

# Datenbanksysteme 2011

## Kapitel 10: Datenbankapplikationen

Oliver Vornberger

Institut für Informatik  
Universität Osnabrück

# Datenbankapplikationen

ODBC

MS Visio

MS Access

Embedded SQL

JDBC Application

SQLJ

SQLite / HSQLDB

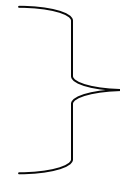
JDBC Applet

Java Servlet

Java Server Pages

PHP

Ruby on Rails



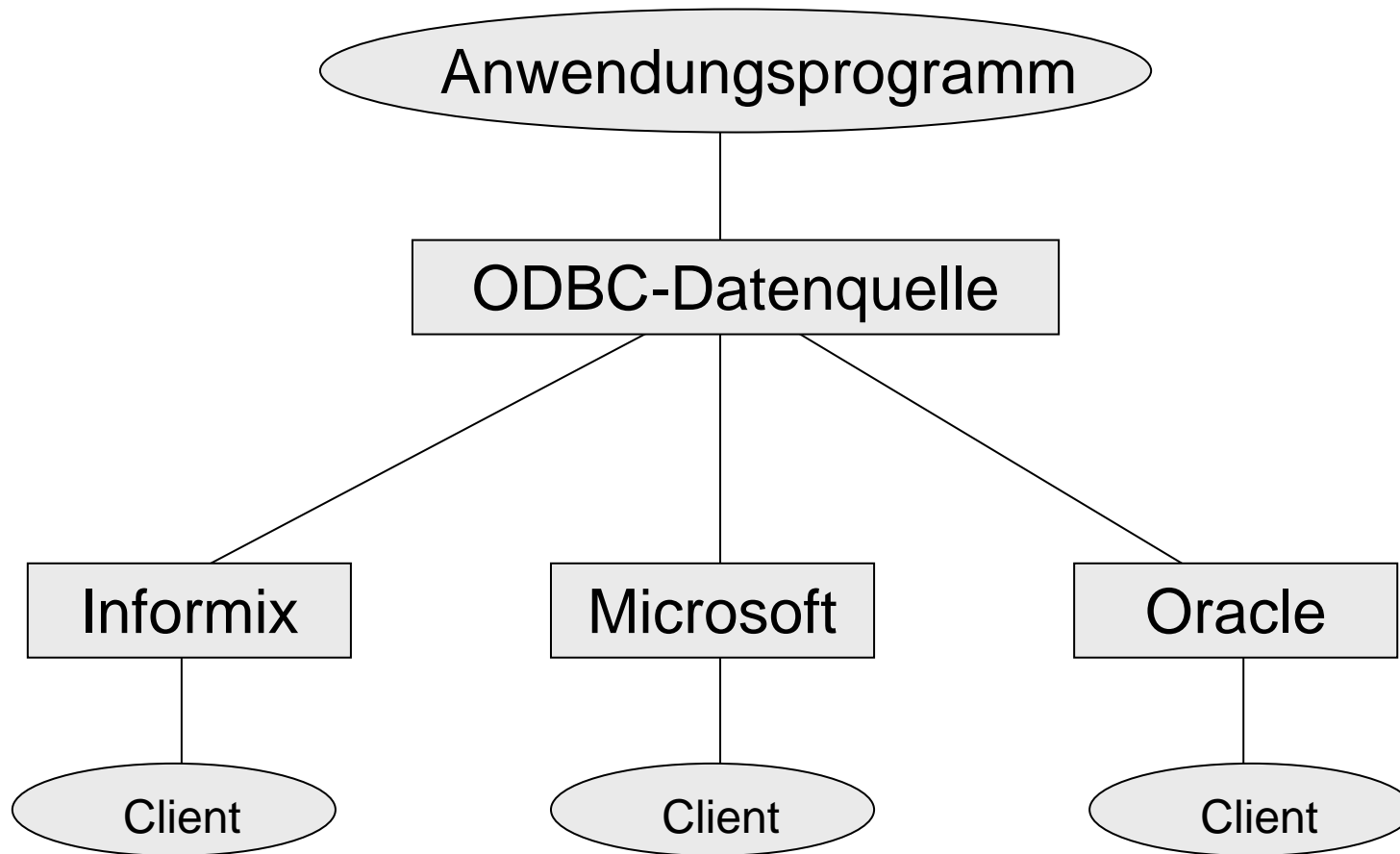
DQ im  
Client



DB-Server-Protokoll

# ODBC

Open Data base connectivity



# MySQL Connector

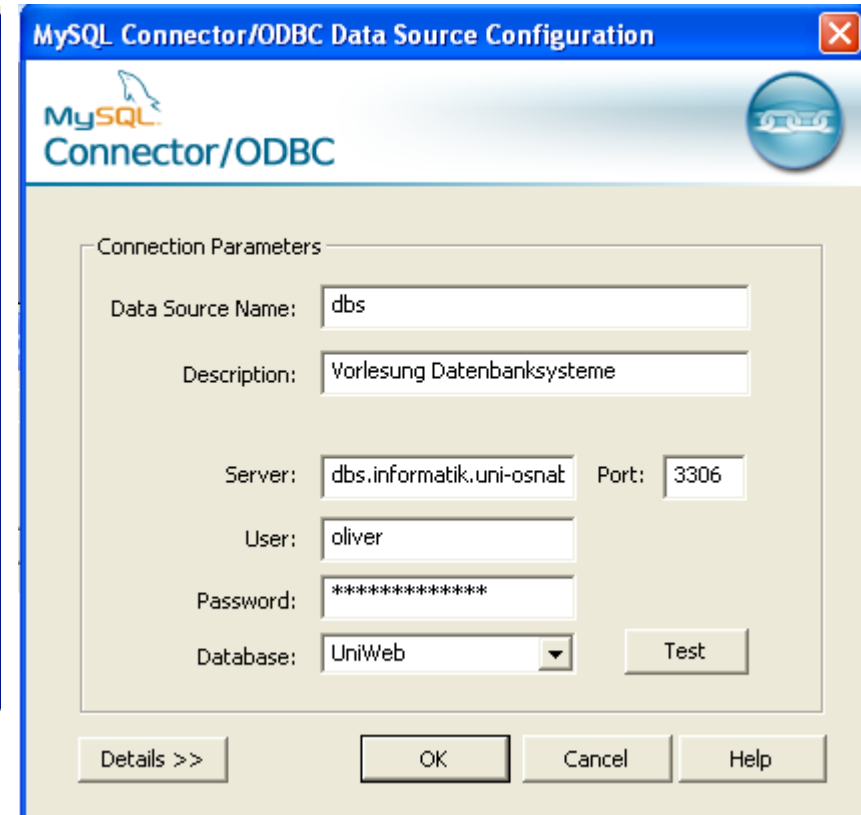
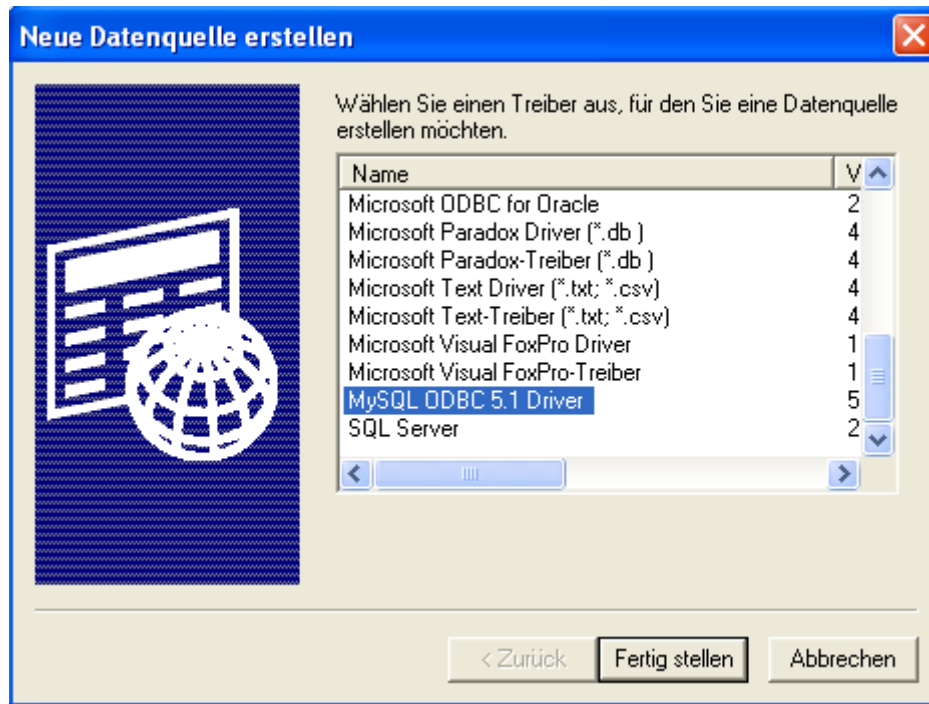
<http://www.mysql.com/products/connector/>

