

# Relationales Datenbankpraktikum 2020ss

*V. Christen, M. Franke, Z. Sehili, Dr. H. Köpcke*

## **3 Teile**

- 1) Datenbankentwurf, Datenimport**
- 2) Formulierung von SQL-Anfragen**
- 3) Entwicklung einer Middleware für eine Applikation unter Verwendung von Hibernate

## **Inhalt Einführung**

- Domäne
- Überblick zur Aufgabenstellung A1+A2 ( + kurz A3)
- Organisatorisches

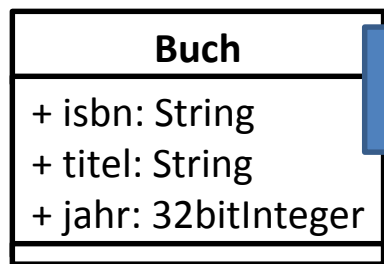


# Aufgabe 1

- Datenbank-Entwurf: Erstellung des konzeptionellen Schemas
- Überführung eines UML-Diagramms in ein valides **Relationenmodell**
  - Relationen:
    - Kardinalitäten berücksichtigen (1:1, 1:n, n:m)
    - Vererbungshierarchien: Modellierung von Kindklassen als eigene Relationen
      - Beispiele: City-[erbt von]->Place, Post-[erbt von]->Message
  - Attribute
  - Datentypen
  - Definition von Integritätsbedingungen
    - Primärschlüsselbedingung
    - Fremdschlüsselbedingungen / referentielle Integrität: Definition sinnvoller Lös- und Updateregeln
    - CHECK Constraints

# Aufgabe 1

- Entwickeln eines SQL-Skripts (DDL Schema-Definition) anhand des Relationenmodells und Erzeugung der Datenbank
  - Erzeugen der Tabellen, Schlüssel-Constraints, CHECK-Constraints ...



UML

Relationenmodell

*buch(isbn, titel, ...)*  
*autor(vorname, nachname, gebjahr, wohnort)*  
*buchaut(isbn, vorname, nachname, gebjahr)*  
isbn FK auf buch;  
vorn..., nac..., gebj.. FK auf autor

0..\*

buchaut

1..\*

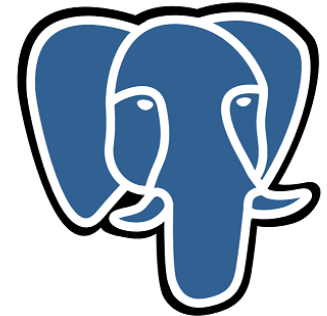
Autor

```
classDiagram
    class Autor {
        + vorname: String
        + nachname: String
        + gebjahr: 32bitInteger
        + wohnort: String
    }
```

SQL

```
Database Browser x Query Editor - script1.sql x Query Edito
Connection: db2 v9.5 leutzsch
1 create table buch (
2 isbn int not null primary key,
3 titel varchar(100) not null,
4 jahr smallint not null );
5
6 create table autor (
7 vorname varchar(50) not null,
8 nachname varchar(50) not null,
9 gebjahr smallint not null,
10 wohnort varchar(50),
11 primary key (vorname, nachname, gebjahr) );
12
13 create table buchaut (
14 isbn int not null references buch on delete
15 vorname varchar(50) not null,
16 nachname varchar(50) not null,
17 gebjahr smallint not null
```

# Aufgabe 1



PostgreSQL

- **Import der Daten in DB**
  - Daten als CSV
  - **SQL-Skript**
    - Transformation mittels der von der Datenbank bereitgestellten Mittel
    - Verwenden temporärer Tabellen via COPY und Transformation in das Zielschema mit SQL
  - **Lade-Programm**
    - Einlesen der CSV-Daten, Konvertierung in das Zielschema und Schreiben in die Datenbank (z.b. mit Java und JDBC)

