

# H2数据库使用 - yixiaoping的专栏 - CSDN博客

## H2数据库介绍

常用的开源数据库：H2，Derby，HSQLDB，MySQL，PostgreSQL。其中H2，HSQLDB类似，十分适合作为嵌入式数据库使用，其它的数据库大部分都需要安装独立的客户端和服务端。

## H2的优势：

1、h2采用纯Java编写，因此不受平台的限制。

2、h2只有一个jar文件，十分适合作为嵌入式数据库试用。

3、性能和功能的优势

H2比HSQLDB的最大的优势就是h2提供了一个十分方便的web控制台用于操作和管理数据库内容，这点比起HSQLDB的swing和awt控制台实在好用多了。

H2和各数据库特征比较。

H2特征比较：

	H2	Derby	HSQLDB	MySQL	PostgreSQL
纯Java	支持	支持	支持	不支持	不支持
内存模式	支持	不支持	支持	不支持	不支持
事务隔离	支持	支持	不支持	支持	支持
基于代价的优化方式	支持	支持	不支持	支持	支持
数据库加密	支持	支持	不支持	不支持	不支持
ODBC驱动	支持	不支持	不支持	支持	支持
全表扫描	支持	不支持	不支持	支持	支持
数据库并发控制	支持	不支持	不支持	不支持	支持
数据库大小	~1MB	~2MB	~600kb	~4MB	~4MB

## 准备工作

1、h2-2011-04-04.zip 下载地址：<http://www.h2database.com/html/download.html>

2、解压文件，这里以%H2\_HOME%表示为解压的文件目录。运行%H2\_HOME%\bin\h2.bat 将会自动打开下面网址。（请确认是否安装了jdk，并设置了JAVA\_HOME环境变量）

<http://192.168.140.1:8082/login.jsp?jsessionid=244e36a683f97f0d4f3b000f33530ed1>

http://192.168.140.1:8082/login.jsp?jsessionid=...

中文 (简体) 配置 Tools 帮助

### 登录

保存的连接设置: Generic H2 (Embedded)

连接设置名称: Generic H2 (Embedded)

---

驱动类: org.h2.Driver

JDBC URL: jdbc:h2:~/test

用户名: sa

密码:

3、点击 connect ，登录。

jdbc:h2:~/test | 自动提交 | 最大行数: 1000 | Auto complete | 帮助

jdbc:h2:~/test | INFORMATION\_SCHEMA | 用户 | H2 1.3.154 (2011-04-04)

执行 (Ctrl+Enter) | 清除 | SQL 语句:

重要的命令

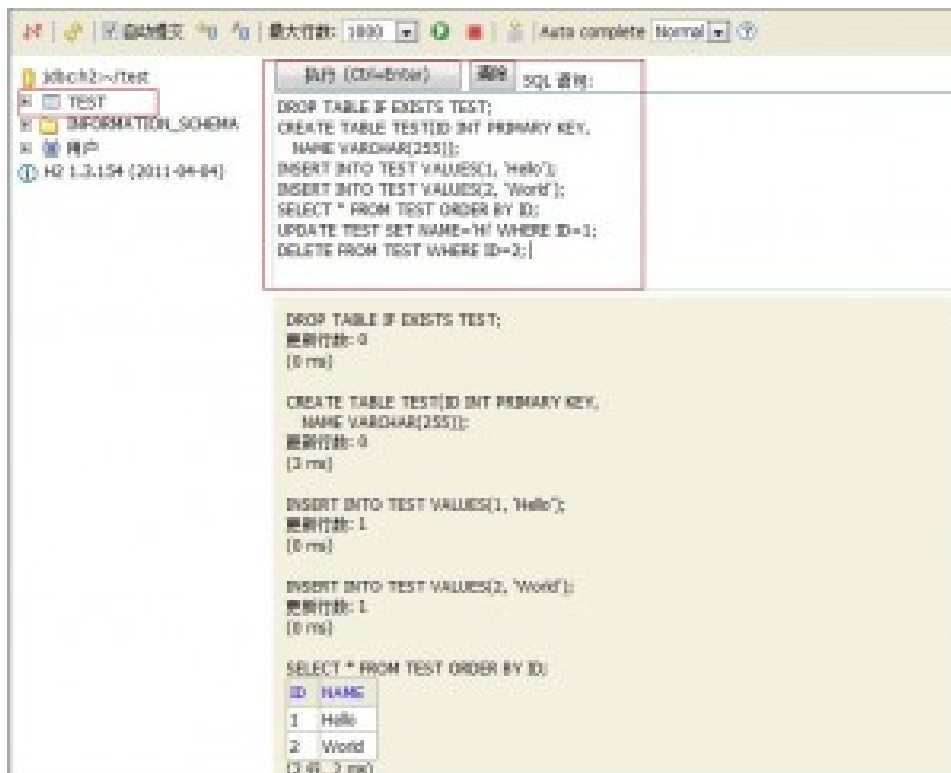
- 显示帮助页
- 显示历史SQL命令
- 执行当前SQL语句
- 离开数据库连接

