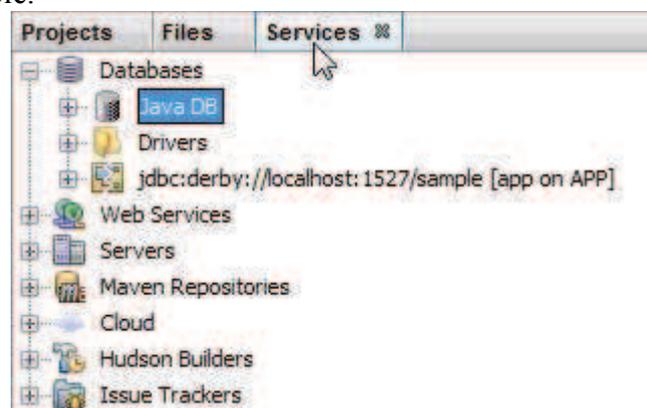


6 NetBeans und Datenbanken (NetBeans 7.2)

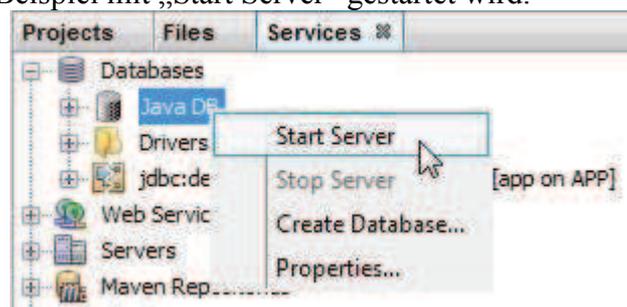
In diesem Kapitel wird davon ausgegangen, dass Java DB zusammen mit Java installiert wurde, was Java 7 immer automatisch der Fall ist. Java DB geht auf die Datenbank Apache Derby zurück, bei der es sich um eine vollwertige Datenbank mit Transaktionssteuerung und der Unterstützung fast aller SQL-Befehle handelt. NetBeans ermöglicht die Erstellung und Bearbeitung von Datenbanken mit einem recht mächtigen Verwaltungssystem. Die Datenbanken werden typischerweise im Zusammenhang mit Web-Applikationen genutzt, stehen aber auch sonst zur Verfügung.

Um sich über den aktuellen Status der Datenbank zu informieren, wird in der Projektverwaltung, die sich standardmäßig auf der linken Seite befindet, der Reiter Services ausgewählt. Hier kann der Bereich Datenbanken aufgeklappt werden. Bei einer Neuinstallation existiert nur die Beispieldatenbank sample.



6.1 Anlegen einer neuen Datenbank

Es soll jetzt eine neue Datenbank eingerichtet werden, die beispielhaft von JDBC aus genutzt werden soll. Die Datenbank läuft dabei auf dem gleichen Rechner, also unter der Adresse localhost. Mit einem Rechtsklick auf Java DB kann man sich über den Status des Datenbanksservers informieren, der für das Beispiel mit „Start Server“ gestartet wird.

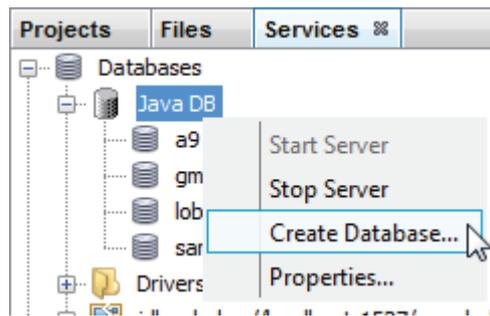


Im Output-Fenster wird ein Text der folgenden Form ausgegeben. Die Port-Nummer ist relevant, wenn man eine Verbindung mit der Datenbank aufnehmen möchte.

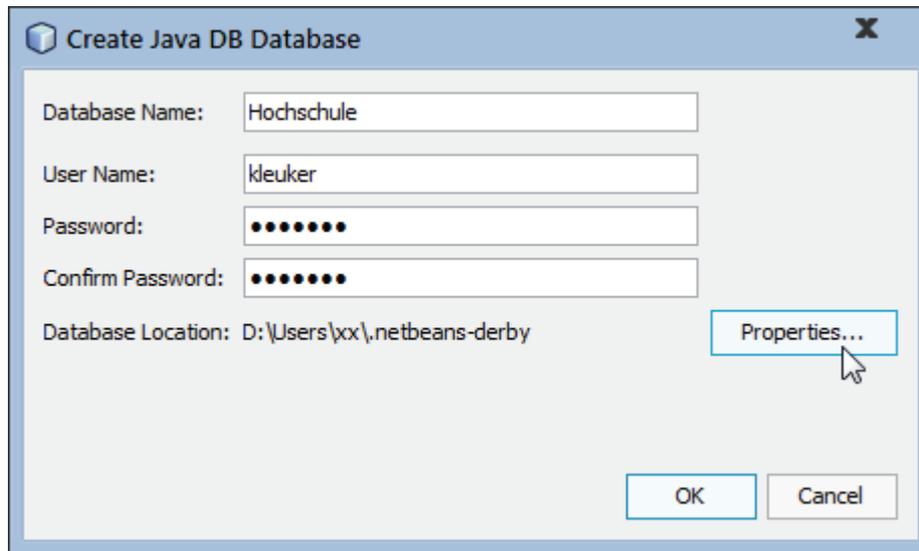
```
2012-08-04 17:37:42.900 GMT : Sicherheitsmanager mit einfacher Server-Sicherheitsrichtlinie installiert.
```

```
2012-08-04 17:37:45.872 GMT : Apache Derby Network Server 10.5.3.0 - (802917) wurde gestartet und ist bereit, Verbindungen am Port 1527 zu akzeptieren.
```

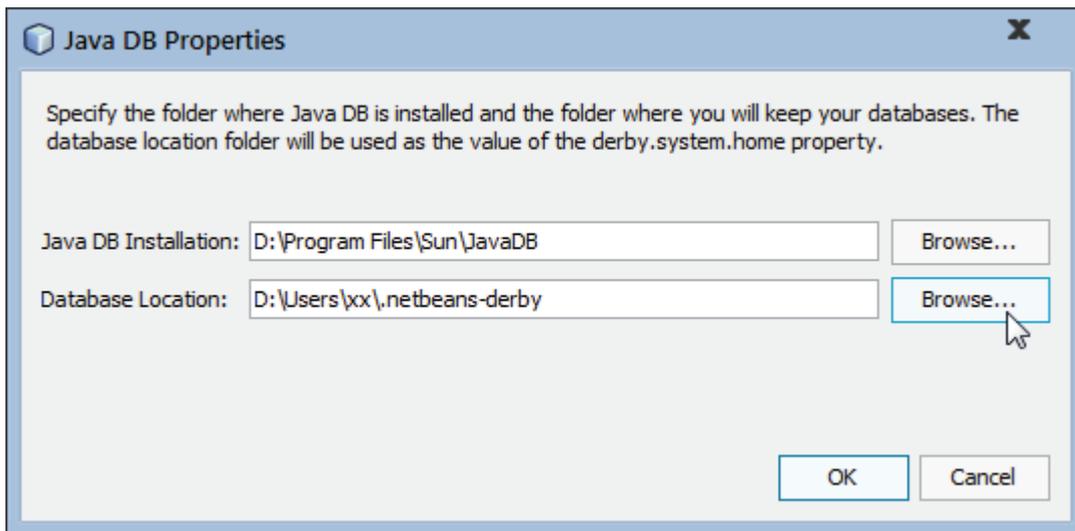
Danach wird im gleichen Menü „Create Database..“ gewählt (was auch ohne Server-Start möglich wäre, der Server wird dann beim Erzeugen gestartet).