The screenshot shows a Mozilla Firefox browser window displaying the MySQL website. The address bar shows the URL <http://www.mysql.com/products/connector/>. The page features the MySQL logo and Sun Microsystems logo, along with the tagline "The world's most popular open source database". A navigation menu includes links for "Products", "Downloads", "Documentation", "Services", "Customers", "Why MySQL?", "News & Events", and "How to Buy". A search bar is also present. The main content area is titled "MySQL Connectors" and includes a description: "MySQL provides standards-based drivers for JDBC, ODBC, and .Net enabling developers to build database applications in their language of choice. In addition, a native C library allows developers to embed MySQL directly into their applications." Below this, a table lists various connectors developed by MySQL, each with links for "Download", "Documentation", and "Discussion".

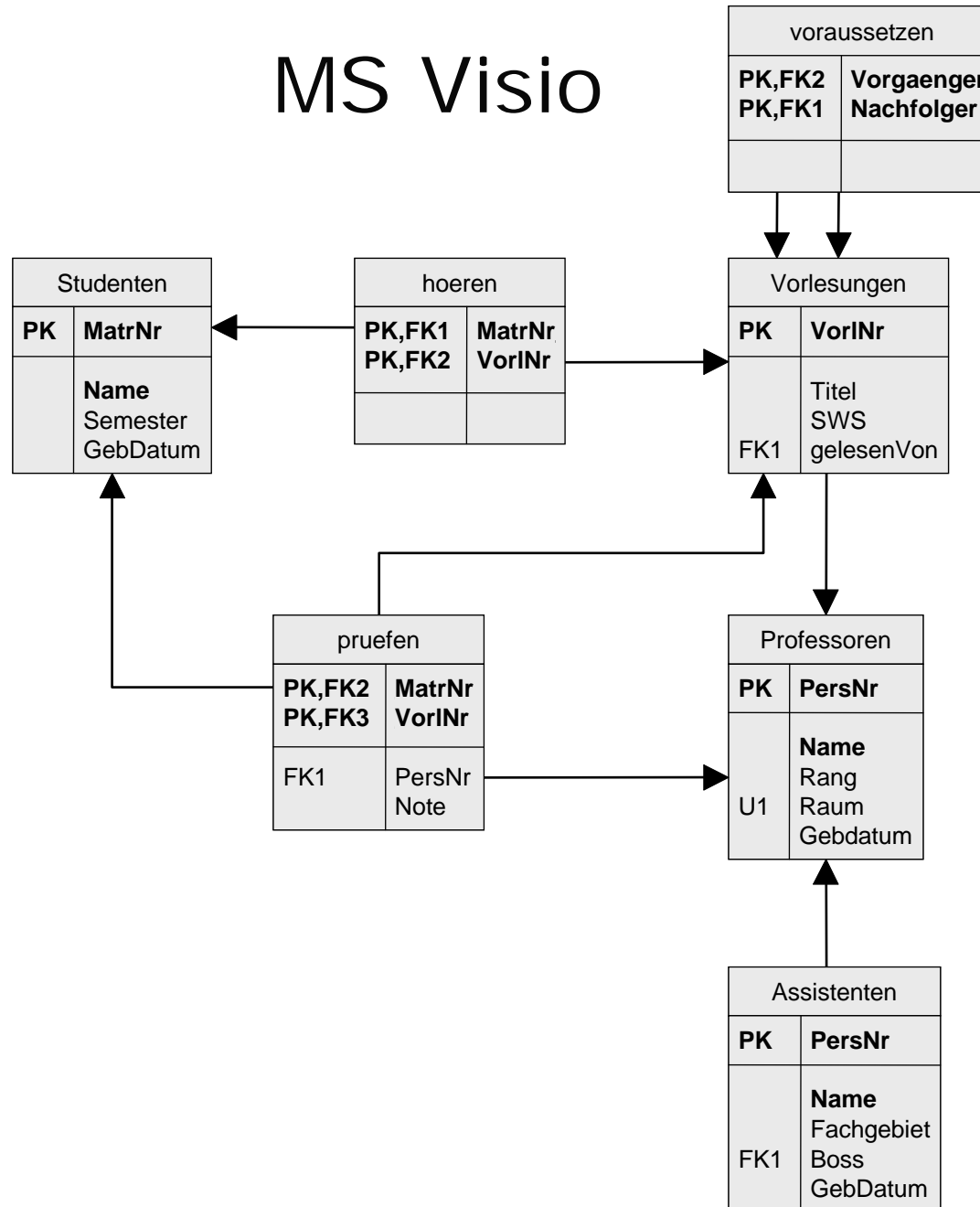
Developed by MySQL			
<b>ADO.NET Driver for MySQL (Connector/NET)</b>	<a href="#">Download</a>	<a href="#">Documentation</a>	<a href="#">Discussion</a>
<b>ODBC Driver for MySQL (Connector/ODBC)</b>	<a href="#">Download</a>	<a href="#">Documentation</a>	<a href="#">Discussion</a>
<b>JDBC Driver for MySQL (Connector/J)</b>	<a href="#">Download</a>	<a href="#">Documentation</a>	<a href="#">Discussion</a>
<b>MXJ Driver for MySQL (Connector/MXJ)</b>	<a href="#">Download</a>	<a href="#">Documentation</a>	<a href="#">Discussion</a>
<b>C++ Driver for MySQL (Connector/C++)</b>	<a href="#">Download</a>	<a href="#">Documentation</a>	<a href="#">Discussion</a>
<b>C Driver for MySQL (Connector/C)</b>	<a href="#">Download</a>	<a href="#">Documentation</a>	<a href="#">Discussion</a>
<b>C API for MySQL (mysqlclient)</b>	<a href="#">Download</a>	<a href="#">Documentation</a>	<a href="#">Discussion</a>

At the bottom of the browser window, the address bar shows the URL <http://dev.mysql.com/downloads/connector/j/>.

# Datenquelle hinzufügen



# MS Visio



# MS Access

Stand-alone Datenbanksystem

Frontend per ODBC für relationale Datenbank

- Schemadesign
- Beziehungen
- Queries
- Berichte
- Formulare
- VBA-Skript

# MS Access Tabelle

The screenshot shows the Microsoft Access 'Person : Tabelle' window. The top part displays a table structure with columns for 'Feldname', 'Felldatentyp', and 'Beschreibung'. The bottom part shows the 'Feldeigenschaften' (Field Properties) pane with the 'Allgemein' (General) tab selected. The 'Gültigkeitsregel' (Validation Rule) is set to '<math>[C2] \text{ Oder } [C3] \text{ Oder } [C4]</math>' and the 'Gültigkeitsmeldung' (Validation Message) is 'Bitte auf Rang achten !!'. A blue text box on the right explains that the validation rule restricts the values that can be entered into the field.

Feldname	Felldatentyp	Beschreibung
persnr	AutoWert	Personalnummer
name	Text	Name der Person
rang	Text	Besoldungsstufe
gebdatum	Datum/Uhrzeit	Geburtsdatum
gehalt	Währung	Gehalt
foto	OLE-Objekt	Passbild

**Feldeigenschaften**

**Allgemein** | Nachschlagen

Feldgröße: 2

Format:

Eingabeformat:

Beschriftung:

Standardwert:

Gültigkeitsregel: '<math>[C2] \text{ Oder } [C3] \text{ Oder } [C4]</math>'

Gültigkeitsmeldung: Bitte auf Rang achten !!

