



Wintersemester 2009/10

# Grundlagen und Konzepte der Softwareentwicklung

## Datenbankzugriffe

### SQL (und JDBC)

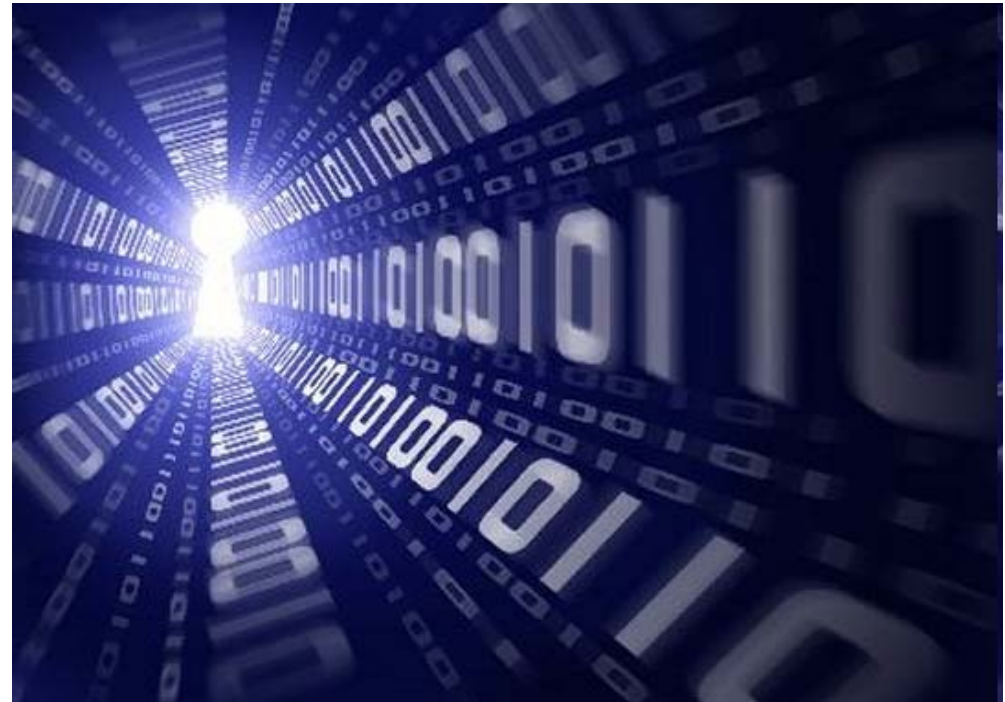
Bernd Resch  
bernd.resch@sbg.ac.at

17. Dezember 2009



# Übersicht

- Organisatorisches
- Summary der letzten Einheit
- SQL
- (JDBC)
- Zusammenfassung



# Organisatorisches – Themen Gruppenarbeit

---

- Java – **22. Oktober 2009**  
→ Lance Firmhofer, Christian Bews
- UML – Unified Modelling Language – **5. November 2009**  
→ Doris Kotzuwan, Georg Goschnik
- GeoTools – **19. November 2009**
- GeoServer – **19. November 2009**  
→ Castellazzi, Fürthauer, Neuwirth
- HTML – **3. Dezember 2009**  
→ Roland Riezinger, Daniela Lassacher
- Java Server Pages – **3. Dezember 2009**  
→ Mariana Belgiu

# Organisatorisches – Themen Gruppenarbeit

---

- Skriptsprachen (Python, Perl) – **17. Dezember 2009**  
→ Judith Huber
- Datenbankzugriffe – **17. Dezember 2009**  
→ Günter Gruber, Stefan Mayer
- Google Earth – **14. Jänner 2010**  
→ David Powell, Gerald Reischenböck, Roman Lindenthaler
- Virtual Earth API – **14. Jänner 2010**  
→ Daniela Voit
- Google Maps API – **14. Jänner 2010**  
→ Michael Haider, Klaus Paccagnel, Andreas Richter