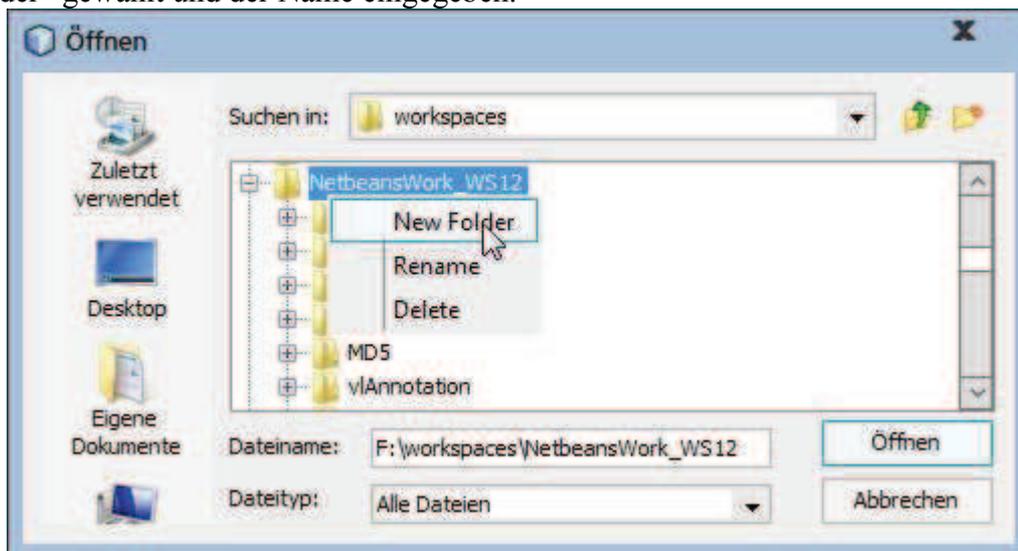
Es werden die vorgegebenen Felder gefüllt, dabei kann auf einen Nutzer und ein Passwort verzichtet werden. Für weitere Einstellungen wird auf „Properties“ geklickt, worauf im Normalfall verzichtet werden kann.



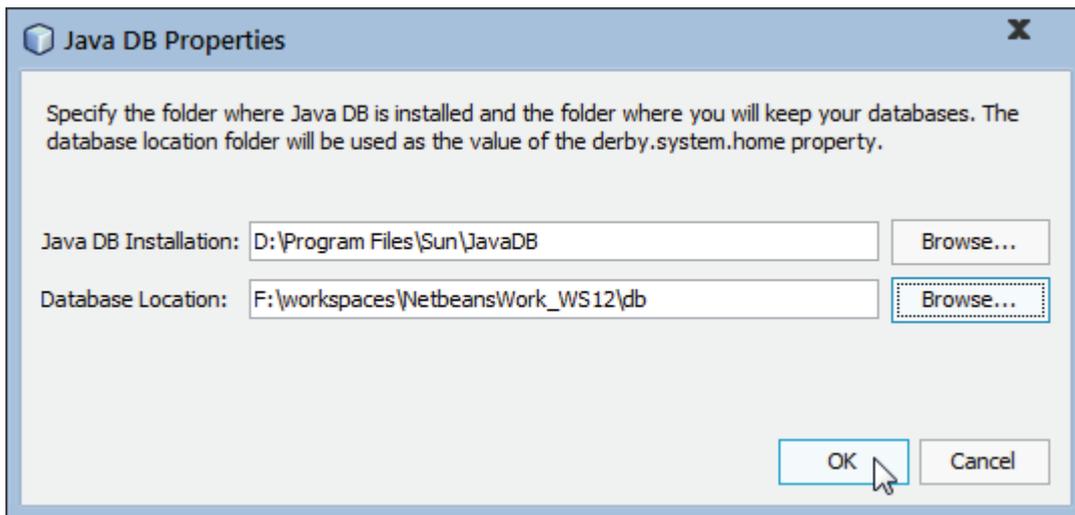
Unter Properties sieht man, welche Datenbanksoftware genutzt wird und wo die Datenbank gespeichert wird. Der letzte Ort sollte passend für alle Programme mit Java DB-Nutzung abgeändert werden. Da das Datenbanksystem dateibasiert arbeitet, kann die jeweilige Datenbank leicht verschoben und auch gesichert werden. In der Hochschule kann die vorgegebene Einstellung übernommen werden, alternativ ist die „Database Location“ auf ein Nutzerlaufwerk umzuändern, in dem man z. B. Datenbanken sammelt. Die hier vorgeschlagene Variante ist, die Datenbank über den Projektordnern abzulegen, in denen sie genutzt werden sollen. Ein direktes Anlegen der Datenbanken in den jeweiligen Projektordnern kann zu Problemen beim Kompilieren und Deployen von Web-Applikationen führen.



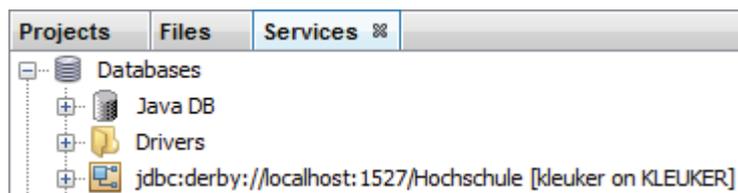
Dazu ist es sinnvoll, im Projekt zunächst einen Ordner anzulegen, was auch hier direkt möglich ist. Dazu wird ein Rechtsklick auf dem Ordner gemacht, dem ein Unterordner hinzugefügt werden soll, „New Folder“ gewählt und der Name eingegeben.



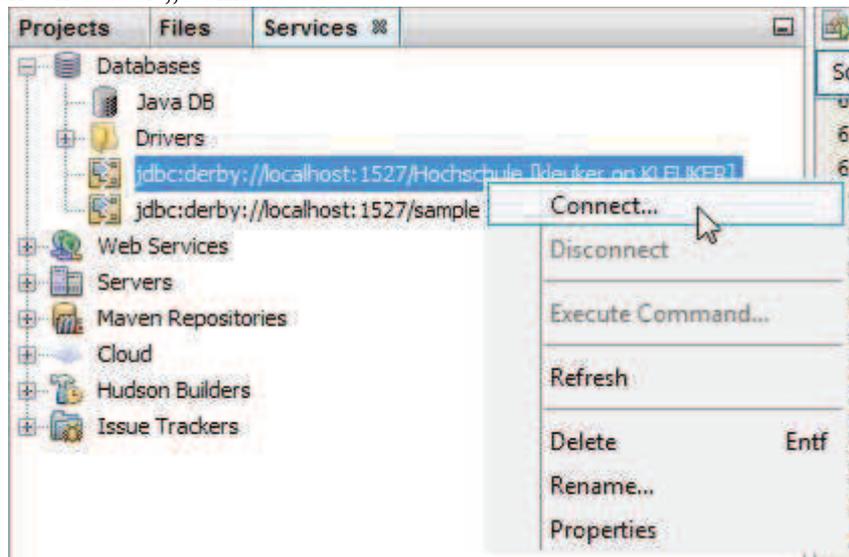
Danach wird die Erstellung mit einem „OK“ im vorherigen Fenster abgeschlossen.



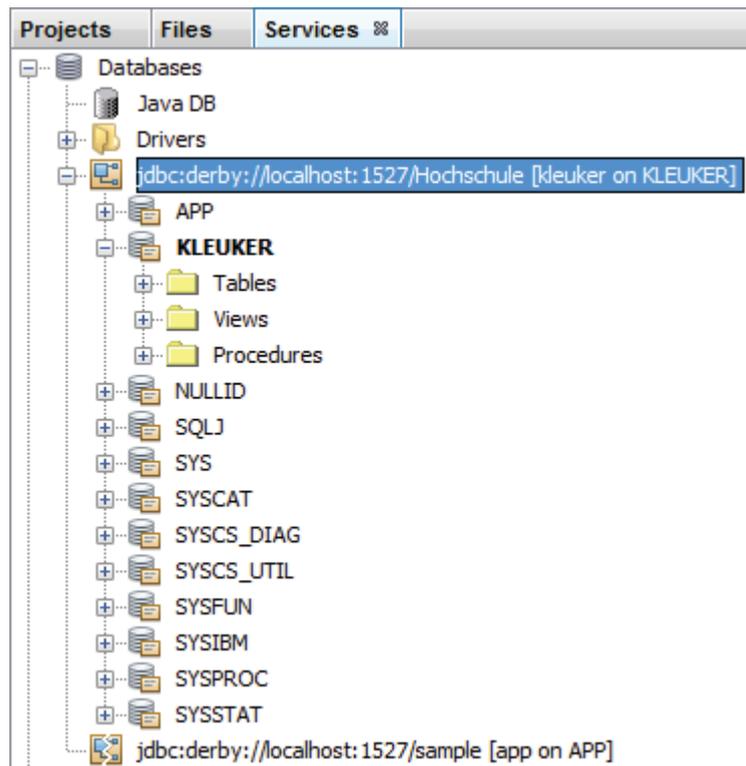
Die neue Datenbank ist auch im Browser sichtbar. Man sieht weiterhin den Nutzernamen und das zugehörige Schema (der Datenbanksbereich). Implizit merkt sich das System das Passwort für den Nutzer.



