

Datenbanksysteme 2015

noch Kapitel 09:
Datenbankapplikationen

Vorlesung vom 02.06.2015

Oliver Vornberger

Institut für Informatik
Universität Osnabrück

Datenbankapplikationen

- MS Visio
- MS Access
- ~~• Embedded SQL im C-Programm~~
- JDBC / SQLJ Application
- SQLite / HSQLDB
- JDBC Applet
- Java Servlet
- Java Server Pages
- PHP
- Ruby on Rails
- CouchDB

JDBC

Java-Programm mit Klassen aus [java.sql](#)

- Application
- Applet

nutzt Datenbankserver oder lokale Datei

- Treiber laden
- Verbindung herstellen
- SQL-Statement ausführen
- Ergebnis verarbeiten

SQL versus Java

BIT	boolean
TINYINT	byte
SMALLINT	short
INTEGER	int
BIGINT	long
FLOAT	float
REAL, DOUBLE	double
NUMERIC, DECIMAL	java.math.Big
CHAR, VARCHAR	String
DATE	java.sql.Date
TIME	java.sql.Time
TIMESTAMP	java.sql.Timestamp
BINARY	byte[]

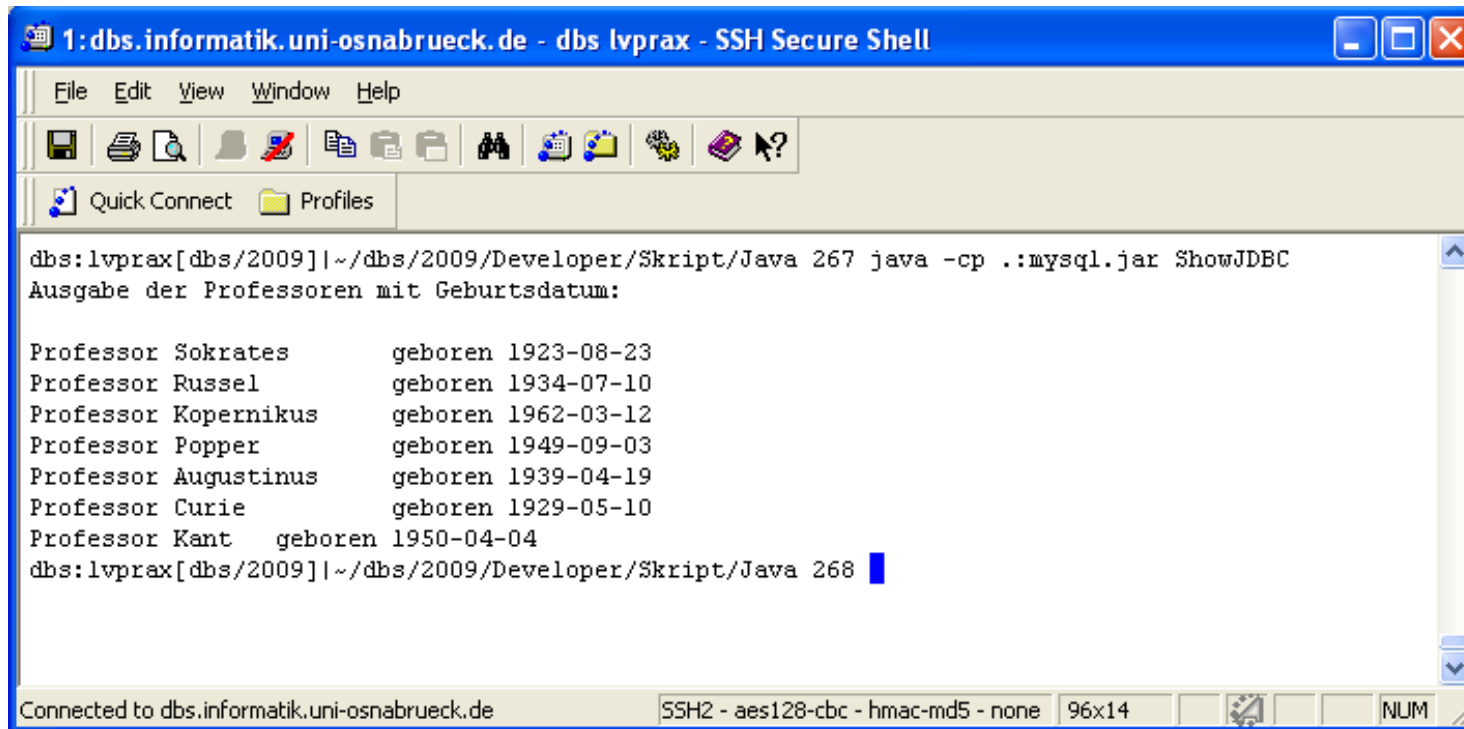
ShowJDBC.java

```
String url      ="jdbc:mysql://dbs.informatik.uos.de/UniWeb";
String user     = "erika";
String passwd   = "mustermann";
Class.forName("com.mysql.jdbc.Driver");
Connection con  = DriverManager.getConnection(url,user,passwd);
Statement stmt  = con.createStatement();
String query    = "select * from Professoren";
ResultSet rs    = stmt.executeQuery(query);
System.out.println("Professoren mit Geburtsdatum: \n");

while(rs.next()) {
    System.out.print("Professor ");
    System.out.print(rs.getString("Name"));
    System.out.print(" \t geboren ");
    System.out.println(rs.getDate("gebdatum"));
}

stmt.close();
con.close();
```

