README FILE FOR DAVISBASE

This Project is implemented in Java using Eclipse IDE.

Instruction for readme to run file application are as follows:

From Eclipse:

1) Extract the compressed folder downloaded from eLearning.
2) Import the folder SXB180124DavisBase into the Eclipse IDE.
3) Left Click on DavisBase and Select Run as Java Application OR Run the DavisBase.java to start the application.

(or)

From local terminal:

4) Extract the compressed folder SXB180124DavisBase
5) Open terminal or command prompt
6) Navigate to the path where the folder has been extracted.
7) Change directory to the SXB180124DavisBase\src\ in terminal and compile all files using the below command:

```
javac *.java
```

8) Change directory to parent directory (SXB180124DavisBase\src)
9) Run DavisBase.java file using the below command:

```
java DavisBase
```

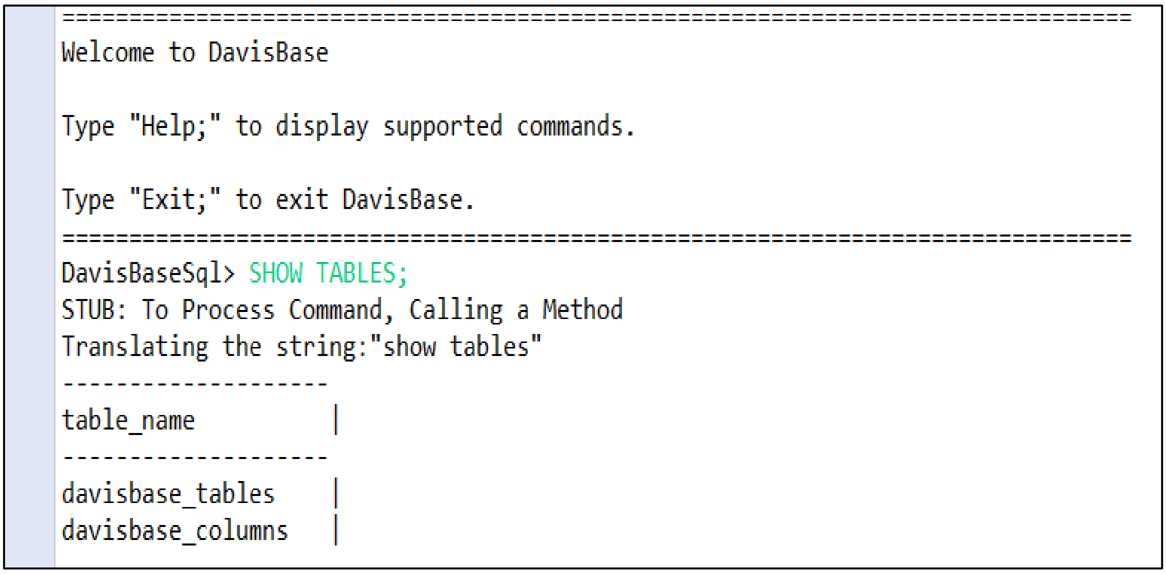
README FILE FOR DAVISBASE

Following are the commands supported by DavisBase:

DDL Commands:

a) SHOW TABLES: To Display a list of all tables in DavisBase.