---

# Summary der letzten Einheit

---

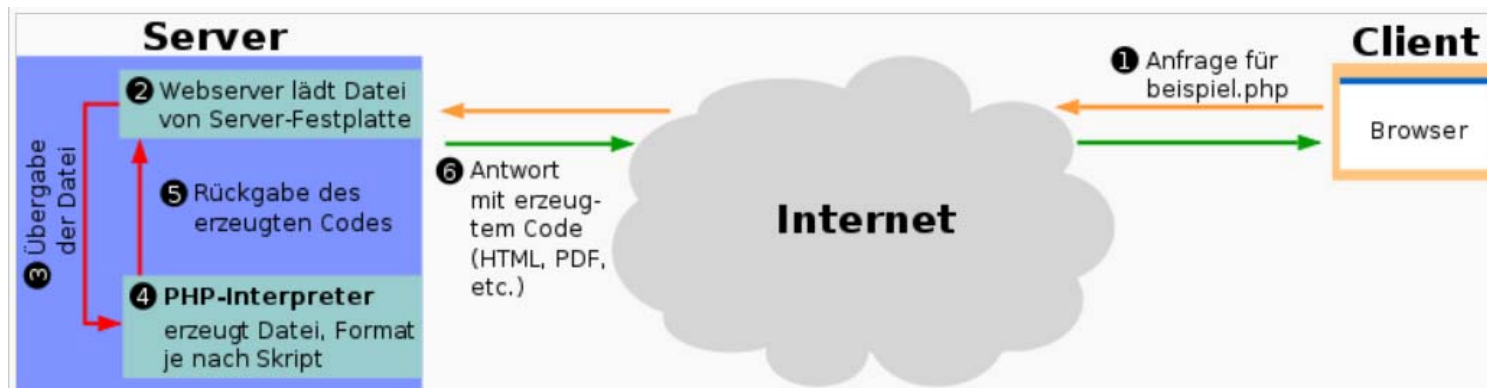
# Skriptsprachen

---

- Für kleine, überschaubare Programmieraufgaben
- Schnelle Erstellung von kleinen Programmen
- Problem: keine Compilermöglichkeit
- „Interpreter“

# Skriptsprachen ::: PHP

- Hypertext Preprocessor
- Syntax an C angelehnt
- Open-source Software



Quelle: <http://tu-chemnitz.de>

---

# Datenbankzugriffe

---

# SQL ::: Basics

---

- Structured Query Language
- Standardisierte Sprache für Datenbankzugriffe und –manipulationen
  - ◆ Data Manipulation Language
  - ◆ Data Definition Language
  - ◆ Data Control Language

# SQL ::: Kommandos

<b>SELECT</b>	<b>Data retrieval</b>
<b>INSERT</b> <b>UPDATE</b> <b>DELETE</b> <b>MERGE</b>	<b>Data manipulation language (DML)</b>
<b>CREATE</b> <b>ALTER</b> <b>DROP</b> <b>RENAME</b> <b>TRUNCATE</b>	<b>Data definition language (DDL)</b>
<b>COMMIT</b> <b>ROLLBACK</b> <b>SAVEPOINT</b>	<b>Transaction control</b>
<b>GRANT</b> <b>REVOKE</b>	<b>Data control language (DCL)</b>

# SQL ::: Funktionalität

---

- Datenabfragen an Datenbanken
- INSERT/UPDATE/DELETE von Datensätzen
- Erstellen einer neuen Datenbank
- Erstellen von Tabellen in einer Datenbank
- Stored Procedures
- Views
- Berechtigungen

# SQL ::: Beispiel

- **CREATE TABLE** Persons  
(  
    P\_Id int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
)

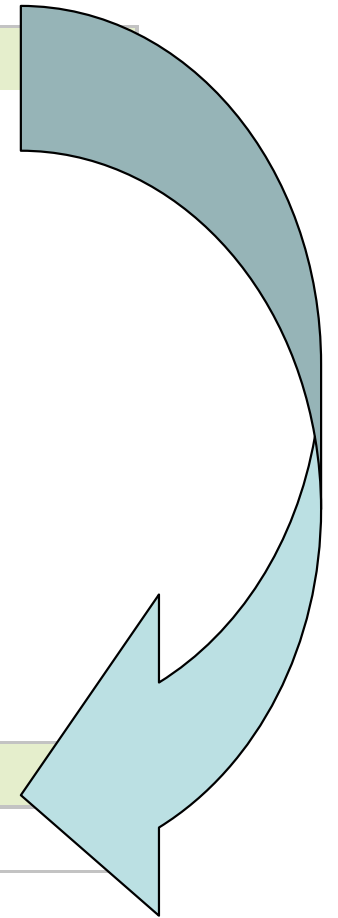
P_Id	LastName	FirstName	Address	City
------	----------	-----------	---------	------