Eingabe erforderlich: Nein

Leere Zeichenfolge: Ja

Indiziert: Nein

Unicode-Kompression: Ja

IME-Modus: Keine Kontrolle

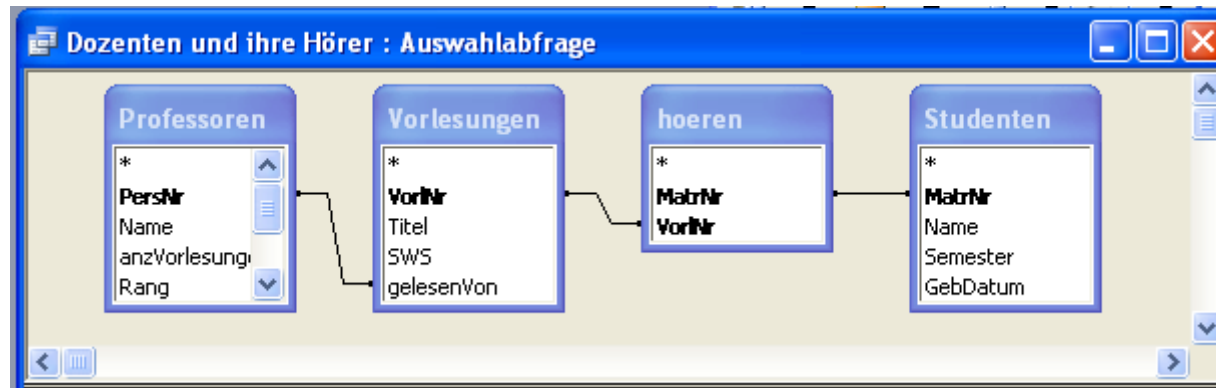
IME-Satzmodus: Keine

Smarttags:

Ein Ausdruck, der die Werte einschränkt, die in das Feld eingegeben werden können.



# MS Access Abfrage



```
SELECT Professoren.Name AS Dozenten, Studenten.Name AS Hörer  
FROM ((Vorlesungen INNER JOIN hoeren ON  
Vorlesungen.VorlNr= hoeren.VorlNr)  
INNER JOIN Studenten ON hoeren.MatrNr=Studenten.MatrNr)  
INNER JOIN Professoren ON  
Vorlesungen.gelesenVon=Professoren.PersNr;
```

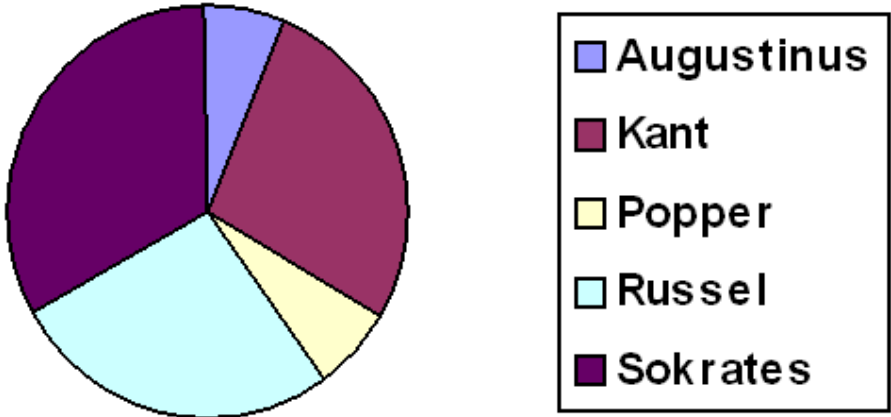
# MS Access Bericht



The screenshot shows a Microsoft Access report window titled "prof+stud.doc". The report is titled "Dozenten und ihre Hörer" in a blue, italicized font. It is divided into two columns: "Professor" and "Student".

<i>Professor</i>	<i>Student</i>
Augustinus	Feuerbach
	Jonas
Kant	Feuerbach
	Fichte
	Schopenhauer
	Theophrastos
Popper	Carnap
Russel	Carnap
Sokrates	Carnap
	Schopenhauer
	Theophrastos

Lehrbelastung



# MS Access Formular

Prof+Assi : Formular

**Stammdaten der Professoren**

PersNr:

Name:

Rang:

Raum:

**Stammdaten der zugehörigen Assistenten**

Personalnummer	Name	Fachgebiet
<input type="text" value="3005"/>	<input type="text" value="Rethikus"/>	<input type="text" value="Planetenbewegung"/>
<input type="text" value="3006"/>	<input type="text" value="Newton"/>	<input type="text" value="Keplersche Gesetze"/>
* <input type="text"/>	<input type="text"/>	<input type="text"/>

Datensatz:  von 7

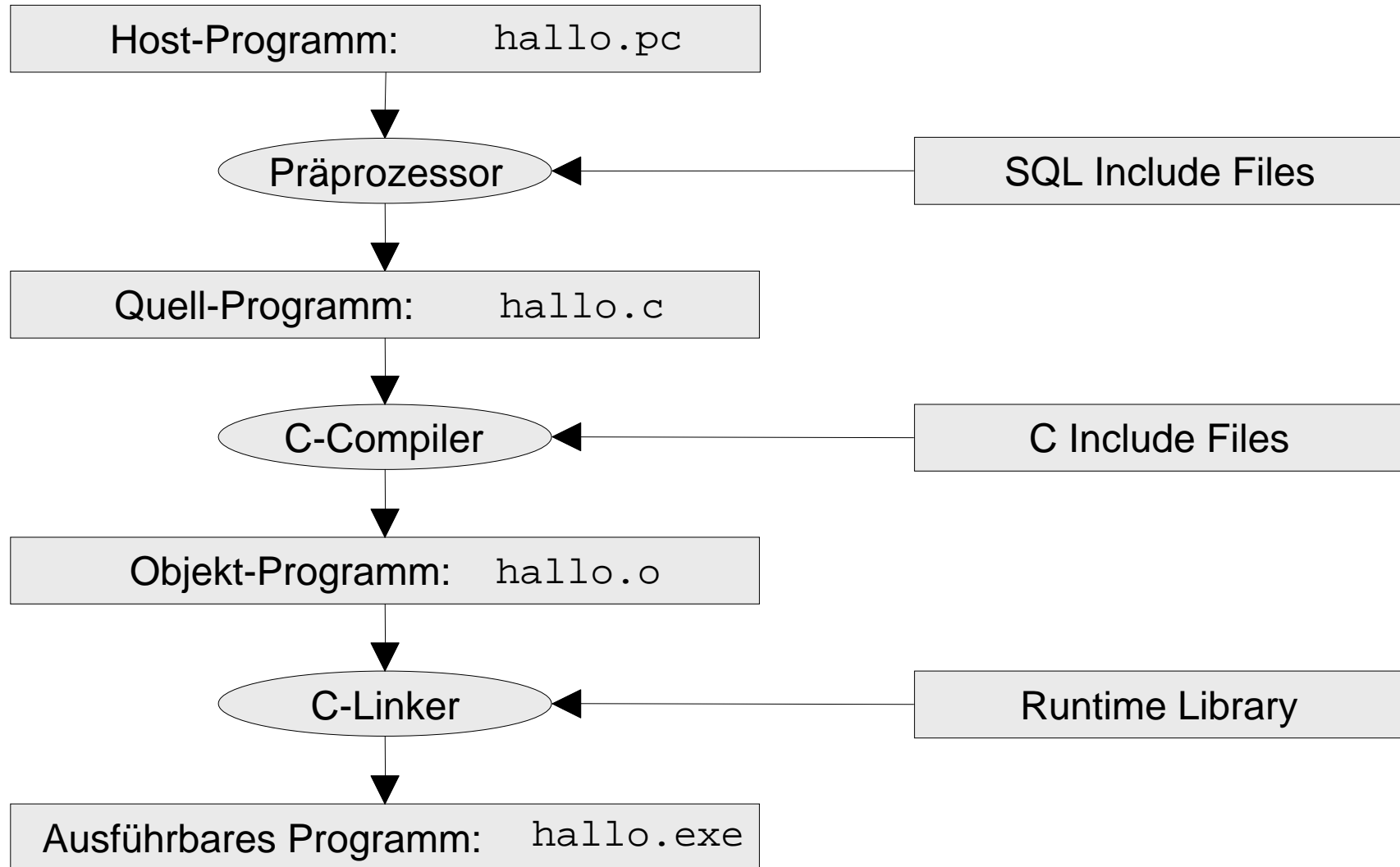
# MS Access Visual Basic for Applications

```
Private Sub berechne_Click()  
Dim rang As String  
Select Case gehaltsgruppe.Value  
Case 1  
    rang = "C2"  
Case 2  
    rang = "C3"  
Case 3  
    rang = "C4"  
Case Else  
    rang = " "  
End Select  
If rang = " "  
    Then MsgBox ("Rang fehlt !!!")  
Else  
Dim rs As Recordset  
Set rs = CurrentDb.OpenRecordset("Select name, gebdatum from professoren " & _  
"where gebdatum = (select min(gebdatum) from professoren " & _  
"where rang = '" & rang & "'")  
ausgabe.Value = rs.Fields("name").Value & ", geboren am " & rs.Fields("gebdatum")  
End If  
End Sub
```

# 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

# Embedded SQL (Microsoft)



# Hostvariable

```
int    persnr;           // Personalnummer
char   name[20];        // Name
char   rang[3];         // Rang
int    raum;            // Raum
char   gebdatum[17];    // Geburtsdatum

short  raum_ind;        // Raum-Indikator
```

Innerhalb von SQL-Statements:  
Doppelpunkt (:) voranstellen !

