

Webbasierte Anwendungen SS 2015

Java Data Base Connectivity

Dozentin: Grit Behrens
<mailto:grit.behrens@fh-bielefeld.de>

Lehrinhaltsübersicht der Vorlesungen zu WBA

1. Einführung in WBA
2. Wiederholung: Grundlagen des WWW, HTML und HTTP
3. Clientseitige Implementierungstechnologien: Javascript, DOM, Ajax, (Java-Applet)
4. Serverseitige Implementierungstechnologien: JSP, Java-Servlet
- 5. JDBC (und WBA-Abschlussprojekt)**
6. Java-WebFramework: Java Server Faces

DB-Anbindung: Inhaltsübersicht

1. Erzeugen einer Beispieldatenbank

2. Datenbanken und JDBC

3. Hinweise zum WBA-Abschlussprojekt

Erzeugen einer Beispieldatenbank

Erzeugen einer MySQL-DB (shop1):

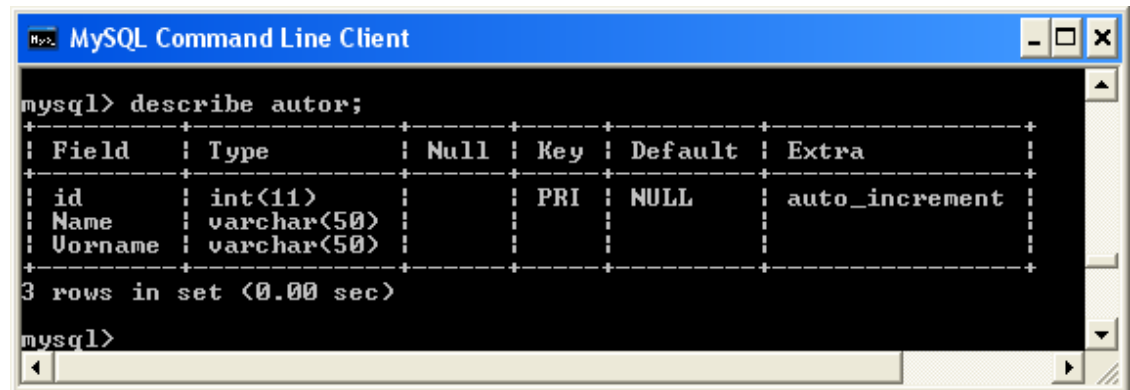
- Aufruf des Interaktionsfensters für MySQL unter Windowsfunktionalität
->programme ->MySQL ->MySQL Server 4.1 ->MySQL Command Line Client
- Befehle: `create database shop1;`
`use shop1;`

Anlegen von 2 Tabellen:

- Autor
- Buch

Autor

```
mysql> drop table Autor;
mysql> create table Autor
->(
-> id INTEGER NOT NULL AUTO_INCREMENT,
-> Name VARCHAR(50) NOT NULL,
-> Vorname VARCHAR(50) NOT NULL,
-> CONSTRAINT id PRIMARY KEY(id)
-> );
```

