

Access Microsoft Sql Server 2005 with Elixir Repertoire 7.x using SQL JDBC driver

Introduction

Java Database Connectivity (JDBC) is a standard SQL database access interface, providing uniform access to a wide range of relational databases. Elixir Repertoire is able to access Microsoft SQL Server 2005 data via JDBC type 4 driver as SQL JDBC Driver.

The prerequisites are

- Elixir Repertoire Designer or Server
- SQL JDBC driver, sqljdbc.jar (Package with your SQL Server 2005)
- Microsoft SQL Server Express 2005 with TCP/IP, port 1433 enabled
- SQL Server Management Studio Express (optional)

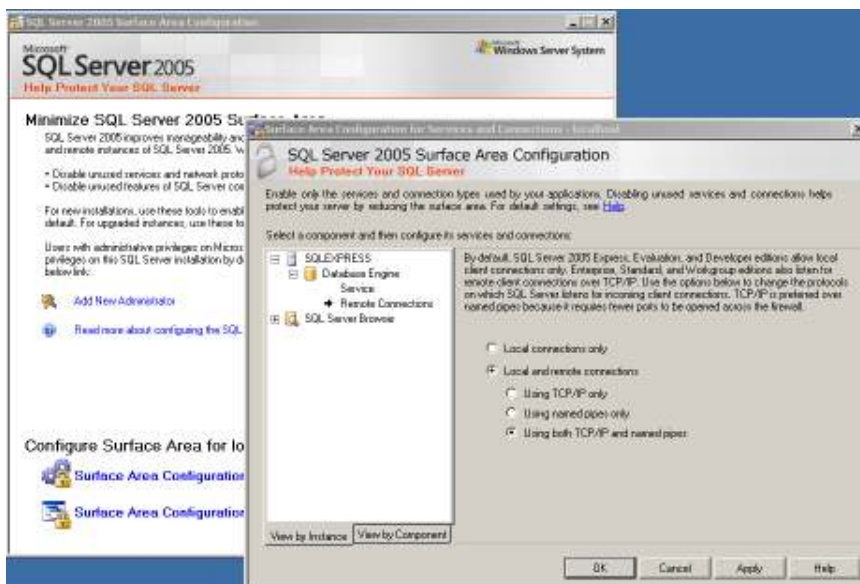
Step by Step

Below is a step by step illustration of how the JDBC data source is defined.

A) Configuring SQL Server (Express Edition)

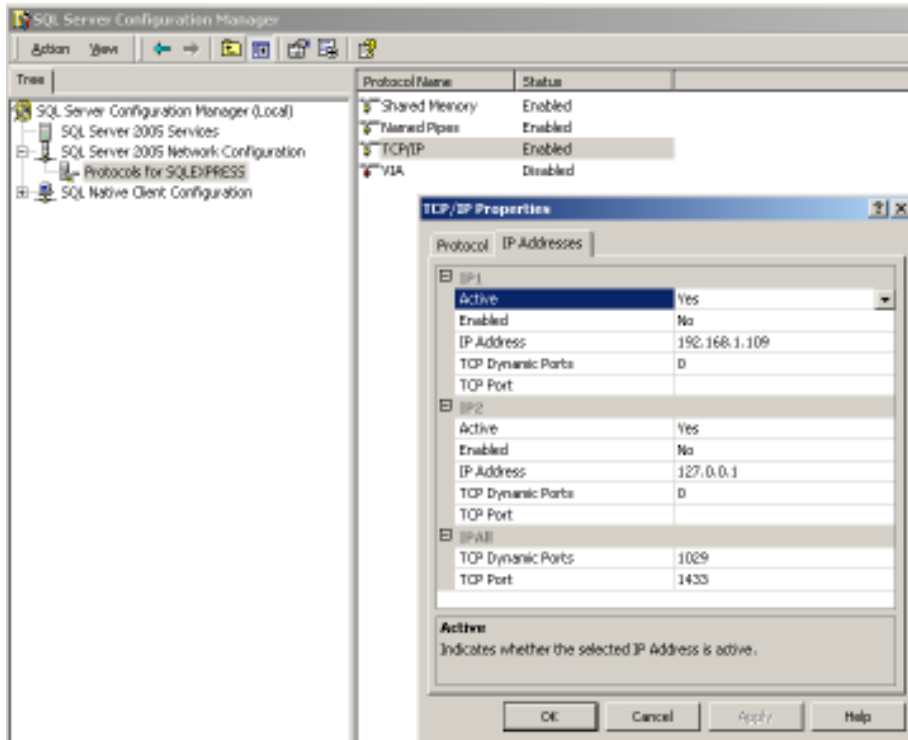
1. Enable TCP/IP connectivity in SQL Server

