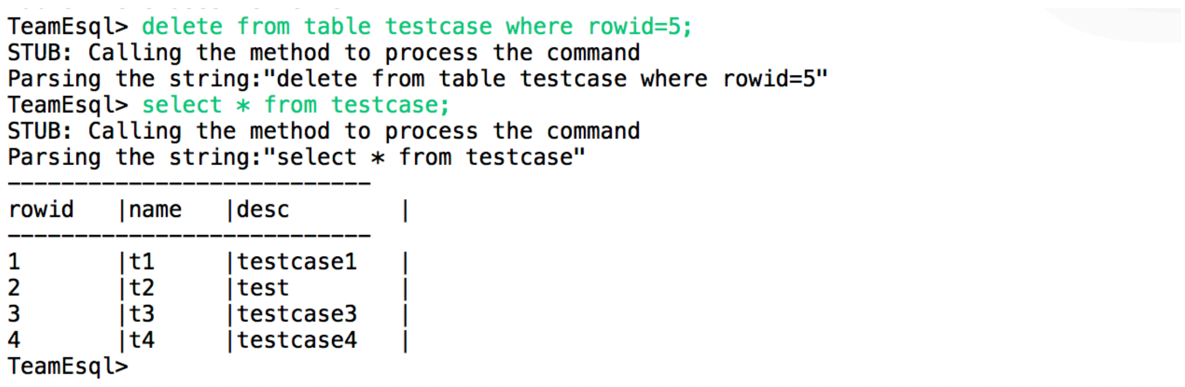
Syntax: UPDATE table_name SET column_name = value WHERE <condition>;



```
TeamEsql> update testcase set desc=test where rowid=2;
STUB: Calling the method to process the command
Parsing the string:"update testcase set desc=test where rowid=2"
TeamEsql> select * from testcase;
STUB: Calling the method to process the command
Parsing the string:"select * from testcase"
---------------------------
rowid   |name   |desc       |
---------------------------
1       |t1     |testcase1  |
2       |t2     |test       |
TeamEsql>
```

(f) Delete records from table

SYNTAX: DELETE FROM TABLE table_name WHERE <condition>;

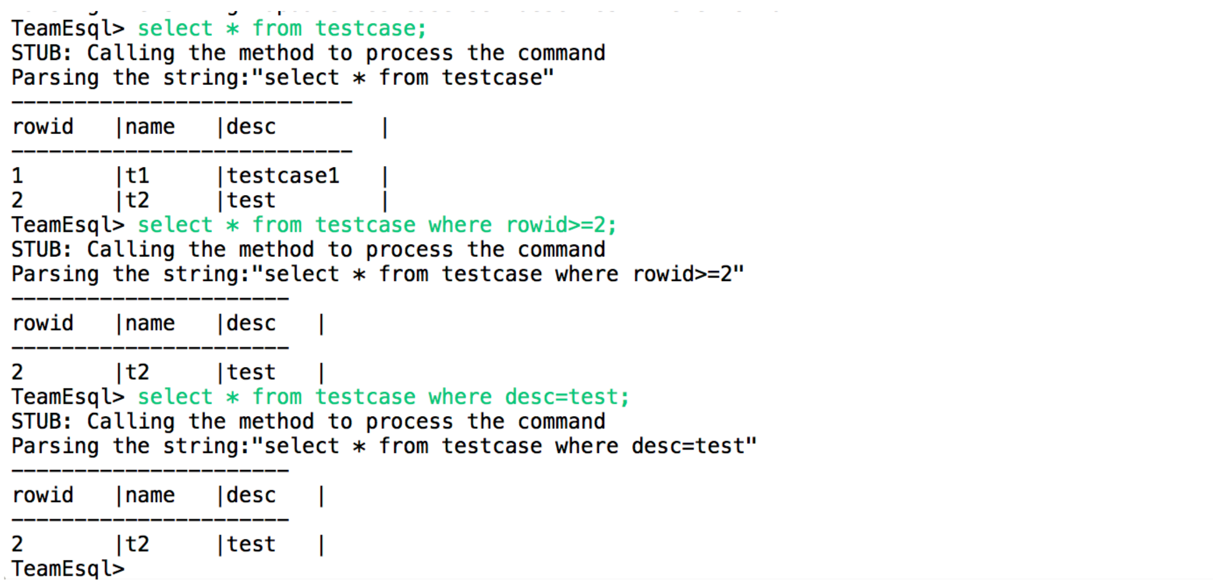


```
TeamEsql> delete from table testcase where rowid=5;
STUB: Calling the method to process the command
Parsing the string:"delete from table testcase where rowid=5"
TeamEsql> select * from testcase;
STUB: Calling the method to process the command
Parsing the string:"select * from testcase"
---------------------------
rowid   |name   |desc       |
---------------------------
1       |t1     |testcase1  |
2       |t2     |test       |
3       |t3     |testcase3  |
4       |t4     |testcase4  |
TeamEsql>
```