Möchte man eine Tabelle anlegen, so macht man einen Rechtsklick auf der Datenbank und verbindet sich zunächst über „Connect“.

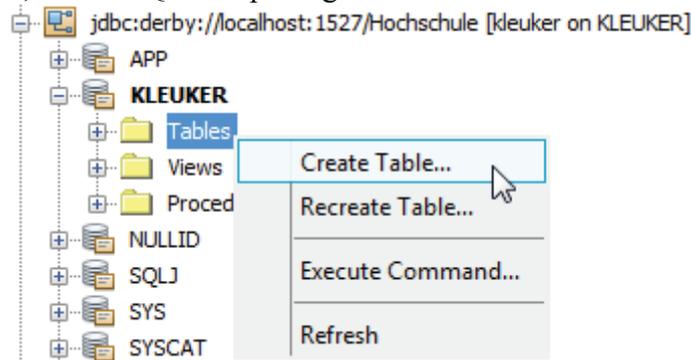


Man sieht dann eine Unterstruktur mit verschiedenen für das Datenbanksystem relevanten Bereichen.

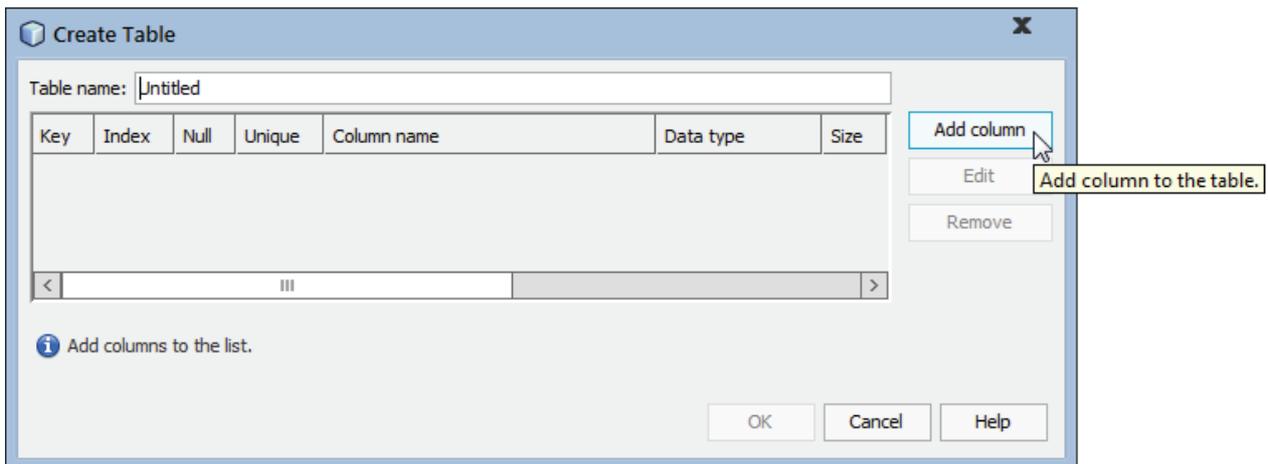


6.2 Anlegen einer Tabelle

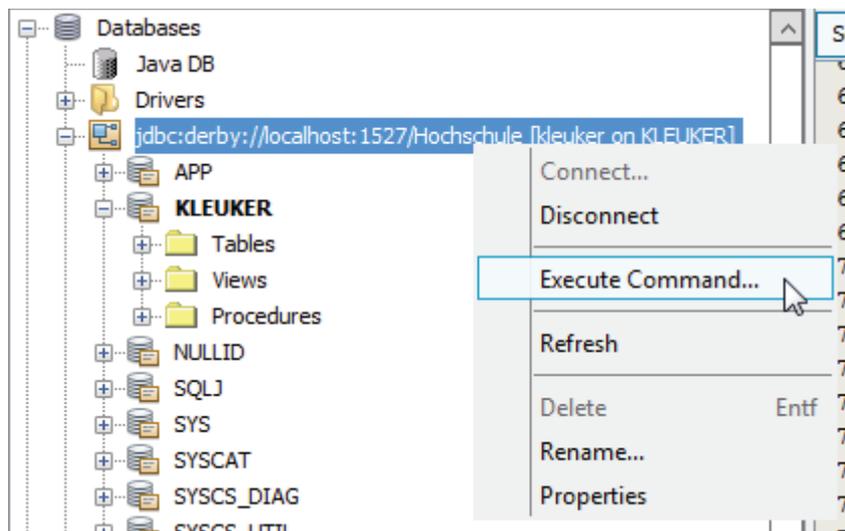
Durch einen Rechtsklick auf den Ordner Tables erhält man mit „CREATE TABLE“ die Möglichkeit, eine Tabelle anzulegen. Diese Möglichkeit wird hier nur vorgeführt, aber im Folgenden nicht genutzt, da ein SQL-Skript eingesetzt werden soll.



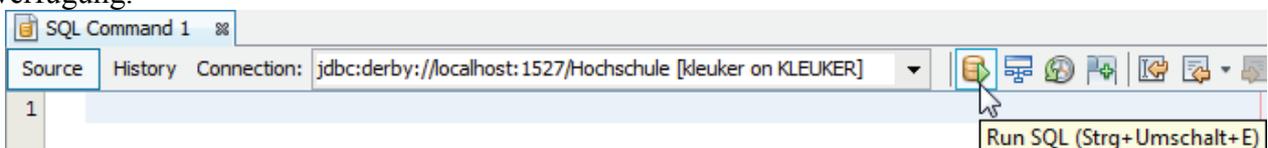
Im darauf sich öffnenden Fenster kann man dann eine Tabelle zusammenstellen. In der folgenden Beschreibung wird allerdings die direkte Befehlseingabe bevorzugt.



Möchte man einfach beliebige SQL-Befehle ausführen, wird bei der Auswahl unter Tables einfach „Execute Command...“ genutzt. Genau der gleiche Befehl steht auch direkt durch einen Rechtsklick auf der Datenbank zur Verfügung.



Es öffnet sich dann im Editorenbereich ein Fenster, mit dem beliebige SQL-Befehle, auch ganze Skripte, ausgeführt werden können. Zum Ausführen steht das Symbol neben der „Connection“ zur Verfügung.



Fehler werden im Output-Bereich unter Nennung des Fensternamens ausgegeben. Im folgenden Beispiel fehlt eine schließende Klammer.

