Datenbanken

Steilkurs

Software-Praktikum im Grundstudium
WS 2004/2005

Dipl.-Inform. Michael Kirchhof
Dipl.-Inform. Bodo Kraft

Department of Computer Science III
Software Engineering
Ahornstr. 55
52074 Aachen, Germany

http://www-i3.informatik.rwth-aachen.de

Gliederung

- Relationales Datenmodell
- Anfragesprache SQL
- Datenbankzugriff unter JAVA mit JDBC

Relationale Datenbanken

- Daten sind organisiert in Relationen (= Tabellen)

key	name	vorname	email
0001	Becker	Simon	sbecker@i3.informatik.rwth-aachen.de
0002	Schleicher	Ansgar	schleich@i3.informatik.rwth-aachen.de

- jeder Spalte in einer Datenbank ist ein Wertebereich zugeordnet (vgl. Relation in der Mathematik)

```
(Long X String[50] X String[50] X String[50])
```

- eine oder mehrere Spalten bilden zusammen den Primärschlüssel, d.h. jeder Wert kommt nur einmal vor, mit diesem können Datensätze identifiziert werden
- eine Datenbank besteht aus mehreren Tabellen

Warum mehrere Tabellen?

- Personen mit mehreren email-Adressen

Tabelle personen		
key	*name*	*vorname*
0001	Becker	Simon
0002	Schleicher	Ansgar

"Referentielle Integrität"

"Trigger"

Tabelle emails		
key	*person_key*	*email*
0001	0001	sbecker@i3.informatik.rwth-aachen.de
0002	0001	sbecker@cs.rwth-aachen.de
003	0002	schleich@i3.informatik.rwth-aachen.de

Anfragen mit SQL

Namensliste erstellen:

gewünschte Felder

Sortierung, DESC=absteigend

```
SELECT name, vorname FROM personen ORDER BY name DESC
```

Quell-Tabelle

Ergebnis	
name	*vorname*
Schleicher	Ansgar
Becker	Simon

email-Adressen zur Person mit key 0002:

```
SELECT email FROM emails WHERE person_key=0002
```

Ergebnis
email
schleich@i3.informatik.rwth-aachen.de

Joins über mehrere Tabellen

alle Personen mit allen email-Adressen:

mehrere Tabellen

```
SELECT name, vorname, email FROM personen, emails
WHERE personen.key=email.person_key ORDER BY name
```

Feldname nicht eindeutig: Tabellennamen mit angeben

Verknüpfung zwischen den Tabellen

Ergebnis		
name	*vorname*	*email*
Schleicher	Ansgar	schleich@i3.informatik.rwth-aachen.de
Becker	Simon	sbecker@i3.informatik.rwth-aachen.de
Becker	Simon	sbecker@cs.rwth-aachen.de

Joins über mehrere Tabellen (2)

alle email-Adressen zu Becker:

```
SELECT email FROM personen, emails WHERE
personen.key=email.person_key and name="Becker"
```

Ergebnis
email
sbecker@i3.informatik.rwth-aachen.de
sbecker@cs.rwth-aachen.de

Datensätze ändern mit SQL

Vor- und Nachnamen für Person 0001 ändern:

```
UPDATE personen SET name='Bäcker', vorname='Simon M.'
WHERE key=0001
```

Neue Person einfügen:

```
INSERT INTO personen (key, name, vorname) VALUES
(0001, "Westfechtel", "Bernhard")
```

Person löschen:

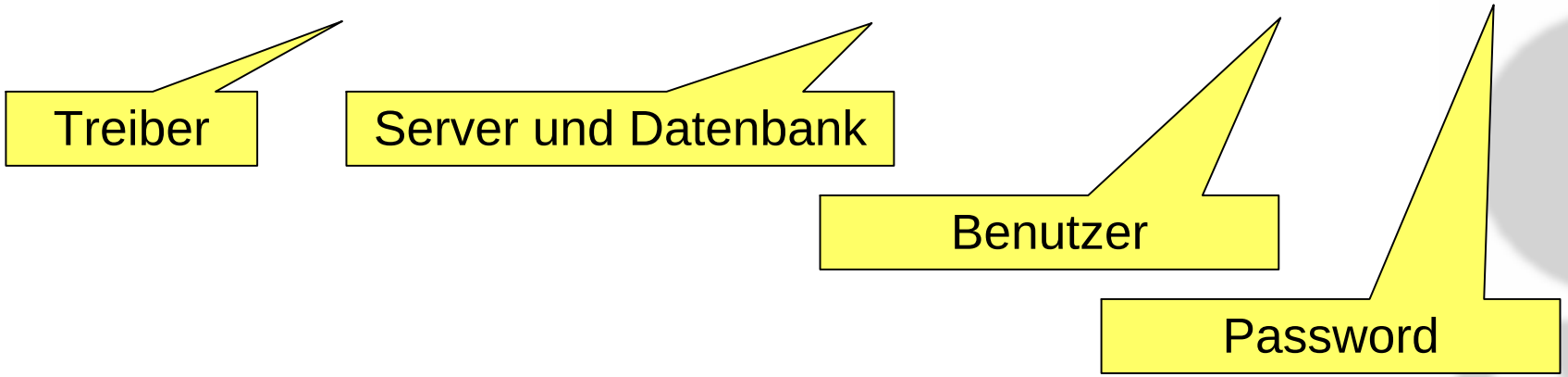
```
DELETE FROM personen WHERE key=0001
```

später: Ändern und Hinzufügen von Tabellendefinitionen

JDBC

- Java Database Connectivity
- Verbindung herstellen (hier für PostgreSQL-Datenbank):

```
Class.forName("org.postgresql.Driver");
con = DriverManager.getConnection(
   "jdbc:postgresql://salieri/spgs0405DB", "user", "pword");
```



- Anmerkung: SQLException muss angefangen werden
- Treiber: ~spgs0405/lib/javax/jar/lib/psql-jdbc.jar ==> /lib

Abfragen mit JDBC

Über die **`Connection con`** können Anfragen an die Datenbank gesendet werden:

```
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT name FROM personen
ORDER BY name");
```

Über das **`ResultSet rs`** kann jetzt auf das Ergebnis zugegriffen werden:

```
while (rs.next()) {
   System.out.println(rs.getString("name"));
}
```

geht einen Datensatz weiter, liefert **`true`**, wenn vorhanden

gibt den Inhalt eines Feldes für den aktuellen Datensatz, gibt es auch für andere Datentypen

Auch hier: Exception-Handling!

Daten ändern mit JDBC

Datensatz ändern:

```
Statement stmt = con.createStatement();
stmt.executeUpdate("UPDATE personen SET name='Bäcker',
   vorname='Simon M.' WHERE key=0001");
```

Datensatz einfügen:

```
Statement stmt = con.createStatement();
stmt.executeUpdate("INSERT INTO personen (key, name,
vorname) VALUES (0001, 'Westfechtel', 'Bernhard') ");
```

Prepared Statements

```
PreparedStatement updatePerson = null;

updatePerson = con.prepareStatement("UPDATE personen SET"
+
    " name=?, vorname=? WHERE key=?");

updatePerson.clearParameters();
updatePerson.setString(1, "Bäcker");
updatePerson.setString(2, "Simon M.");
updatePerson.setLong(3, 1);
updatePerson.executeUpdate();
```

speichert vorbereitete SQL-Anweisung

SQL-Anweisung, Parameter über "?"

über Reihenfolge setzen

Ausführen

- Werden von der Datenbank vorbereitet ==> Performance
- Hier: Setzen der Parameter über komfortable Funktionen