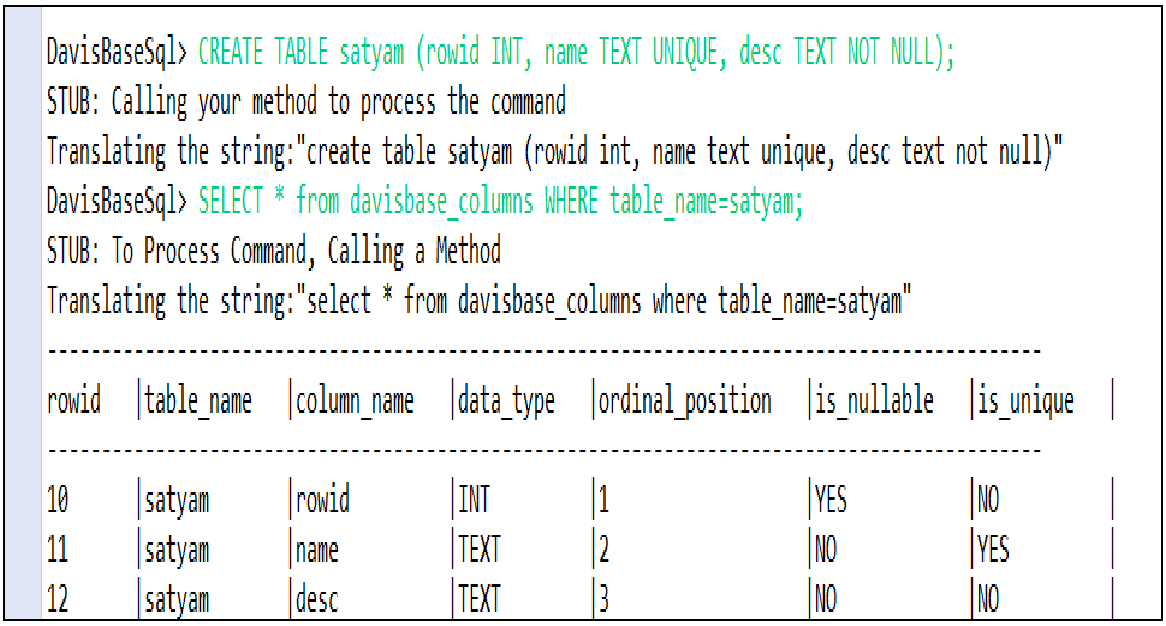
Syntax: SHOW TABLES;



```
================================================================================
Welcome to DavisBase

Type "Help;" to display supported commands.

Type "Exit;" to exit DavisBase.
================================================================================
DavisBaseSql> SHOW TABLES;
STUB: To Process Command, Calling a Method
Translating the string:"show tables"
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
```

b) CREATE TABLE: Creates a new table schema, i.e. a new empty table.

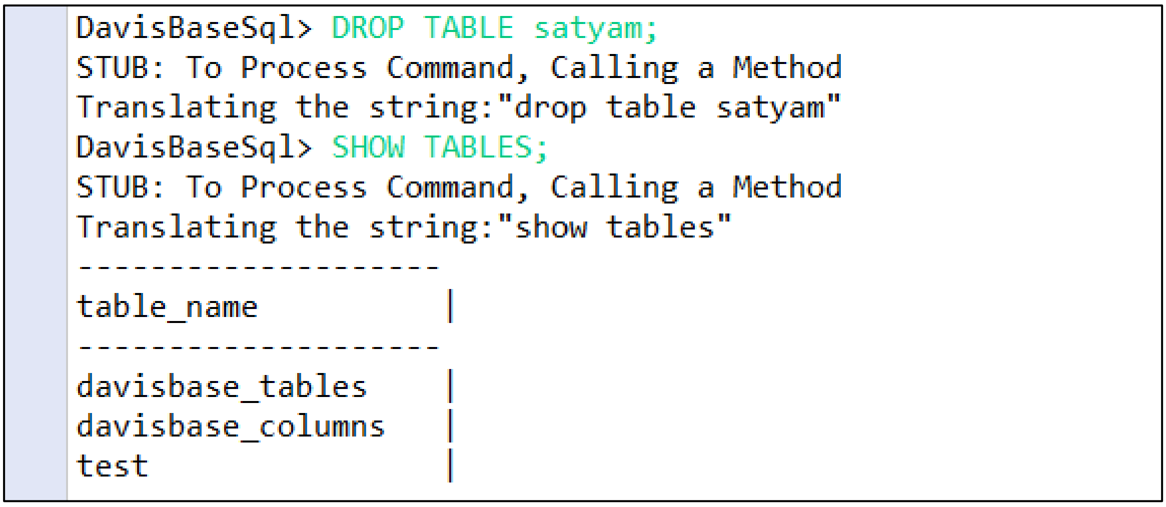
Syntax: CREATE TABLE <table_name> (<column name> <constraint>, ...);



```
DavisBaseSql> CREATE TABLE satyam (rowid INT, name TEXT UNIQUE, desc TEXT NOT NULL);
STUB: Calling your method to process the command
Translating the string:"create table satyam (rowid int, name text unique, desc text not null)"
DavisBaseSql> SELECT * from davisbase_columns WHERE table_name=satyam;
STUB: To Process Command, Calling a Method
Translating the string:"select * from davisbase_columns where table_name=satyam"
------------------------------------------------------------------------------------------
rowid   |table_name   |column_name   |data_type   |ordinal_position   |is_nullable   |is_unique   |
------------------------------------------------------------------------------------------
10      |satyam       |rowid         |INT         |1                  |YES           |NO          |
11      |satyam       |name          |TEXT        |2                  |NO            |YES         |
12      |satyam       |desc          |TEXT        |3                  |NO            |NO          |
```