Nutzung von NetBeans

The screenshot shows the NetBeans IDE with the SQL editor open. The SQL command is:

```
1 CREATE TABLE Student (  
2     matnr INTEGER,  
3     name VARCHAR(16),  
4     semester VARCHAR(6),  
5     CONSTRAINT PK_Student  
6     PRIMARY KEY(matnr  
7 );  
8
```

The Output window shows the following error message:

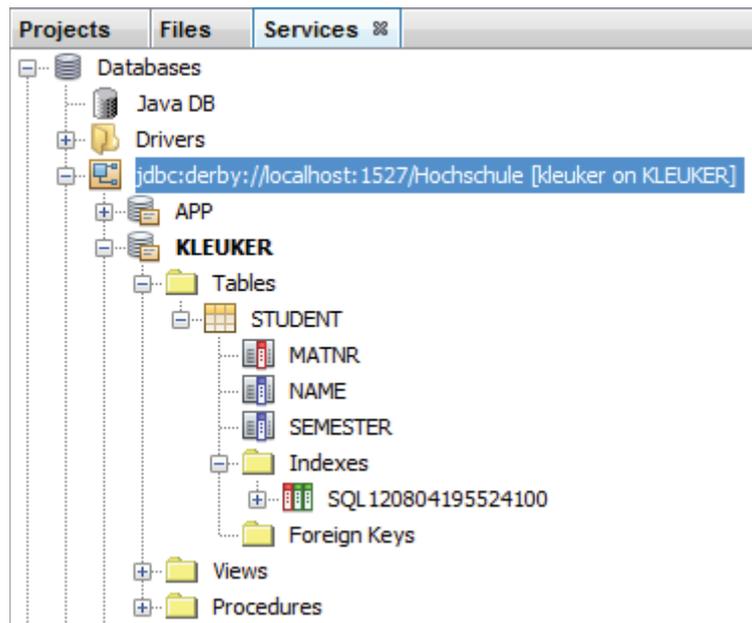
```
vJDBC (run) | Java DB Database Process | SQL Command 1 execution  
Error code -1, SQL state 42X01: Syntaxfehler: Encountered "<EOF>" at line 7, column 1.  
Line 1, column 1  
Execution finished after 0 s, 1 error(s) occurred.
```

Nachdem die Tabellenerzeugung erfolgreich abgeschlossen wurde, wird eine Information im Output-Fenster ergänzt. Damit die Tabelle auch im Browser auf der linken Seite angezeigt wird, muss mit einem Rechtsklick auf der Datenbank ein „Refresh“ aufgerufen werden.

The screenshot shows the NetBeans IDE with the 'Databases' tree on the left. The 'KLEUKER' database is selected, and a context menu is open with the 'Refresh' option highlighted. The SQL editor on the right shows the same CREATE TABLE statement as in the previous screenshot. The Output window shows the following message:

```
vJDBC (run) | Java DB Database Process | SQL Command 1 execution  
Executed successfully in 0,186 s, 0 rows affected.  
Line 1, column 1  
Execution finished after 0,186 s, 0 error(s) occurred.
```

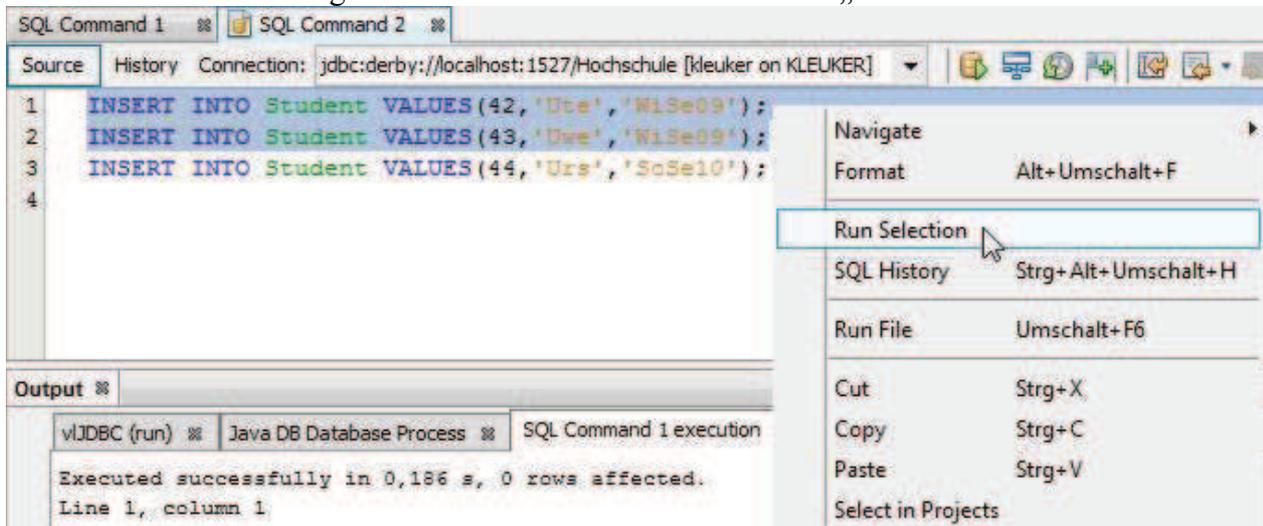
Man erkennt, dass die Tabelle unter dem aktuellen Nutzer eingerichtet wurde.



6.3 Ausführung von SQL-Befehlen

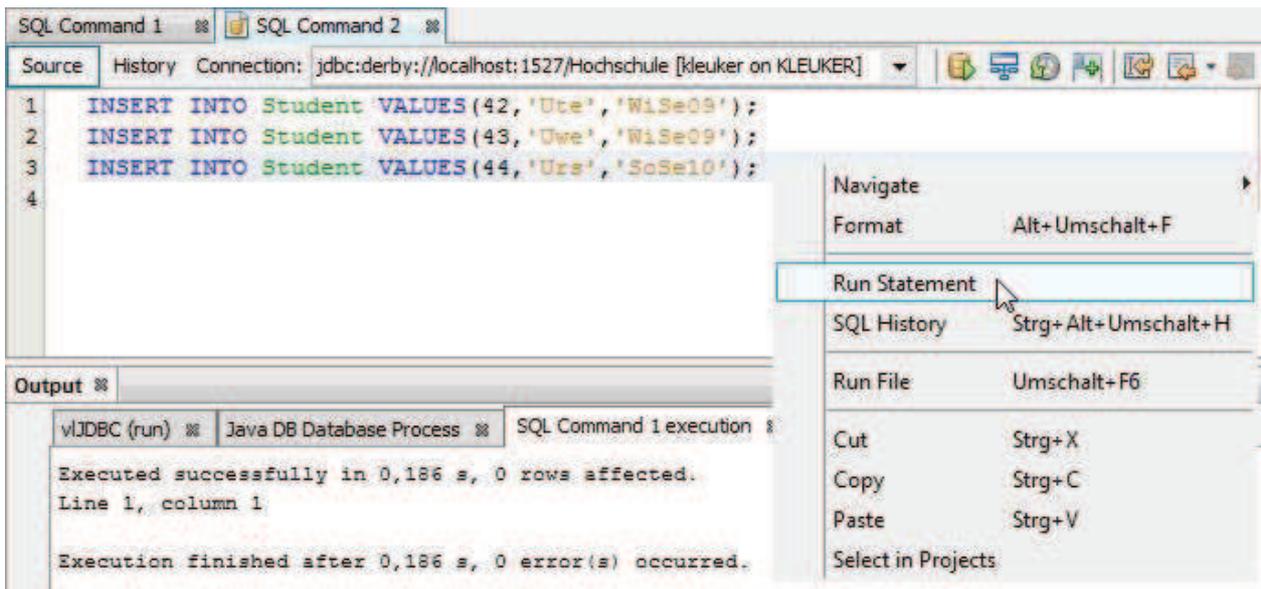
Möchte man Datensätze ergänzen, kann man dies entweder durch einen Rechtsklick auf der Tabelle einleiten, oder direkt SQL-Befehle im Editor-Fenster eingeben. Man kann dabei durchaus mit mehreren Fenstern arbeiten, um die Übersicht zu behalten. Es wird also wieder „Execute Command“ ausgewählt.

Im Editor besteht die Möglichkeit Zeilen zu markieren und über „Run Selection“ auszuführen.