# SQL ::: Beispiel

P_Id	LastName	FirstName	Address	City
------	----------	-----------	---------	------

- **INSERT INTO** Persons  
**VALUES** (  
    1,  
    'Hansen',  
    'Ola',  
    'Timoteivn 10',  
    'Sandnes'  
)

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes



# SQL ::: Beispiel

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

- **SELECT \***  
**FROM Persons**  
**WHERE City='Sandnes'**

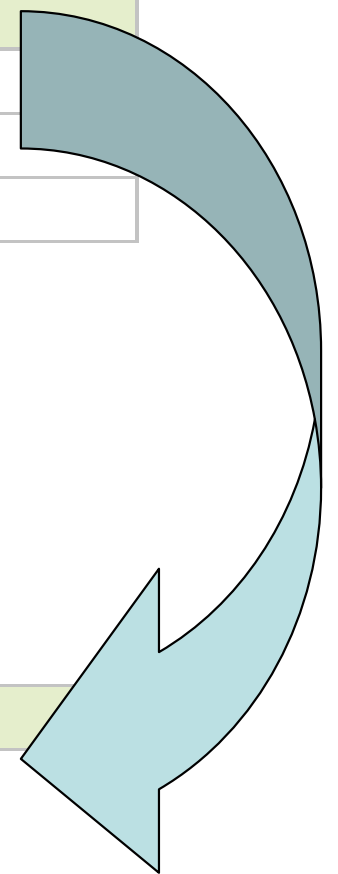
P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes

# SQL ::: Beispiel

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

- SELECT** \*  
**FROM** Persons  
**WHERE** City='Sandnes'  
**AND** LastName='Hansen'

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes



# JDBC ::: Verbindungsaufbau

```
1  import java.sql.DriverManager;
2  import java.sql.SQLException;
3  import com.mysql.jdbc.Connection;
4
5  public class JDBCConnect {
6      public static void main(String[] args) {
7          try {
8              Class.forName("com.mysql.jdbc.Driver");
9          } catch (ClassNotFoundException e) {
10             System.err.println("Driver class not found");
11             e.printStackTrace();
12         }
13         Connection con = null;
14         try {
15             con =
16                 (Connection) DriverManager.getConnection(
17                     "jdbc:mysql://localhost/jdbctest/",
18                     "mario",
19                     "thePassword");
20         } catch (SQLException e1) {
21             System.err.println("Error establishing database connection");
22             Throwable t = e1;
23             while (t != null) {
24                 System.err.println("Type: " + t.getClass().getName());
25                 System.err.println("Message: " + t.getMessage());
26                 System.err.println("-----");
27                 t = t.getCause();
28             }
29         }
30     }
31 }
32
```

# JDBC ::: Abfragebeispiel

```
import java.sql.*;
class JDBCQuery {
    public static void main( String args[] ){

        // Verbindungsaufbau → s. vorhergehende Folie
        ...

        // Erzeugen des „Statement“-Objektes für diese Verbindung
        Statement stmt = conn.createStatement();

        // Absetzen der Abfrage
        ResultSet rs = stmt.executeQuery("SELECT * FROM Persons WHERE
        City='Sandnes'" );

        // Auslesen der Ergebnisdatensätze
        while(rs.next())
            System.out.println(rs.getString(1));

        // Schließen von ResultSet, Statement und Connection
        rs.close();
        stmt.close();
        conn.close();
    }
}
```

---

# Zusammenfassung

---

# SQL ::: Funktionalität

---

- Datenabfragen an Datenbanken
- INSERT/UPDATE/DELETE von Datensätzen
- Erstellen einer neuen Datenbank
- Erstellen von Tabellen in einer Datenbank
- Stored Procedures
- Views
- Berechtigungen

## Referenzen

---

- Forta, Ben (2004) Sams Teach Yourself SQL in 10 Minutes. 3rd Edition, 246 pp., ISBN-10 0672325675, Sams, 10. April 2004.



Wintersemester 2009/10

# Grundlagen und Konzepte der Softwareentwicklung

## Datenbankzugriffe

### SQL (und JDBC)

Bernd Resch  
bernd.resch@sbg.ac.at

17. Dezember 2009

