



UNIVERSITÄT
LEIPZIG

Relationales Datenbank-Praktikum SS 2019

Universität Leipzig, Institut für Informatik
Abteilung Datenbanken

V.Christen, M.Franke, M. Nentwig, Z. Sehili, J. Zschache



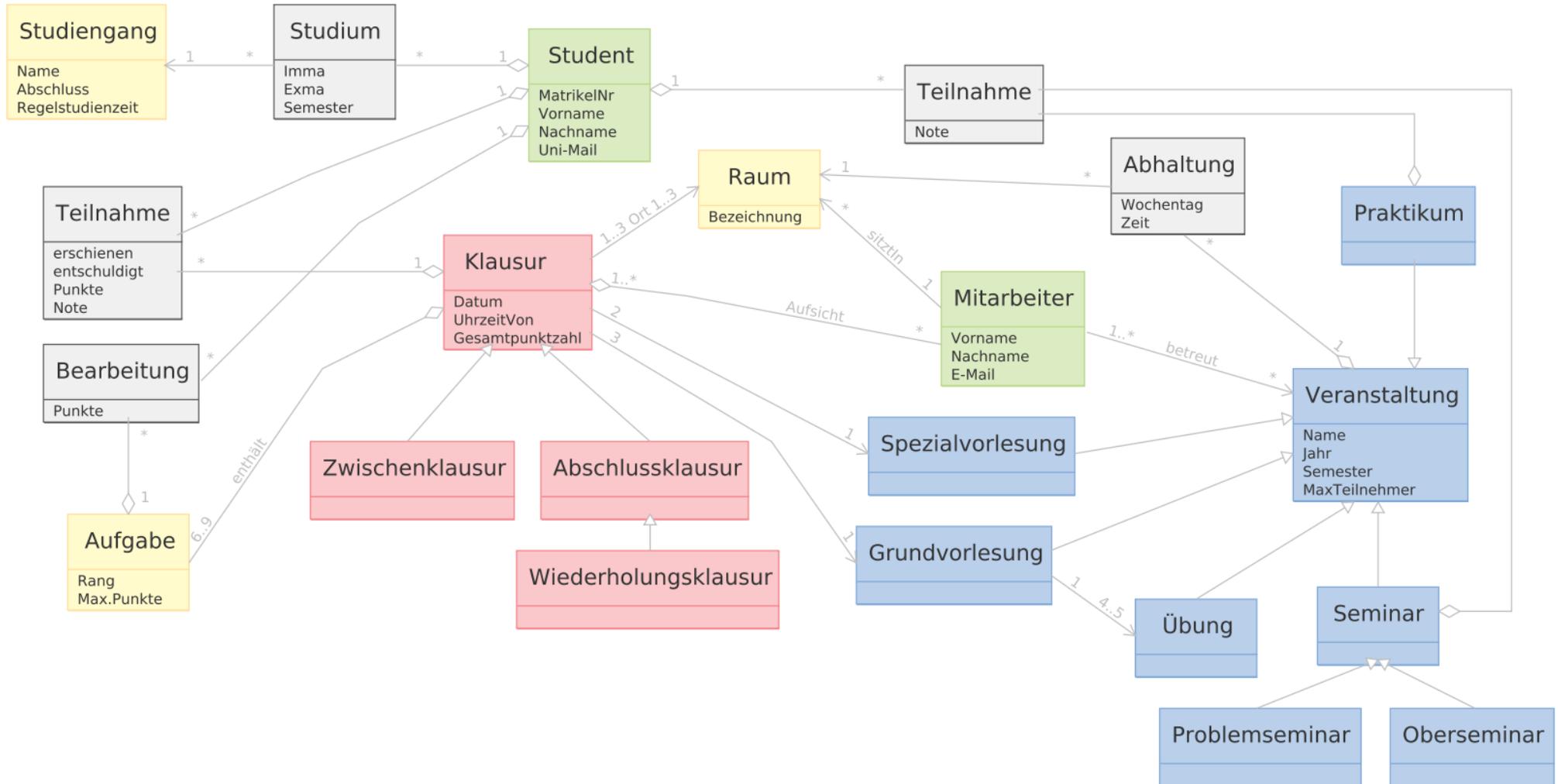
Database Group Leipzig

Department of Computer Science

Aufbau

- ▶ **Ziel:** Datenbank-Anwendungsprogrammierung, praktische DB-Kenntnisse
- ▶ **Ablauf:**
 - ▶ Selbstständige Bearbeitung der Aufgaben in **2er-Gruppen**
 - ▶ **Testat 1:** Datenbankentwurf (UML → RM), Datenimport
 - ▶ Deadline: 20.05.2019 - 24.05.2019
 - ▶ **Testat 2:** Formulierung von SQL-Anfragen
 - ▶ Deadline: 24.06.2019 - 28.06.2019
 - ▶ **Testat 3:** Anwendungsentwicklung unter Verwendung von Hibernate
 - ▶ Deadline: 30.09.2019
 - ▶ **Anwesenheitspflicht aller Praktikumssteilnehmer zu allen Testaten!**

World of Lehre



Aufgabe 1: DB-Entwurf und Datenimport

- ▶ Datenbank-Entwurf: Erstellung des konzeptionellen Schemas
- ▶ Überführung eines UML-Diagramms in ein valides **Relationenmodell**
 - ▶ **Relationen:**
 - ▶ Kardinalitäten berücksichtigen (1:1, 1:n, n:m)
 - ▶ Vererbungshierarchien auflösen
 - ▶ **Attribute + Datentypen**
 - ▶ **Definition von Integritätsbedingungen**
 - ▶ Primärschlüssel
 - ▶ Fremdschlüssel + sinnvolle Lösch- und Update-Regeln
 - ▶ CHECK-Constraints

Aufgabe 1: DB-Entwurf und Datenimport

- ▶ Entwicklung eines **SQL-DDL-Skripts** anhand des Relationenmodells und Erzeugung der Datenbank