Quelltext von ShowJDBC



The screenshot shows an SSH terminal window titled "1: dbs.informatik.uni-osnabrueck.de - dbs lvprax - SSH Secure Shell". The terminal displays the command `java -cp ./mysql.jar ShowJDBC` and its output, which lists professors and their birth dates. The terminal interface includes a menu bar (File, Edit, View, Window, Help), a toolbar with various icons, and a status bar at the bottom showing connection details and window size.

```
dbs:lvprax[dbs/2009]|~/dbs/2009/Developer/Skript/Java 267 java -cp ./mysql.jar ShowJDBC
Ausgabe der Professoren mit Geburtsdatum:

Professor Sokrates      geboren 1923-08-23
Professor Russel       geboren 1934-07-10
Professor Kopernikus    geboren 1962-03-12
Professor Popper        geboren 1949-09-03
Professor Augustinus    geboren 1939-04-19
Professor Curie         geboren 1929-05-10
Professor Kant          geboren 1950-04-04
dbs:lvprax[dbs/2009]|~/dbs/2009/Developer/Skript/Java 268
```

```
javac ShowJDBC.java
java -cp ./mysql.jar ShowJDBC.java
jar tf mysql.jar
```

SSH

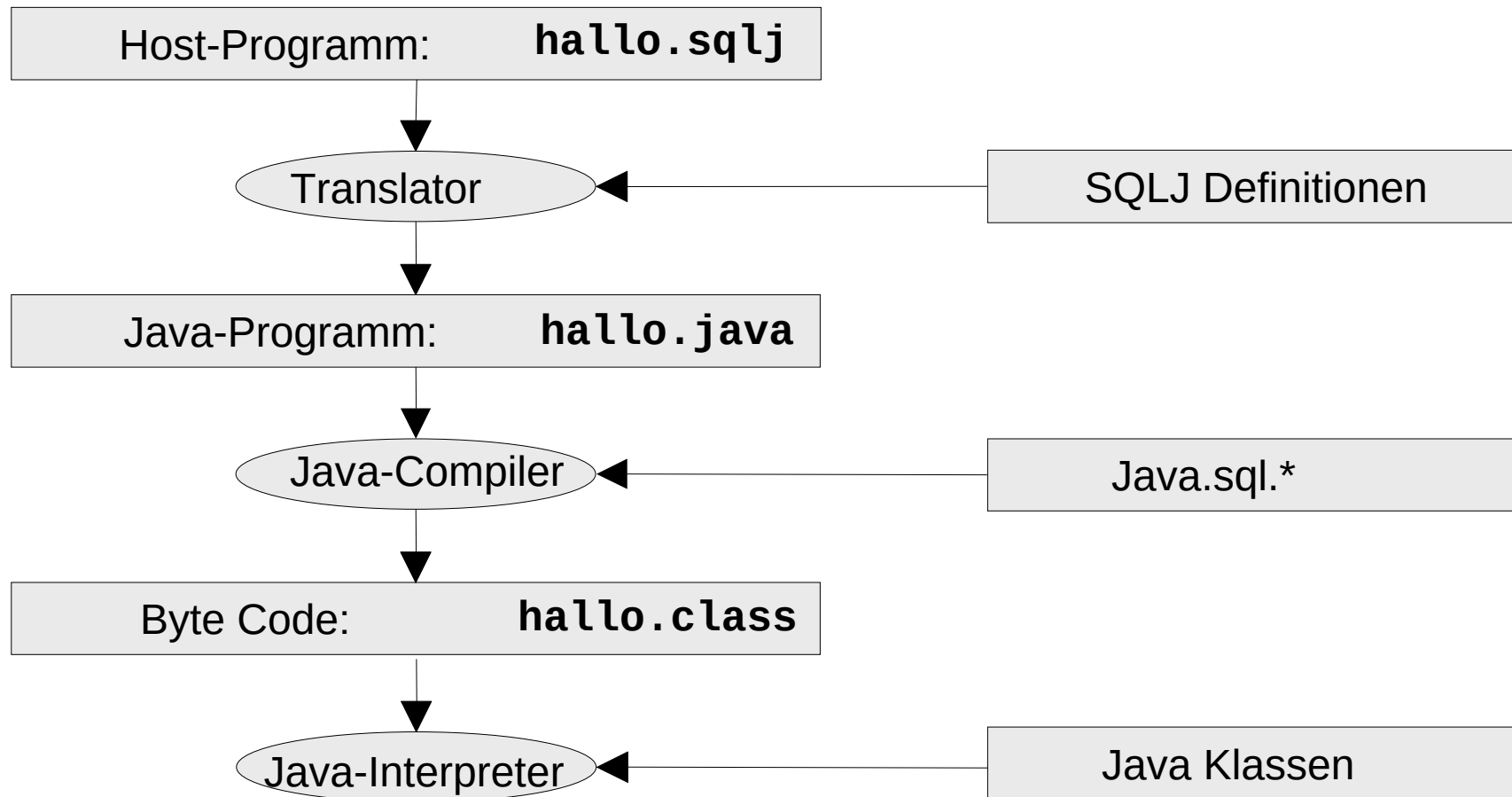
JDBC: Update

```
...  
Statement stmt = con.createStatement();  
String query = "UPDATE professoren " +  
               "SET rang='C2' WHERE rang IS NULL";  
int x = stmt.executeUpdate(query);  
