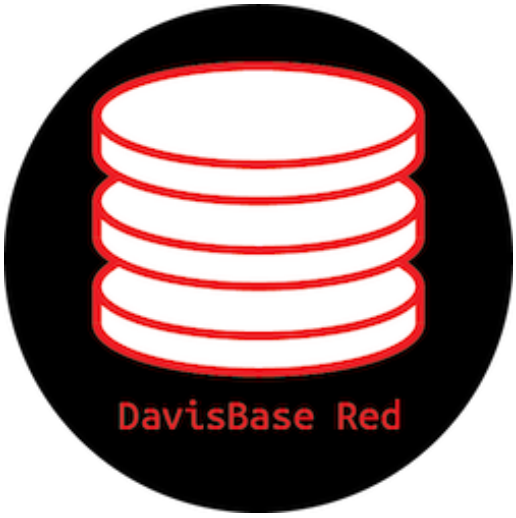
# DavisBase Red




DavisBase Red is a simple relational database management system (RDBMS) created for educational purposes. It was authored by Sangeetha Tatineni, Ameya Potey, Ben Monical, Aarthi Gunasekaran, and Andrew Byers for a summer database design course at the University of Texas at Dallas. Due to stringent time constraints, only a subset of the initial project requirements have been implemented, however the DBMS remains minimally functional. The application supports table creation/deletion, record insertion/manipulation/deletion, and querying using logical comparisons on the `rowid` field.

## Setup

Download the and unzip .zip file to the location of your choice, then in a fresh terminal window, `cd` over to the project directory and run

```
$ javac *.java
$ java DavisBasePrompt
```

to compile the `.class` files and run the command line shell.

## Dependencies

This project was built and tested on `Java SE >16.0.0`, however it may still function on older versions.

## Usage

When `java DavisBasePrompt` is run, you will be greeted with the splash screen and DavisBase command line prompt.  From here you can issue the command `help;` to

view a list of the available commands, which includes their function signatures and a short description about what they do. All identifiers in DavisBase are case insensitive, so you don't have to worry about case when referring to commands, table names, or column names.

Say you want to create a table to keep track of a collection of tasks that must be completed within some set timeframe. Each task entity will be represented by a table called `Task`, which has name, description, due date, and due time attributes. We can accomplish creation of such a table with the command

```
davisql> CREATE TABLE Task (
           Name         TEXT NOT NULL,
           Description  TEXT NOT NULL,
           DueDatetime  DATETIME NOT NULL,
         );
```

See the DavisBase documentation for descriptions of the available data types. From here, we can insert individual task records like

```
davisql> INSERT INTO TABLE (Name, Description, DueDatetime)
         Task VALUES (
           'Name of the task',
           'Task description',
           '2021-05-21'
         );
```

DavisBase automatically gives each record a special column called `rowid`. For a given table, values of `rowid` are monotonically increasing, starting at 1. The `rowid` value, along with the other column values, can be queried using

```
davisql> SELECT rowid, *
         FROM Task;
```

The `SELECT` command also supports a `WHERE` clause, using a logical comparison on the `rowid` field. The `rowid` value can then be used to refer to the individual row. If we wanted to rename our task, we could call

```
davisql> UPDATE Task
         SET Name = 'New task name'
         WHERE rowid < 2;
```

When each task is completed, we can remove it using

```
davisql> DELETE FROM TABLE Task
         WHERE rowid = 1;
```

Then finally, once all our tasks are completed, we can remove the `Task` table with

```
davisql> DROP TABLE Task;
```