

# DavisBase

The goal of this project is to implement a (very) rudimentary database engine that is loosely based on a hybrid between MySQL and SQLite, called DavisBase. DavisBase created using java programming language, supports only one database and one user which are used by default. User can create multiple tables in the database. Supported commands are as follows.

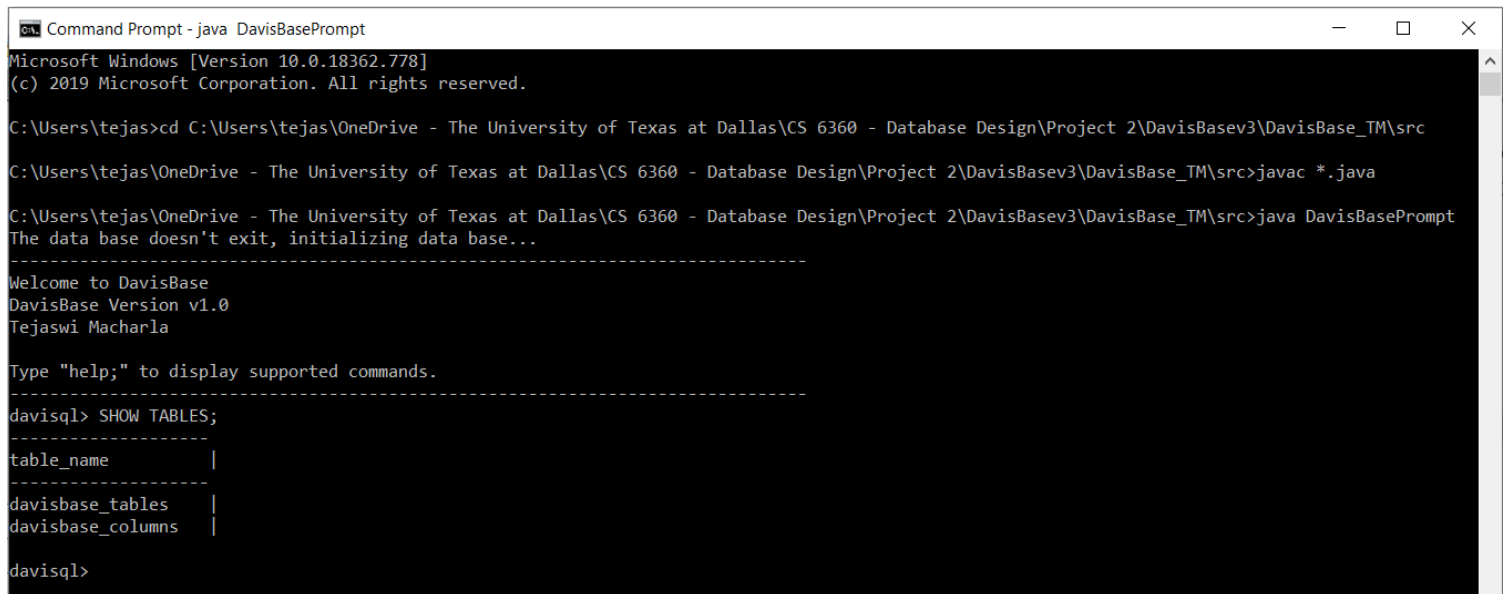
All the commands are case insensitive and must end with ;.

- HELP - Show this help information.

```
davisql> help;
*****
List of all DavisBase commands:
All commands below are case insensitive
1.DDL Commands:
  (a)SHOW TABLES;           Displays a list of all tables in DavisBase
  (b)CREATE TABLE <table_name>; Creates a new table schema, i.e. a new empty table
  (c)DROP TABLE <table_name>; Remove a table schema, and all of its contained data
2.DML Commands:
  (a)INSERT INTO table_name [column_list] VALUES value_list; Inserts a single record into a table
3.VDL Commands:
  (a)SELECT * FROM <table_name>; Display all records in the table
  (b)SELECT * FROM <table_name> WHERE column_name = <value>; Display records satisfying a particular condition
4.EXIT;                      Exit the program and save
5.HELP;                      Show this help information
*****
```

## DDL

- SHOW TABLES – Displays a list of all tables in DavisBase.



```
Command Prompt - java DavisBasePrompt
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\tejas>cd C:\Users\tejas\OneDrive - The University of Texas at Dallas\CS 6360 - Database Design\Project 2\DavisBasev3\DavisBase_TM\src
C:\Users\tejas\OneDrive - The University of Texas at Dallas\CS 6360 - Database Design\Project 2\DavisBasev3\DavisBase_TM\src>javac *.java
C:\Users\tejas\OneDrive - The University of Texas at Dallas\CS 6360 - Database Design\Project 2\DavisBasev3\DavisBase_TM\src>java DavisBasePrompt
The data base doesn't exit, initializing data base...
-----
Welcome to DavisBase
DavisBase Version v1.0
Tejaswi Macharla

Type "help;" to display supported commands.
-----
davisql> SHOW TABLES;
-----
table_name      |
-----
davisbase_tables |
davisbase_columns |
-----
davisql>
```

- CREATE TABLE – Creates a new table schema, i.e. a new empty table.

```
davisql> CREATE TABLE Persons (PersonID INT, LastName TEXT, FirstName TEXT);
davisql> CREATE TABLE Employees (ID INT NOT NULL, LastName TEXT, FirstName TEXT NOT NULL);
davisql> SHOW TABLES;
-----
table_name      |
-----
davisbase_tables |
davisbase_columns |
persons         |
employees       |
```

- DROP TABLE – Remove a table schema, and all of its contained data.

```
davisql> SHOW TABLES;
-----
table_name      |
-----
davisbase_tables |
davisbase_columns |
employees       |
persons         |

davisql> DROP TABLE Persons;
davisql> SHOW TABLES;
-----
table_name      |
-----
davisbase_tables |
davisbase_columns |
employees       |

davisql>
```

## DML

- INSERT INTO TABLE – Inserts a single record into a table.

```

davisql> INSERT INTO Persons VALUES (1, Macharla, Tejaswi);
davisql> INSERT INTO Employees VALUES (1, Macharla, null);
NULL-value constraint violation

davisql> INSERT INTO Employees VALUES (1, Macharla, Tejaswi);
davisql> SELECT * FROM Persons;
-----
personid  |lastname  |firstname  |
-----
1         |macharla  |tejaswi    |
davisql> SELECT * FROM Employees;
-----
id  |lastname  |firstname  |
-----
1   |macharla  |tejaswi    |
davisql> INSERT INTO Persons VALUES (2, Chris, Davis);

```

VDL

- “SELECT-FROM-WHERE” -style query.

```

davisql> SELECT PersonID, LastName, FirstName FROM Persons WHERE PersonID = 1;
-----
personid  |lastname  |firstname  |
-----
1         |macharla  |tejaswi    |
davisql>

```

- EXIT – Cleanly exits the program and saves all table information in non-volatile files.

```

davisql> exit;
Exiting...
DavisBase Exited

C:\Users\tejas\OneDrive - The University of Texas at Dallas\CS 6360 - Database Design\Project 2\DavisBasev3\DavisBase_TM\src>

```