

## Übung 10

### Advanced SQL 1: Aggregate, Joins und Rekursion

Für dieses Übungsblatt nutzen Sie bitte das vorbereitete SQL Script  
DB - KH -Uebung10 Testdaten.sql aus der Community.

#### Aufgabe 1: Aggregatfunktionen

- a) Nutzen Sie die Aggregatfunktion MIN, um die beste Note zu jeder Veranstaltung eines Dozenten auszugeben.
- b) Nutzen Sie die Aggregatfunktion MAX um die schlechteste Note zu jedem Dozenten auszugeben.
- c) Nutzen Sie die Aggregatfunktion COUNT, um die Anzahl der Studenten zu jeder Veranstaltung eines Dozenten auszugeben.
- d) Kombinieren Sie die Aggregatfunktionen COUNT und AVG, um die durchschnittliche Anzahl an Studenten in den Veranstaltungen eines Dozenten anzugeben.

#### Aufgabe 2: Joins

In dieser Aufgabe interessieren wir uns nicht für die Projektionsliste, geben Sie daher einfach alle Attribute mit Hilfe von `select *` aus. Geben zu jedem nachfolgend verwendetem join an, ob es sich um einen natural, equi, theta oder semi join handelt.

- a) Geben Sie eine Liste aus von allen Veranstaltungen eines Dozenten, für die sich noch kein Student registriert hat.
- b) Nutzen Sie einen left join, um eine Liste aller Studenten, die zusätzlich die Informationen der jeweils besuchten Veranstaltungen enthält zu erzeugen.
- c) Wandeln Sie den left join aus der Aufgabe zuvor in einen right join um.

#### Aufgabe 3: Rekursion

In dieser Aufgabe beschäftigen wir uns mit Rekursionen. Da in SQL eine Rekursion anders aufgebaut ist, als in gängigen imperativen Programmiersprachen, gehen wir ähnlich zu dem Bus Beispiel in der Vorlesung schrittweise vor. Betrachten Sie zunächst in dem .sql-File zu dieser Übung die Tabelle `Ahnen`. Dort ist zu einer Person jeweils Vater und Mutter angegeben.

- a) Geben Sie eine List aus, in der in der ersten Spalte der Name einer Person aufgeführt ist und in der zweiten Spalte der Name eines direkten Ahnen, also jeweils Vater oder Mutter.
- b) Verwenden Sie nun den zuvor erstellten Query in einem `with` und nennen Sie ihn etwa `ersteVerwandtschaftsverhaeltnisse`. Erweitern Sie dann um zweite und dritte Verwandtschaftsverhältnisse.
- c) Entwickeln Sie aus den ersten beiden Aufgaben den Rekursionsanfang und den Rekursionsschritt und geben Sie nun mittels Rekursion eine Liste aus, die alle Verwandtschaftsverhältnisse enthält.
- d) Erweitern Sie die zuvor erstellte Rekursion um den Grad der Verwandtschaft. Es kann hilfreich sein, dazu wieder zunächst die Queries aus den ersten beiden Aufgaben zu erweitern. Geben Sie dann alle Verwandtschaftsverhältnisse aus und sortieren Sie ansteigend nach Name und absteigend nach Verwandtschaftsgrad.