复制SQL脚本

如果表存在删除它	DROP TABLE IF EXISTS TEST;
创建一个新表	CREATE TABLE TEST(ID INT PRIMARY KEY,
用名称NAME列	NAME VARCHAR(255));
添加新的一行	INSERT INTO TEST VALUES(1, 'Hello');
添加另一行	INSERT INTO TEST VALUES(2, 'World');
查询表	SELECT * FROM TEST ORDER BY ID;
设置一行数据	UPDATE TEST SET NAME='hi' WHERE ID=1;
删除一行	DELETE FROM TEST WHERE ID=2;
帮助	HELP ...

增加数据库驱动

4、执行上图中红色部分sql语句，成功创建test表。



因为没有指定数据库文件位置，会自动输出到输出到C:\Users\Administrator下。

test.h2.db	2021/4/27 15:50	Data Base File	2,080 KB
test.lock.db	2021/4/27 15:50	Data Base File	1 KB
test.trace.db	2021/4/27 15:48	Data Base File	1 KB
h2server.properties	2021/4/27 15:50	PROPERTIES 文件	2 KB

H2文件结构

%H2\_HOME%

-h2

-bin

h2-1.3.154.jar //jar包

h2.bat //Windows控制台启动脚本

h2.sh //Linux控制台启动脚本

h2w.bat //Windows控制台启动脚本（不带黑屏窗口）

+docs 帮助文档

+service //通过wrapper包装成服务。

+src //源代码

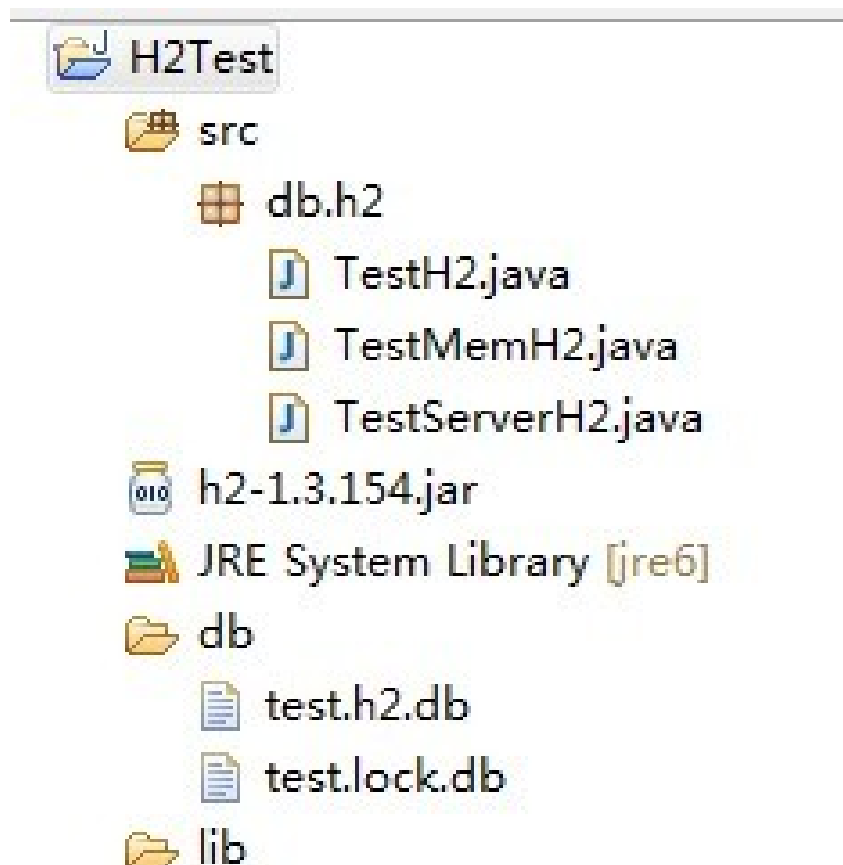
build.bat windows构建脚本

build.sh linux构建脚本

H2的使用

支持Embedded，server和in-memory模式以及内存模式。

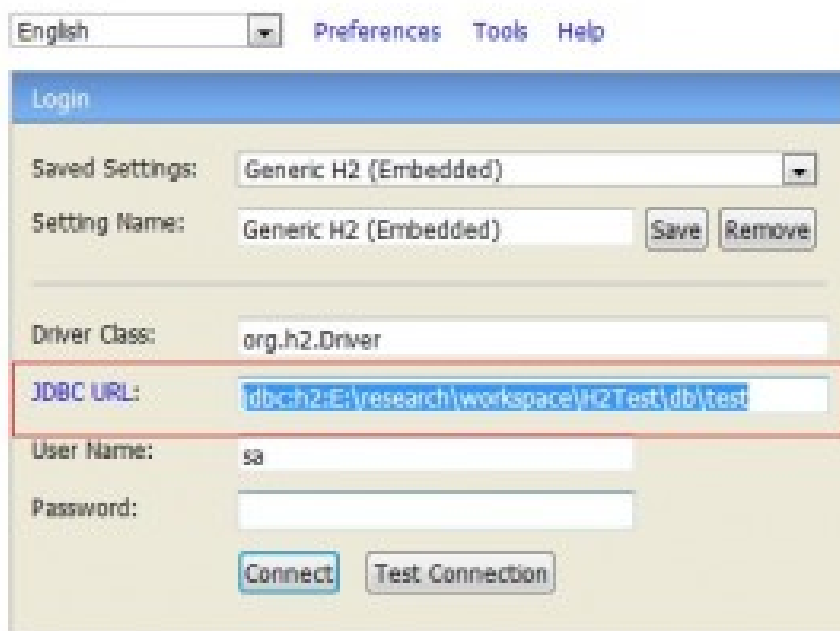
1、新建java project工程 H2Test。



www.lifeba.org

2、%H2\_HOME%\bin\h2-1.3.154.jar 复制到 \H2Test\lib下，并加入工程引用。

3、新建Generic H2 (Embedded)数据库，指定：JDBC URL: jdbc:h2:E:\research\workspace\H2Test\db\test，然后执行上面的test sql语句，来创建一个test表。



