INSTRUCTIONS TO RUN DAVISBASE

- Open Command Prompt.
- Go to the current folder where the "**DavisBase.java"** is present.
- Compile using the command: "**javac DavisBase.java".** It'll create all the classes.
- Run the program by the command: "**java DavisBase".** It'll create the data folder for storage.

```
Microsoft Windows [Version 10.0.18363.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\userr>cd C:\Users\userr\OneDrive\Desktop\Summer2020\DbDesign\Project\test\c

C:\Users\userr\OneDrive\Desktop\Summer2020\DbDesign\Project\test\c>javac DavisBase.java

C:\Users\userr\OneDrive\Desktop\Summer2020\DbDesign\Project\test\c>java DavisBase
```

The program will start with the following display:

```
--------------------------------------------------------------------------------


Welcome to DavisBase.

Type "help;" to display supported commands.
Use ";" at the end of each command.

--------------------------------------------------------------------------------
```

Type **"help;"** to display the supported commands:

```
RKsql> help;

********************************************************************************

SUPPORTED COMMANDS :

SHOW TABLES;
        Display all the tables in the database.

CREATE TABLE table_name (<column_name datatype> <NOT NULL/UNIQUE>);
        Create new table in the database.

DROP TABLE table_name;
        Remove table.

INSERT INTO table_name VALUES (value1,value2,....);
        Insert new row into table. Column1 is always primary key
        which is inbuilt and increments automatically.

DELETE FROM TABLE table_name WHERE row_id = key_value;
        Delete a record from the table whose rowid is <key_value>.

UPDATE table_name SET column_name = value WHERE condition;
        Modifies the records in the table.

SELECT * FROM table_name;
        Display all records from table.

SELECT * FROM table_name WHERE column_name operator value;
        Display records from table where given condition is satisfied.

HELP;
        Show supported commands.

EXIT;
        Exit program.

********************************************************************************
```

COMMANDS:

SHOW TABLES;

This will list all the tables stored in the data.

CREATE TABLE table_name (<column_name datatype> <NOT NULL/UNIQUE>);

This will create a table with the name provided in **table_name** and with columns as described.

```
RKsql> SHOW TABLES;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |

RKsql> CREATE TABLE Dogs ( ID SHORT , NAME TEXT , WEIGHT TEXT , AGE INT);
RKsql> SHOW TABLES;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
dogs                |

RKsql>
```

SELECT * FROM Dogs;

This will display the contents of the table.

Note : The program will notify if the table is empty.

INSERT INTO DOGS VALUES (933,Rover,20.6,4);

This will insert values into the table.

```
RKsql> SELECT * FROM Dogs;
Table is empty
RKsql> INSERT INTO DOGS VALUES (933,Rover,20.6,4);
RKsql> SELECT * FROM Dogs;
----------------------------
id    |name    |weight   |age   |
----------------------------
933   |rover   |20.6     |4     |
RKsql>
```

Insert command also checks for unique ID for the entries. It won't accept duplicate ID's

```
RKsql> INSERT INTO DOGS VALUES (933,Rover,20.6,4);
RKsql> INSERT INTO DOGS VALUES (933,Denver,25.0,3);
Uniqueness constraint violation
RKsql>
```

DROP TABLE Dogs;

This will delete the table with the given name.

```
RKsql> SELECT * FROM Dogs;
-----------------------------
id    |name    |weight   |age   |
-----------------------------
933   |rover   |20.6     |4     |
RKsql> DROP TABLE Dogs;
RKsql> SHOW TABLES;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
```

Insert all the values in the table one by one.

Check the table by SELECT command.

```
RKsql> SELECT * FROM Dogs;
-----------------------------
id    |name    |weight   |age   |
-----------------------------
933   |rover   |20.6     |4     |
RKsql> INSERT INTO DOGS VALUES (8326,Spot,10.8,7);
RKsql> INSERT INTO DOGS VALUES (5359,Lucky,31.2,5);
RKsql> INSERT INTO DOGS VALUES (10355,Dinky,4.8,11);
RKsql> INSERT INTO DOGS VALUES (7757,Bruiser,42.0,6);
RKsql> INSERT INTO DOGS VALUES (3597,Patch,29.6,9);
RKsql> INSERT INTO DOGS VALUES (202,Prince,16.6,7);
RKsql> INSERT INTO DOGS VALUES (1630,Bubbles,7.1,1);
RKsql> INSERT INTO DOGS VALUES (11223,Peanut,14.3,2);
RKsql> SELECT * FROM Dogs;
--------------------------------
id      |name      |weight   |age   |
--------------------------------
10355   |dinky     |4.8      |11    |
933     |rover     |20.6     |4     |
8326    |spot      |10.8     |7     |
11223   |peanut    |14.3     |2     |
202     |prince    |16.6     |7     |
7757    |bruiser   |42.0     |6     |
3597    |patch     |29.6     |9     |
1630    |bubbles   |7.1      |1     |
5359    |lucky     |31.2     |5     |
RKsql>
```

SELECT * FROM Dogs WHERE AGE=7;

This will display all the dogs whose age=7;

```
RKsql> SELECT * FROM Dogs;
---------------------------------
id      |name      |weight   |age   |
---------------------------------
10355   |dinky     |4.8      |11    |
933     |rover     |20.6     |4     |
8326    |spot      |10.8     |7     |
11223   |peanut    |14.3     |2     |
202     |prince    |16.6     |7     |
7757    |bruiser   |42.0     |6     |
3597    |patch     |29.6     |9     |
1630    |bubbles   |7.1      |1     |
5359    |lucky     |31.2     |5     |
RKsql> SELECT * FROM Dogs WHERE AGE=7;
-------------------------------
id     |name     |weight   |age   |
-------------------------------
8326   |spot     |10.8     |7     |
202    |prince   |16.6     |7     |
RKsql>
```

EXIT;

This will exit from the program.

```
RKsql> EXIT;
Exiting...

C:\Users\userr\OneDrive\Desktop\Summer2020\DbDesign\Project\test\c>
```