c) DROP TABLE: Remove a table schema, and all its contained data.

Syntax: DROP TABLE <table_name>;



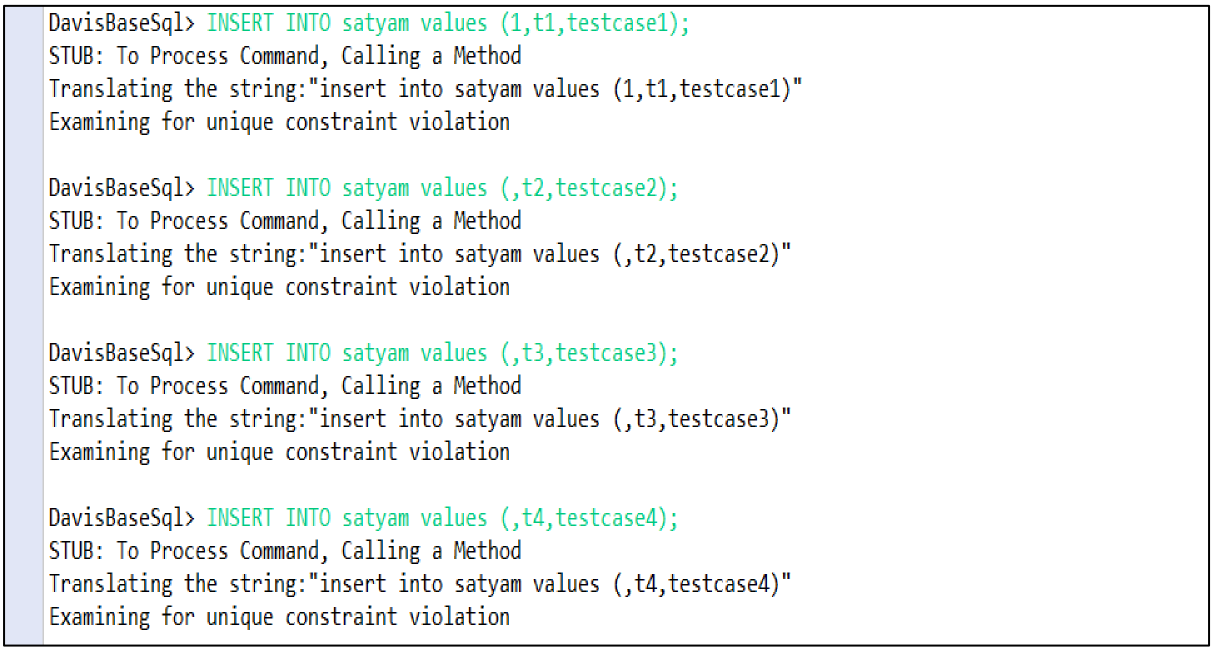
```
DavisBaseSql> DROP TABLE satyam;
STUB: To Process Command, Calling a Method
Translating the string:"drop table satyam"
DavisBaseSql> SHOW TABLES;
STUB: To Process Command, Calling a Method
Translating the string:"show tables"
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
test                |
```

DML Commands:

INSERT INTO: Inserts a single record into a table.