4、新建 TestH2类 主要代码

```

public static void main(String[] a)

    throws Exception {

    Class.forName("org.h2.Driver");

    Connection conn = DriverManager.

        getConnection("jdbc:h2:E:\\research\\workspace\\H2Test\\db\\test", "sa", "");

    // add application code here

    Statement stmt = conn.createStatement();

    ResultSet rs = stmt.executeQuery("SELECT * FROM TEST ");

    while(rs.next()) {

        System.out.println(rs.getInt("ID")+", "+rs.getString("NAME"));

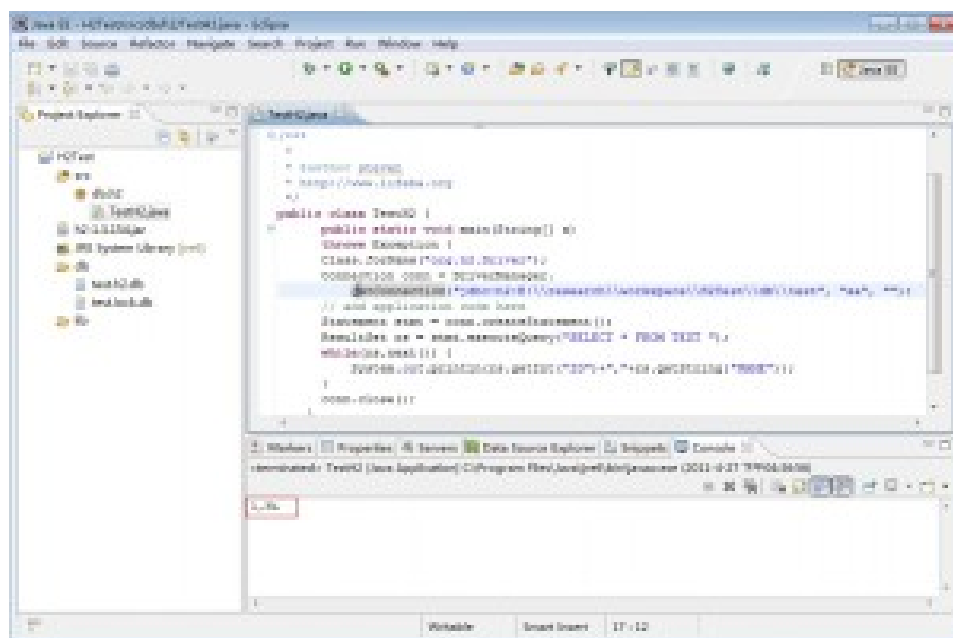
    }

    conn.close();

}

```

控制台打印出：1, Hi



## server模式

1、直接将jdbc url 改为：jdbc:h2:tcp://localhost/~ /test 就行了。因为我们在上面第一步的时候已经在 C:\Users\Administrator创建了test数据库。

你也可以再创建新的数据库，默认都是保存在C:\Users\Administrator下的。

注意：你必须启动服务：%H2\_HOME%\bin\h2.bat 或者 以服务模式启动：%H2\_HOME%\service\0\_run\_server\_debug.bat ，里面有好几个脚本把H2部署为服务模式。每次机器启动后自动启动H2服务。

## 2、新建 TestServerH2类 主要代码

```
public static void main(String[] a)

    throws Exception {

    Class.forName("org.h2.Driver");

    Connection conn = DriverManager.

        getConnection("jdbc:h2:tcp://localhost/~ /test", "sa", "");

    // add application code here

    Statement stmt = conn.createStatement();

    ResultSet rs = stmt.executeQuery("SELECT * FROM TEST ");

    while(rs.next()) {

        System.out.println(rs.getInt("ID")+", "+rs.getString("NAME"));

    }

    conn.close();

}
```

运行的结果和上面一样。

**内存模式**（数据只保存在内存中）

## 1、新建 TestMemH2类 主要代码

```
public static void main(String[] a)

    throws Exception {

    Class.forName("org.h2.Driver");

    Connection conn = DriverManager.

        getConnection("jdbc:h2:tcp://localhost/mem:test2", "sa", "");

    // add application code here

    Statement stmt = conn.createStatement();

    stmt.executeUpdate("CREATE TABLE TEST_MEM(ID INT PRIMARY KEY, NAME VARCHAR(255));");

    stmt.executeUpdate("INSERT INTO TEST_MEM VALUES(1, 'Hello_Mem');");

    ResultSet rs = stmt.executeQuery("SELECT * FROM TEST_MEM");

    while(rs.next()) {
```

