

How to query HyperSQL database from C#

[HSQLDB](#) (HyperSQL DataBase) is the SQL relational database software written in Java.

It offers a small, multithreaded and transactional database engine with in-memory and disk-based tables and supports embedded and server modes.

There are 2 alternatives to query HSQLDB from a .NET application:

- You can use an [ODBC to JDBC Bridge driver](#).

Downside: all driver implementations cost a lot of money.

- The stingy ones, including me, prefer another solution: *IKVM.NET*.

[IKVM.NET](#) includes a tool that translates Java bytecode (JAR files) to .NET IL (DLLs or EXE files).

We will take an HSQLDB java driver and convert it using IKVM.NET to an NET dll. The .NET dll does not require Java runtime.

Starting HSQLDB in server mode

If you have an existing HSQLDB instance running you can skip this chapter.

But let's suppose you don't have a running HSQLDB server in your development environment.

In this chapter we learn how to launch a HSQLDB server instance.

Later we will create a C# project to connect to the database instance and query the database.

Requirements: installed Java Runtime

[Download HSQLDB driver](#)

Extract the zip file in a folder. Let's call this folder *current folder*.

Create new sub-folder database in current folder.

You should now have 2 folders in current folder:

```
database
hsqldb-2.4.1
```

In cmd.exe change to this folder: `cd database`

To start database in server mode type in cmd.exe:

```
java -cp ..\hsqldb-2.4.1\hsqldb\lib\hsqldb.jar org.hsqldb.server.Server --database.0 file:mydb --dbname.0 xdb --port 9999
```

Here my cmd.exe:

```
d:\temp\HSQLDB-to-.NET\database>java -cp ..\hsqldb-2.4.1\hsqldb\lib\hsqldb.jar org.hsqldb.server.Server --database.0 file:mydb --dbname.0 xdb --port 9999
[Server@ea2f77]: Startup sequence initiated from main() method
[Server@ea2f77]: Could not load properties from file
[Server@ea2f77]: Using cli/default properties only
[Server@ea2f77]: Initiating startup sequence...
[Server@ea2f77]: Server socket opened successfully in 4 ms.
[Server@ea2f77]: Database [index=0, id=0, db=file:mydb, alias=xdb] opened successfully in 422 ms.
[Server@ea2f77]: Startup sequence completed in 428 ms.
[Server@ea2f77]: 2018-06-18 10:42:10.427 HSQLDB server 2.4.1 is online on port 9999
[Server@ea2f77]: To close normally, connect and execute SHUTDOWN SQL
[Server@ea2f77]: From command line, use [Ctrl]+[C] to abort abruptly
```

HSQLDB server is now running.

hsqldb.jar command line parameter:

- file - database file name on disk. HSQLDB creates a few files with base name mydb.* with these extensions: *.lock, *.data, *.properties, *.script
- dbname.0 - alias name to use in Connection String.

The database listens by default on port 9001 for new connections. We specified the port 9999 on the command line.

HSQLDB created in database folder a few files:

```
mydb.lock
mydb.log
mydb.properties
mydb.script
```

mydb.tmp

Don't touch this files, they are managed by the HSQLDB engine.

Manually checking connection to database

To check the connection we will start *HSQL Database Manager*

The jar file hsqldb.jar contains a JDBC driver and a GUI query tool at the same time. Nice!

To start *HSQL Database Manager* double-click on hsqldb-2.4.1\hsqldb\lib\hsqldb.jar.