System.out.println("Es wurden " + x + " befoerdert.");  
...
```

JDBC: Prepared Statement

```
String query = "select * from Studenten" +  
               "where semester < ? ";  
  
PreparedStatement pstmt = con.prepareStatement(query);  
  
System.out.print("Bitte Semesterobergrenze: ");  
Scanner sc = new Scanner(System.in);  
int vorgabe = sc.nextInt();  
  
pstmt.setInt(1, vorgabe);  
  
ResultSet rs = pstmt.executeQuery();
```


SQLJ



Queries können zur Übersetzungszeit überprüft werden !

JDBC + SQLJ (Hostvariablen)

```
int vorgabe = IO.readInt("Bitte Personalnr: ");  
String name, rang;
```

```
#sql {select name, rang into :name, :rang  
      from professoren  
      where persnr = :vorgabe};
```

```
System.out.print("Die Personalnr. gehoert zu");  
System.out.println(name + " mit Rang " + rang);
```

<http://www-lehre.inf.uos.de/~dbs/2015/Java/ShowSqljHost.sqlj>

<http://www-lehre.inf.uos.de/~dbs/2015/Java/ShowSqljHost.java>

SQLJ Iterator

```
#sql iterator MyIter(String name, String semester);  
  
MyIter iter;  
  
#sql iter = {select name, semester from Studenten};  
  
while (iter.next()) {  
    System.out.print(iter.name() + " studiert im ");  
    System.out.println(iter.semester() + ". Semester");  
}
```