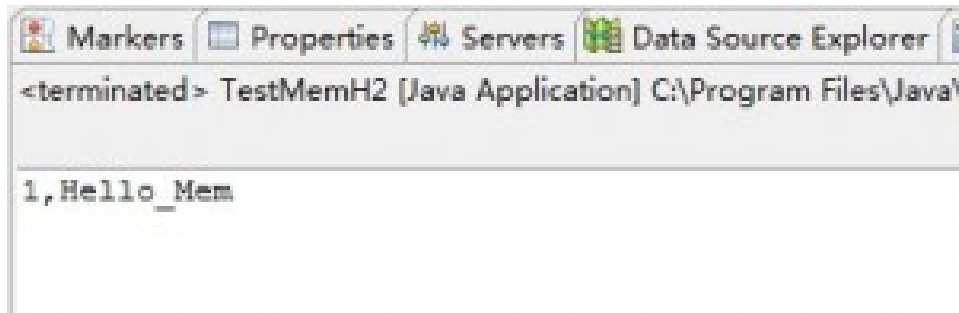
```
System.out.println(rs.getInt("ID")+", "+rs.getString("NAME"));
```

```
}
```

```
conn.close();
```

```
}
```

控制台打印出: 1,Hello\_Mem



## 集群 / 高可用性

数据库支持简单的集群/高可用性机制。架构是：两个数据库服务运行在两台不同的计算机上，两台计算机有同样数据库的副本，如果两个服务器都处于运行状态，每个数据库操作都被在两台计算机上执行，如果一台服务器宕机（断电、硬件故障、网络故障等），另外一台计算机仍能提供服务，从这一刻开始，数据库操作仅在一台服务器上执行，直到另外一台服务器恢复运行。

集群仅能用于服务器模式（内嵌模式并不支持集群）。可以在数据库运行状态下恢复集群，但是要求在第二个数据库恢复期间没有应用在改变第一个数据库的数据，因此恢复集群是一个手工的过程。

初始化集群，使用下面的步骤：

- 创建数据库
- 使用CreateCluster工具创建一个数据库副本到另外的地方，并且初始化集群，这样就得到了同样数据的两个数据库
- 启动两个数据库服务（每个数据库的副本）
- 现在可以通过应用客户端连接到数据库

### 使用创建集群工具

要了解集群如何工作，请尝试下面的例子，在这个例子里，两个数据库驻留在同一台计算机上，但通常，两个数据库在不同的计算机上。

- 创建两个目录：server1, server2。每个目录将模拟一台计算机
- 在第一个目录启动TCP服务，你可以运行下面的命令：

```
java org.h2.tools.Server -tcp-tcpPort 9101 -baseDirserver1
```
- 在第二个目录启动TCP服务，模拟第二个服务器（冗余运行），你能使用下面的命令：

```
java org.h2.tools.Server -tcp-tcpPort 9102 -baseDirserver2
```
- 使用 CreateCluster 工具初始化集群，如果数据库不存在，将创建一个新的空数据库，运行下面命令行：

```
java org.h2.tools.CreateCluster -urlSourcejdbc:h2:tcp://localhost:9101/~ /test
```

- `-urlTarget jdbc:h2:tcp://localhost:9102/~ /test`

- `-user sa`

- `-serverList localhost:9101,localhost:9102`

- 应用或者是H2控制台可以通过下面的JDBC的URL连接数据库：

`jdbc:h2:tcp://localhost:9101,localhost:9102/~ /test`

- 如果你停止一个服务（通过杀进程），你注意到另一个机器继续工作，数据库仍能提供访问。
- 恢复集群，你需要先删掉宕机的数据库，然后重启宕机的数据库的服务，再重新运行CreateCluster集群工具。