The SQL Server Express is installed with TCP/IP connection disabled. You can enable TCP/IP connection from the SQL Server Surface Area Configuration.



2. Set the port number, 1433 (Default port)

The port number for SQL Server can be defined, we will use port 1433 for now. It can be set from the SQL Server Configuration Manager at TCP/IP protocol property – TCP Port. Restart the SQL Server.



3. To verify the port access

To verify if the port is available for connection. You may use Telnet to connect to port 1433 in MS command console. The command is

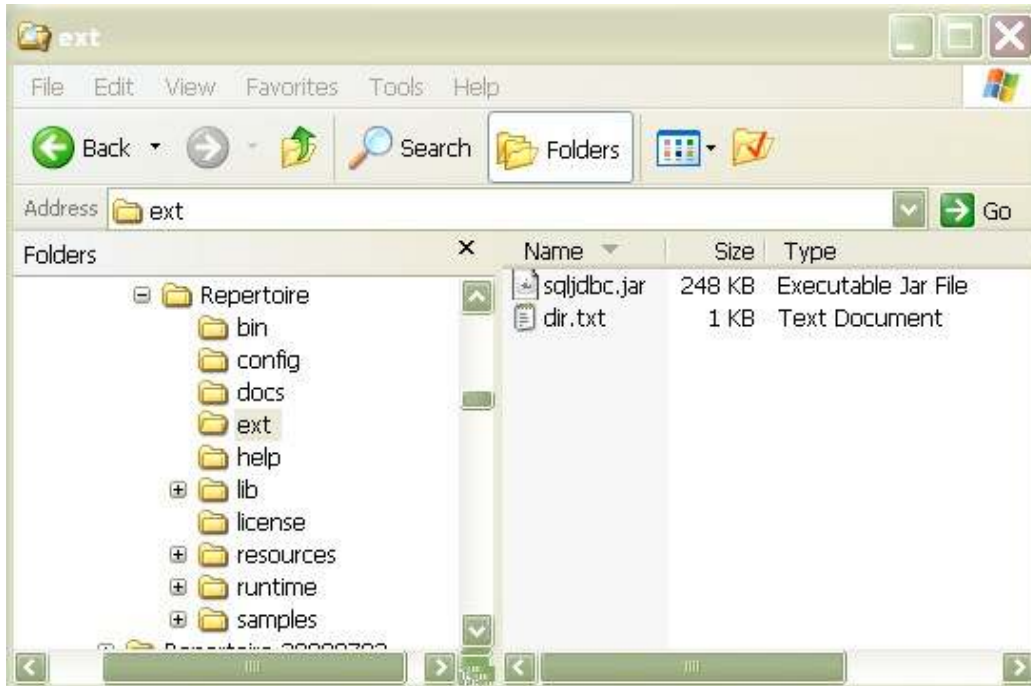
```
telnet localhost 1433
```

4. Authentication

The SQL Server is set to use mixed mode to simplify logon access from the JDBC driver. You may refer to the JDBC driver documentation if you like to use SQL Server NT authentication.

