DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF MANITOBA

Using the HSQLDB System

This document provides a brief introduction to using the HSQLDB system. The HSQLDB system implements a minimal relational dbms that can be accessed and manipulated using SQL commands. The HSQLDB system is primarily designed for small to medium size databases although it has been used for larger databases that do not require the functionality of a "complete" database management system (such as Oracle).

Setting Up HSQLDB:

The version of HSQLDB that is included in this folder is HSQLDB 1.7.2.4. A complete download of the HSQLDB system is available from www.hsqldb.org or from http://hsqldb.sourceforge.net/. The complete download includes the source code (in Java) for HSQLDB plus a variety of other files. However, the only file that is actually needed to run HSQLDB is the file hsqldb.jar Prior to running the HSQLDB system, ensure that Java 1.4 is installed. It is not necessary to install the HSQLDB system, simply unzip the hsqldb.zip file to a folder. A collection of Windows batch scripts that illustrate the use of the HSQLDB system have been included in the folder. (The HSQLDB system runs under Linux but the batch scripts would have to be modified slightly first.)

In order to run HSQLDB, the file hsqldb.jar must be available. This jar file has been made available by adding it to the Windows classpath at the beginning of each script using the statement:

```
set classpath=.;.\hsqldb.jar;%classpath%
```

If you prefer, you can add the jar file to the Java external runtime library which typically is in the following location:

```
C:\Program Files\Java\j2re1.4.1_02\lib\ext
```

Note: ensure that you add the jar file to the Java runtime folder, not to the SDK folder.

Alternately, you could add the jar file to your own classpath variable by modifying the classpath variable at Settings→Control Panel→System→Advanced→Environment Variables.

If you make the jar file available globally, you can remove the SET CLASSPATH statement from the scripts.

Two documents that describe the HSQLDB system are included in this folder. The first document, guide1.pdf, was included with the HSQLDB distribution and was generated in "portrait" mode which caused some command lines to be truncated. The second document, guide2.pdf, was generated from the html documentation in "landscape" mode so that characters are not truncated. Guide1.pdf is 111 printed pages while guide2.pdf is 156 printed pages.

Defining an HSQLDB Database:

The HSQLDB system is unique in that databases created with the system are not stored using a proprietary physical representation. Instead, each database is represented by the SQL commands that were used to create it. For example, the Students database that is included in the folder is defined in the text file Students.script. This file contains the following statements:

```
CREATE TABLE STUDENTS(STUDENTID INTEGER NOT NULL PRIMARY KEY,STUDENTNAME VARCHAR(20))
CREATE TABLE COURSES(COURSEID INTEGER NOT NULL PRIMARY KEY,COURSENAME VARCHAR(20))
CREATE TABLE SC(STUDENTID INTEGER NOT NULL,COURSEID INTEGER NOT NULL,GRADE
   VARCHAR(20),CONSTRAINT SYS_PK_SC PRIMARY KEY(STUDENTID,COURSEID))
CREATE USER SA PASSWORD "" ADMIN
INSERT INTO STUDENTS VALUES(100,'Joe Dirt')
INSERT INTO STUDENTS VALUES(200,'Clint Eastwood')
INSERT INTO STUDENTS VALUES(300,'Neo Matrix')
INSERT INTO STUDENTS VALUES(400,'Michael Zapp')
INSERT INTO STUDENTS VALUES(500,'Alan Marshall')
INSERT INTO COURSES VALUES(74302,'Human-Computer Interaction')
INSERT INTO COURSES VALUES(74335,'Software Engineering')
INSERT INTO COURSES VALUES(74338,'Databases')
INSERT INTO SC VALUES(100,74302,'B')
INSERT INTO SC VALUES(100,74335,'A')
INSERT INTO SC VALUES(100,74338,'A+')
INSERT INTO SC VALUES(200,74302,'A+')
INSERT INTO SC VALUES(200,74335,'A')
INSERT INTO SC VALUES(200,74338,'A+')
INSERT INTO SC VALUES(300,74302,'B')
INSERT INTO SC VALUES(300,74338,'A+')
```

When changes are made to the database, this script is updated to reflect the changes.