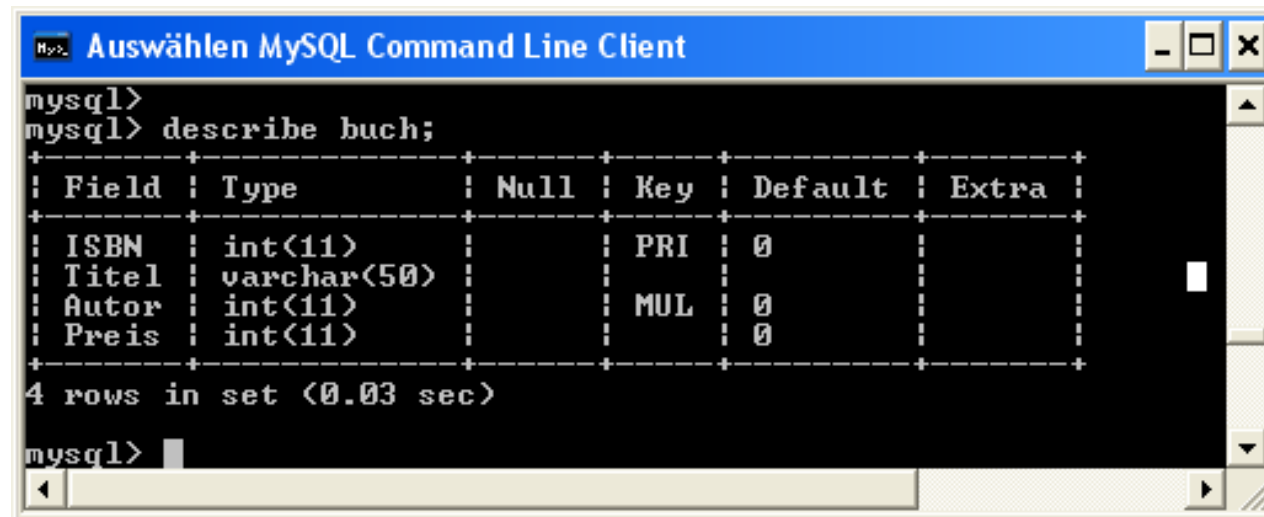


Erzeugen einer Beispieldatenbank

Anlegen von 2 Tabellen (weiter)

Buch

```
mysql> create table Buch
-> (
-> ISBN INTEGER NOT NULL,
-> Titel VARCHAR(50) NOT NULL,
-> Autor Integer NOT NULL,
-> Preis Integer NOT NULL,
-> CONSTRAINT id PRIMARY Key(ISBN) ,
-> CONSTRAINT fk_autor FOREIGN KEY(Autor) REFERENCES Autor(id)
-> );
```



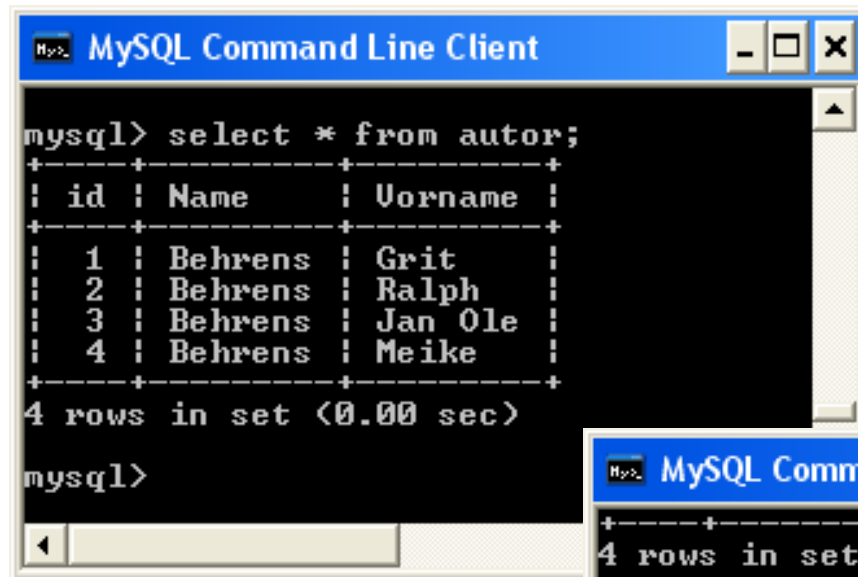
The screenshot shows a MySQL Command Line Client window titled "Auswählen MySQL Command Line Client". The command prompt shows the user has entered 'mysql> describe buch;'. The output is a table with 6 columns: Field, Type, Null, Key, Default, and Extra. It lists the fields ISBN, Titel, Autor, and Preis with their respective data types and constraints. The output indicates 4 rows in the set and took 0.03 seconds to execute.