B) Deploying the SQL JDBC Driver in Elixir Repertoire

1. Copy the SQL JDBC Driver, `sqljdbc.jar`, into Elixir Repertoire `ext` directory.



2. Restart Repertoire so that the driver may be loaded into the memory.

C) Defining Elixir Repertoire JDBC data source

The next step is to define the JDBC data source in Elixir Repertoire. The JDBC parameters are follows

- i. JDBC Driver Class Name:

com.microsoft.sqlserver.jdbc.SQLServerDriver

- ii. JDBC Driver URL:

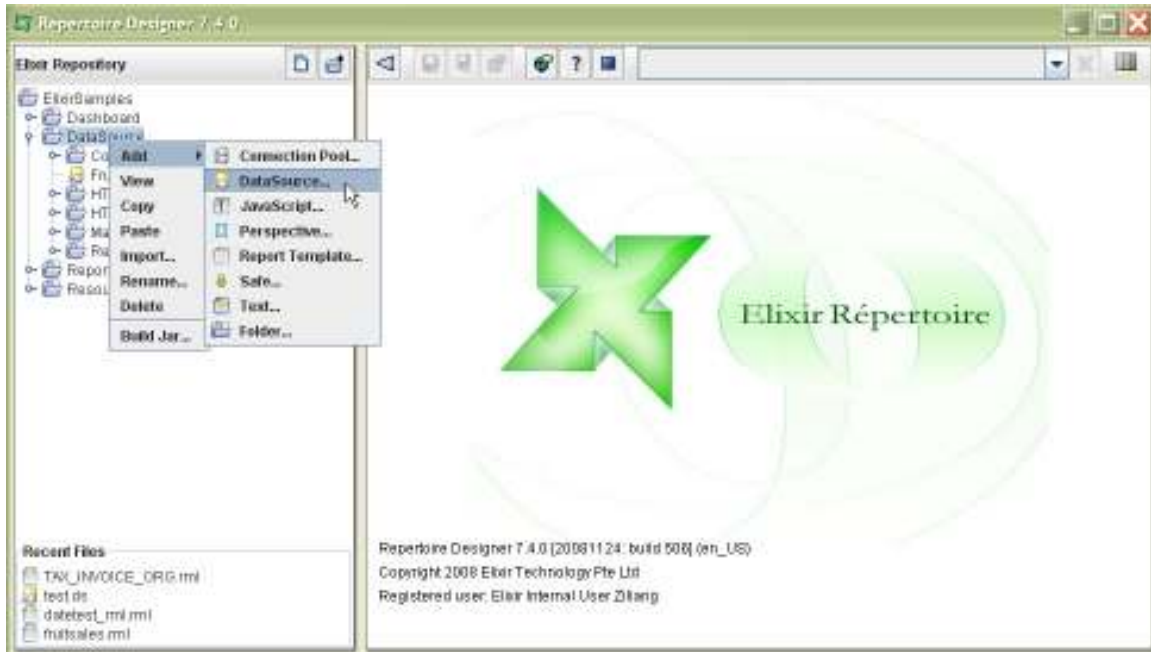
jdbc:sqlserver://serverName:port;DatabaseName=Sample