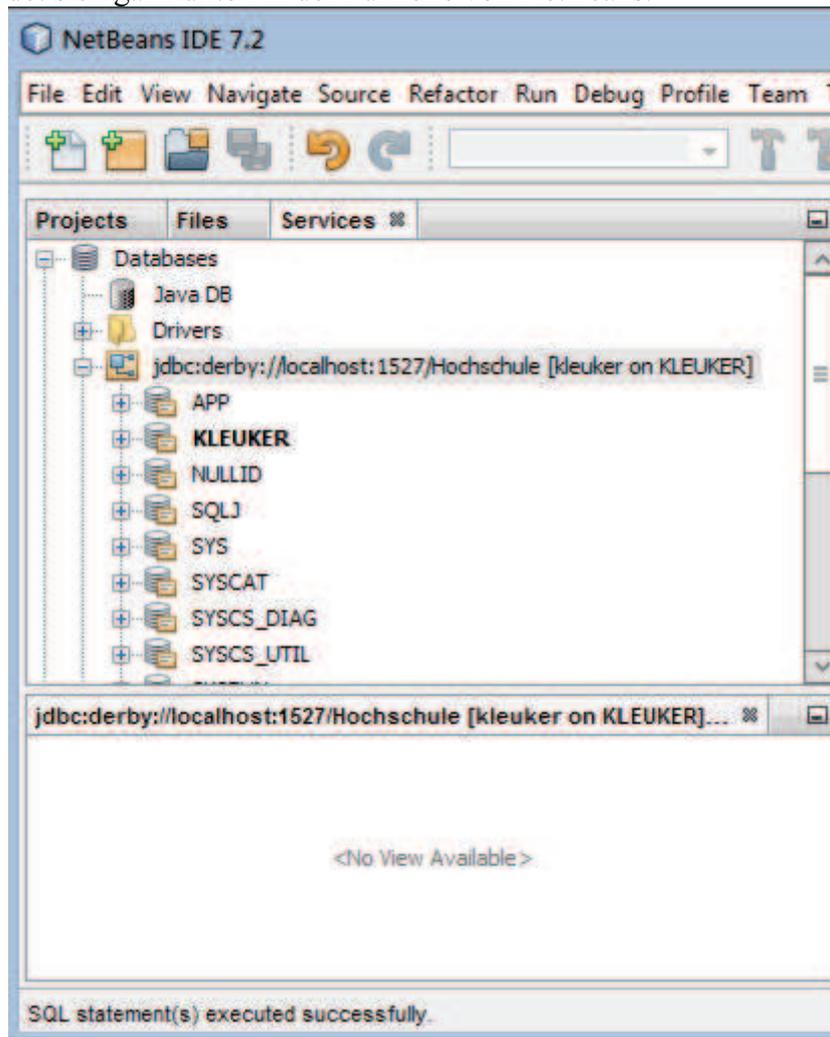


Alternativ besteht auch die Möglichkeit, irgendwo in einem SQL-Befehl einen Rechtsklick auszuführen und „Run Statement“ zu benutzen.

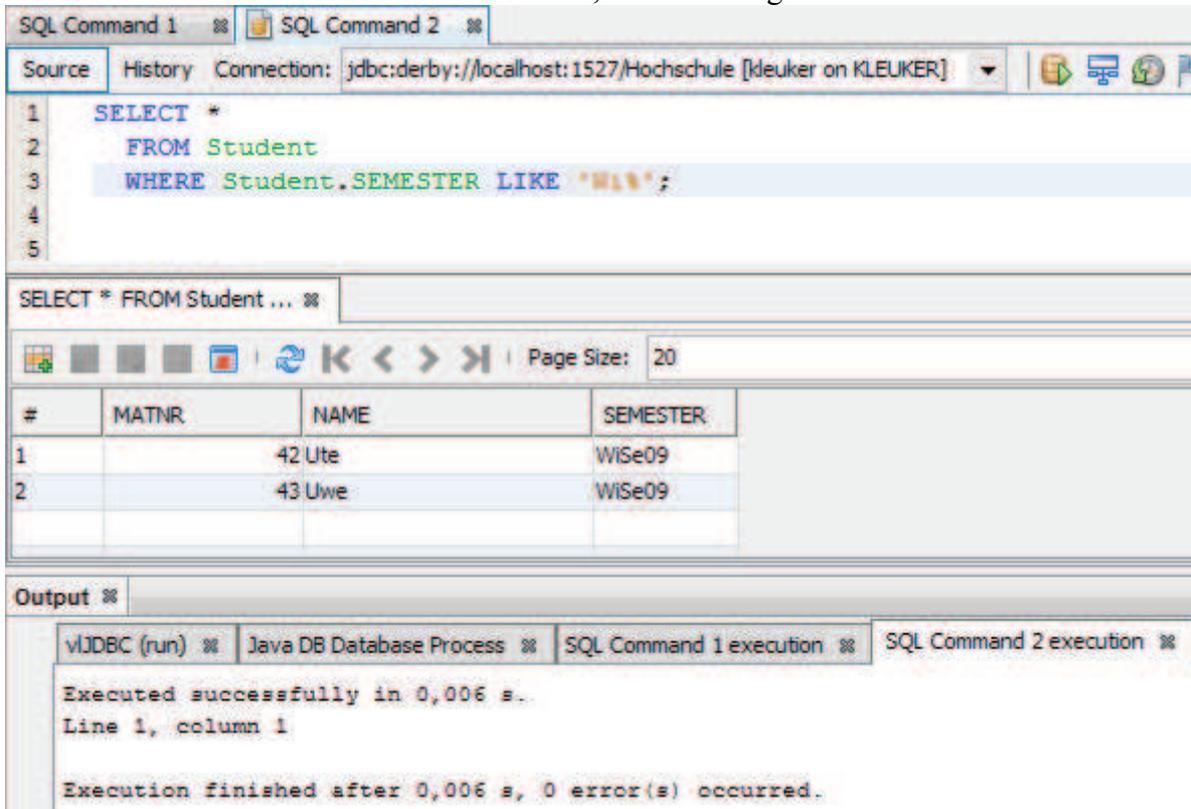
Nutzung von NetBeans



Generell erhält man bei erfolgreich ausgeführten SQL-Befehlen sehr wenig Feedback. Die einzige Information befindet sich ganz unten in der Fußzeile von NetBeans.



Führt man einen SELECT-Befehl aus, wird zwischen dem Editor-Fenster und dem Output-Fenster ein zusätzliches Fenster mit Reitern eingblendet, die die Ergebnisse enthalten. Evtl. muss man die Kanten zwischen den Fenstern etwas verschieben, damit die Ergebnisse sichtbar werden.



The screenshot shows the NetBeans IDE interface. At the top, there are two tabs for SQL Command windows. The active window, 'SQL Command 2', contains the following SQL query:

```
1 SELECT *
2 FROM Student
3 WHERE Student.SEMESTER LIKE 'WiSe09';
4
5
```

Below the editor, a results window titled 'SELECT * FROM Student ...' displays the query results in a table. The table has four columns: '#', 'MATNR', 'NAME', and 'SEMESTER'. The results are as follows:

#	MATNR	NAME	SEMESTER
1		42 Ute	WiSe09
2		43 Uwe	WiSe09

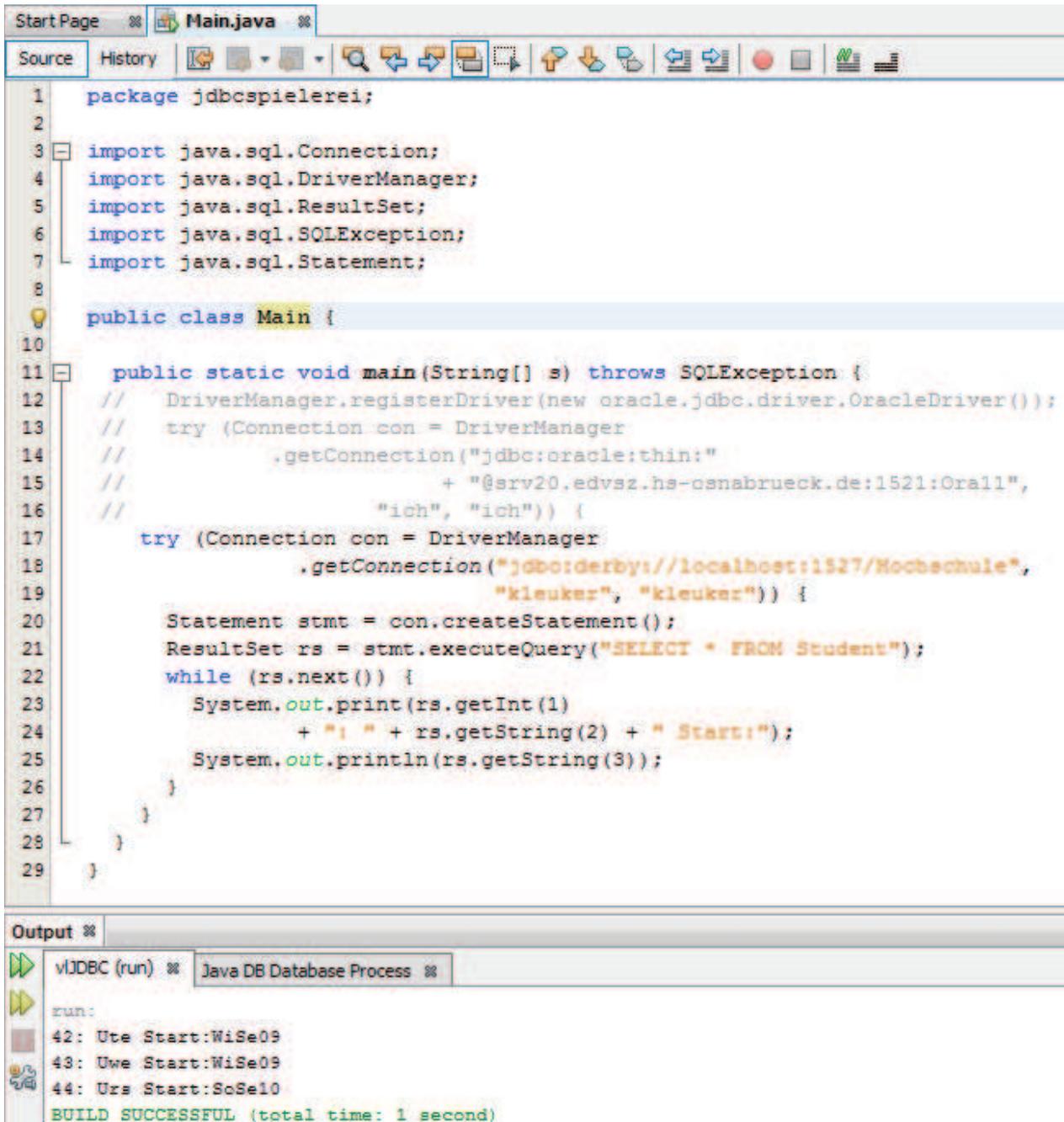
At the bottom, the 'Output' window shows the execution log:

```
vJDBC (run) | Java DB Database Process | SQL Command 1 execution | SQL Command 2 execution
Executed successfully in 0,006 s.
Line 1, column 1
Execution finished after 0,006 s, 0 error(s) occurred.
```

6.4 Zugriff mit JDBC

Der Zugriff mit Hilfe von JDBC erfolgt mit den üblichen Sprachkonstrukten, der „Connection-String“ wurde bereits bei jedem SQL-Command-Fenster ausgegeben. Ein Beispielprogramm sieht mit zugehöriger Ausgabe wie folgt aus. Man beachte, dass in den Kommentaren die Verbindung zu einer externen Oracle-Datenbank beschrieben ist, für die man zunächst einen JDBC-Treiber herunterladen muss. Weiterhin fällt auf, dass für Oracle der Treiber erst registriert werden muss. Dies ist für Java DB nicht der Fall, da der Treiber bereits in NetBeans integriert ist. Damit das Programm auch problemlos auf anderen Rechnern läuft, kann man die Zeile

`DriverManager.registerDriver(new org.apache.derby.jdbc.ClientDriver());`
ergänzen.

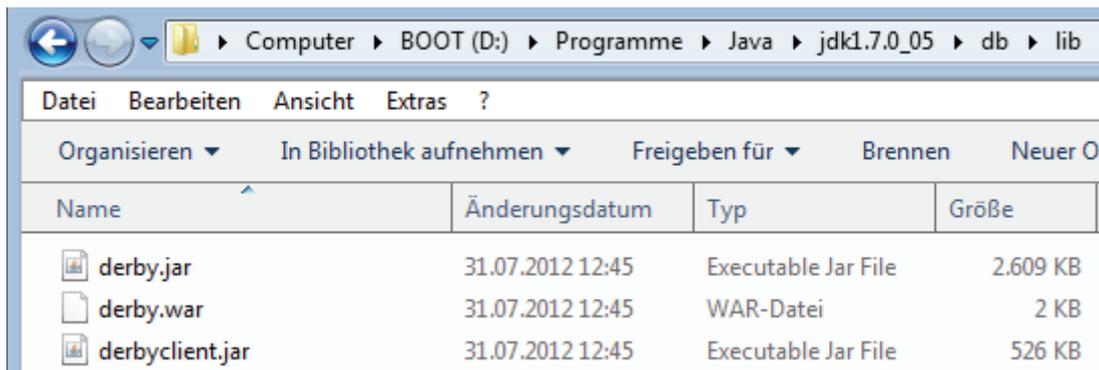


```
1 package jdbcspielerei;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8
9 public class Main {
10
11     public static void main(String[] s) throws SQLException {
12         // DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
13         // try (Connection con = DriverManager
14         //     .getConnection("jdbc:oracle:thin:"
15         //         + "@srv20.edvsz.hs-osnabrueck.de:1521:Orall",
16         //         "ich", "ich")) {
17         try (Connection con = DriverManager
18             .getConnection("jdbc:derby://localhost:1527/Hochschule",
19                 "kleuker", "kleuker")) {
20             Statement stmt = con.createStatement();
21             ResultSet rs = stmt.executeQuery("SELECT * FROM Student");
22             while (rs.next()) {
23                 System.out.print(rs.getInt(1)
24                     + " | " + rs.getString(2) + " Start:");
25                 System.out.println(rs.getString(3));
26             }
27         }
28     }
29 }
```

Output

```
vJDBC (run) Java DB Database Process
run:
42: Ute Start:WiSe09
43: Uwe Start:WiSe09
44: Urs Start:SoSe10
BUILD SUCCESSFUL (total time: 1 second)
```

Damit das Projekt erfolgreich ausgeführt werden kann, muss der zur Datenbank passende JDBC-Treiber in das Projekt eingebunden werden. Für Java DB findet man diesen Treiber im Installationsverzeichnis; typischerweise C:\Program Files\Java\jdk1.7.0_05\db\lib. Er heißt derbyclient.jar. Die Nutzung von Jar-Dateien wird im Abschnitt „4.5 Nutzung von Bibliotheken/Jar-Files (NetBeans 7.2)“ beschrieben.



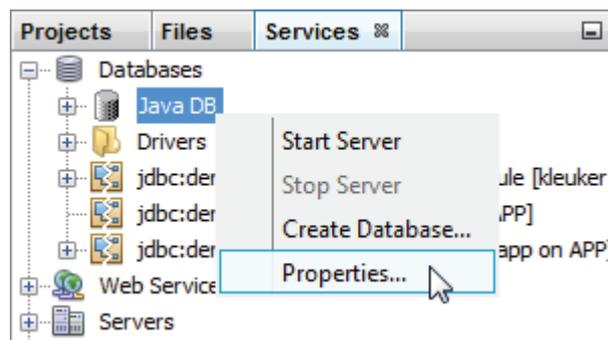
6.5 Einbindung existierender Java DB-Datenbanken

NetBeans erlaubt die flexible Speicherung und erneute Einbindung von Projekten. Ähnlich verhält es sich mit Datenbank-Dateien. Dies ermöglicht z. B. die Speicherung von Datenbanken auf USB-Sticks.