Field	Type	Null	Key	Default	Extra
ISBN	int(11)		PRI	0	
Titel	varchar(50)				
Autor	int(11)		MUL	0	
Preis	int(11)			0	

4 rows in set (0.03 sec)

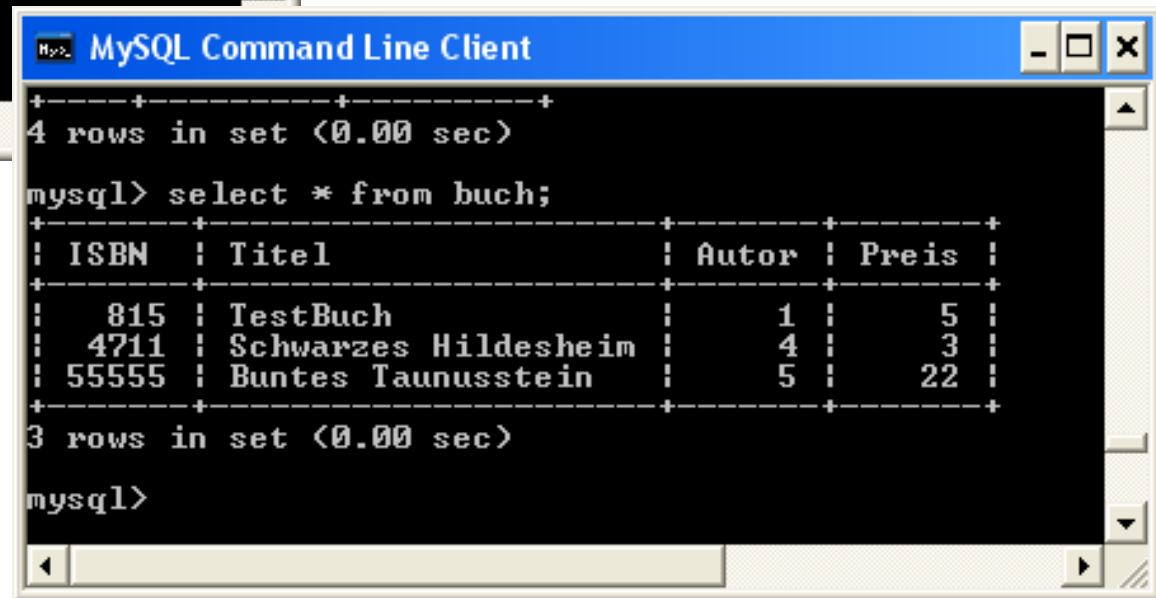
Erzeugen einer Beispieldatenbank

Tabelleninhalt für Struts-Beispielanwendung „Buchladen“



```
mysql> select * from autor;
+----+-----+-----+
| id | Name   | Vorname |
+----+-----+-----+
| 1  | Behrens | Grit    |
| 2  | Behrens | Ralph   |
| 3  | Behrens | Jan Ole |