Example as

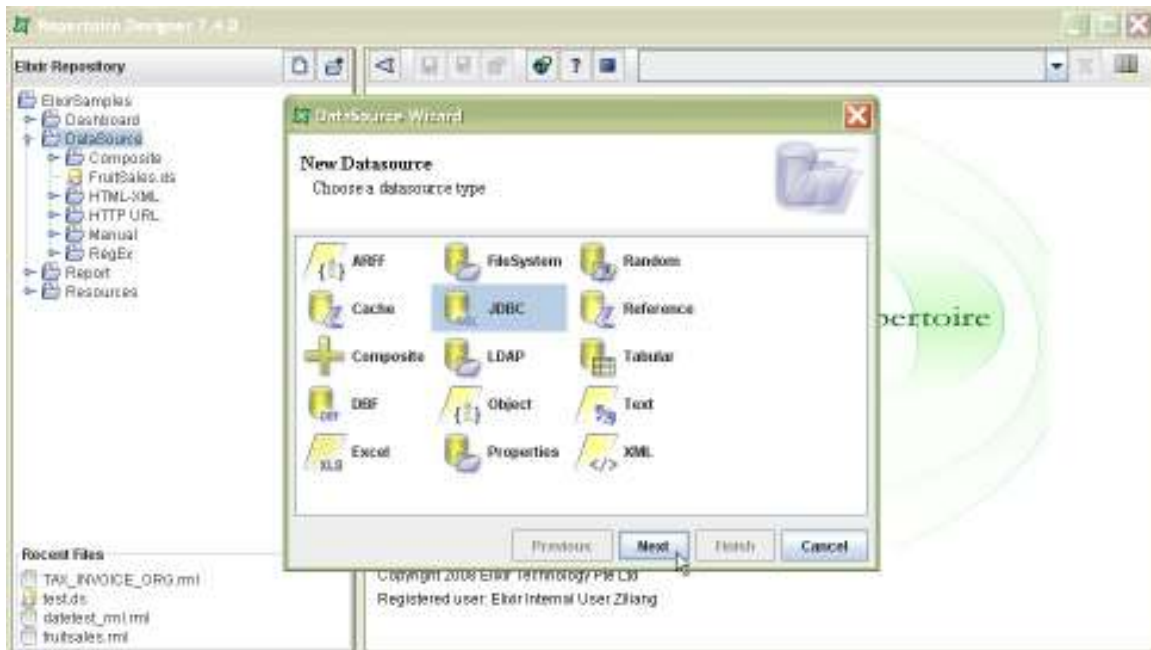
jdbc:sqlserver://localhost:1433;DatabaseName=TestDB

Define JDBC data source

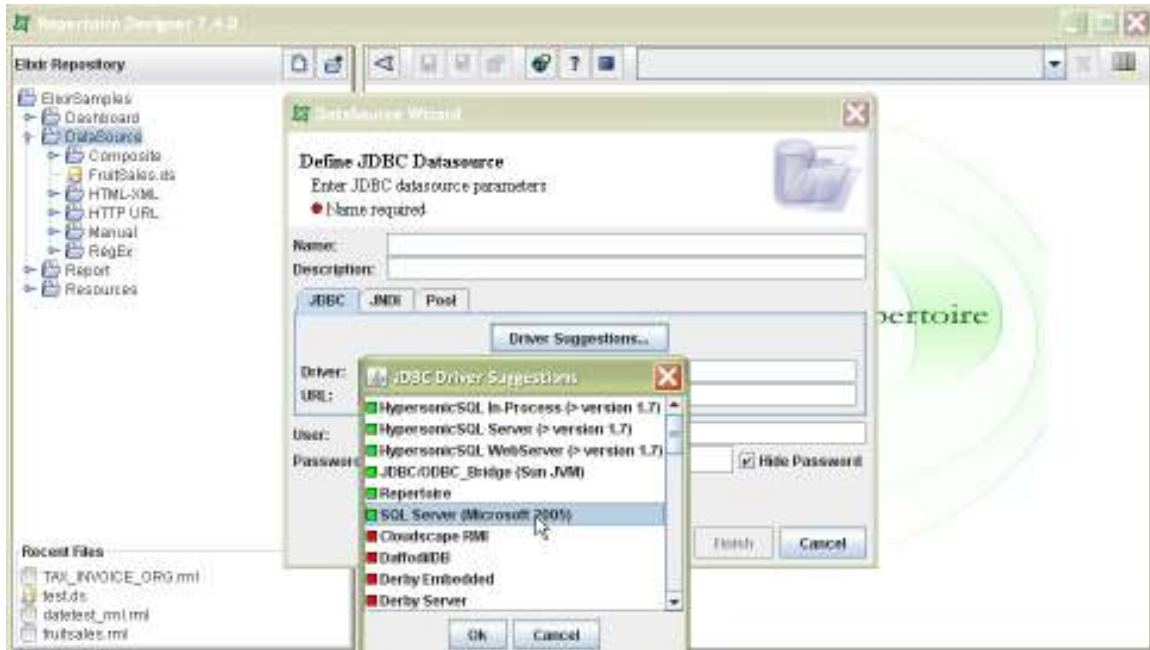
1. Add -> DataSource...



