

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 von zwei 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 für Daten bei dem **Einfügen**, die die **Fremdschlüsselintegritätsregel nicht erfüllen und darum nicht eingefügt werden können**.
 - d. **Ändern und Löschen** von Daten. Benutze dabei 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, und die auch Sinn machen für die ausgewählte Datenbank. **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** in einer Abfrage wo ein Inner Join nicht dasselbe Ergebnis ergibt, und erkläre wofür das Outer Join gut ist
 - 1 Abfrage mit einer Unterabfrage verbunden mit **ALL** (geschachtelte Abfrage)
 - 1 Abfrage mit einer Unterabfrage verbunden mit **ANY**
 - 3 Abfragen mit **GROUP BY**
 - 2 Abfragen mit **HAVING**

- 3 Abfragen mit Aggregatfunktionen (**COUNT, SUM, AVG, MIN, MAX**); benutze dabei **wenigsten zwei unterschiedliche Aggregatfunktionen**
- 2 **unterschiedliche** Abfragen mit der **Vereinigung** Operation zwischen zwei Mengen (ohne Konstanten Mengen, die Menge sind Abfrageergebnisse); benutze **zwei Methoden: UNION, OR**
- 2 **unterschiedliche** Abfragen mit der **Durchschnitt** Operation zwischen zwei Mengen (ohne Konstanten Mengen, die Menge sind Abfrageergebnisse); benutze **zwei Methoden: INTERSECT, IN**
- 2 **unterschiedliche** Abfragen mit der **Differenz** Operation zwischen zwei Mengen (ohne Konstanten Mengen, die Menge sind Abfrageergebnisse); benutze **zwei Methoden: EXCEPT, NOT IN**
- 1 Abfrage mit **DISTINCT** (wo DISTINCT auch einen Unterschied macht – erkläre!)
- 1 Abfrage mit **TOP** (erkläre welche Tupeln ausgegeben werden und warum)
- 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?