

Datenbanken II  
SS 2021  
**Praktikum 1**

(Ausgabe: 12.05.2021 – Abgabe: 9.06.2021)

Die unten aufgeführten Aufgabenstellungen können in Kleingruppen bearbeitet werden. Die Lösungen schicken Sie bitte zur Abgabefrist per Mail an [andreas.behrend@th-koeln.de](mailto:andreas.behrend@th-koeln.de). Die Funktionalität der Software werden ausgewählte Studierende via Videokonferenz demonstrieren müssen. Bitte fügen Sie ihrer Lösung bereits (auskommentierte) Beispielanfragen bzw. Befehle hinzu, damit man die Software direkt damit testen kann. Ihre Software muss unter PostgreSQL lauffähig sein. Für Fragen zur Aufgabenstellung bzw. zu den Praktikumsmodalitäten nutzen Sie bitte das Forum bzw. kontaktieren Sie mich per Mail.

## Vorbereitung

Im Rahmen dieses Praktikums soll eine einfache Anwendung zur Unterstützung der Universitätsverwaltung erstellt werden. Damit sollen Professoren, Studierende, Angestellte und wissenschaftliche Mitarbeiter sowie das Lehrangebot und Prüfungsdaten verwaltet werden. Sie können sich dabei an dem Schema aus dem Kemper-Buch „Datenbanksysteme“ ([Kapitel 2 und 3](#)) orientieren:

Studenten :	{{ <u>MatrNr</u> , Name, Semester}}	hoeren :	{{ <u>MatrNr</u> , <u>VorlNr</u> }}
Vorlesungen :	{{ <u>VorlNr</u> , Titel, SWS, gelesenVon}}	voraussetzen :	{{ <u>Vorgaenger</u> , <u>Nachfolger</u> }}
Assistenten :	{{ <u>PersNr</u> , Name, Fachgebiet, Boss}}	Professoren :	{{ <u>PersNr</u> , Name, Rang, Raum}}
pruefen :	{{ <u>MatrNr</u> , <u>VorlNr</u> , PersNr, Note}}		

Die unterstrichenen Attribute (Spaltenbezeichner) sind die Schlüsselattribute der entsprechenden Relation. Wichtig ist, dass sie *objektrelationale Features* abdecken. So muss in Ihrer Anwendung mindestens ein mengenwertiges Attribut, eine IS-A Hierarchie sowie eine sinnvolle UDF verwendet werden.

## Aufgabe 1

Erstellen Sie ein EERD für die Universitätsverwaltung mit den verschiedenen Beziehungstypen und kennzeichnen Sie die Schlüsselattribute. Führen Sie anschließend die Abbildung in ein relationales Schema durch. Erläutern Sie, nach welchem Prinzip Sie die **IS-A Beziehung** umgesetzt haben und beschreiben Sie kurz die Vor- bzw. Nachteile Ihrer Lösung. Eine IS-A Beziehung bietet sich z.B. für die verschiedenen Personen in Ihrer Anwendung an (also Angestellter, Professoren, Mitarbeiter, Studierende etc.) aber sie können auch andere Hierarchien modellieren und das Schema um eigene (aber sinnvolle) Entity-Typen erweitern.

Überlegen Sie sich eine Anfrage an die Supertabelle, mit der Tupel aus der Supertabelle und aus den Spezialisierungen (Subtabellen) zurückgegeben werden. Mit dem internen Attribut *tableoid* kann man die Tabellenherkunft der einzelnen Resultate als numerischen Wert sehen. Programmieren Sie ihre Anfrage um, so dass nun auch in jeder Antwortzeile der Name der Tabelle angegeben wird, aus der das jeweilige Tupel stammt. Nutzen Sie dafür die interne Meta-Tabelle *pg\_class*.

## Aufgabe 2

Nehmen Sie an, dass in der Tabelle der Studierenden ein mengenwertiges Attribut *Noten* existiert (z.B. vom Datentyp Array), mit dem man für einen Studierenden seinen Notenspiegel bestimmen kann. Dabei soll sichergestellt sein, dass man für jede Note auch die dazugehörige Vorlesung (z.B. über VorlNr) kennt:

MatrNr	Name	...	Noten
100		...	{{1.0,1234},{2.3,1345},...}
101		...	{{3.7,1345},{1.3,2001},...}
102		...	..
103		...	

Die Noten sind hier als zweidimensionales Array modelliert, wobei in der ersten Dimension die Note und in der zweiten die dazugehörige Vorlesungsnummer angegeben wird. Überlegen Sie sich eine *Integritätsbedingung* für die Tabelle Studenten, mit der sichergestellt wird, dass eine Vorlesungsnummer höchstens dreimal im Notenspiegel eines Studierenden auftritt.

## Aufgabe 3

Jedem Professor ist eine Vorlesung eindeutig zugeordnet. Erstellen Sie eine Übersicht (mittels einem View und/oder einer UDF), in der die erzielte Durchschnittsnote der Studierenden pro Professor angegeben ist.

## Abgabe und Demonstration

Um Ihre Anwendung zu testen, fügen Sie mit Hilfe einer Transaktion neue Studierende, Professoren, Angestellte sowie Vorlesungen der Datenbank hinzu. Erstellen Sie dafür ein SQL-Skript, mit dessen Hilfe ihre Tabellen eine definierte Ausgangspopulation bekommen.

Die zur Abgabefrist 9.6 elektronisch einzureichenden Dokumente sind mindestens

- 1) ein EER-Diagramm ihres DB-Schemas
- 2) ein SQL Skript zum Erstellen und Füllen ihrer Datenbank-Anwendung mit Beispieldaten
- 3) eine Dokumentation mit Erläuterung Ihrer Anfragen, ihrer Integritätsbedingung sowie der programmierten UDF anhand von Beispielen.

Falls Ihr Team zur Online-Präsentation ausgewählt werden sollte, bereiten Sie bitte eine max. 10-minütige Präsentation ihrer Software vor. Es ist notwendig, dass alle Gruppenmitglieder anwesend sind. Jedes Gruppenmitglied muss sich an der Präsentation angemessen beteiligen bzw. in der Lage sein, Fragen zu allen Teilen des Projektes zu beantworten.

Achtung: Bitte halten Sie sich an den Zeitrahmen. Es wird sehr empfohlen, die Demo vorher zu proben!