

1. Conexión JDBC con Sqlite

Driver a incluir en el Build Path: sqlite-jdbc-3.8.11.2.jar

Nombre del driver para la conexión a la BD: org.sqlite.JDBC

Para facilitar el poner la URL copio el fichero de la BD Sqlite a un directorio con una ruta más corta, en este caso D:\BBDD\ud2_sqlite.db

```
UD2_JDBC_Sqlite.java
// Conexión a BD Sqlite
// Paso 1. Importar las clases necesarias
import java.sql.*;

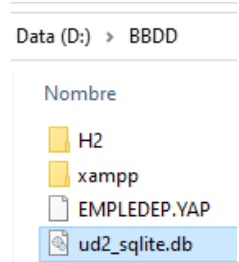
public class UD2_JDBC_Sqlite {

    public static void main(String[] args) {

        try {
            // Paso 2. Carga el driver JDBC
            Class.forName("org.sqlite.JDBC");

            // Paso 3. Identifico el origen de datos
            String url = "jdbc:sqlite:D:/BBDD/ud2_sqlite.db";

            // Paso 4. Crea objeto Connection
            Connection conexion = DriverManager.getConnection(url);
        }
    }
}
```



2. Conexión JDBC con Apache Derby

Driver a incluir en el Build Path: derby.jar

Nombre del driver para la conexión a la BD: org.apache.derby.jdbc.EmbeddedDriver

Para facilitar el poner la URL copio el fichero de la BD Apache Derby a un directorio con una ruta más corta, en este caso D:\BBDD\ud2_derby.db

```
UD2_JDBC_ApacheDerby.java
// Conexión a BD Apache Derby
// Paso 1. Importar las clases necesarias
import java.sql.*;

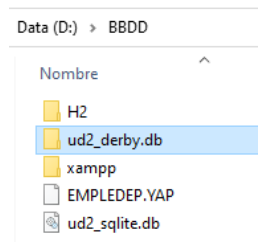
public class UD2_JDBC_ApacheDerby {

    public static void main(String[] args) {

        try {
            // Paso 2. Carga el driver JDBC
            Class.forName("org.apache.derby.jdbc.EmbeddedDriver");

            // Paso 3. Identifico el origen de datos
            String url = "jdbc:derby:D:/BBDD/ud2_derby.db";

            // Paso 4. Crea objeto Connection
            Connection conexion = DriverManager.getConnection(url);
        }
    }
}
```

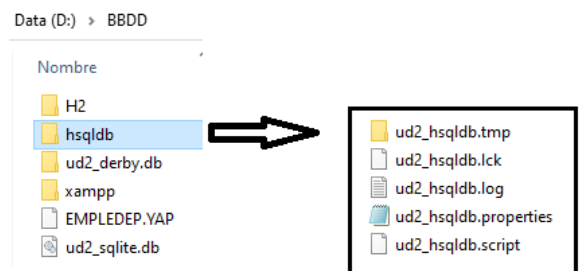


3. Conexión JDBC con HSQLDB

Driver a incluir en el Build Path: hsqldb.jar

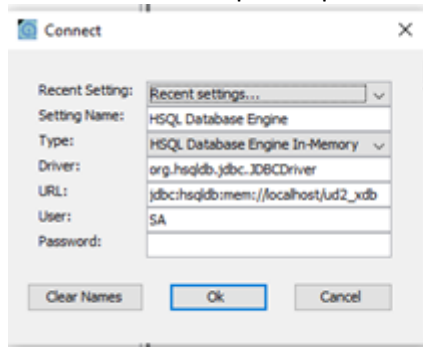
Nombre del driver para la conexión a la BD: org.hsqldb.jdbcDriver

Para facilitar el poner la URL copio toda la carpeta de la BD HSQLDB, que incluye los diferentes ficheros de la BD, a un directorio con una ruta más corta, en este caso D:\BBDD\hsqldb. Dentro de ese directorio está la BD ud2_hsqldb formada por varios ficheros.