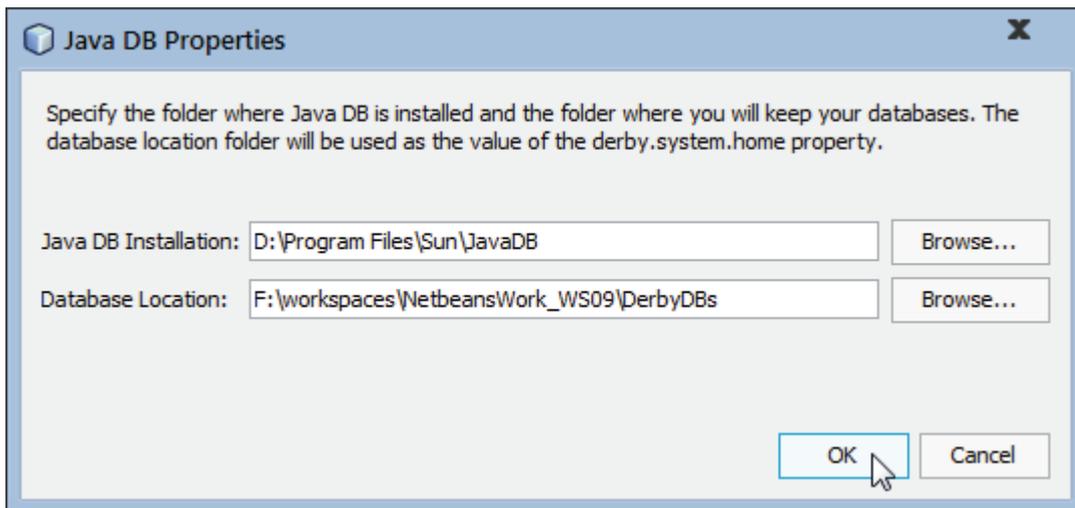
Die relevante Datenbank sei z. B. im Verzeichnis `F:\workspaces\NetBeansWork_WS09\DerbyDBs\Aufgabe05` gespeichert. Dabei befindet sich die Datenbank selbst im Verzeichnis `DerbyDBs`.

Generell arbeitet Java DB nur direkt mit den Datenbanken, die in der Database Location unter „Properties“, siehe auch die nächsten beiden Abbildungen, eingetragen sind. Gibt es fachlich keine anderen Notwendigkeiten, ist es der einfachste Schritt, die zu nutzende Datenbank in dieses Verzeichnis zu kopieren. Die Datenbank wird dann direkt unter „Java DB“ angezeigt. Dies ist für die meisten Programme kein Problem, da nicht der physikalische Ort der gespeicherten Datenbank, sondern der JDBC-Connection-String relevant ist.

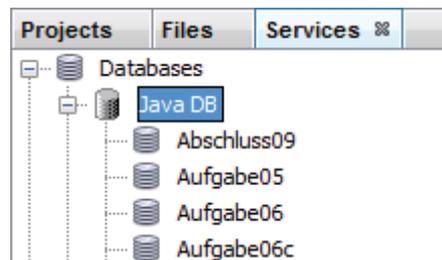
Um die Datenbank direkt in ihrem Ursprungsverzeichnis zu nutzen, muss zunächst der Pfad zur Datenbank im System eingetragen werden. Dazu wird ein Rechtsklick auf „Java DB“ unter dem Reiter Services gemacht.



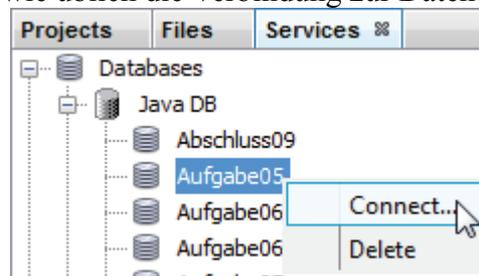
Bei den Properties wird der Pfad zur Datenbank unter „Database Location“ eingetragen.



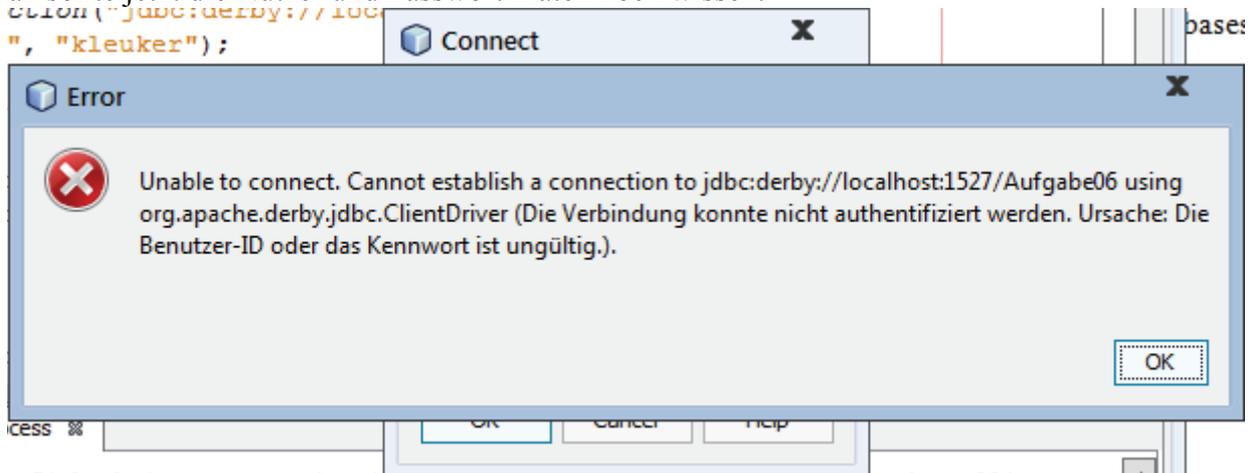
Danach stehen unter Java DB alle im Ordner DerbyDBs vorhandenen Datenbanken zur Verfügung.



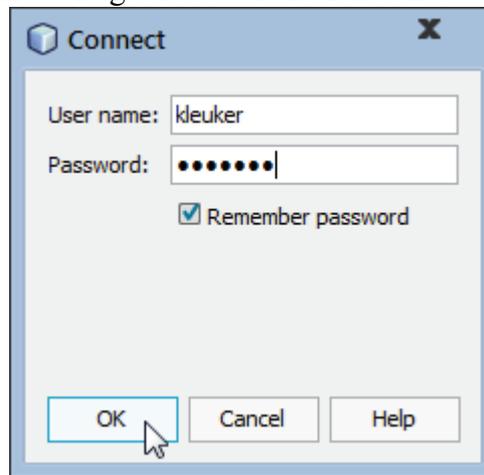
Durch einen Rechtsklick kann wie üblich die Verbindung zur Datenbank aufgebaut werden.



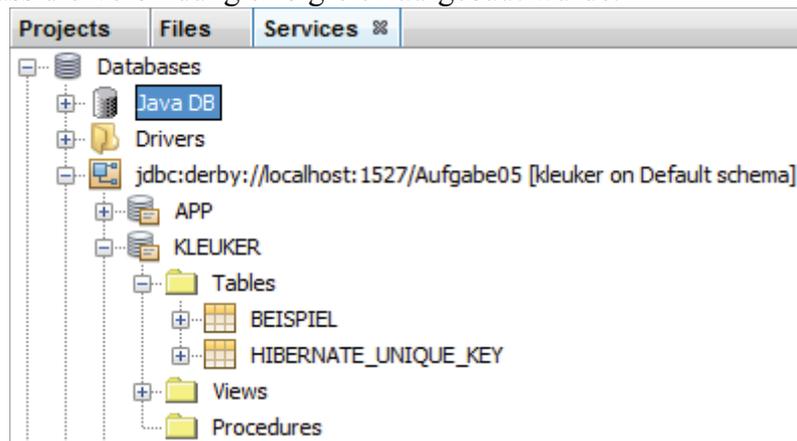
Man sollte jetzt die Nutzer und Passwort-Daten noch wissen.



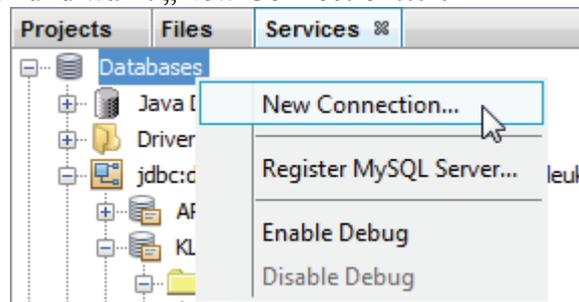
Die Meldung kann man einfach schließen, danach muss der Nutzer und das Passwort eingegeben werden. Sollte weder Nutzer noch Passwort eingetragen sein, klickt man einfach auf „OK“. Lässt man den Haken bei „Remember Password“, wird man nicht wieder gefragt. Wie immer gilt, dass man beim Experimentieren kein wichtiges Passwort nutzen sollte.