# Select

```
EXEC SQL SELECT persnr, raum
      INTO :persnr, :raum INDICATOR :raum_ind
      FROM Professoren
      WHERE PersNr = 2125;
```

```
if (raum_ind == -1)
printf("Personalnummer %d ohne Raumangabe \n",persnr);
```

<http://www-lehre.inf.uos.de/~dbs/2011/ESQL-Microsoft/beispiel.sqc>

<http://www-lehre.inf.uos.de/~dbs/2011/ESQL-Microsoft/beispiel.c>



# beispiel.exe

```
C:\WINNT\System32\cmd.exe
Datenträgernummer: 0112-0BC0

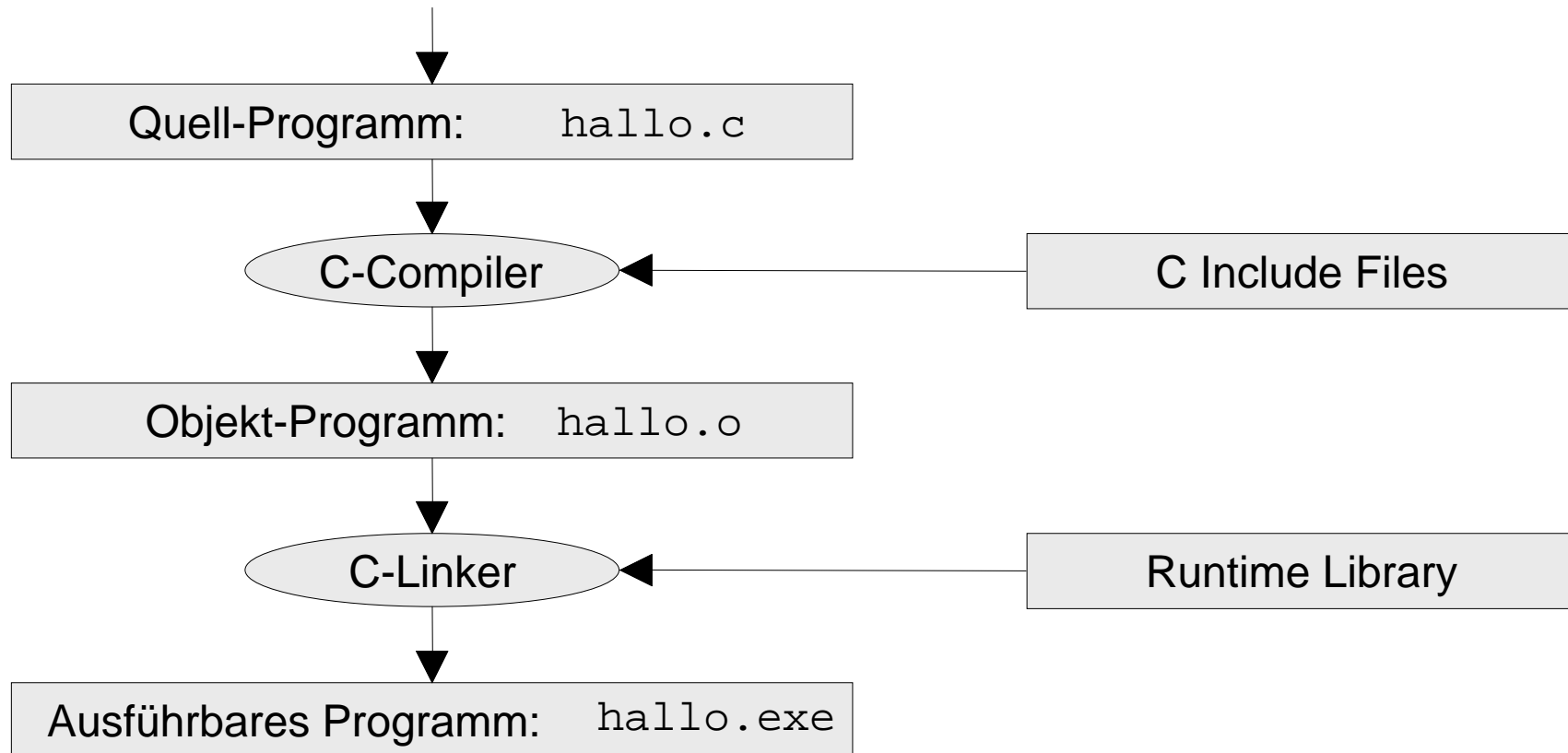
Verzeichnis von L:\dbs\2003\Developer\Skript\ESQL-Microsoft

26.05.2003  11:07      <DIR>          .
28.05.2003  09:56      <DIR>          ..
23.01.2003  14:38                272  readme.txt
23.01.2003  14:38           126.976  sqlakw32.dll
23.01.2003  14:38           162.733  unzip_esqlc.exe
23.01.2003  14:38             3.244  beispiel.sqc
23.01.2003  14:38           45.056  beispiel.exe
                5 Datei(en)          338.281 Bytes
                2 Verzeichnis(se), 10.792.992.768 Bytes frei

L:\dbs\2003\Developer\Skript\ESQL-Microsoft>beispiel
Verbindung zum SQL Server aufgebaut!
Bitte Rang eingeben: C4
Mit Rang C4 gespeichert:
2125 Sokrates      C4 226 23 08 1923  0:00
2126 Russel       C4 232 10 07 1934  0:00
2136 Curie        C4  36 10 05 1929  0:00
2137 Kant         C4   7 04 04 1950  0:00

L:\dbs\2003\Developer\Skript\ESQL-Microsoft>
```

# Embedded SQL (MySQL)



# beispiel.c [Teil 1]

```
#include <mysql.h>
#include <stdio.h>
main() {
    MYSQL      *conn;
    MYSQL_RES  *res;
    MYSQL_ROW  row;
    char *server    = "dbs.informatik.uni-osnabrueck.de";
    char *user      = "erika";
    char *password  = "mustermann";
    char *database  = "UniWeb";
    char query[80] = "select * from Professoren where rang =";
    char rang[3];
    conn = mysql_init(NULL);
    mysql_real_connect(conn, server, user, password,
                       database, 0, NULL, 0));
```