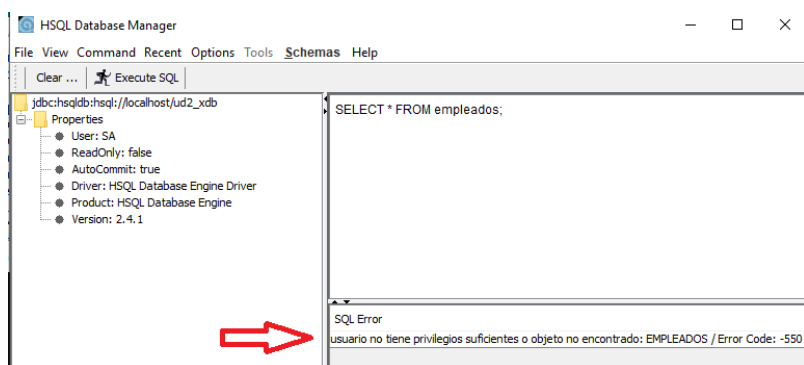
Trato de conectarme con el proyecto Java, pero me da el error de que el objeto (departamentos) no existe o no tengo privilegios suficientes. Para comprobar que los datos están en la BD arranco el servidor, ya que la BD hsqldb la cree en

modo servidor, pero sigue apareciendo el mismo error (el problema no es que esté parado el servidor). Entonces abro HSQL Database Manager para conectarme a la BD y me doy cuenta de que en su momento la conexión que hice fue en modo memoria por lo que los datos no se guardaron en disco.

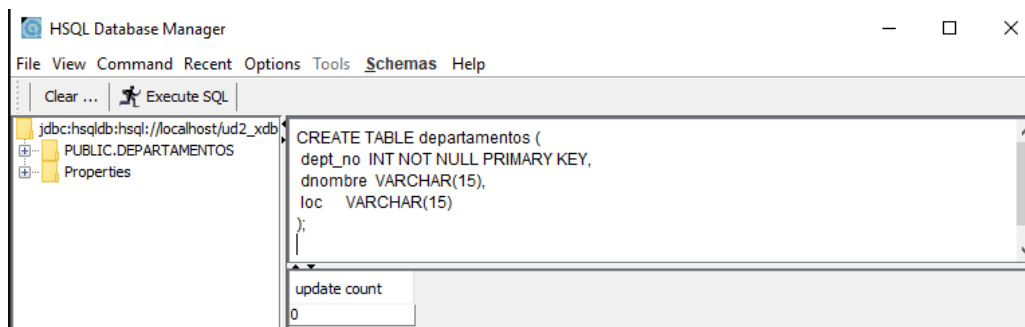


La conexión se hizo en modo memoria por lo que los datos no se guardaron.

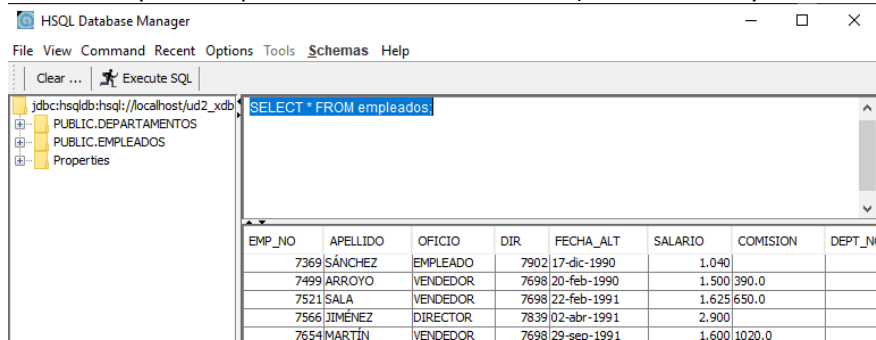
Lo compruebo desde HSQL Database Manager y efectivamente, los datos no están en la BD. Veo el mismo error que me aparecía desde el proyecto Java.



Vuelvo a crear las tablas (empleados y departamentos) y le inserto los valores desde la consola.



Ahora compruebo que los datos están en la BD (veo las tablas y consulto los valores)



