

**HOMEWORK 1 REPORT**  
**Submitted by: Team Runtime Terror**  
**Submitted on: 1/30/2020**

**About the project h2 database:**

H2 is an embeddable RDBMS written in Java. It has both Embedded and server modes.

**Feature 1: Create Table**

First, we started off by searching for the string “createtable” in the codebase using Find Usage functionality of IntelliJ. We found CreateTable class which represents the CREATE TABLE command. CreateTable class makes use of CreateTableData which represents the data that is used to create the table like schema, column names, table name, whether table would be temporary or global, should be persisted or not, etc. The update method calls the createTable in Schema class which in turn calls the createTable method in MVTableEngine class. This then calls createTable method in Store which is an inner class in MVTableEngine. In this method, we create an instance of the Table object and store the table in tableMap which is a ConcurrentHashMap and then return the table. Now an object table of type MVTable is returned through a return hierarchy to the initial CreateTable class thus creating a table.

**Feature 2: How to set user privileges (admin)**

First, we started off by searching for the string “user” in the codebase because we believed all the user related meta would be present there. We found User class which has a setAdmin method. We believe this User class represents each User of the h2 database. For example, User A could be an admin and have master privileges whereas the other users might just have select query privileges.

We then checked for the usages of setAdmin to better understand where the admin privileges are being set for the user. Then we came across the openSession method in Engine.java. There we found out if the database is not created or is new, the current session user would be granted admin privileges. And will also be set as the master user for the database in consideration.

Then we came across the class called CreateUser which represents the statement CREATE USER. The parsing of this statement takes place in a file called Parser.java in which we create an instance of CreateUser. Now based on the parsed tokens if we find an admin token, then we

use the setAdmin method in CreateUser with a TRUE parameter making the user admin to the database.

Next, we came across AlterUser class where we found out that Admin flag of a user can be toggled between admin and non-admin. Following which we update the meta data for the particular user.