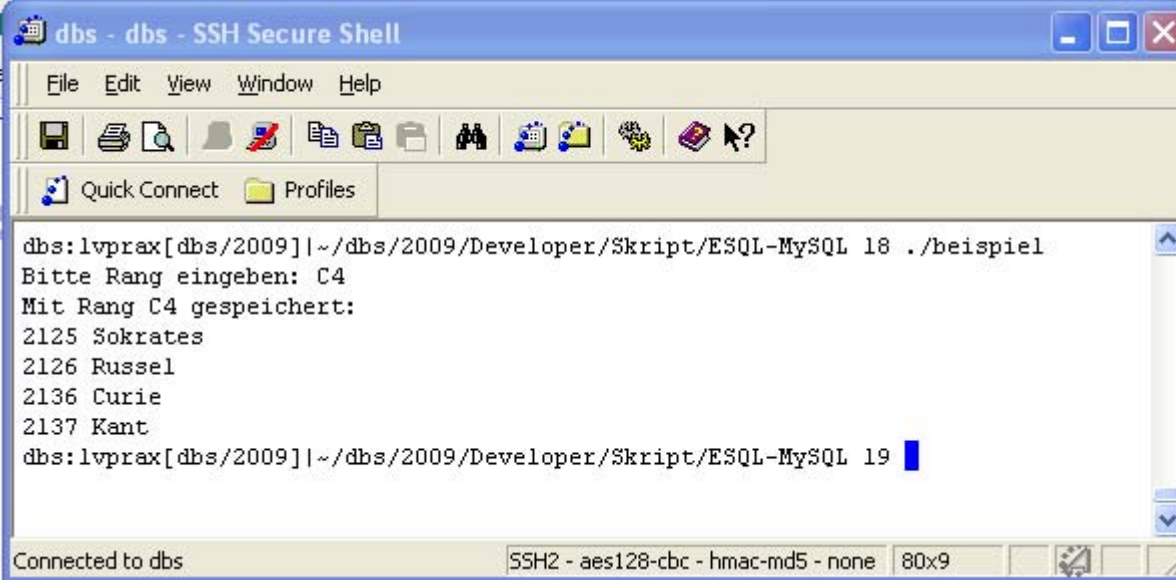
## beispiel.c [Teil 2]

```
printf("Bitte Rang eingeben: ");
scanf("%s", rang);
printf("Mit Rang %s gespeichert:\n", rang);
strcat(query, "'"); strcat(query, rang);
strcat(query, "'");
mysql_query(conn, query);
res = mysql_use_result(conn);
while ((row = mysql_fetch_row(res)) != NULL)
    printf("%s %s \n", row[0], row[1]);
mysql_free_result(res);
mysql_close(conn);
}
```

<http://www-lehre.inf.uos.de/~dbs/2011/ESQL-MySQL/beispiel1.c>

# beispiel.exe

```
gcc -o beispiel `mysql_config --cflags` beispiel.c `mysql_config --libs`
```



The screenshot shows a terminal window titled "dbs - dbs - SSH Secure Shell". The terminal output is as follows:

```
dbs:lvprax[dbs/2009]|~/dbs/2009/Developer/Skript/ESQL-MySQL 18 ./beispiel
Bitte Rang eingeben: C4
Mit Rang C4 gespeichert:
2125 Sokrates
2126 Russel
2136 Curie
2137 Kant
dbs:lvprax[dbs/2009]|~/dbs/2009/Developer/Skript/ESQL-MySQL 19
```

The terminal window includes a menu bar (File, Edit, View, Window, Help), a toolbar with various icons, and a status bar at the bottom indicating "Connected to dbs" and "SSH2 - aes128-cbc - hmac-md5 - none 80x9".

SSH

# JDBC

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

- Application
- Applet

nutzt Datenbankserver oder lokale Datei

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

# SQL versus Java

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

# ShowJDBC.java

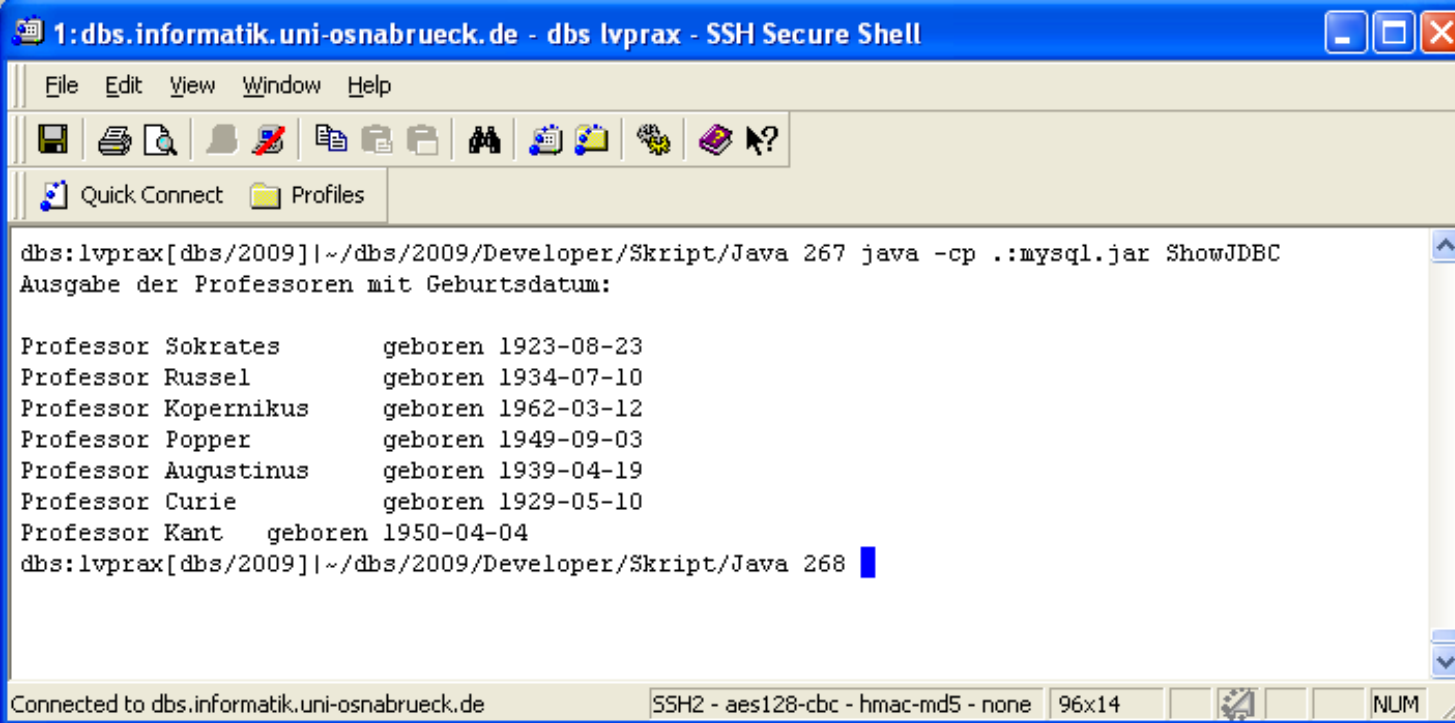
```
String url      ="jdbc:mysql://dbs.informatik.uos.de/UniWeb";
String user    ="erika";
String passwd  ="mustermann";
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection(url,user,passwd);
Statement stmt = con.createStatement();
String query   ="select * from Professoren";
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();
```



# Ausgabe von ShowJDBC



The screenshot shows a 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 prompt is `dbs:lvprax[dbs/2009]|~/dbs/2009/Developer/Skript/Java 267`. The status bar at the bottom indicates the connection is to `dbs.informatik.uni-osnabrueck.de` using `SSH2 - aes128-cbc - hmac-md5 - none` with a resolution of `96x14`.

```
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
```