<http://www-lehre.inf.uos.de/~dbs/2015/Java/ShowSqljIter.sqlj>

SQLite

- in C geschriebene Programmbibliothek für SQL-Abfragen
- doppelt so schnell wie MySQL
- Java-Anbindung möglich (emuliert C in Java VM)
- Treiber sqlitejdbc-v054.jar
- kein DB-Server erforderlich
- nur ein User
- unterstützt Transaktionen
- meist genutzte Datenbank der Welt
- z.B. iPhone, Android
- <http://www.sqlite.org/>

SQLiteSchreibTest.java [Teil 2]

```
        stat.executeUpdate("insert into Profs
                           values(1111, 'Susi', 'W3', 226)");
        stat.executeUpdate("insert into Profs
                           values (2222, 'Paul', 'W3', 312)");
        stat.executeUpdate("insert into Profs
                           values (3333, 'Gerd', 'W3', 434)");
        stat.executeUpdate("insert into Profs
                           values (4444, 'Hans', 'W3', 128)");

        stat.close();

    }catch(SQLException ex) {
        System.err.println("SQLException: " + ex.getMessage());
    }catch(ClassNotFoundException ex) {
        System.err.println("SQLException: " + ex.getMessage());
    }
}
}
```

<http://www-lehre.inf.uos.de/~dbs/2015/Java/SQLiteSchreibTest.java>

SQLiteLeseTest.java [Teil 1]

```
import java.sql.*;

public class SQLiteLeseTest{

    public static void main(String args[]) {

        String url      = "jdbc:sqlite:meine-sqlite-datenbank";
        String user      = "erika";
        String passwd    = "mustermann";
        try {
            Class.forName("org.sqlite.JDBC");
            Connection con;
            con = DriverManager.getConnection(url,user,passwd);
            Statement stmt = con.createStatement();
            String query    = "select * from Profs";
            ResultSet rs    = stmt.executeQuery(query);
```

SQLiteLeseTest.java [Teil 2]

```
System.out.println("Ausgabe der Professoren\n");
while(rs.next()) {
    System.out.print("Professor ");
    System.out.print(rs.getString("Name"));
    System.out.print(" hat den Rang ");
    System.out.println(rs.getString("Rang"));
}

stmt.close();
con.close();
}catch(SQLException ex) {
    System.err.println("SQLException: " + ex.getMessage());
}catch(ClassNotFoundException ex) {
    System.err.println("SQLException: " + ex.getMessage());
}
}
```

<http://www-lehre.inf.uos.de/~dbs/2015/Java/SQLiteLeseTest.java>

HSQldb

in Java geschriebene relationale Datenbank
hsqldb.jar nur 600 KB

Datenbank erzeugbar

- nur zur Laufzeit im Arbeitsspeicher
- im file-Modus ähnlich zu SQLite
- als Server für mehrere Clienten

<http://hsqldb.org/>

HSQLDBSchreibTest.java [Teil 2]

```
        stat.executeUpdate("insert into Profs
                           values(1111, 'Susi', 'W3', 226)");
        stat.executeUpdate("insert into Profs
                           values (2222, 'Paul', 'W3', 312)");
        stat.executeUpdate("insert into Profs
                           values (3333, 'Gerd', 'W3', 434)");
        stat.executeUpdate("insert into Profs
                           values (4444, 'Hans', 'W3', 128)");

        stat.execute("SHUTDOWN");
        stat.close();
    }catch(SQLException ex) {
        System.err.println("SQLException: " + ex.getMessage());
    }catch(ClassNotFoundException ex) {
        System.err.println("SQLException: " + ex.getMessage());
    }
}
```

<http://www-lehre.inf.uos.de/~dbs/2015/Java/HSQLDBSchreibTest.java>

HSQLEseTest.java [Teil 1]

```
import java.sql.*;

public class HSQLEseTest{

    public static void main(String args[]) {

        String url      = "jdbc:hsqldb:file:meine-hsqldb-datenbank";
        String user      = "sa";
        String passwd    = "";
        try {
            Class.forName("org.hsqldb.jdbcDriver");
            Connection con;
            con = DriverManager.getConnection(url,user,passwd);
            Statement stmt = con.createStatement();
            String query    = "select * from Profs";
            ResultSet rs     = stmt.executeQuery(query);
```