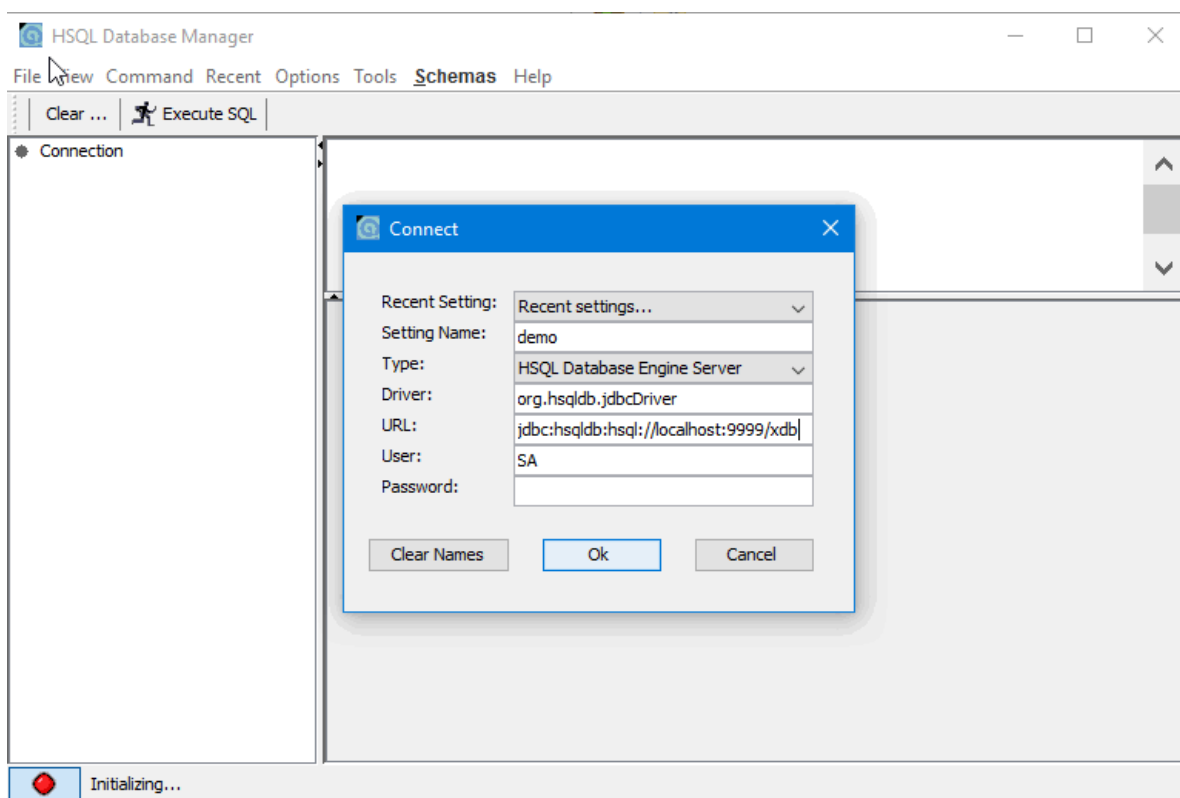
The 'Connect' dialog of the *HSQL Database Manager* should pop up.

Type in this connection string in field URL: jdbc:hsqldb:hsq://localhost:9999/xd

Choose in field Type: HSQL Database Engine Server

Let field Password empty

Click on OK



The database is empty

Let's create a table

Type in this SQL in the top text box:

```
CREATE TABLE Customer (id INT NOT NULL, name VARCHAR(100) NOT NULL, age INT NOT NULL, address VARCHAR(20), Salary INT, PRIMARY KEY (id));
```