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

SSH

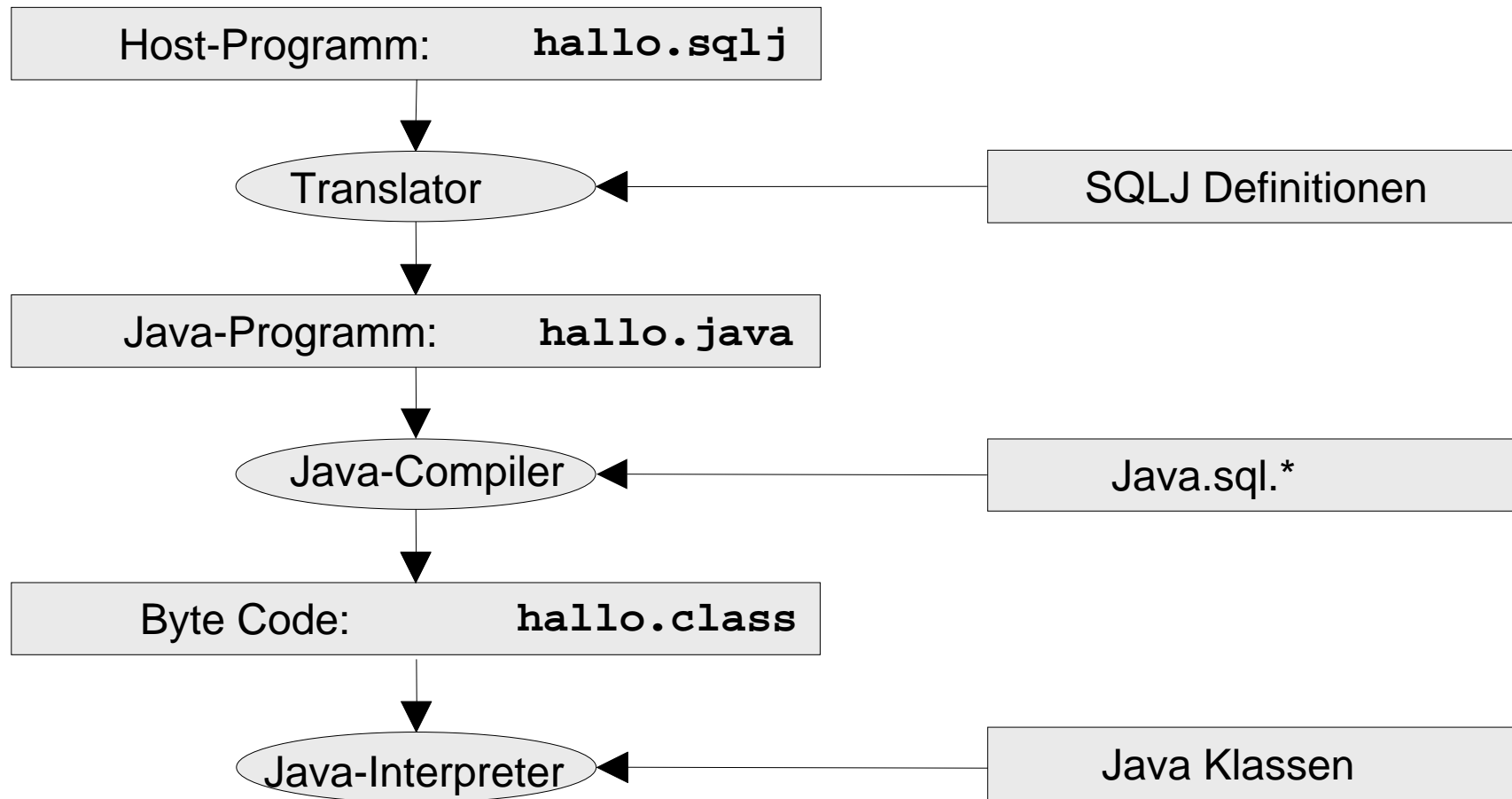
# JDBC: Update

```
...  
Statement stmt = con.createStatement();  
String query = "UPDATE professoren " +  
               "SET rang='W1' 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/2011/Java/ShowSqljHost.sqlj>

<http://www-lehre.inf.uos.de/~dbs/2011/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");  
}
```

# 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 1]

```
import java.sql.*;
public class SQLiteSchreibTest{
    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 stat = con.createStatement();
            stat.executeUpdate("drop table if exists Profs");

            stat.executeUpdate("CREATE TABLE Profs(PersNr INT PRIMARY KEY, "
                + "Name VARCHAR(20) NOT NULL, "
                + "Rang CHAR(2), "
                + "Raum INTEGER);");
```



# 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/2011/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/2011/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 1]

```
import java.sql.*;
public class HSQLSchreibTest{
    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 stat = con.createStatement();
            stat.executeUpdate("drop table if exists Profs");
            stat.executeUpdate("CREATE TABLE Profs(PersNr INT PRIMARY KEY, "
                + "Name VARCHAR(20) NOT NULL, "
                + "Rang CHAR(2), "
                + "Raum INTEGER);");
        }
    }
}
```

# 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/2011/Java/HSQLDBSchreibTest.java>

# HSQLEDBLeseTest.java [Teil 1]

```
import java.sql.*;

public class HSQLEDBLeseTest{

    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 auf Webbrowser
- Applet darf nur Verbindung aufbauen zu Rechner, von dem es gezogen wurde
- ⇒ 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/>

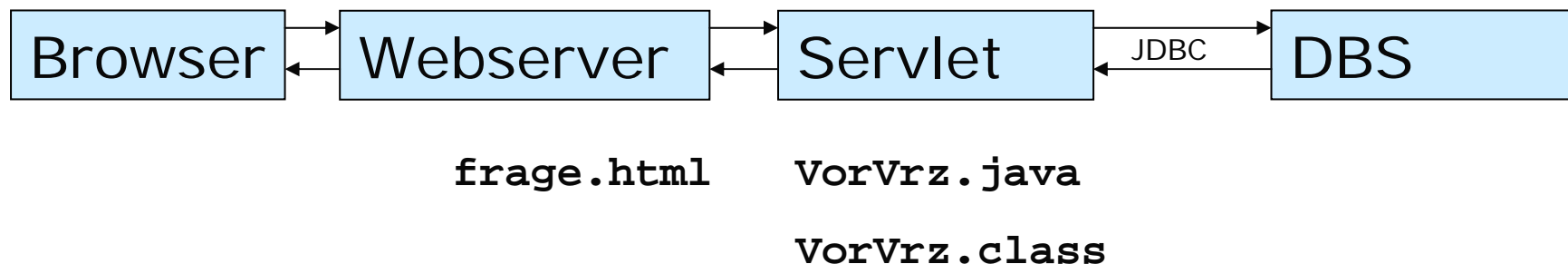
# Datenbankapplikationen

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

# Java Servlet

Java-Programm mit Klassen aus javax.servlet.\*  
läuft auf Server

- HTML-Form ausfüllen
- Argumente an Servlet schicken
- Servlet beantwortet Query mit JDBC
- Ergebnis wird als HTML zurückgeschickt



Apache Tomcat Server <http://dbs.informatik.uos.de:8180/>

# frage.html

```
<html>
  <head>
    <title>Vorlesungsverzeichnis mit Java Servlet</title>
  </head>

  <body>
    <form method="GET" action="VrlVrz">
      Bitte geben Sie den Namen eines Professors ein:
      <p><input name="professor_name" size="40">
      <p><input type="submit" value="Vorlesungen ermitteln">
    </form>

  </body>
</html>
```

# VorVrz.java (Anfang ... Ende)

```
public class VrlVrz extends HttpServlet {

    public void doGet (HttpServletRequest request,
                      HttpServletResponse response)
                      throws ServletException, IOException {
        Connection con = null;
        Statement stmt = null;
        ResultSet rs = null;
        response.setContentType("Text/html");
        PrintWriter out = response.getWriter();
        try {
            ...
            ...
        } catch(ClassNotFoundException e) {
            out.println("Datenbanktreiber nicht gefunden: " + e.getMessage());
        }
        catch(SQLException e) {
            out.println("SQLException: " + e.getMessage());
        }
        finally {
            try { if (con != null ) con.close();
            } catch (SQLException ignorieren) {}
        }
    }
}
```



# VorVrz.java (Mitte)

```
Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection(
    "jdbc:mysql://dbs.informatik.uni-osnabrueck.de/UniWeb", "erika", "mustermann");
stmt = con.createStatement();
String query = "select v.vorlnr, v.titel, v.sws " +
    "from vorlesungen v, professoren p " +
    "where v.gelesenvon = p.persnr and p.name ='" +
    request.getParameter("professor_name") + "'";
rs = stmt.executeQuery(query);
out.println("<HTML>");
out.println("<HEAD><TITLE>Java Servlet</TITLE></HEAD>");
out.println("<BODY>");
out.println("<H1>Vorlesungen von Prof. " +
    request.getParameter("professor_name") + ": </H1>");
out.println("<UL>");
while (rs.next())
    out.println("<LI>" +
        rs.getInt("VorlNr") + ": " +
        rs.getString("Titel") + " (mit " +
        rs.getInt("SWS") + " SWS)" + "</LI>");
out.println("</UL>");
out.println("<BODY></HTML>");
```

[VorVrz.java](#)

[servlet.html](#)

# SQL-Injection

Obacht: User-Eingabe nicht ungefiltert an den SQL-Interpreter weiterreichen !