 - ▶ Erzeugen der Tabellen, Schlüssel-Constraints, CHECK-Constraints, ...
- ▶ **Schritte:**
 - ▶ 1. UML → Relationenmodell
 - ▶ 2. Relationenmodell → SQL-DDL-Skript
- ▶ Verwendung von **PostgreSQL**
 - ▶ Open-source DBMS
 - ▶ Top-4 im DBMS-Popularität-Ranking



Aufgabe 1: DB-Entwurf und Datenimport

▶ Import der Daten in die DB

- ▶ Daten als CSV-Files
- ▶ Daten müssen entsprechend dem Relationenmodell *transformiert* bzw. *aufbereitet* werden
- ▶ **Varianten:**
 - ▶ **SQL-Skript:**
 - ▶ Transformation mit Mitteln des DBS / SQL
 - ▶ Verwendung temporärer Tabellen via COPY und Transformation in das Zielschema mit SQL-Befehlen
 - ▶ **Java-Programm:**
 - ▶ Einlesen der CSV-Daten, z. B. mit Apache Commons CSV
 - ▶ Konvertierung in Zielschema
 - ▶ Schreiben der Daten mit JDBC oder Hibernate (Wiederverwendbar für Aufgaben 2 und 3)

Aufgabe 2: SQL-Anfragen, Views, Trigger

- ▶ Formulieren von **SQL-Anfragen** auf der erzeugten DB, u.a. zu folgenden Themen
 - ▶ Wie viele Teilnehmer... ?
 - ▶ Welche Mitarbeiter haben noch nie eine ZK beaufsichtigt?
 - ▶ Welche Studenten haben mehrere Veranstaltungen zusammen besucht?
 - ▶ Wer sind die Top-Studierenden im Jahr ...?
- ▶ **View**
 - ▶ Liste möglicher Lernpartner
- ▶ **Trigger**
 - ▶ Automatische Anmeldung von Studierenden zu Klausuren in Abhängigkeit von verschiedenen Ereignissen