The HSQLDB system modifies the parameters of some CREATE statements by adding the keyword CONTSTRAINT followed by a corresponding constraint name. For example, in the statement `CREATE TABLE SC`, the primary key was initially identified using the statement:

```
      PRIMARY KEY(STUDENTID,COURSEID)
```

but the HSQLDB system modified the statement so that it became:

```
      CONSTRAINT SYS_PK_SC PRIMARY KEY(STUDENTID,COURSEID)
```

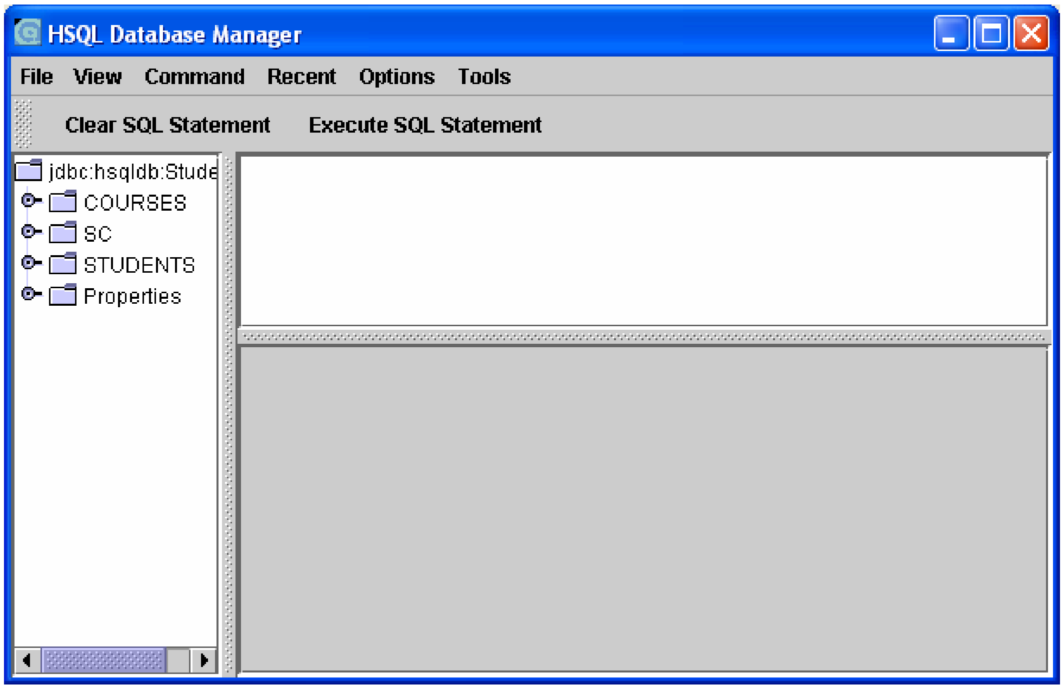
Once the organization of an HSQLDB database is understood, the user can either create a database interactively or simply create a file of SQL commands that define the database in a .script file. The user can also modify a database by editing the corresponding .script file.

Running HSQLDB Interactively:

Once a database has been created, it can be queried and/or modified in several ways. The simplest is to use the DatabaseManager utility that is included with the HSQLDB system. This utility is invoked using the following statement (see the script named GUI):

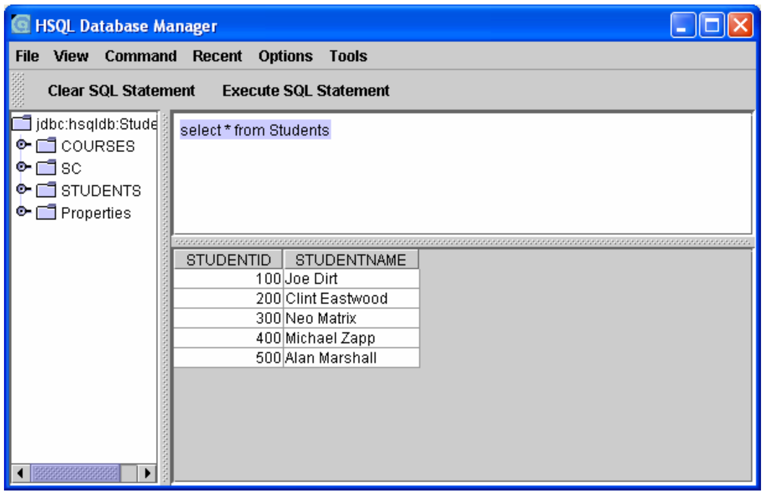
```
java org.hsqldb.util.DatabaseManagerSwing -url "jdbc:hsqldb:Students"
```

This utility allows the user to enter SQL commands interactively. The window initially looks like the following:



The left pane of the window identifies the relations that have been created; in this example, the relations are named: Students, Courses, and SC.