```
select ... where p.name = ' " +  
request.getParameter("professor_name") + "' ;
```

Eingabe **Sokrates**

erzeugt **select ... where name='Sokrates' ;**

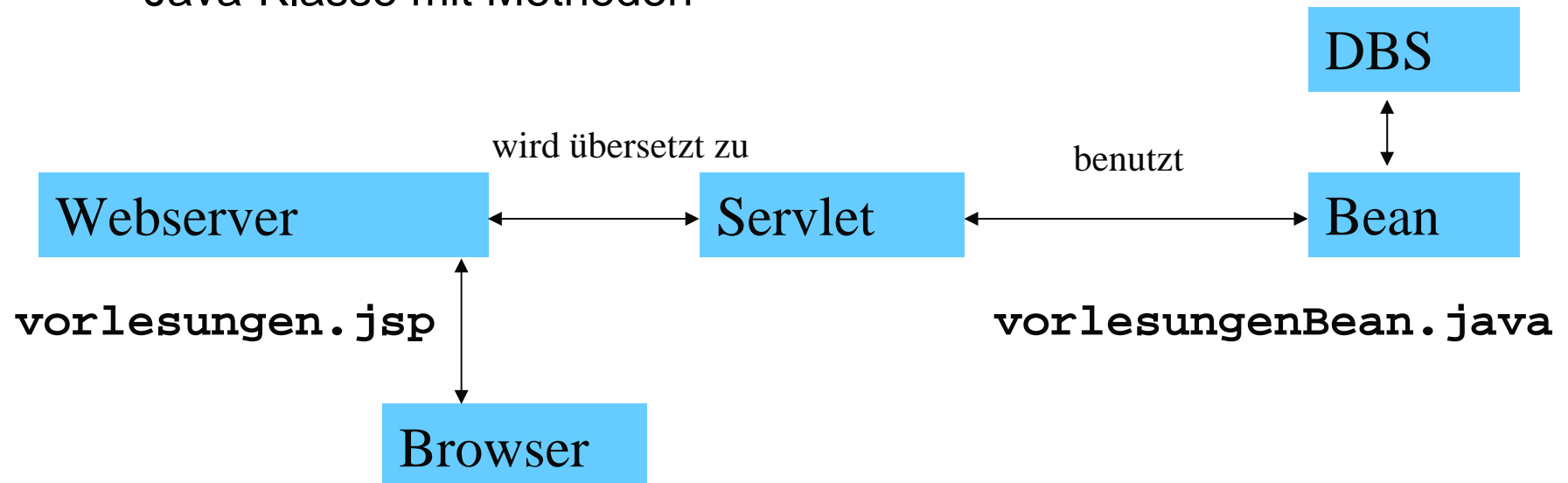
Eingabe **Sokrates';delete from Professoren where name='Kant**

erzeugt **select ... where name='Sokrates' ;  
delete from Professoren where name='Kant' ;**

# Java Server Pages

Trennung von Form und Funktionalität:

- HTML-Seite mit Aufruf von Java-Methoden
- Java-Klasse mit Methoden



# Syntax der JSP-Seite

In den HTML-Code ist eingebettet ...

<code>&lt;%@</code>	Page Direktive	<code>%&gt;</code>
<code>&lt;%!</code>	Java-Deklaration	<code>%&gt;</code>
<code>&lt;%=</code>	Java-Ausdruck	<code>%&gt;</code>
<code>&lt;%</code>	Java-Codefragment	<code>%&gt;</code>
<code>&lt;%--</code>	Kommentar	<code>%&gt;</code>

...

```
<html>
  <head><title>Hello World</title></head>
  <body>
    <h1><% prg.generiereVorlListe() %></h1>
  </body>
</html>
```

# vorlesungen.jsp

```
<%@ page import = "dbs.VorlesungenBean" %>
<jsp:useBean id="prg" class="dbs.VorlesungenBean" scope="request"/>
<jsp:setProperty name="prg" property="*" />

<html>
  <% if (prg.getProfname() == null) { %>
    <head><title>Professoren-Namen erfassen</title></head>
    <body bgcolor="DDDDDD">
      <FORM METHOD="GET">
        Bitte geben Sie den Namen eines Professors ein:<P>
        <INPUT TYPE=TEXT NAME=profname><P>
        <INPUT TYPE=SUBMIT VALUE="Vorlesungen ermitteln!">
      </FORM>
    </body>
  <% } else { %>
    <head><title>Vorlesungen ausgeben</title></head>
    <body bgcolor="DDDDDD">
      Die Vorlesungen von <%= prg.getProfname() %> lauten: <P>
      <%= prg.generiereVorlListe() %>
    </body>
  <% } %>
</html>
```

<http://dbs.informatik.uos.de:8180/vorlesungen/vorlesungen.jsp>

# VorlesungenBean.java (Teil 1)

```
package dbs; import java.sql.*;

public class VorlesungenBean {
    Connection con;
    String error;
    String profname;

    public VorlesungenBean() {
        String url = "jdbc:mysql://dbs.informatik.uos.de/UniWeb";
        String user = "erika"; String passwd = "mustermann";
        try { Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection(url, user, passwd);
        } catch(Exception e) { error = e.toString(); } }

    public void setProfname(String name) { profname = name; }

    public String getProfname() { return profname; }
```

# VorlesungenBean.java (Teil 2)

```
public String generiereVorlListe() {
    Statement stmt = null;
    ResultSet rs = null;
    if (con==null) return "Probleme mit der Datenbank: "+error+"<br/>";
    StringBuffer result = new StringBuffer();
    try { stmt = con.createStatement();
        String query = "select v.VorlNr, v.Titel, v.SWS "
        + " from Vorlesungen v, Professoren p " + " where v.gelesenVon = p.PersNr "
        + " and p.name =' " + profname + "'";
        rs = stmt.executeQuery(query);
        result.append("<ul>");
        while (rs.next())
            result.append("<li>" + rs.getInt("VorlNr") + ": "
            + rs.getString("Titel") + " (mit " + rs.getInt("SWS") + " SWS)"
            + "</li>");
        result.append("</ul>"); }
    catch(SQLException e)
    { result.append("Bei der Abfrage fuer " +profname+ " trat ein Fehler auf: "
    + e.getMessage() + "<br />"); } return result.toString();
}
```

<http://www-lehre.inf.uos.de/~dbs/2011/Java/VorlesungenBean.java>

# Active Server Pages (ASP)

- Microsoft Internet Information Server
- Visual Basic for Applications
- jetzt ASP.NET

```
<html>  
  <body>  
    <%  
      response.write("Hello World !")  
    %>  
  </body>  
</html>
```



# PHP

- Personal Home Page (seit 1995)
- Server-basierte Scriptsprache (75 % aller Serverscripte)
- Integriert in HTML-Seiten
- PEAR (PHP Extension and Application Repository, 1999)
- GD-Library (in C geschrieben, Schnittstelle zu PHP)
- JpGraph-Library (in PHP geschrieben)

<http://www.selfphp.de/>

<http://pear.php.net/>

<http://www.boutell.com/gd/>

<http://jpgraph.net/>

# matrix.php

```
<HTML>
  <HEAD><TITLE>Matrix</TITLE></HEAD>
  <BODY BGCOLOR="DDDDDD">
    <CENTER>
      <H1>Matrix</H1>
      <TABLE>
        <?
          for ($i=0; $i<10; $i++){
            echo "<tr>";
            for ($j=0; $j<10; $j++){
              echo "<TD>", $i, $j, "</TD>";
            }
            echo "</TR>\n";
          }
        ?>
      </TABLE>
    </CENTER>
  </BODY>
</HTML>
```

<http://www-lehre.inf.uos.de/~dbs/2011/PHP/matrix.php>

# eingabe.html

```
<HTML>
  <HEAD><TITLE>Eingabe</TITLE></HEAD>
  <BODY BGCOLOR="DDDDDD">
    <center>
      <H1>Hier kommt die Eingabe ...</H1>

      <FORM METHOD="GET" ACTION="ausgabe.php" >
        Bitte geben Sie eine Zahl ein:<P>
        <INPUT name="zahl" size="2" >
        <INPUT TYPE="submit" VALUE="Abschicken !">
      </FORM>

    </center>
  </BODY>
</HTML>
```

<http://www-lehre.inf.uos.de/~dbs/2011/PHP/eingabe.html>

# ausgabe.php

```
<HTML>
  <HEAD><TITLE>Ausgabe</TITLE>
</HEAD>
<BODY BGCOLOR="DDDDDD">
  <CENTER>
<H1>... und hier kommt die Ausgabe:</H1>
  <?
    $z = $_GET['zahl'];
    for ($i=1; $i<=10; $i++) {
      echo $i, " mal ", $z, " = ", $i*$z, "<BR>";
    }
  ?>
</CENTER>
</BODY>
</HTML>
```

# frage.html