Aufgabe 3: Java-Anwendung

- ▶ Objekt-relationales Mapping mit Hibernate
 - ▶ Entwicklung einer Middleware zum Zugriff auf die Datenbank
 - ▶ Hibernate als quelloffenes Objekt-relationales Persistenz-Framework
 - ▶ Verwendung von Java-Annotationen

- ▶ Anwendungsprogrammierung mit Java
 - ▶ Realisierung von verschiedenen Use Cases, inklusive aller notwendigen Ein- und Ausgabe- sowie Datenbank-Operationen
 - ▶ Implementierung einer GUI (JavaFX) oder Konsolen-Logi zur Abarbeitung / Ausführung der Use Cases
 - ▶ Use Case, z. B. Klausurergebnisse eintragen

Organisation

- ▶ <https://dbs.uni-leipzig.de/de/stud/2019ss/dbprak/>
 - ▶ UML-Diagramm (Schema), CSV-Dateien
 - ▶ Genaue Aufgabenstellungen
 - ▶ Fristen, Abgaben
 - ▶ Anmeldung als Student auf der DBS-Website erforderlich!
- ▶ **Installation:**
 - ▶ Lokal auf eigenem Rechner (Laptop für Testat notwendig)
 - ▶ PostgreSQL + pgAdmin III
 - ▶ Eclipse / IntelliJ
 - ▶ Java
- ▶ **Dokumentation:**
 - ▶ Eigenständige Recherche

Organisation

▶ **Testate:**

- ▶ Präsentation der lauffähigen Programme / Ausführung der SQL-Anfragen
- ▶ Website: Hinweise zur Abgabe und Darstellung der Lösung
- ▶ Beantwortung weitergehender Fragen zu der jeweiligen Thematik
- ▶ Begründung von Design-Entscheidungen

▶ **Bewertung:**

- ▶ Anwesenheit zu jedem Testat ist zwingend erforderlich!
- ▶ Jedes Testat muss bestanden werden!
- ▶ Durchschnitt ergibt Gesamtnote

Organisation

▶ **Bei Fragen:**

- ▶ Aufgabe genau lesen
- ▶ Möglichkeiten und deren Folgen abwägen und dementsprechend handeln
 - ▶ Lösung muss nachvollziehbar bzw. begründbar sein
- ▶ FAQs lesen
- ▶ Betreuer kontaktieren: nachname @ informatik.uni-leipzig.de
 - ▶ Nicht erwünscht: “Ist das richtig, wenn wir im Sachverhalt X wie folgt vorgehen?”

▶ **E-Mail-Kommunikation**

- ▶ Betreff: dbprak + Gruppennummer
- ▶ CC an Praktikumpartner

▶ **Infoveranstaltung Teil 3:**

- ▶ 24.06.2019, 11:15 Uhr

Organisation

▶ Platzvergabe:

- ▶ Vorrangig an Studierende, welche folgende Kriterien erfüllen
 - ▶ Bestehende **Einschreibung im AlmaWeb** für das Modul
 - ▶ **Heutige Anwesenheit**
- ▶ **Selbsteinschreibung in Gruppen via Moodle**
 - ▶ <https://moodle2.uni-leipzig.de/course/view.php?id=21137>
 - ▶ **Einschreibeschlüssel notwendig**
 - ▶ **Deadline: 12.04.2019**
 - ▶ **Falls #Teilnehmer > #Praktikumsplätze**
 - ▶ **Gruppenauflösung nach o. g. Kriterien**
- ▶ **Vereinbarung Testat-Terminen**
 - **Moodle**

Relationales Datenbank-Praktikum SS 2019

Home Mein Moodle Kalender Meine Kurse Dieser Kurs Autorenhilfekurs

Meine Kurse > DB-Prak-SS19 > Allgemeines > Gruppeneinschreibung

Gruppeneinschreibung

Getrennte Gruppen

Alle Teilnehmer/innen

- Gruppe 1
- Gruppe 2
- Gruppe 3
- Gruppe 4
- Gruppe 5