2. Select JDBC Data Source in Data Source Selection Wizard



3. Select the driver available, this is determined by the driver in *ext* directory.



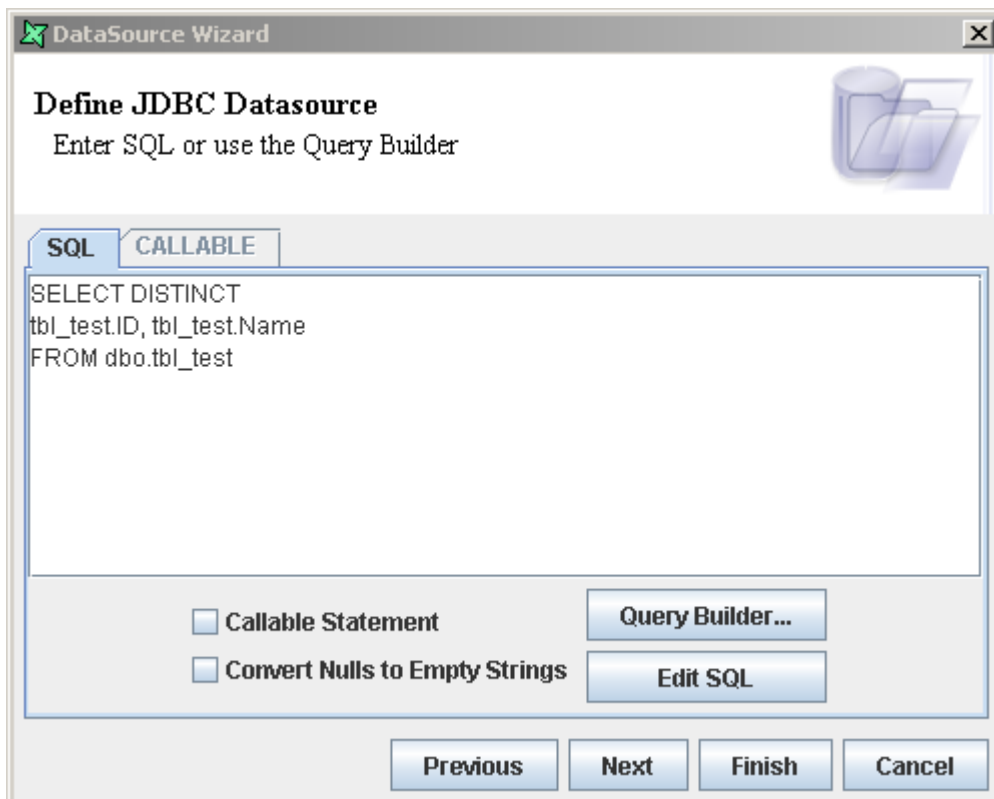
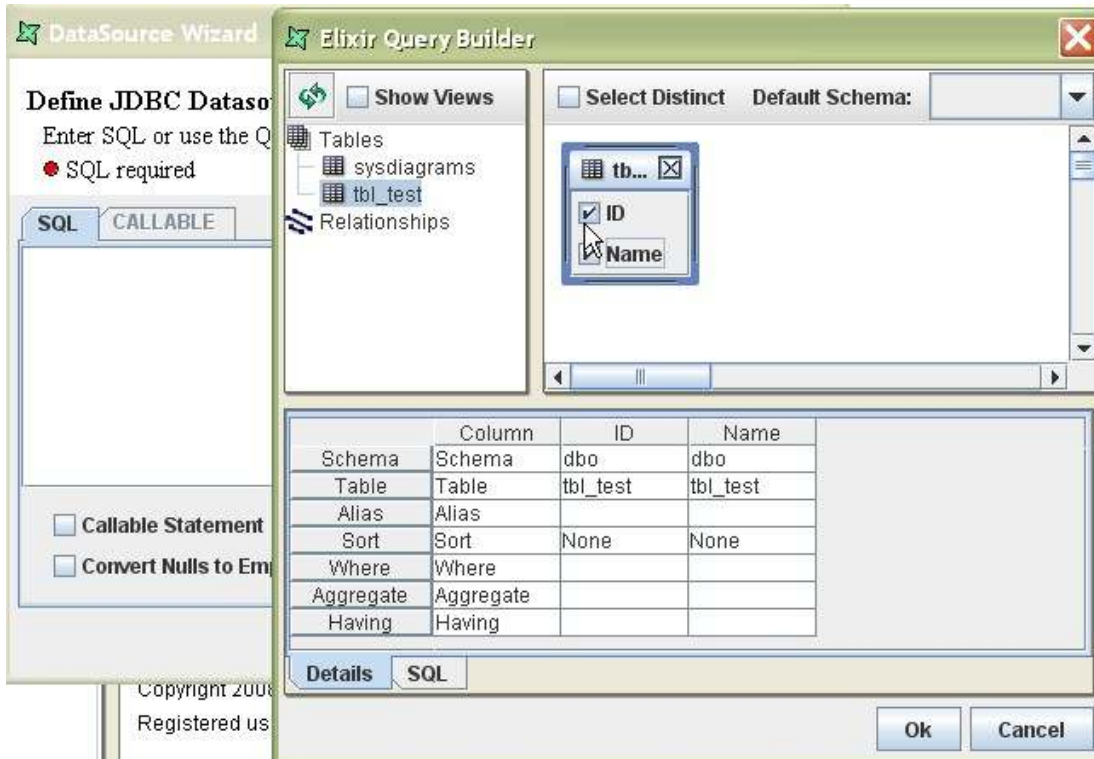
4. Define the JDBC connection properties in the JDBC data source wizard