Inicio el servidor indicando la ubicación y nombre de la BD

```

Simbolo del sistema - java -classpath lib/hsqldb.jar org.hsqldb.server.Server --database.0 file:D:/BBDD/hsqldb/ud2_hsqldb --dbname.0 ud2_xdb

>java -classpath lib/hsqldb.jar org.hsqldb.server.Server --database.0 file:D:/BBDD/hsqldb/ud2_hsqldb --dbname.0 ud2_xdb
[Server@af3868]: Startup sequence initiated from main() method
[Server@af3868]: Loaded properties from [D:\ALBERTO\Curso 2021-22\486-Acceso a datos\miAccesoDatos\UD2 - Manejo de conectores\SW_BBDD_Embebidas\hsqldb-2.4.1\hsqldb\server.properties]
[Server@af3868]: Initiating startup sequence...
[Server@af3868]: Server socket opened successfully in 8 ms.
[Server@af3868]: Database [index=0, id=0, db=file:D:/BBDD/hsqldb/ud2_hsqldb, alias=ud2_xdb] opened successfully in 296 ms.
[Server@af3868]: Startup sequence completed in 305 ms.
[Server@af3868]: 2021-10-31 11:20:04.109 HSQLDB server 2.4.1 is online on port 9001
[Server@af3868]: To close normally, connect and execute SHUTDOWN SQL
[Server@af3868]: From command line, use [Ctrl]+[C] to abort abruptly

```

Pruebo a conectarme desde la aplicación con la JDBC URL `"jdbc:hsqldb:hsq1://localhost/ud2_xdb"` pero me aparece error que puedo ver también en el cmd (se debe a la versión del driver que estoy empleando. Estaba usando el archivo hsqldb.jar que estaba en los recursos que es versión 2.3.3 y estoy usando versión de BD 2.4.1

```

Simbolo del sistema - java -classpath lib/hsqldb.jar org.hsqldb.server.Server --database.0 file:D:/BBDD/hsqldb/ud2_hsqldb --dbname.0 ud2_xdb

>java -classpath lib/hsqldb.jar org.hsqldb.server.Server --database.0 file:D:/BBDD/hsqldb/ud2_hsqldb --dbname.0 ud2_xdb
[Server@af3868]: Startup sequence initiated from main() method
[Server@af3868]: Loaded properties from [D:\ALBERTO\Curso 2021-22\486-Acceso a datos\miAccesoDatos\UD2 - Manejo de conectores\SW_BBDD_Embebidas\hsqldb-2.4.1\hsqldb\server.properties]
[Server@af3868]: Initiating startup sequence...
[Server@af3868]: Server socket opened successfully in 8 ms.
[Server@af3868]: Database [index=0, id=0, db=file:D:/BBDD/hsqldb/ud2_hsqldb, alias=ud2_xdb] opened successfully in 296 ms.
[Server@af3868]: Startup sequence completed in 305 ms.
[Server@af3868]: 2021-10-31 11:20:04.109 HSQLDB server 2.4.1 is online on port 9001
[Server@af3868]: To close normally, connect and execute SHUTDOWN SQL
[Server@af3868]: From command line, use [Ctrl]+[C] to abort abruptly
org.hsqldb.HsqlException: Se requiere una versión de protocolo de red HSQLDB '2.4.1' La versión del cliente '2.1.0.0' es incompatible.
    at org.hsqldb.error.Error.error(Unknown Source)
    at org.hsqldb.server.ServerConnection.init(Unknown Source)
    at org.hsqldb.server.ServerConnection.run(Unknown Source)
    at java.lang.Thread.run(Unknown Source)

```

Cambio el fichero jar en el proyecto por el que viene en la carpeta hsqldb 2.4.1 y vuelvo a probar desde el proyecto Java.

```

UD2_JDBC_HSqlDb.java
1 // Conexión a BD HSQLDB
4
5 import java.sql.*;
6
7 public class UD2_JDBC_HSqlDb {
8
9     public static void main(String[] args) {
10
11         try {
12             // Paso 2. Carga el driver JDBC
13             Class.forName("org.hsqldb.jdbcDriver");
14
15             // Paso 3. Identifico el origen de datos
16             // String url = "jdbc:hsqldb:file:D:/BBDD/hsqldb/ud2_xdb";
17             String url = "jdbc:hsqldb:hsq1://localhost/ud2_xdb";
18             String usuario = "SA";
19             String passwd = "";
20
21             // Paso 4. Crea objeto Connection
22             Connection conexion = DriverManager.getConnection(url, usuario, passwd);
23
24         } catch (Exception e) {
25             e.printStackTrace();
26         }
27     }
28 }

```

Ahora modifico el programa para ver si me puedo conectar a la BD HSqlDb, pero en lugar de en modo Servidor, hacerlo en modo Stand-alone (dedicado o aislado). Para ello se cambia el JDBC URL indicando que apunte al fichero concreto de la BD (ud2_hsqldb), en lugar de al nombre de la base de datos (ud2_xdb). Hay que tener en cuenta que para que la BD funcione en modo stand-alone, el servidor debe estar parado. En caso contrario da porque no puede bloquear la BD **Fallo al obtener un bloqueo en la base de datos:**

Compruebo la conexión desde el proyecto y funciona, aunque en la consola también aparecen los logs de la BD. Estos logs los puedo quitar indicándolo en la BD o a través de Java.

