

Derby 数据库使用简介

Derby 数据库是由 Apache 下的一个开源的 DB 项目，与其它 DB 相比，它有如下的优点：

1. 完全用 Java 实现
2. 支持 JDBC 及 SQL 标准
3. 全部文件只有 2M 左右
4. 提供嵌入式的支持，可以简单地嵌入到其它的 Java 项目
5. 同时提供网络数据库的功能，可以用 C/S 方式访问

本篇将对 Derby 的使用方式进行简单的介绍：

一、下载

Derby 项目的首页为：<http://db.apache.org/derby/index.html>，大家可以在该网站找到下载路径及相关文档，现在的最新版本为 10.5.3.0。

二、安装

Derby 下载文件为一个 Zip 压缩档，只需将该压缩档解压到一个目录(这里用 DERBY_HOME 代替)即可。

三、本地访问

Derby 可以作为一个本地数据库使用，这点就像是 Access，下面将介绍如何使用 Derby 提供的 ij 工具来建立一个新的 DB。

新建一个 demo 目录，在 demo 目录下运行：

```
DERBY_HOME\bin\ij
```

然后输入

```
connect 'jdbc:derby:myDemo;create=true';
```

建立一个新的 DB 名字为 myDemo，保存在当前目录，关于 ij 的用法可以输入 help;查看相关帮助。

四、网络访问

在 demo 目录下，运行

```
java -jar DERBY_HOME\lib\derbyrun.jar server start
```

这时，Derby 服务器已启动成功，默认的端口为 1527，打开另外一个 Console，运行

```
ij
connect 'jdbc:derby://localhost:1527/myDemo';
```

则可成功的连接到刚才建立的 myDemo 数据库，为便于下面测试，运行以下语句新建一个 test 表，并插入两笔数据。

```
create table test(id int, name varchar(20));
insert into test values(0, 'Jack');
insert into test values(1, 'Tom');
```

五、Java 代码访问

新建 Java 文件，代码如下：

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class DerbyDemo {

    static String EMBEDDED_DRIVER = "org.apache.derby.jdbc.EmbeddedDriver";
    static String NET_DRIVER = "org.apache.derby.jdbc.ClientDriver";

    static String DB_NAME = "myDemo";
    static String HOST_URL = "localhost:1527/";

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
```

```

    try {
        conn = getNetwrokConnect();
        stmt = conn.createStatement();
        rs = stmt.executeQuery("select * from test");
        while (rs != null && rs.next()) {
            System.out.println(rs.getString(2));
        }
    } catch (Exception ex) {
        System.out.println(ex.getMessage());
    } finally {
        try {
            rs.close();
        } catch (Exception e) {
        }
        rs = null;
        try {
            stmt.close();
        } catch (Exception e) {
        }
        stmt = null;
        try {
            conn.close();
        } catch (Exception e) {
        }
        conn = null;
    }
}

public static Connection getEmbeddedConnect() {
    try {
        Class.forName(EMBEDED_DRIVER);
    } catch (ClassNotFoundException e) {
        // TODO Auto-generated catch block
        throw new RuntimeException(e.getMessage());
    }
    try {
        return DriverManager.getConnection("jdbc:derby:" + DB_NAME);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        throw new RuntimeException(e.getMessage());
    }
}

public static Connection getNetwrokConnect() {

```

```

    try {
        Class.forName(NET_DRIVER);
    } catch (ClassNotFoundException e) {
        // TODO Auto-generated catch block
        throw new RuntimeException(e.getMessage());
    }
    try {
        return DriverManager.getConnection("jdbc:derby:" + HOST_URL
            + DB_NAME);
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        throw new RuntimeException(e.getMessage());
    }
}
}
}

```

将 DerbyDemo 保存到 demo 目录下，运行如下代码编译并运行即可通过网络访问(需按照第四步启动服务器)

```

javac -cp DERBY_HOME\lib\derbyclient.jar DerbyDemo.java
java -cp .;DERBY_HOME\lib\derbyclient.jar DerbyDemo

```

使用嵌入式访问只需将 main 中的 getNetwrokConnect 改为 getEmbededConnect 即可。