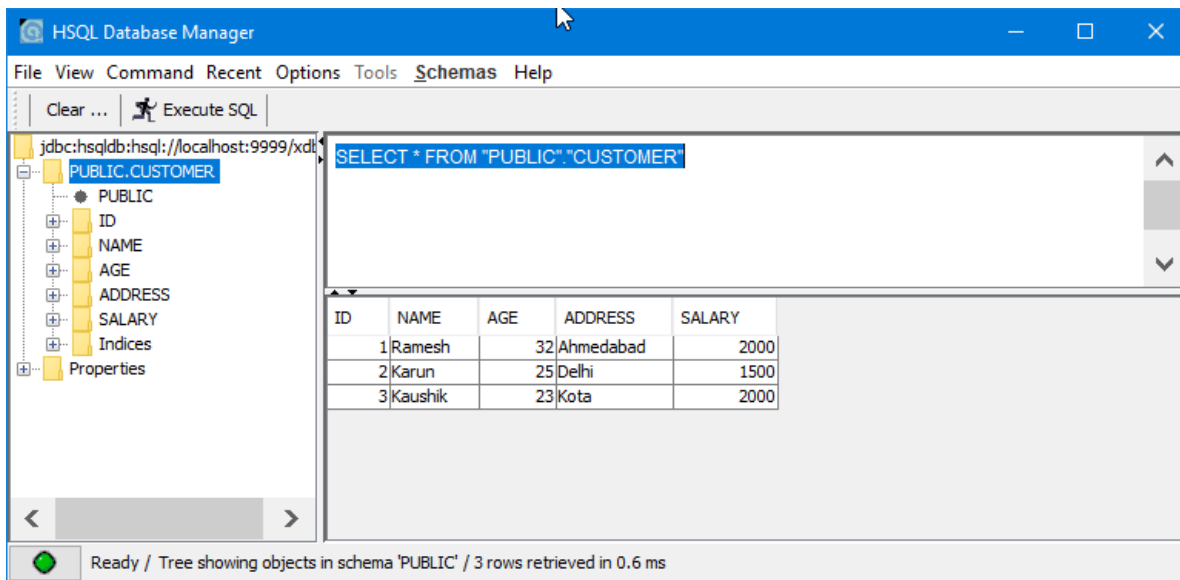
Click on Execute SQL

Let's put some data to table 'demo':

```
Insert into Customer values (1, 'Ramesh', 32, 'Ahmedabad', 2000);  
Insert into Customer values (2, 'Karun', 25, 'Delhi', 1500);  
Insert into Customer values (3, 'Kaushik', 23, 'Kota', 2000);
```

Click on Execute SQL

At last let's select all rows from table 'demo':



At this point we are sure that HyperSQL is working.

Convert HSQLDB Java driver to .NET DLL

1. [Download IKVM.NET](http://www.frijters.net/ikvmbin-8.1.5717.0.zip) <http://www.frijters.net/ikvmbin-8.1.5717.0.zip>
2. Unzip the downloaded file. Your current folder should have these 2 folders:

```
hsqldb-2.4.1\
ikvm-8.1.5717.0\
```

3. Open cmd.exe in the current folder and type:
`ikvm-8.1.5717.0\bin\ikvmc.exe -target:library -version:2.4.1 hsqldb-2.4.1\hsqldb\lib\hsqldb.jar`

Here how it looks in my cmd.exe:

```
sdf
```

```
d:\temp\HSQLDB-to-.NET>ikvm-8.1.5717.0\bin\ikvmc.exe -target:library
-version:2.4.1 hsqldb-2.4.1\hsqldb\lib\hsqldb.jar
IKVM.NET Compiler version 8.1.5717.0
Copyright (C) 2002-2015 Jeroen Frijters
http://www.ikvm.net/
```

```
note IKVMC0002: Output file is "hsqldb.dll"
warning IKVMC0100: Class "javax.servlet.http.HttpServlet" not found
warning IKVMC0100: Class "org.hsqldb.util.Transfer" not found
```

Ignore the warnings.

4. The generated file hsqldb.dll should be now in the current folder:

```
d:\temp\HSQLDB-to-.NET>dir /b
hsqldb-2.4.1
hsqldb.dll
ikvm-8.1.5717.0
```

C# Project talks to HyperSQL

In Visual Studio create new *Console App (.NET Framework)*.

Compile time dependencies

In Solution Explorer, References, choose 'Add Reference...', section Browse, click on 'Browse...' browse to current folder and select 3 DLLs:

- hsqldb.dll
- \ikvm-8.1.5717.0\bin\IKVM.OpenJDK.Core.dll
- \ikvm-8.1.5717.0\bin\IKVM.OpenJDK.Jdbc.dll

In Reference Manager, section Assemblies, add a reference to System.Configuration.dll.