HSQMDBLeseTest.java [Teil 2]

```
System.out.println("Ausgabe der Professoren\n");
while(rs.next()) {
    System.out.print("Professor ");
    System.out.print(rs.getString("Name"));
    System.out.print(" hat den Rang ");
    System.out.println(rs.getString("Rang"));
}

stmt.close();
con.close();
}catch(SQLException ex) {
    System.err.println("SQLException: " + ex.getMessage());
}catch(ClassNotFoundException ex) {
    System.err.println("SQLException: " + ex.getMessage());
}
}
```

JDBC Applet

- Javaprogramm auf Webserver
- gezogen von und ausgeführt im Webbrowser
- Applet darf nur Socket-Verbindung aufbauen zu Rechner, von dem es gezogen wurde
- $\Rightarrow$ Datenbankserver und Webserver auf einem Rechner

getMetaData()

```
...  
stmt = con.createStatement();  
ResultSet rs = stmt.executeQuery(query);  
int spalten = rs.getMetaData().getColumnCount();  
StringBuilder b = new StringBuilder();  
for (int i = 1; i <= spalten; i++) {  
    String lab = rs.getMetaData().getColumnLabel(i);  
    int max = rs.getMetaData().getColumnDisplaySize(i) + 4;  
    int cur = b.length();  
    b.setLength(cur+max);  
    b.replace(cur, cur+lab.length(), lab);  
}  
b.append("\n");  
...
```

[JDBCApplet.java](#)

[Aufruf von JDBCApplet](#)

einfache Webseite für JDBCApplet.class

```
<html>
  <head>
    <title>JDBC-Applet UniWeb</title>
  </head>
  <body>
    <applet
      code      ="applet.JDBCApplet.class"
      archive   ="mysql-connector-java-5.1.7-bin.jar"
      width     ="700"
      height    ="400" >
    </body>
</html>
```

<http://dbs.informatik.uos.de/media2mult/applet/index2.html>

korrekte Webseite für JDBCApplet.class

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html><head><title>JDBC-Applet UniWeb</title></head>
<body> <h1>Demo-Applet f&uuml;r JDBC-Datenbankzugriff</H1>
<div>
  <object classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
    codetype="application/java-vm" width="700" height="400"
    data="applet.JDBCApplet.class">
    <param name="archive" value="mysql-connector-java-5.1.7-bin.jar">
    <param name="code" value="applet.JDBCApplet.class">
    <param name="type" value="application/java-vm">
    <comment>
      <object classid="java:applet.JDBCApplet.class"
        codetype="application/java-vm" width="700" height="400">
        <param name="archive" value="mysql-connector-java-5.1.7-bin.jar">
      </object>
    </comment>
  </object>
</div>
</body>
</html>
```

<http://dbs.informatik.uos.de/media2mult/applet/>

Javascript und PHP

```
<?php
mysql_connect("dbs.informatik.uos.de", "thorben", "4711");
mysql_select_db("opi");
$query_result = mysql_query("SELECT * FROM pfad");
$result = array();
while($data_point = mysql_fetch_assoc($query_result)){
    $result[] = $data_point;
}
print json_encode($result);
?>
```

<http://www-lehre.inf.uos.de/~dbs/2015/PHP/data.php.txt>

```
$.getJSON('data.php', function(data) {
...
    drawCircle(x,y,context);
})
```

<http://www-lehre.inf.uos.de/~dbs/2015/javascript/studenten-timeout-circle.html.txt>

<http://project2.informatik.uos.de/opi/Loesungen/studenten-timeout-circle.html>

Datenbankapplikationen

- MS Visio
- MS Access
- ~~• Embedded SQL im C Programm~~
- JDBC / SQLJ Application
- SQLite / HSQLDB
- JDBC Applet
- Javascript & PHP
- Java Servlet
- Java Server Pages
- PHP
- Ruby on Rails
- CouchDB