| 4  | Behrens | Meike   |
+----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

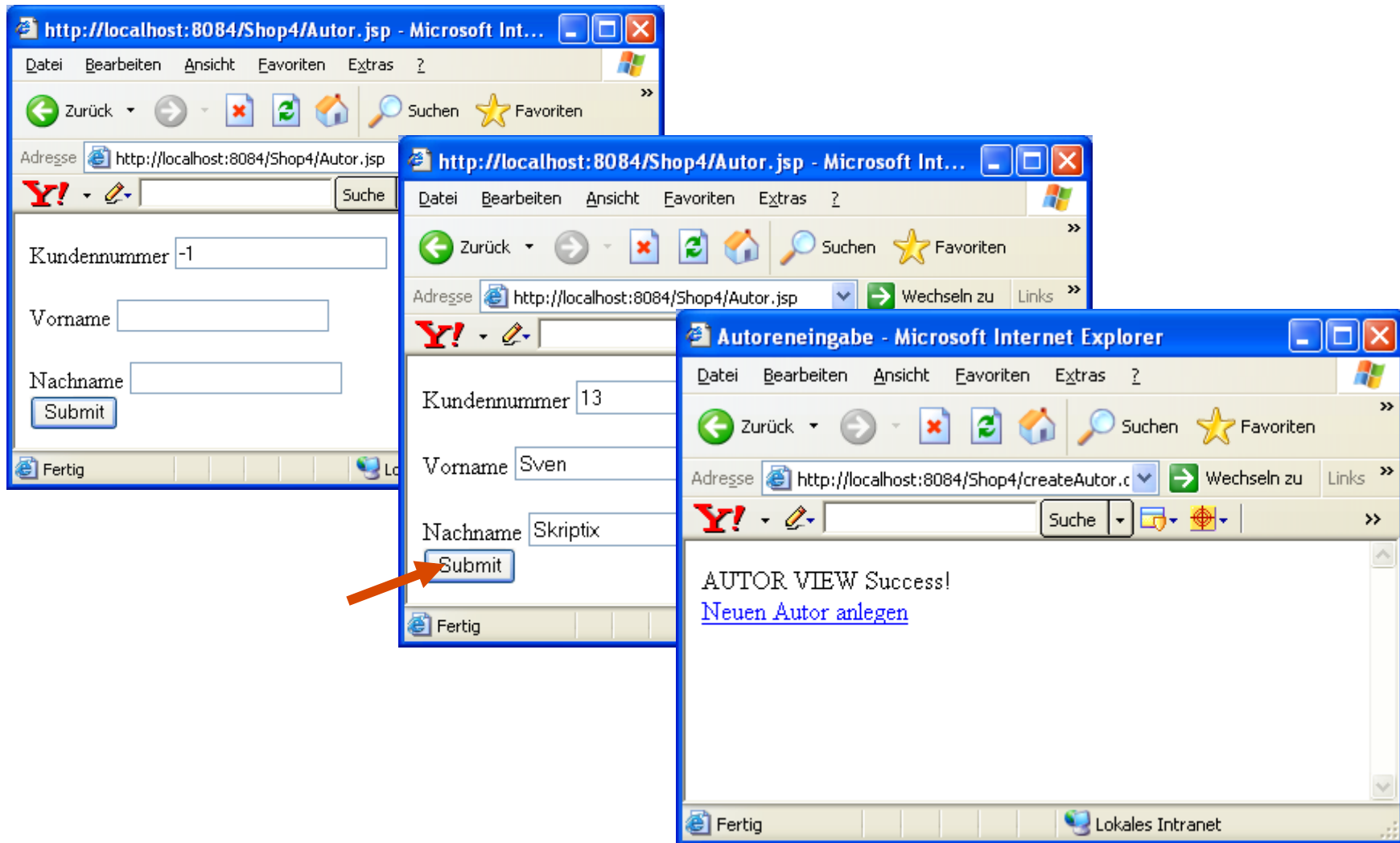


```
mysql> select * from buch;
+-----+-----+-----+-----+
| ISBN | Titel                | Autor | Preis |
+-----+-----+-----+-----+
| 815  | TestBuch             | 1     | 5     |
| 4711 | Schwarzes Hildesheim | 4     | 3     |
| 55555 | Buntes Taunusstein  | 5     | 22    |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