Runtime dependencies

Copy these 4 DLLs from \ikvm-8.1.5717.0\bin\ to your project's bin\Debug or bin\Release folders.

- IKVM.OpenJDK.Localedata.dll
- IKVM.OpenJDK.Text.dll

- IKVM.OpenJDK.Util.dll
- IKVM.Runtime.dll

Connection string

Add connectionStrings section and one connection string to app.config:

```
<configuration>
<connectionStrings>
  <add name="HyperSQL"
    connectionString="jdbc:hsqldb:hsqldb://localhost:9999/xdb;user=SA;password=" />
</connectionStrings>
<startup>
  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
</startup>
</configuration>
```

C# Code

In Program.cs add following C# code:

```
using System;
using System.Configuration;

namespace HyperSQL
{
    class Program
    {
        readonly static string CONNECTION_STRING = ConfigurationManager.ConnectionStrings["HyperSQL"].ConnectionString;
        const string SQL = "SELECT * FROM customer";

        static void Main(string[] args)
        {
            java.sql.DriverManager.registerDriver(new org.hsqldb.jdbcDriver());
            using (java.sql.Connection conn = java.sql.DriverManager.getConnection(CONNECTION_STRING))
            {
                java.sql.PreparedStatement ps = conn.prepareStatement(SQL);
                using (java.sql.ResultSet rs = ps.executeQuery())
                {
                    while (rs.next())
                    {
                        Console.WriteLine($"ID={rs.getInt("id")}");
                        Console.WriteLine($"NAME={rs.getString("name")}");
                        Console.WriteLine($"AGE={rs.getInt("age")}");
                        Console.WriteLine($"ADDRESS={rs.getString("address")}");
                        Console.WriteLine($"SALARY={rs.getInt("salary")}");

                        Console.WriteLine("-----");
                    }
                }
            }

            Console.ReadLine();
        }
    }
}
```

[Download Visual Studio project](#) or browse [repository](#).

Start the project

Console output:

```
ID=1
NAME=Ramesh
AGE=32
ADDRESS=Ahmedabad
SALARY=2000
-----
ID=2
NAME=Karun
AGE=25
ADDRESS=Delhi
SALARY=1500
-----
```

ID=3
NAME=Kaushik
AGE=23
ADDRESS=Kota
SALARY=2000

Done. We can talk to HyperSQL. You can try other SQL statement: INSERT, DELETE etc.

Troubleshooting:

If HyperSQL instance is not running or the connection string in app.config is wrong, you get 'connection' errors:

java.sql.SQLException: **java.net.ConnectException**: Connection refused: connect'

If you forget to add *runtime dependencies* to the build folder of your project then you can get these errors:

- Could not load file or assembly 'IKVM.OpenJDK.Util, Version=8.1.5717.0, Culture=neutral, PublicKeyToken=13235d27fcbfff58' or one of its dependencies. The system cannot find the file specified.
- TypeInitializationException: The type initializer for 'org.hsqldb.jdbc.JDBCDriver' threw an exception:
FileNotFoundException: Could not load file or assembly 'IKVM.Runtime, Version=8.1.5717.0, Culture=neutral, PublicKeyToken=13235d27fcbfff58' or one of its dependencies.
- 'The type initializer for 'sun.util.locale.provider.LocaleProviderAdapter' threw an exception.'
FileNotFoundException: Could not load file or assembly 'IKVM.OpenJDK.Text, Version=8.1.5717.0, Culture=neutral, PublicKeyToken=13235d27fcbfff58' or one of its dependencies.
- 'The type initializer for 'org.hsqldb.HsqlDateTime' threw an exception.'
InvalidCastException: Unable to cast object of type 'java.util.PropertyResourceBundle' to type 'sun.util.resources.OpenListResourceBundle'.