*place\_0\_0.csv*

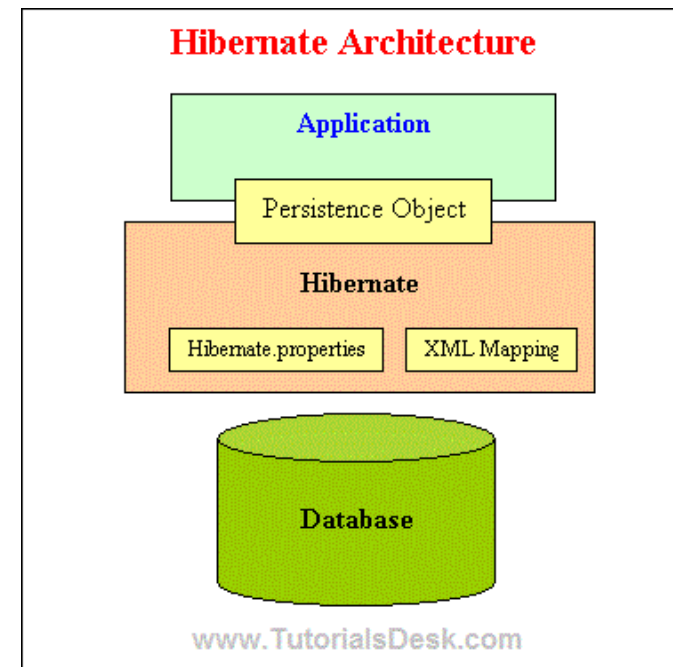
```
id|name|url|type|isPartOf|
0|India|http://dbpedia.org/resource/India|country|1460|
1|China|http://dbpedia.org/resource/China|country|1460|
2|Angola|http://dbpedia.org/resource/Angola|country|1461|
3|Austria|http://dbpedia.org/resource/Austria|country|1462|
```

# Aufgabe 2

- Formulieren von SQL Anfragen auf der erzeugten DB, u.a. zu folgenden Themen
  - Wie viele Forenbeiträge ...?
  - Wie viele Kommentare zu Posts ... ?
  - Woher kommen die meisten Nutzer?
  - Mitglied in den gleichen Foren?
  - Interessanteste Themen, Like/Post-Anzahl..?
  - Freunde, Freundesfreunde, Pfade zwischen Nutzern?
  - **Genauere Formulierung der Fragen siehe Praktikumswebseite**
- Änderungen in der erzeugten DB
  - Beendigung eines Arbeitsverhältnisses
  - Automatische Dokumentation des Löschvorgangs

# Vorschau - Aufgabe 3

- Java-Anwendung + DB
  - Implementieren einer Middleware unter Verwendung von Hibernate
  - Objekt-Relationales Persistenz-Framework
    - Open-Source-Projekt:  
[www.hibernate.org](http://www.hibernate.org)
    - "Java-Objekt in relationaler Datenbank speichern und laden"



# Organisatorisches (1)

- <https://dbs.uni-leipzig.de/de/stud/2020ss/dbprak/teil1>
  - Miniweltbeschreibung, UML-Diagramm, Daten
- <https://dbs.uni-leipzig.de/de/stud/2020ss/dbprak/teil2>
  - Beschreibung der zu formulierenden Anfragen, Mechanismus zur Änderungsverfolgung
- Installationen
  - Auf eigenem Rechner (..Laptop für Testat)
    - Alternativen: individuelle Absprache
  - PostgreSQL
  - ggf. SQL-Tool
  - ggf. IDE, Java, ...
- Dokumentationen
  - [PostgreSQL Dokumentation](#) + eigenständige Recherche



# Organisatorisches (2)

- Testate

- Präsentation der lauffähigen Programme/  
Ausführen der Anfragen im Testat
- **Webseite: Hinweise zur Abgabe und Darstellung der Lösung!**

- Verbindliche Termine siehe Praktikumswebseite

- Testat 1: 18.05.2020 - 22.05.2020
- Testat 2: 22.06.2020- 26.06.2020
- Testat 3: September (TBA)

- Bewertung

- Anwesenheit analog jedem Testat oder digital falls es die Situation erfordert
- Jedes Testat muss bestanden werden!
- Durchschnitt ergibt Gesamtnote

# Organisatorisches (3)

- Bei Fragen
  - 1) **Aufgabe genau lesen**
  - 2) (Bis dahin ggf. erweiterte) **FAQs** lesen
  - 3) Betreuer fragen: [nachname@informatik.uni-leipzig.de](mailto:nachname@informatik.uni-leipzig.de)

## **Email-Kommunikation:**

- Angabe **dbprak + Gruppennummer**
- **im CC: Email des Praktikumpartners**

- Infoveranstaltung Teil 3
  - tba