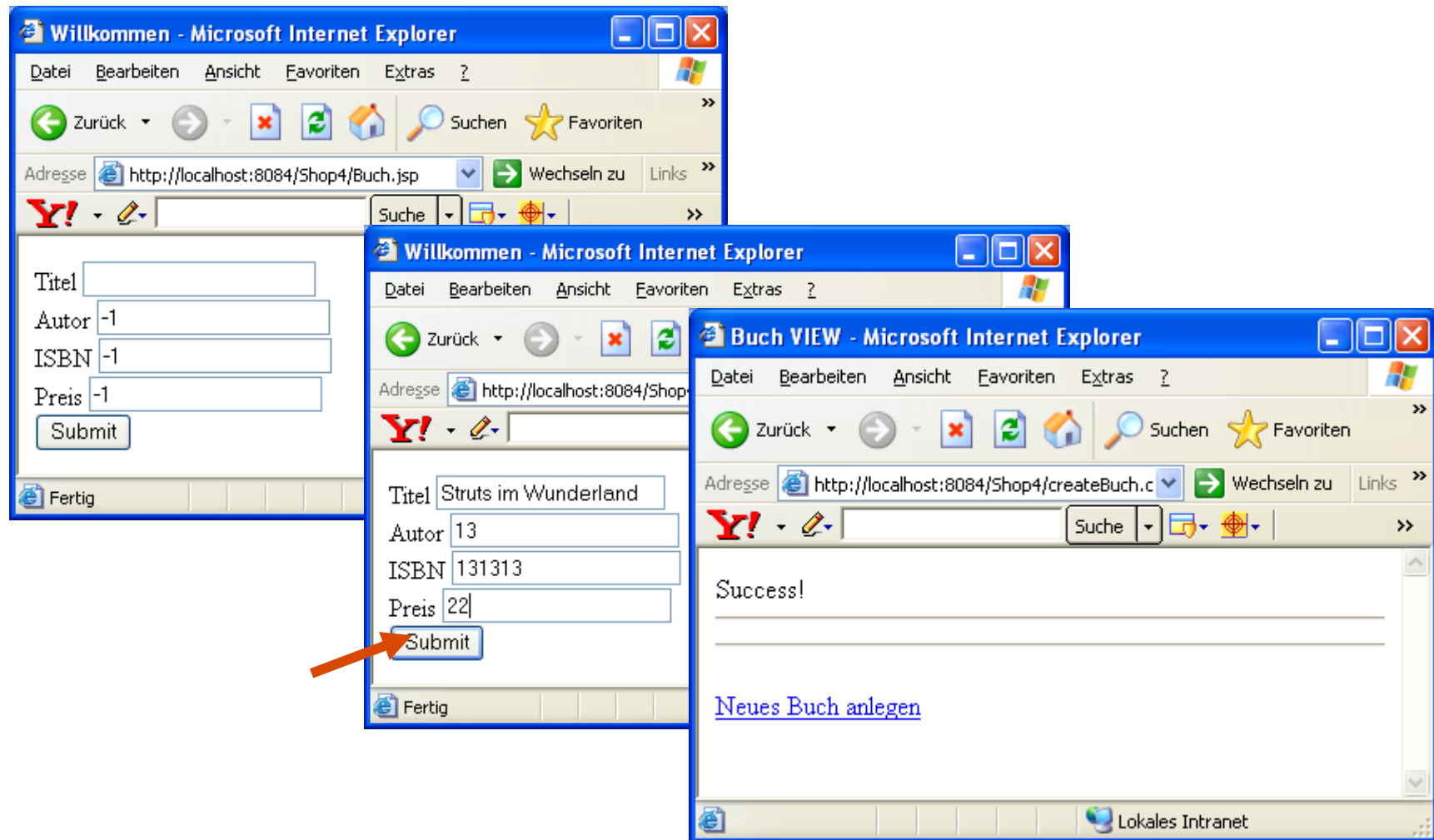
Erzeugen einer Beispieldatenbank

Eingabe in Tabelle Autor in Struts-Beispielanwendung „Buchladen“



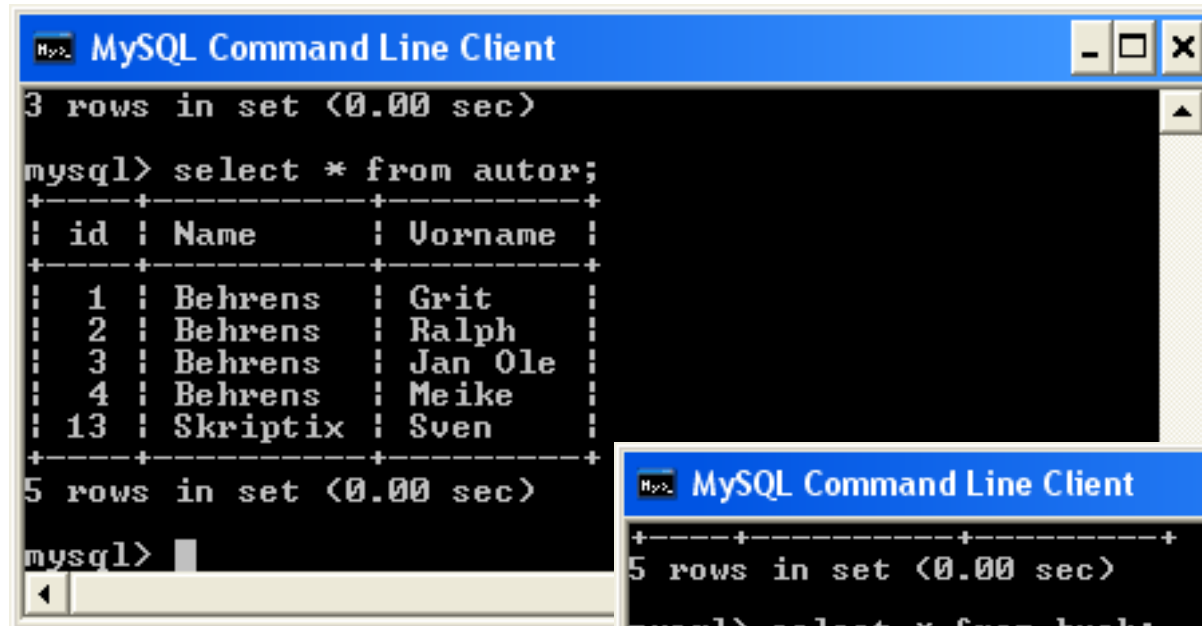
Erzeugen einer Beispieldatenbank

Eingabe in Tabelle Buch in Struts-Beispielanwendung „Buchladen“



Erzeugen einer Beispieldatenbank

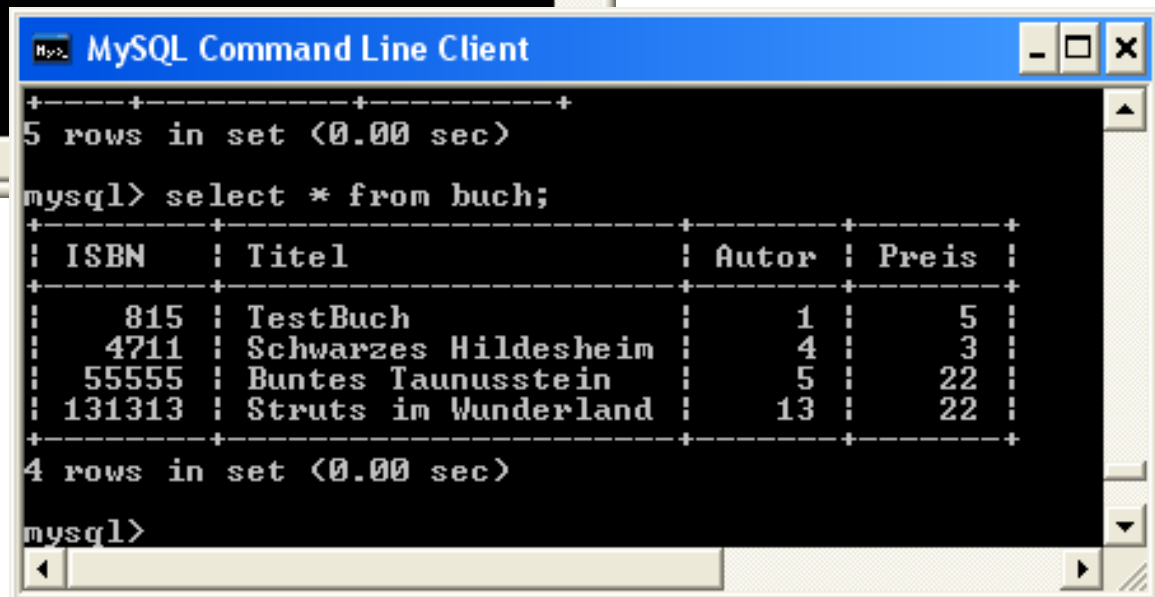
Neuer Tabelleninhalt für Struts-Beispielanwendung „Buchladen“



```
MySQL Command Line Client
3 rows in set (0.00 sec)

mysql> select * from autor;
+----+-----+-----+
| id | Name   | Vorname |
+----+-----+-----+
| 1  | Behrens | Grit    |
| 2  | Behrens | Ralph   |
| 3  | Behrens | Jan Ole |
| 4  | Behrens | Meike   |
| 13 | Skriptix | Sven    |
+----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```