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

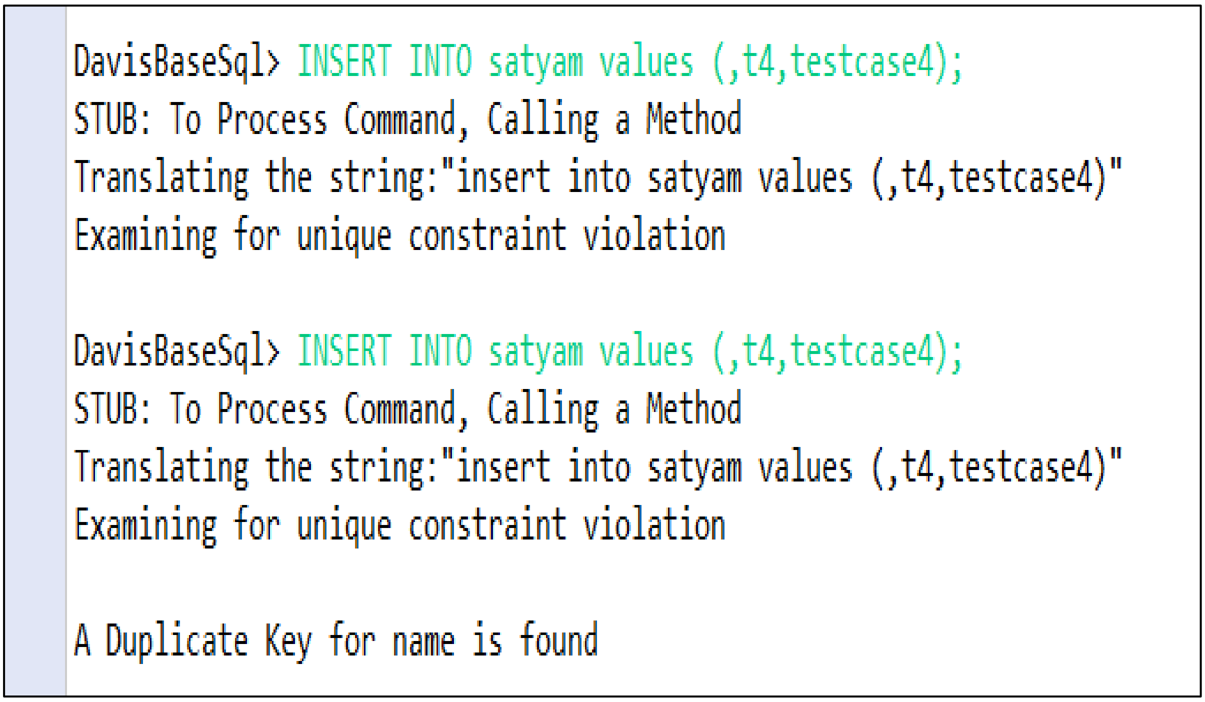
Note1: Here the Value for the Column can be skipped as it can be generated dynamically with use of auto increment feature implemented on Primary Key.



```
DavisBaseSql> INSERT INTO satyam values (1,t1,testcase1);
STUB: To Process Command, Calling a Method
Translating the string:"insert into satyam values (1,t1,testcase1)"
Examining for unique constraint violation

DavisBaseSql> INSERT INTO satyam values (,t2,testcase2);
STUB: To Process Command, Calling a Method
Translating the string:"insert into satyam values (,t2,testcase2)"
Examining for unique constraint violation

DavisBaseSql> INSERT INTO satyam values (,t3,testcase3);
STUB: To Process Command, Calling a Method
Translating the string:"insert into satyam values (,t3,testcase3)"
Examining for unique constraint violation

DavisBaseSql> INSERT INTO satyam values (,t4,testcase4);
STUB: To Process Command, Calling a Method
Translating the string:"insert into satyam values (,t4,testcase4)"
Examining for unique constraint violation
```