<https://stackoverflow.com/questions/9931156/is-there-a-way-to-silence-hsqldb-logging>

```
UD2_JDBC_Hsqldb.java
4
5 import java.sql.*;
6
7 public class UD2_JDBC_Hsqldb {
8
9     public static void main(String[] args) {
10
11         try {
12             // Paso 2. Carga el driver JDBC
13             Class.forName("org.hsqldb.jdbcDriver");
14
15             // Paso 3. Identifico el origen de datos
16             String url = "jdbc:hsqldb:file:D:/B800/hsqldb/ud2_hsqldb"; // conexión en modo stand-alone (ud2_hsqldb)
17             // String url = "jdbc:hsqldb:hsqldb://localhost/ud2_xdb"; // conexión en modo server (ud2_xdb)
18
19             String usuario = "SA";
20             String passwd = "";
21
22             // Paso 4. Crea objeto Connection
23             Connection conexion = DriverManager.getConnection(url, usuario, passwd);
24
25         } catch (Exception e) {
26             e.printStackTrace();
27         }
28     }
29 }
```

@ Javadoc Declaration Console

```
<terminated> UD2_JDBC_Hsqldb [Java Application] C:\Users\JoséAlberto\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149
oct 31, 2021 11:58:46 A. M. org.hsqldb.persist.Logger logInfoEvent
INFO: checkpointClose start
oct 31, 2021 11:58:46 A. M. org.hsqldb.persist.Logger logInfoEvent
INFO: checkpointClose synched
oct 31, 2021 11:58:46 A. M. org.hsqldb.persist.Logger logInfoEvent
INFO: checkpointClose script done
oct 31, 2021 11:58:46 A. M. org.hsqldb.persist.Logger logInfoEvent
INFO: checkpointClose end
10, CONTABILIDAD, SEVILLA
20, INVESTIGACIÓN, MADRID
30, VENTAS, BARCELONA
40, PRODUCCIÓN, BILBAO
```

4. Conexión JDBC con H2

Driver a incluir en el Build Path: h2-1.4.191.jar

Nombre del driver para la conexión a la BD: org.h2.Driver

La ubicación de la BD es en el directorio home de mi usuario y de nombre test, es decir ~/test, con el usuario sa y sin password. Por ello en la jdbc url pondremos jdbc:h2: ~/test como hicimos desde la consola para acceder a la BD H2.

Nota: virgulilla (~) en un teclado español corresponde a la tecla ALT Gr + 4

Login

Saved Settings: Generic H2 (Embedded)

Setting Name: Generic H2 (Embedded) Save Remove

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:~/test

User Name: sa

Password:

Connect Test Connection

Connecting to an Embedded (Local) Database

The database URL for connecting to a local database is `jdbc:h2:[file:][<path>]<databaseName>`. The prefix `file:` is optional. If no or only a relative path is used, then the current working directory is used as a starting point. The case sensitivity of the path and database name depend on the operating system, however it is recommended to use lowercase letters only. The database name must be at least three characters long (a limitation of `File.createTempFile`). The database name must not contain a semicolon. To point to the user home directory, use `~/`, as in: `jdbc:h2:~/test`.

```
UD2_JDBC_H2.java
// Conexión a BD H2

// Paso 1. Importar las clases necesarias
import java.sql.*;

public class UD2_JDBC_H2 {

    public static void main(String[] args) {

        try {
            // Paso 2. Carga el driver JDBC
            Class.forName("org.h2.Driver");

            // Paso 3. Identifico el origen de datos
            String url = "jdbc:h2:~/test"; // Equivalente a url = "jdbc:h2:file:C:/Users/JoséAlberto/test";
            String usuario = "sa";
            String passwd = "";

            // Paso 4. Crea objeto Connection
            Connection conexion = DriverManager.getConnection(url, usuario, passwd);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```