The user then types SQL commands into the command area and then clicks on the Execute SQL Statement button (or uses the CTRL-E shortcut). For example, executing the statement `Select * from Students` causes all information about each student to be generated.



```
select * from Students
```

STUDENTID	STUDENTNAME
100	Joe Dirt
200	Clint Eastwood
300	Neo Matrix
400	Michael Zapp
500	Alan Marshall

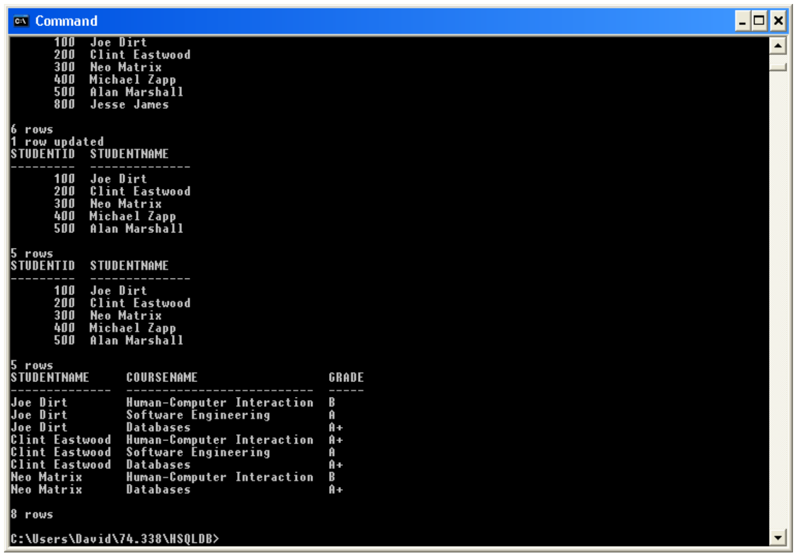
The user may update the database as well as query the database using this utility.

Running HSQLDB in Batch Mode:

Alternatively, if you have a collection of SQL commands already defined that you want to have executed, you can use the SqlTool utility program (see the script SQL):

```
java org.hsqldb.util.SqlTool --rcfile sqltool.rc Students TestStudents.sql
```

This utility executes the SQL statements in a file (in this example, TestStudents.sql). The results are generated on the system console, as is shown below:



```
      100  Joe Dirt
      200  Clint Eastwood
      300  Neo Matrix
      400  Michael Zapp
      500  Alan Marshall
      800  Jesse James

6 rows
1 row updated
STUDENTID  STUDENTNAME
---------  -------------
      100  Joe Dirt
      200  Clint Eastwood
      300  Neo Matrix
      400  Michael Zapp
      500  Alan Marshall

5 rows
STUDENTID  STUDENTNAME
---------  -------------
      100  Joe Dirt
      200  Clint Eastwood
      300  Neo Matrix
      400  Michael Zapp
      500  Alan Marshall

5 rows
STUDENTNAME     COURSENAME                  GRADE
--------------  --------------------------  -----
Joe Dirt        Human-Computer Interaction  B
Joe Dirt        Software Engineering        A
Joe Dirt        Databases                   A+
Clint Eastwood  Human-Computer Interaction  A+
Clint Eastwood  Software Engineering        A
Clint Eastwood  Databases                   A+
Neo Matrix      Human-Computer Interaction  B
Neo Matrix      Databases                   A+

8 rows

C:\Users\David\74.338\HSQLDB>
```

This utility requires that an additional file, sqltool.rc, be available. This file must identify each database that will be accessed using SqlTool. To access the Students database, the following statements must be included in the sqltool.rc file. (Note: these statements have already been defined in the sqltool.rc file.)

```
# A personal Memory-Only database.
urlid Students
url jdbc:hsqldb:Students
username sa
password
```

Running HSQLDB in Java Mode:

Java programs may also access any database that has a corresponding JDBC driver available. For example, to connect to the Students database, the following two Java statements are used to define the connection.

```
Class.forName("org.hsqldb.jdbcDriver").newInstance();
url = "jdbc:hsqldb:Students"; // stored on disk mode
```

The Java program can then manipulate the database by issuing normal SQL statements. See the program DBTest.java in this folder for an example of a simple Java program that manipulates the Students database.

General Comments:

Once a database has been created, it is a good idea to make a copy of the *database*.script file so that if the database is accidentally trashed, it can be recovered by restoring the original script file. In this folder, the Restore script restores the Student database to its original form using the statement:

```
copy originalstudents.script students.script
```

The HSQLDB system supports most of the SQL '92 standard. See the documentation in the HSQLDB guides for specific information on SQL commands.

The HSQLDB examples illustrated in this document were run in single-user mode but HSQLDB may also be run in multi-user mode by installing an HSQLDB server.

David Scuse
Department of Computer Science
University of Manitoba

Sunday, September 19, 2004