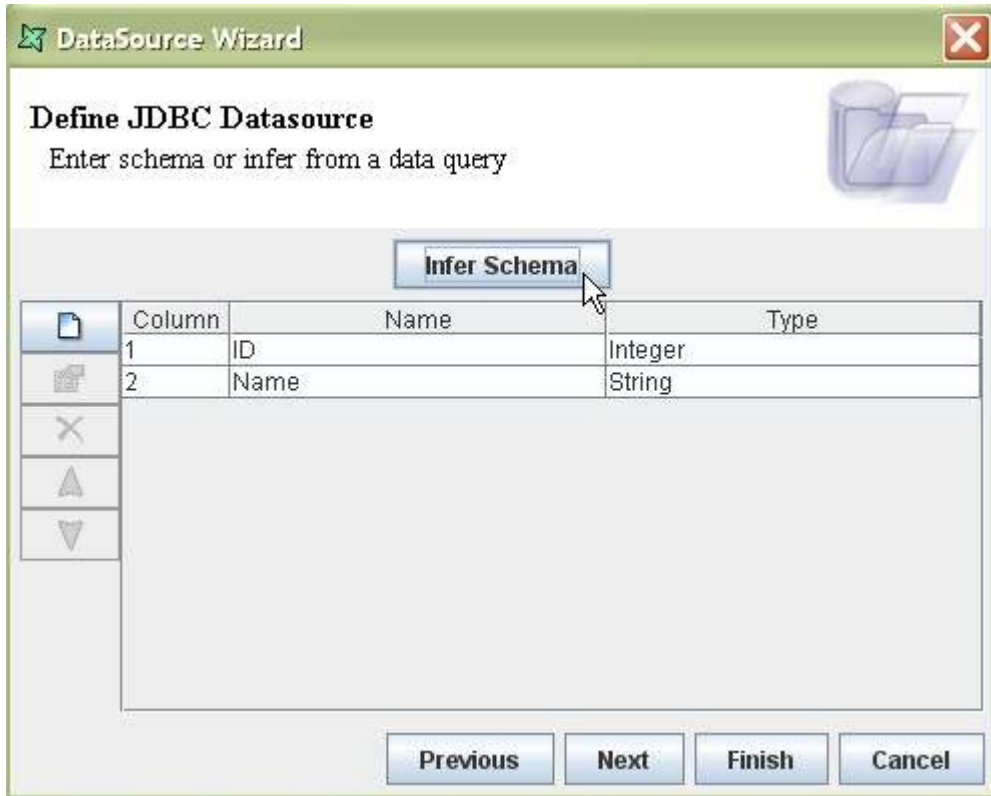
5. Click 'Query Builder' to build SQL query.