### 检测运行状态下的集群

查找哪些节点当前正在运行，通过执行下面的SQL语句：

```
SELECT VALUE FROM INFORMATION_SCHEMA.SETTINGS WHERE NAME='CLUSTER'
```

结果返回为 ''（两个单引号），说明集群模式被屏蔽，否则，集群服务器列表将被单引号包括着返回，如 'server1:9191,server2:9191'。

2、上面的 URL 改为 `jdbc:h2:~/mem:test` 也是可以的。如果是localhost必须启动服务。

---

---

### 数据库连接 URL说明

数据库支持多种连接模式和连接设置，不同的连接模式和连接设置是通过不同的URL来区分的，URL中的设置是不区分大小写。



Topic	URL Format and Examples
<a href="#">嵌入式（本地）连接</a>	jdbc:h2:[file:][ jdbc:h2:~/test jdbc:h2:file:/data/sample jdbc:h2:file:C:/data/sample (Windows only)
<a href="#">内存数据库（私有）</a>	jdbc:h2:mem:
<a href="#">内存数据库（被命名）</a>	jdbc:h2:mem: jdbc:h2:mem:test_mem
<a href="#">使用TCP/IP的服务器模式（远程连接）</a>	jdbc:h2:tcp://[:]/[ jdbc:h2:tcp://localhost/~/test jdbc:h2:tcp://dbserve:8084/~/sample
<a href="#">使用SSL/TLS的服务器模式（远程连接）</a>	jdbc:h2:ssl://[:]/ jdbc:h2:ssl://secureserv:8085/~/sample;
<a href="#">使用加密文件</a>	jdbc:h2::CIPHER=[AES XTEA] jdbc:h2:ssl://secureserv/~/testdb;CIPHER=AES jdbc:h2:file:~/secure;CIPHER=XTEA
<a href="#">文件锁</a>	jdbc:h2::FILE_LOCK={NO FILE SOCKET} jdbc:h2:file:~/quickAndDirty;FILE_LOCK=NO jdbc:h2:file:~/private;CIPHER=XTEA;FILE_LOCK=SOCKET
<a href="#">仅打开存在的数据库</a>	jdbc:h2::IFEXISTS=TRUE jdbc:h2:file:~/sample;IFEXISTS=TRUE
<a href="#">当虚拟机退出时并不关闭数据库</a>	jdbc:h2::DB_CLOSE_ON_EXIT=FALSE
<a href="#">用户名和密码</a>	jdbc:h2:[;USER=][;PASSWORD=] jdbc:h2:file:~/sample;USER=sa;PASSWORD=123
<a href="#">更新记入索引</a>	jdbc:h2::LOG=2 jdbc:h2:file:~/sample;LOG=2
<a href="#">调试跟踪项设置</a>	jdbc:h2::TRACE_LEVEL_FILE= jdbc:h2:file:~/sample;TRACE_LEVEL_FILE=3
<a href="#">忽略位置参数设置</a>	jdbc:h2::IGNORE_UNKNOWN_SETTINGS=TRUE
<a href="#">指定文件读写模式</a>	jdbc:h2::ACCESS_MODE_LOG=rws;ACCESS_MODE_DATA=rws
<a href="#">在Zip文件中的数据库</a>	jdbc:h2:zip:!/ jdbc:h2:zip:~/db.zip!/test
<a href="#">兼容模式</a>	jdbc:h2::MODE= jdbc:h2:~/test;MODE=MYSQL
<a href="#">自动重连接</a>	jdbc:h2::AUTO_RECONNECT=TRUE jdbc:h2:tcp://localhost/~/test;AUTO_RECONNECT=TRUE
<a href="#">自动混合模式</a>	jdbc:h2::AUTO_SERVER=TRUE jdbc:h2:~/test;AUTO_SERVER=TRUE
<a href="#">更改其他设置</a>	jdbc:h2::=[;=...] jdbc:h2:file:~/sample;TRACE_LEVEL_SYSTEM_OUT=3