```
<HTML>
  <HEAD>
    <TITLE>Frage</TITLE>
  </HEAD>
  <BODY>
    <FORM METHOD="POST" ACTION="./antwort.php">
      Bitte geben Sie Ihre SQL-Query ein:
      <P><INPUT NAME="frage" SIZE="70">
      <P>
      <INPUT TYPE="submit" VALUE="Query absetzen">
    </FORM>
  </BODY>
</HTML>
```

<http://dbs.informatik.uos.de/media2mult/php/frage.html>

<http://dbs.informatik.uos.de/media2mult/php/antwort.php?frage=select+name+from+Professoren>

# antwort.php [Teil 1]

```
<html>
  <head>
    <title>Antwort auf DB-Query</title>
  </head>

  <body bgcolor="#dddddd">
    <?php require_once 'MDB2.php';
    $dsn = array( 'phptype' => 'mysql',
                 'username' => 'erika',
                 'password' => 'mustermann',
                 'hostspec' => 'dbs',
                 'database' => 'UniWeb', );
    $con =& MDB2::connect($dsn);
    if (PEAR::isError($con)) die($con->getMessage());
    $result = $con->query($_REQUEST['frage']);
    if (PEAR::isError($result)) die($result->getMessage());
```

hat Zugriff auf GET und POST

```
$s = $result->numCols();
$header = $result->getColumnNames();
echo "<table border=\"2\" cellpadding=\"3\">\n";
echo "<tr>";
for ($i = 0; $i < $s; $i++) {
    echo "<td>".ucfirst(key($header))."</td>";
    next($header);
}
echo "</tr>\n";
while($row = $result->fetchRow()) {
    echo "<tr>";
    for($i = 0; $i < $s; $i++) echo "<td>$row[$i]</td>";
    echo "</tr>\n";
}
echo "</table>\n";
$result->free();
$con->disconnect();

?>
</body>
</html>
```

antwort.php

[Teil 2]

# assistenten.ihtml

```
<HTML>
  <HEAD><TITLE>Auflistung der Assistenten</TITLE>
</HEAD>
<BODY BGCOLOR="#BCDEEF">
  <TABLE border=1 cellspacing=4 >
    <TR>
      <TH>Name</TH><TH>Fachgebiet</TH><TH>Betreuer</TH>
    </TR>
    <!-- BEGIN row -->
    <TR>
      <TD>{Name}</TD>
      <TD>{Fachgebiet}</TD>
      <TD>{Betreuer}</TD>
    </TR>
    <!-- END row -->
  </TABLE>
</BODY>
</HTML>
```

Block

Variable



# assistenten.php [Teil 1]

```
<?php require_once 'MDB2.php';
require_once 'HTML/Template/PHPLIB.php';

$dsn = array ( "phptype" => "mysql",
               "username" => "erika",
               "password" => "mustermann",
               "hostspec" => "dbs",
               "database" => "UniWeb",
             );
$con =& MDB2::connect($dsn);
if (PEAR::isError($con)) die ($con->getMessage());
$sql = "SELECT a.name AS Name,
           a.fachgebiet AS Fachgebiet,
           p.name AS Betreuer
        FROM Assistenten a, Professoren p
        WHERE a.boss = p.persnr";

$result = $con->query($sql);
if (PEAR::isError($result)) die ($result->getMessage());
```

# assistenten.php [Teil 2]

handle für Datei

Block

handle für Block

```
$template = new HTML_Template_PHPLIB();
$template->setFile("assistenten", "assistenten.ihtml");
$template->setBlock("assistenten", "row", "rows");

while($row = $result->fetchRow(MDB2_FETCHMODE_ASSOC)) {
    foreach($row as $field => $value) {
        $template->setVar($field, $value);
    }
    $template->parse("rows", "row", true);
}
$template->pParse("output", "assistenten");
$result->free();
$con->disconnect();
?>
```

<http://dbs.informatik.uos.de/media2mult/php/assistenten.php>

# semester.ihtml

```
<html>
  <head>
    <title>Berechnung von dynamischen Grafiken</title>
  </head>
  <body>
    <table border="1">
      <tr>
        <th>Student</th><th>Studiendauer</th>
      </tr>
      <!-- BEGIN row -->
      <tr>
        <td>{name}</td><td>{grafik}</td>
      </tr>
      <!-- END row -->
    </table>
  </body>
</html>
```

# balken.php

```
<?php
$breite = $_GET['zahl'] * 10;
$hoehe = 30;
$bild = imagecreate($breite, $hoehe);
$farbe_balken = imagecolorallocate($bild, 0, 0, 255);
$farbe_schrift = imagecolorallocate($bild, 255, 255, 255);
ImageString($bild, 3, $breite - 16, 8, $_GET['zahl'],
$farbe_schrift);
header ("Content-Type: image/png");
imagepng($bild);
?>
```

Font                      x-Pos      y-Pos

<http://dbs.informatik.uos.de/media2mult/php/balken.php?zahl=8>

# semester.php [Teil 1]

```
<?php
require_once 'MDB2.php';
require_once 'HTML/Template/PHPLIB.php';
$dsn = array( 'phptype' => 'mysql',
              'username' => 'erika',
              'password' => 'mustermann',
              'hostspec' => 'dbs',
              'database' => 'UniWeb',
            );
$con =& MDB2::connect($dsn);
if (PEAR::isError($con)) die($con->getMessage());

$sql = "SELECT name, semester FROM Studenten ORDER BY name";
$result = $con->query($sql);
if (PEAR::isError($result)) die($result->getMessage());
```

# semester.php [Teil 2]

```
$template = new HTML_Template_PHPLIB();
$template->setFile("semester", "semester.ihtml");
$template->setBlock("semester", "row", "rows");

while($row = $result->fetchRow()) {
    $template->setVar("name", $row[0]);
    $template->setVar("grafik",
        "<img src=\"./balken.php?zahl=$row[1]\">");
    $template->parse("rows", "row", true);
}
$template->pParse("output", "semester");
$result->free();
$con->disconnect();
?>
```

<http://dbs.informatik.uos.de/media2mult/php/semester.php>

# lehre.html

```
<html>

  <head>
    <title>Lehrbelastung der Professoren</title>
  </head>

  <body bgcolor="silver">
    
  </body>

</html>
```

# torte.php [Teil 1]

```
<?php
require_once './jpgraph-2.3.4/src/jpgraph.php';
require_once './jpgraph-2.3.4/src/jpgraph_pie.php';
require_once 'MDB2.php';
$dsn = array( 'phptype' => 'mysql',
              'username' => 'erika',
              'password' => 'mustermann',
              'hostspec' => 'dbs',
              'database' => 'UniWeb',
            );
$con =& MDB2::connect($dsn);
if (PEAR::isError($con)) die($con->getMessage());

$sql = "SELECT name, SUM(sws) FROM Vorlesungen, Professoren
        WHERE persnr = gelesenvon GROUP BY name";
$result = $con->query($sql);
if (PEAR::isError($result)) die($result->getMessage());
```



## torte.php [Teil 2]

```
$i = 0;
while($row = $result->fetchRow()) {
    $namen[$i] = $row[0];
    $daten[$i] = $row[1];
    $i++;
}
$result->free();
$con->disconnect();
$graph = new PieGraph(600, 400, "auto");
$graph->SetShadow();
$graph->title->Set("Lehrbelastung der Professoren");
$graph->title->SetFont(FF_FONT1, FS_BOLD);
$pl = new PiePlot($daten);
$pl->SetLegends($namen);
$graph->Add($pl);
$graph->Stroke();
?>
```

<http://dbs.informatik.uos.de/media2mult/php/lehre.html>

# Zusammenfassung

MS Access



Erstellt eine Abfrage unter Verwendung des Assistenten

Embedded SQL

```
EXEC SQL select name from Professoren into :name;
```

JDBC Application

```
ResultSet rs = stmt.execute(query);
```

JDBC Applet

```
outputArea.append(rs.getString("name"));
```

Java Servlet

```
out.println("<TD>" + rs.getString("name") + "</TD>");
```

Java Server Pages

```
<TD>Vorlesung <%= prg.getProfname() %> </TD>
```

PHP

```
echo "<td>$row[$i]</td>";
```

Nächste Woche:

Ruby on Rails