6. Select the Table to be used with Elixir Repertoire Query Designer or the SQL statement can be entered directly into the SQL text field.

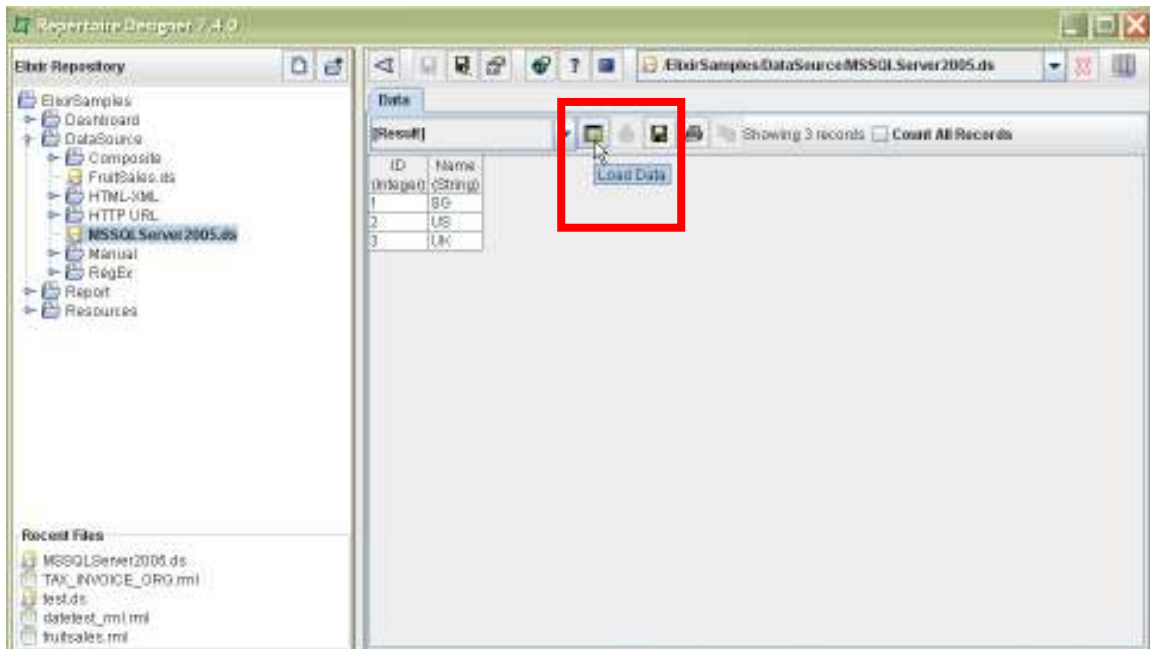


7. Infer schema by click in the Infer Schema button and finish the definition.



Now the field has been bind, you can use this data source to query for the records in the table.

8. Click on Load Data button to test the connectivity and view the data.



FAQs

Q: Why is the driver not showing in the 'JDBC Driver Suggestions' dialog box ?

A: Either this is a incorrect driver or the driver is not found in *ext* directory

Q: Using Elixir Repertoire Server/Designer, the data are able to load but running .jsp with same setting and driver I got the below error.

HTTP ERROR: 500

```
access denied (java.net.SocketPermission localhost:1433 connect,resolve)
```

```
RequestURI=/ReadRecordsfromMSSQL05.jsp
```