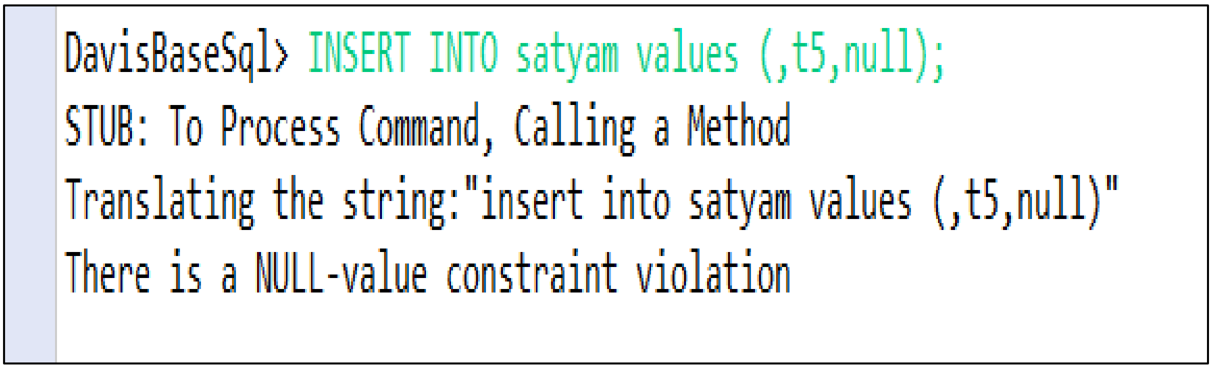
README FILE FOR DAVISBASE

Note2: If the duplicate value is passed for column, then it throws an error "Duplicate Key for column_name is found".



```
DavisBaseSql> INSERT INTO satyam values (,t4,testcase4);
STUB: To Process Command, Calling a Method
Translating the string:"insert into satyam values (,t4,testcase4)"
Examining for unique constraint violation

DavisBaseSql> INSERT INTO satyam values (,t4,testcase4);
STUB: To Process Command, Calling a Method
Translating the string:"insert into satyam values (,t4,testcase4)"
Examining for unique constraint violation

A Duplicate Key for name is found
```

Note3: If a null value is passed in NOT NULL value column, then it throws error "There is a NULL-value constraint violation".



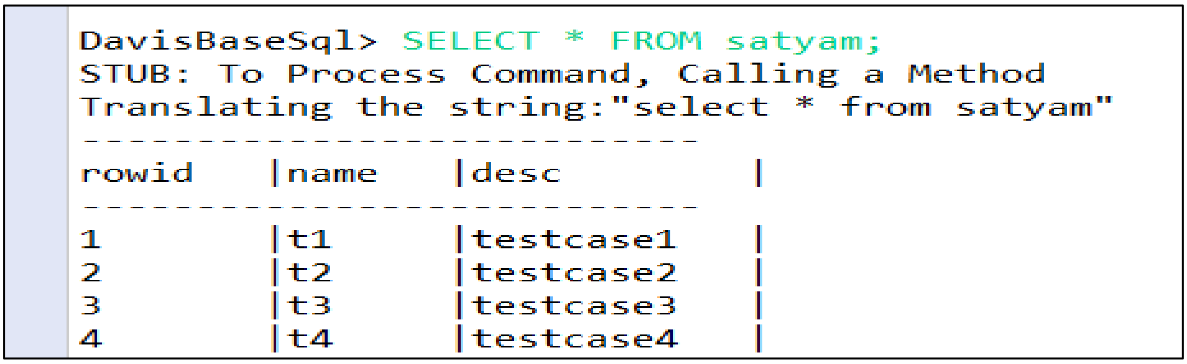
```
DavisBaseSql> INSERT INTO satyam values (,t5,null);
STUB: To Process Command, Calling a Method
Translating the string:"insert into satyam values (,t5,null)"
There is a NULL-value constraint violation
```

README FILE FOR DAVISBASE

VDL Command:

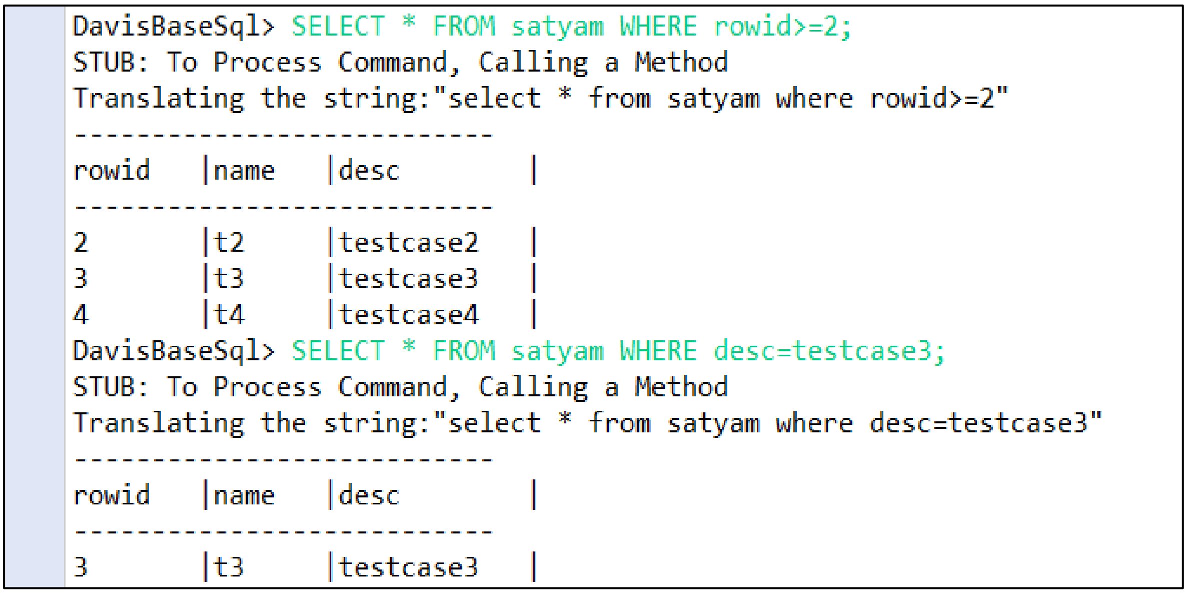
a) SELECT-FROM-WHERE: style query

Syntax: SELECT * FROM <table_name>;



```
DavisBaseSql> SELECT * FROM satyam;
STUB: To Process Command, Calling a Method
Translating the string:"select * from satyam"
---------------------------
rowid   |name   |desc        |
---------------------------
1       |t1     |testcase1   |
2       |t2     |testcase2   |
3       |t3     |testcase3   |
4       |t4     |testcase4   |
```

