

2. Abfragen /Queries

Gegeben in: Woche 3

Abgabe:

Woche 4 - Teil 1 + wenigstens eine komplexe Abfrage aus Teil 2

Woche 5 – der Rest aus Teil 2

Die Endnote wird nur in Woche 5 berechnet.

1. (3p) Für die Datenbank, die ihr bei Aufgabe 1 erstellt habt, schreibe SQL-Anweisungen für:

- a. Das **Erstellen zweier Tabellen**, von denen eine der Tabellen **ein zusammengesetztes Primärschlüssel** hat. Erstelle gleichzeitig auch die **Fremdschlüssel**.
- b. Das **Einfügen von Daten (wenigstens in zwei Tabellen)**.
- c. Zeige auch ein Beispiel von Daten bei dem **Einfügen**, welche die **Fremdschlüsselintegritätsregel nicht erfüllen und darum nicht eingefügt werden können**.
- d. **Ändern und Löschen** von Daten. Dabei benutze die **WHERE** Klausel, wobei wenigstens einmal Folgendes benutzt wird:
 - Eine zusammengesetzte Bedingung mit logischen Operatoren
 - IS [NOT] NULL
 - IN (ohne geschachtelte Abfrage)
 - BETWEEN
 - LIKE

2. (6p) Schreibe **10 Abfragen** auf die Struktur der Datenbank, die ihr bei Aufgabe 1 erstellt habt. **D.h. ihr müsst komplexe Anfragen haben, die mehrere der unteren Befehle benutzen, und nicht eine separate Anfrage für jeden Befehl!**

In diesen 10 Abfragen müssen folgende Befehle vorkommen (eine Abfrage kann mehrere der folgenden Bedingungen erfüllen):

- 5 Abfragen mit **WHERE**
- 7 Abfragen **JOINING** mehr als 2 Tabellen; benutze wenigstens ein **OUTER JOIN** und erkläre wofür das gut ist
- 1 Abfrage mit **ALL**
- 1 Abfrage mit **ANY**
- 3 Abfragen mit **GROUP BY**
- 2 Abfragen mit **HAVING**
- 3 Abfragen mit Aggregatfunktionen (**COUNT, SUM, AVG, MIN, MAX**)
- 2 Abfragen mit der **Vereinigung** Operation; benutze **zwei Methoden: UNION, OR**
- 2 Abfragen mit der **Durchschnitt** Operation; benutze **zwei Methoden: INTERSECT, IN**

- 2 Abfragen mit der **Differenz** Operation; benutze **zwei Methoden: EXCEPT, NOT IN**
- 1 **geschachtelte Abfrage**
- 1 Abfrage mit **DISTINCT** (wo DISTINCT auch einen Unterschied macht – erkläre!)
- 1 Abfrage mit **TOP**
- 1 Abfrage mit **ORDER BY**

Die Abfragen müssen:

- für das Thema der entworfenen Datenbank relevant sein.
- komplex sein, aber nicht unnötig kompliziert
- verschiedene Methoden aus dem Seminar verknüpfen oder neue Methoden benutzen
- relevant sein, aus einem Business-Sichtpunkt (nützlich für den Benutzer dieses Systems)
- **erklärt** werden. Ihr müsst jede Abfrage **erklären können** – was gibt diese Abfrage aus?