

Praktikum 1

(Ausgabe: 15.12.2020 – Abgabe: 06.01.2021)

Die unten aufgeführten Aufgabenstellungen können in Kleingruppen (mit bis zu 4 Personen) bearbeitet werden. Die Lösungen schicken Sie bitte zur Abgabefrist per Mail an andreas.behrend@th-koeln.de. Die Abgaben werden im ILIAS für jede Praktikumsgruppe gespeichert. Die Funktionalität der Software werden ausgewählte Studierende via Videokonferenz demonstrieren müssen. Bitte fügen Sie Kommentare ihren Beispielanfragen bzw- befehle hinzu, damit erkennbar ist, dass Sie die Lösungen auch verstanden haben. 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.

Aufgabe 1

Programmieren Sie DDL-Ausdrücke zum Anlegen des oben aufgeführten Schemas. Überlegen Sie sich geeignete *Datentypen* für die Attribute und sinnvolle *Column Constraints* (Default-Wert, Nullability, Check-Bedingungen) für ausgewählte Spalten. Finden Sie heraus, was **referentielle Integritätsbedingungen** sind und wie diese eingesetzt werden. Modellieren Sie mit deren Hilfe folgende Bedingungen: Jede Vorlesung muss einem bestimmten Dozent zugeordnet sein, der selbst Professor an der Hochschule ist. Jeder Professor muss mindestens für eine Vorlesung verantwortlich sein.

Aufgabe 2

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. Im ILIAS-Ordner Praktikum der Vorlesung finden Sie eine Datei mit ersten Beispieldaten. Erweitern Sie die Daten in sinnvoller Weise, so dass Sie für folgende SQL-Anfragen ein auswertbares Ergebnis erhalten:

1. Finden Sie alle Studenten (MatrNr ausgeben), die die Vorlesung *Ethik* hören.
2. Welche Studenten haben schon mal mit dem Studenten *Schopenhauer* gemeinsam eine Vorlesung gehört? Geben Sie die Namen dieser Studenten duplikatfrei aus. Schopenhauer selbst darf nicht im Ergebnis vorkommen.
3. Welche Studenten nehmen an allen Vorlesungen teil, die auch der Student *Schopenhauer* hört?
4. Welche Vorlesungen (VorlNr genügt) haben mindestens zwei andere Vorlesung als Voraussetzung?
5. Geben Sie eine absteigend sortierte Liste aus allen Vorlesungen und der Anzahl pro Vorlesung abgehaltener Prüfungen aus, etwa so:

VorlNr	Anzahl
4630	2
5001	1
5041	1
5022	0
...	...

6. Finden Sie den/die Professor(en) (Name ausgeben) mit den meisten Assistenten.
7. Welche Studierenden hören alle Vorlesungen?
8. Wie oft wurde eine Prüfung mit der Note 1 oder 2 bewertet?
9. Erstellen Sie eine Übersicht, in der die MatrNr und der Name der Studierenden zusammen mit der von ihnen erreichten Durchschnittsnote sowie dem dazugehörigen Varianz-Wert angeben werden.
10. Gibt es Namen von Personen, die in mindestens zwei verschiedenen Tabellen auftreten?
11. Erstellen Sie eine Übersicht, welche Vorlesung (VorlNr genügt) welche anderen Vorlesungen direkt oder indirekt als Voraussetzung hat.

Aufgabe 3

Mit den Befehlen **EXPLAIN** sowie **EXPLAIN ANALYZE** kann man die Ausführungspläne von PostgreSQL für SQL-Anfragen einsehen. Finden Sie eine Anfrage aus Aufgabe 2, die nur eine mäßige Performanz aufweist und beschleunigen Sie deren Ausführung durch das Anlegen eines geeigneten Hilfsindex (mittels **CREATE INDEX**).

Abgabe und Demonstration

Die zur Abgabefrist 6.1.2021 elektronisch einzureichenden Dokumente sind mindestens

- 1) ein SQL-Skript zum Erstellen und Füllen ihrer Datenbank-Anwendung mit Erläuterungen der verwendeten Integritätsbedingungen
- 2) ein SQL-Skript zur Beantwortung der Anfragen aus Aufgabe 2 zusammen mit den Ergebnissen der Anfragen
- 3) ein SQL-Skript zum Anlegen eines Hilfsindexes sowie ein Vergleich der Performanz

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.