Caused by:

```
java.security.AccessControlException: access denied (java.net.SocketPermission localhost:1433 connect,resolve)
    at java.security.AccessControlContext.checkPermission(AccessControlContext.java:323)
    at java.security.AccessController.checkPermission(AccessController.java:546)
    at java.lang.SecurityManager.checkPermission(SecurityManager.java:532)
    at com.elixirtech.arch.ElxSecurityManager.checkPermission(Unknown Source)
    at java.lang.SecurityManager.checkConnect(SecurityManager.java:1034)
    at com.microsoft.sqlserver.jdbc.SQLServerConnectionSecurityManager.checkConnect(Unknown Source)
    at com.microsoft.sqlserver.jdbc.SQLServerConnection.connect(Unknown Source)
    at com.microsoft.sqlserver.jdbc.SQLServerDriver.connect(Unknown Source)
    at java.sql.DriverManager.getConnection(DriverManager.java:582)
    at java.sql.DriverManager.getConnection(DriverManager.java:185)
    at org.apache.jsp.ReadRecordsfromMSSQL05_jsp._jspService(org.apache.jsp.ReadRecordsfromMSSQL05_jsp:61)
    at org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:111)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
    at org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:373)
    at org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:464)
    at org.apache.jasper.servlet.JspServlet.service(JspServlet.java:358)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
    at org.mortbay.jetty.servlet.ServletHolder.handle(ServletHolder.java:491)
    at org.mortbay.jetty.servlet.ServletHandler.handle(ServletHandler.java:367)
    at org.mortbay.jetty.handler.HandlerWrapper.handle(HandlerWrapper.java:139)
    at com.elixirtech.jetty.handler.CallWrapper.handle(Unknown Source)
    at org.mortbay.jetty.handler.HandlerWrapper.handle(HandlerWrapper.java:139)
    at com.elixirtech.jetty.handler.ClientServerSecurityHandler.doHandle(Unknown Source)
    at com.elixirtech.jetty.handler.ClientServerSecurityHandler.handle(Unknown Source)
    at org.mortbay.jetty.servlet.SessionHandler.handle(SessionHandler.java:181)
    at org.mortbay.jetty.handler.ContextHandler.handle(ContextHandler.java:689)
    at org.mortbay.jetty.servlet.Dispatcher.forward(Dispatcher.java:224)
    at org.mortbay.jetty.servlet.Dispatcher.forward(Dispatcher.java:126)
    at com.elixirtech.web.Controller.forwardToJSP(Unknown Source)
    at com.elixirtech.web.Controller.doGet(Unknown Source)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:707)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
    at org.mortbay.jetty.servlet.ServletHolder.handle(ServletHolder.java:491)
    at org.mortbay.jetty.servlet.ServletHandler.handle(ServletHandler.java:367)
    at org.mortbay.jetty.handler.HandlerWrapper.handle(HandlerWrapper.java:139)
    at com.elixirtech.jetty.handler.CallWrapper.handle(Unknown Source)
    at org.mortbay.jetty.handler.HandlerWrapper.handle(HandlerWrapper.java:139)
    at com.elixirtech.jetty.handler.ClientServerSecurityHandler.doHandle(Unknown Source)
    at com.elixirtech.jetty.handler.ClientServerSecurityHandler.handle(Unknown Source)
    at org.mortbay.jetty.servlet.SessionHandler.handle(SessionHandler.java:181)
    at org.mortbay.jetty.handler.ContextHandler.handle(ContextHandler.java:689)
    at org.mortbay.jetty.handler.HandlerList.handle(HandlerList.java:49)
    at org.mortbay.jetty.handler.HandlerWrapper.handle(HandlerWrapper.java:139)
    at org.mortbay.jetty.Server.handle(Server.java:285)
    at org.mortbay.jetty.HttpConnection.handleRequest(HttpConnection.java:457)
    at org.mortbay.jetty.HttpConnection$RequestHandler.headerComplete(HttpConnection.java:751)
    at org.mortbay.jetty.HttpParser.parseNext(HttpParser.java:500)
    at org.mortbay.jetty.HttpParser.parseAvailable(HttpParser.java:209)
    at org.mortbay.jetty.HttpConnection.handle(HttpConnection.java:357)
    at org.mortbay.jetty.bio.SocketConnector$Connection.run(SocketConnector.java:217)
    at org.mortbay.thread.BoundedThreadPool$PoolThread.run(BoundedThreadPool.java:475)
```

[Powered by Jetty://](#)

A: This error indicate that this port 1433 is blocked.

To grant access to port 1433, the below code is required in 'RepertoireServer\config\java2.policy'.

Example:

```
grant
{
    permission java.net.SocketPermission "localhost:1433", "connect,resolve";
    permission java.util.PropertyPermission "java.home", "read";
};
```