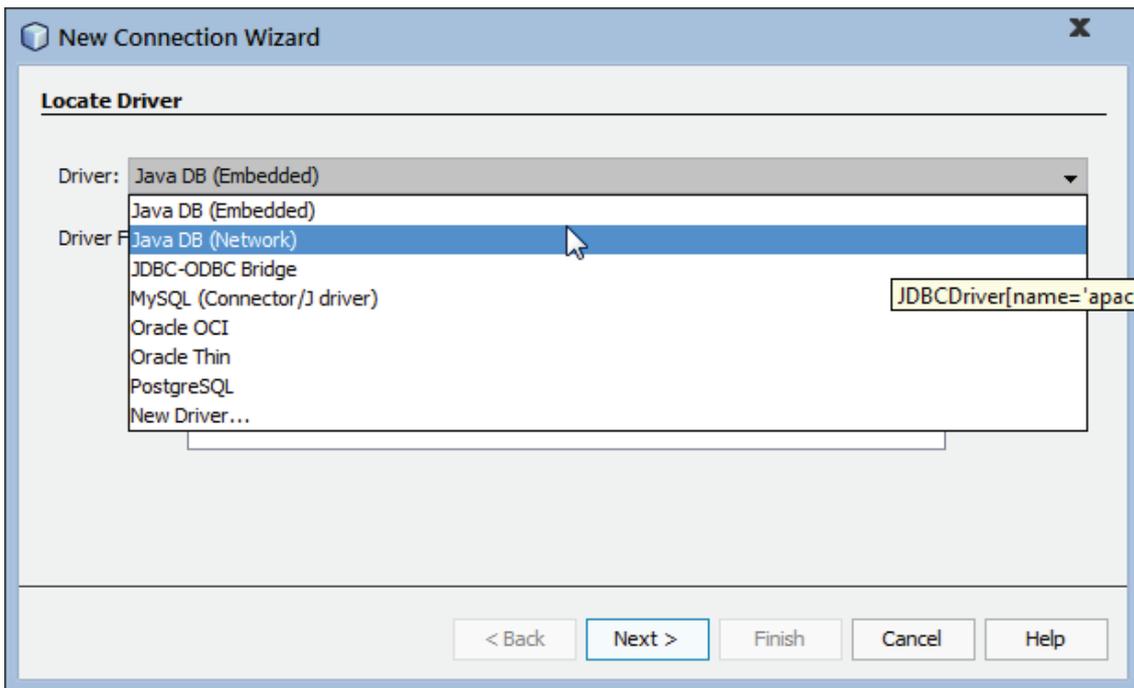
Man sieht dann, dass die Verbindung erfolgreich aufgebaut wurde.



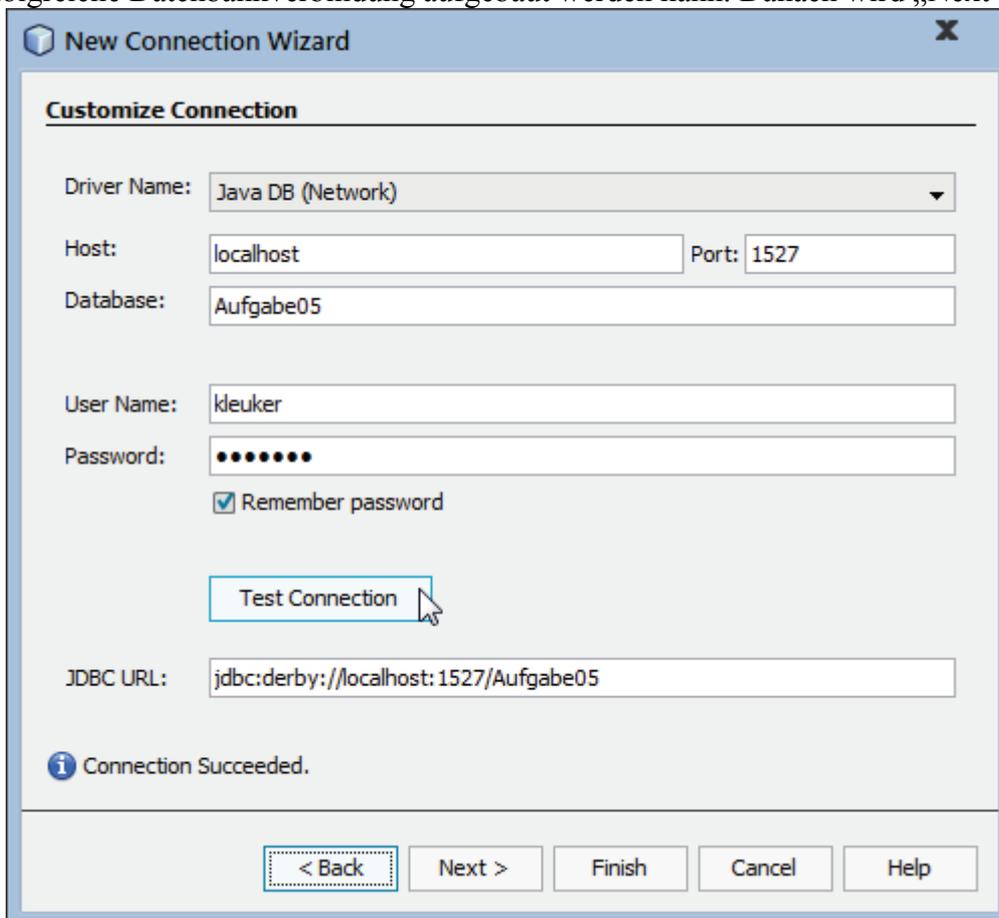
Will man die Verbindung zu anderen existierenden Datenbanken aufbauen, macht man einen Rechtsklick auf „Databases“ und wählt „New Connection...“.



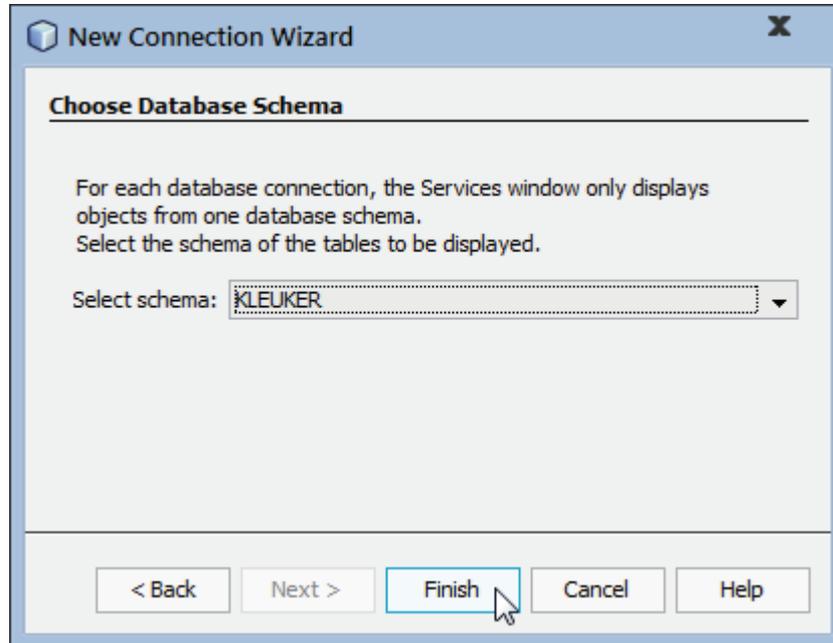
Dann muss oben der passende Treiber ausgewählt werden, NetBeans hat bereits einige installiert, hier können weitere integriert werden. Nach der Auswahl wird „Next>“ gedrückt.



Hier müssen jetzt die Verbindungsdaten eingetragen werden. Vereinfachend wird die gerade vorher eingebundene Datenbank genutzt. Man beachte die Möglichkeit über „Test Connection“ zu prüfen, ob eine erfolgreiche Datenbankverbindung aufgebaut werden kann. Danach wird „Next>“ gedrückt.



Danach muss evtl. noch ein Schema ausgewählt werden. Dies muss das sein, in dem die Daten liegen.



Danach steht die Datenbank zur Verfügung.

