

## Übungen Datenbanksysteme Serie 6

### Besprechung am 14.5. bzw. 24.5.

Verwenden Sie für die folgenden Übungen die Datenbank DOZ.

In Abbildung 1 ist das Datenbankschema dargestellt. Die Datenbank enthält Dozenten (abgekürzt Doz), denen ein Fachbereich (Fb) zugeordnet ist.

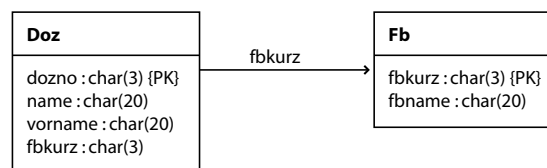


Abbildung 1: Datenbankschema für Dozenten

Die Besonderheit besteht darin, dass es sein kann, dass einem Dozenten gar kein Fachbereich zugeordnet ist. Das Feld `fbkurz` ist dann `null`. Oder es kann sein, dass es einen Fachbereich gibt, dem keine Dozenten zugeordnet sind.

#### 1. DOZ01

Ermitteln Sie Namen, Vornamen und zugeordneten Fachbereich der Dozenten. Es sollen nur diejenigen Dozenten angegeben werden, die tatsächlich einem Fachbereich zugeordnet sind.

#### 2. DOZ02

Ermitteln Sie Namen, Vornamen der Dozenten, die keinem Fachbereich zugeordnet sind.

#### 3. DOZ03

Ermitteln Sie Namen, Vornamen der Dozenten und Fachbereich von *allen* Dozenten - auch jenen, die keinem Fachbereich zugeordnet sind.

Was passiert, wenn Sie in der Anweisung die Position der beiden Tabellen **Doz** und **Fb** vertauschen? Was müssen Sie tun, damit Sie das gleiche Ergebnis wie vorher erhalten?

#### 4. DOZ04

Erstellen Sie eine Liste aller Fachbereiche mit ihren Dozenten, geben Sie dabei auch diejenigen Fachbereiche an, die keine Dozenten haben.

**5. DOZ05**

Erstellen Sie eine Liste aller Fachbereiche und geben Sie für jeden Fachbereich die Zahl der Dozenten an, die ihm zugeordnet sind.

**6. DOZ06**

Erstellen Sie eine Liste der Bezeichnungen der Fachbereiche, die keine Dozenten haben.

**7. DOZ07**

Erstellen Sie eine Liste der Bezeichnungen der Fachbereiche und der Namen der Dozenten, die den Fachbereichen zugeordnet sind. In der Liste sollen auch die Fachbereiche erscheinen, die keine Dozenten haben und die Dozenten, die keinem Fachbereich zugeordnet sind.

**8. DOZ08**

Formulieren Sie die SQL-Anweisungen für Aufgabe DOZ03 und DOZ07 *ohne* „outer join“.

**9. Join01**

Gegeben seien die beiden folgende Tabellen:

Mitarbeiter		Gehalt	
PId	Name	PId	Betrag
1001	Hans	1582	2500
1582	Peter	1672	4000
1703	Eva	1703	3200
		1005	2200

Was ergeben folgende SQL-Anweisungen?

- a) `select * from Mitarbeiter natural join Gehalt`
- b) `select * from Mitarbeiter left outer join Gehalt using(PId)`
- c) `select * from Mitarbeiter right outer join Gehalt using(PId)`
- d) `select * from Mitarbeiter full outer join Gehalt using(PId)`

**10. Join02**

Gegeben seien drei Tabellen T1, T2 und T3:

T1	<table><tr><th>a</th><th>b</th></tr><tr><td>1</td><td>2</td></tr></table>	a	b	1	2	T2	<table><tr><th>b</th><th>c</th></tr><tr><td>2</td><td>3</td></tr></table>	b	c	2	3	T3	<table><tr><th>a</th><th>c</th></tr><tr><td>4</td><td>5</td></tr></table>	a	c	4	5
a	b																
1	2																
b	c																
2	3																
a	c																
4	5																

Zeigen Sie am Beispiel dieser drei Tabellen, dass der *full outer join*-Operator von SQL nicht assoziativ ist.

Wie verhält es sich mit den „natural inner join“?

## 11. Join03

Diskutieren Sie die folgenden Aussagen:

(...) you should understand the basic types of join clauses:

- **Join**  
Joins are powerful SQL commands for creating tables from data that is retrieved from multiple sources.
- **Equi-Join**  
Equi-Joins are formed by retrieving all the data from two separate sources and combining it into one, large table.
- **Inner/Outer Join**  
Inner Joins are joined on the inner columns of two tables. Outer Joins are joined on the outer columns of two tables.
- **Left/Right Join**  
Left Joins are joined on the left columns of two tables. Right Joins are joined on the right columns of two tables.
- **Compound/Complex Joins**  
There are also other kinds of joins – left/inner, left/outer, right/inner, and right/outer.

Wieviele der Aussagen in diesem Text, der aus einem Internet-Portal für IT- und Datenbankspezialisten stammt, sind zutreffend?

## 12. Join04

In einem Blog im Internet findet man folgende Aussagen zum Verbund (*join*):

„Ein *inner join* liefert im Prinzip die Schnittmenge aus beiden Grundmengen.“

und weiter über den äußeren Verbund:

„Ein *full outer join* gibt eigentlich alle Datensätze von Tabelle A und Tabelle B aus, wobei die Schnittmenge zusammengefasst wird.“

- Am Ende des Blog-Eintrags reagiert der Autor auf einen Einwand eines seiner Leser:  
„Wie der Kommentator Steffen korrekterweise moniert, ist ein JOIN tatsächlich keine Schnittmenge, sondern vielmehr ein kartesisches Produkt in Verbindung mit einer Selektion. Datenbanktheoretiker mögen

mich strafen, aber ich bleibe dennoch bei meiner starken Vereinfachung in diesem Artikel!“

Was halten Sie von dieser Reaktion? Handelt es sich bei den obigen Ausführungen um eine *Vereinfachung*?

- Es gibt tatsächlich eine sehr spezielle Situation, in der der natürliche Verbund zweier Relationen ihre Schnittmenge zum Ergebnis hat. Welche?