README file

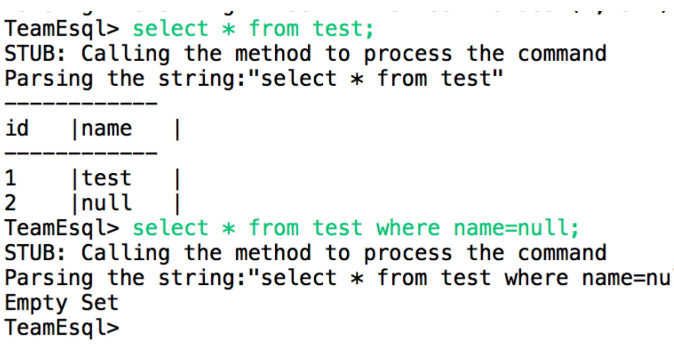
(g) Display records from a table

Syntax: SELECT * FROM <table_name> [WHERE <condition>];



```
TeamEsql> select * from testcase;
STUB: Calling the method to process the command
Parsing the string:"select * from testcase"
---------------------------
rowid    |name    |desc        |
---------------------------
1        |t1      |testcase1   |
2        |t2      |test        |
TeamEsql> select * from testcase where rowid>=2;
STUB: Calling the method to process the command
Parsing the string:"select * from testcase where rowid>=2"
----------------------
rowid    |name    |desc   |
----------------------
2        |t2      |test   |
TeamEsql> select * from testcase where desc=test;
STUB: Calling the method to process the command
Parsing the string:"select * from testcase where desc=test"
----------------------
rowid    |name    |desc   |
----------------------
2        |t2      |test   |
TeamEsql>
```

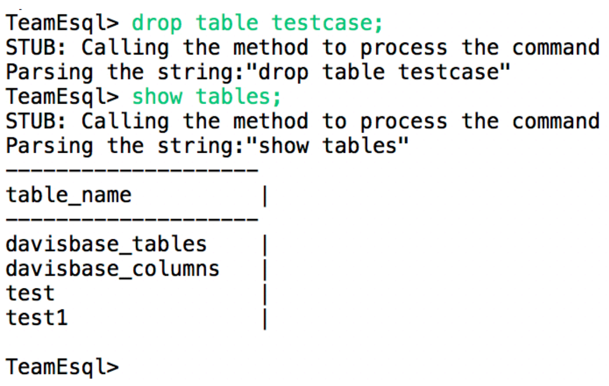
When NULL is used in where condition, the database engine equate null value in data with the where condition.



```
TeamEsql> select * from test;
STUB: Calling the method to process the command
Parsing the string:"select * from test"
------------
id    |name    |
------------
1     |test    |
2     |null    |
TeamEsql> select * from test where name=null;
STUB: Calling the method to process the command
Parsing the string:"select * from test where name=null"
Empty Set
TeamEsql>
```

(h) Drop table from the database

Syntax: DROP TABLE <table_name>;



```
TeamEsql> drop table testcase;
STUB: Calling the method to process the command
Parsing the string:"drop table testcase"
TeamEsql> show tables;
STUB: Calling the method to process the command
Parsing the string:"show tables"
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
test                |
test1               |

TeamEsql>
```

README file

(i) To get help on the command and syntax
Syntax: help;

(j) Exit from the database
Syntax: exit;