```
MySQL Command Line Client
5 rows in set (0.00 sec)

mysql> select * from buch;
+-----+-----+-----+-----+
| ISBN | Titel                | Autor | Preis |
+-----+-----+-----+-----+
| 815  | TestBuch             | 1     | 5     |
| 4711 | Schwarzes Hildesheim | 4     | 3     |
| 55555 | Buntes Taunusstein  | 5     | 22    |
| 131313 | Struts im Wunderland | 13    | 22    |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

DB-Anbindung: Inhaltsübersicht

1. Erzeugen einer Beispieldatenbank

2. Datenbanken und JDBC

3. Hinweise zum WBA-Abschlussprojekt

Datenbanken und JDBC

Datenbanken und *JDBC* (Java Database Connectivity):

- *JDBC-API* setzt auf Java-Basisklassen auf (Paket `java.sql`)
 - im *JDK* (Java Development Kit) enthalten
 - stellt relationale Datenbankobjekte bereit
 - ermöglicht Datenbankzugriff aus Java-Applikationen
-
- Vorgehen in 7 Schritten:
 1. Importieren der notwendigen Klassen
 2. Laden des JDBC-Treibers
 3. Connect zur Datenbank
 4. Erzeugen eines Statements
 5. Ausführen eines Statements
 6. Auswerten des Ergebnisses
 7. Abmelden von der DB

Datenbanken und JDBC

1) Importieren der notwendigen Klassen:

Syntax: `import java.sql.*`

2) Laden des JDBC-Treibers:

Syntax: `Class.forName("oracle.jdbc.driver.OracleDriver");`

- Laden eines Datenbanktreibers, welcher die Anweisungen für jeweiliges DB-System umsetzt
- Classloader mit Methode `Class.forName();`

3) Connect zur Datenbank:

Syntax: `Connection my_con = DriverManager.getConnection(db_url, username, password);`

- Methode `getConnection` der Klasse `DriverManager` aus `java.sql.*`
- Parameter: `db_url`- url der DB, zu der Verbindung erstellt werden soll
`username`- Anmeldenname auf der DB
`password`- Passwort für die DB-Anmeldung

Datenbanken und JDBC

4) Erzeugen eines Statements:

Syntax: `Statement my_stmt = my_con.createStatement();`

- Methode `createStatement` erfolgt für Objekt der Klasse `Connection` (bestehende Verbindung)

5) Ausführen eines Statements:

Syntax:

`ResultSet my_result = my_stmt.executeQuery („SELECT * FROM TAB“);`
od. `int my_result = my_stmt.executeUpdate („UPDATE TAB SET ...“);`

- `executeQuery()` für Abfragen
- `executeUpdate()` für Update, Insert oder Delete
- bezieht sich immer auf Objekt der Klasse `statement` (bestehendes Statement `my_stmt`)
- `ResultSet` – Ergebnis in Form einer Tabelle

Datenbanken und JDBC

6) Auswerten des Abfrageergebnisses:

- Typ **ResultSet** beinhaltet spezifische Ergebnistabelle für jeweilige konkrete Anfrage
- **ResultSet** stellt *get-Methoden* zur Datenauswertung bereit

z.B.:

```
String sql = "SELECT * FROM autor WHERE id=" + id;
ResultSet r =createConnection().createStatement().executeQuery(sql);
if (r.next())
{
    name      = r.getString("Name");
    vorname   = r.getString("Vorname");
}
```

7) Abmelden von der DB:

- Aufrufen der Methode close für das entsprechende Verbindungsobjekt :
my_con.close();