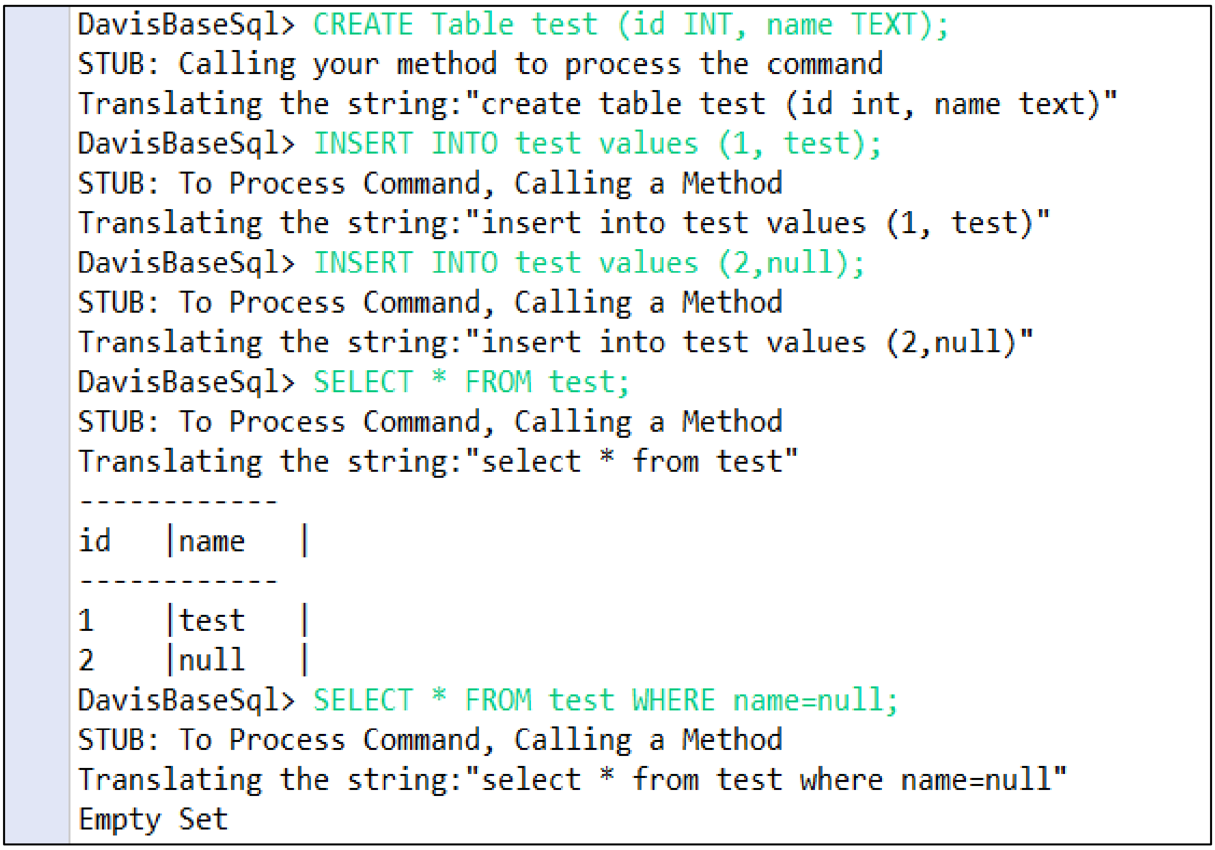
Syntax: SELECT * FROM <table_name> [WHERE <condition>];



```
DavisBaseSql> SELECT * FROM satyam WHERE rowid>=2;
STUB: To Process Command, Calling a Method
Translating the string:"select * from satyam where rowid>=2"
---------------------------
rowid   |name   |desc        |
---------------------------
2       |t2     |testcase2   |
3       |t3     |testcase3   |
4       |t4     |testcase4   |
DavisBaseSql> SELECT * FROM satyam WHERE desc=testcase3;
STUB: To Process Command, Calling a Method
Translating the string:"select * from satyam where desc=testcase3"
---------------------------
rowid   |name   |desc        |
---------------------------
3       |t3     |testcase3   |
```

README FILE FOR DAVISBASE

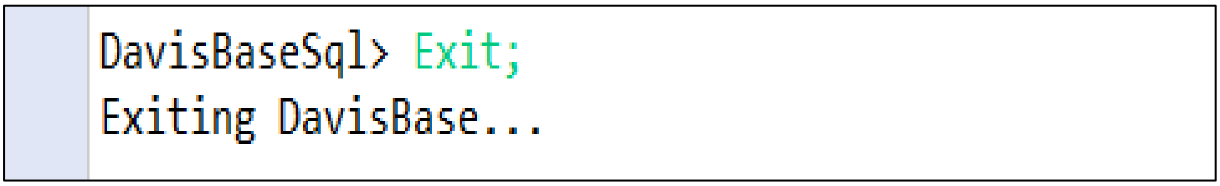
Note: When NULL is used in where condition, the database engine equate null value in data with the where condition.



```
DavisBaseSql> CREATE Table test (id INT, name TEXT);
STUB: Calling your method to process the command
Translating the string:"create table test (id int, name text)"
DavisBaseSql> INSERT INTO test values (1, test);
STUB: To Process Command, Calling a Method
Translating the string:"insert into test values (1, test)"
DavisBaseSql> INSERT INTO test values (2,null);
STUB: To Process Command, Calling a Method
Translating the string:"insert into test values (2,null)"
DavisBaseSql> SELECT * FROM test;
STUB: To Process Command, Calling a Method
Translating the string:"select * from test"
------------
id   |name    |
------------
1    |test    |
2    |null    |
DavisBaseSql> SELECT * FROM test WHERE name=null;
STUB: To Process Command, Calling a Method
Translating the string:"select * from test where name=null"
Empty Set
```

b) Exit: Cleanly exits the program and saves all table information in non-volatile files.

Syntax: exit;



```
DavisBaseSql> Exit;
Exiting DavisBase...
```