

AG Datenbanken und Informationssysteme

Wintersemester 2006 / 2007

Prof. Dr.-Ing. Dr. h. c. Theo Härder
Fachbereich Informatik
Technische Universität Kaiserslautern



<http://www.dvs.informatik.uni-kl.de>

14. Übungsblatt

Für die Übung am Donnerstag, **15. Februar 2007**,
von 15:30 bis 17:00 Uhr in 13/222.

Aufgabe 1: Rekursion in SQL:1999 am Beispiel "Flugstrecke"

Eine Fluggesellschaft verwaltet die von ihr angebotenen Flugstrecken in einer Datenbank, aus der wir folgenden (hier relevanten) Ausschnitt der Tabelle „Flüge“ zur Verfügung gestellt bekommen haben:

FlugNr	Start	Ziel	Preis
HY120	DFW	JFK	255
HY130	DFW	LAX	200
HY140	DFW	ORD	100
HY150	DFW	SFO	300
HY210	JFK	DFW	225
HY240	JFK	ORD	250
HY310	LAX	DFW	200
HY350	LAX	SFO	50
HY410	ORD	DFW	100
HY420	ORD	JFK	250
HY450	ORD	SFO	275
HY510	SFO	DFW	300
HY530	SFO	LAX	50
HY540	SFO	ORD	275

Entwickeln Sie in SQL:1999 Anfragen für folgende Fragestellungen:

- Finde den billigsten Flug von San Fransisco (SFO) nach New York (JFK).
- Finde den Flug von San Fransisco nach New York, bei dem man am wenigsten umsteigen muß.

Aufgabe 2: Rekursion in SQL:1999 am Beispiel "Autobahnkarte"

In der Karte rechts wird ein Ausschnitt aus dem Autobahnnetz dargestellt. Die einzelnen Autobahnteilstücke sind durch die Strecken S1 bis S8 mit entsprechenden Längen in Kilometern gegeben. Die Strecken verbinden jeweils zwei Navigationspunkte miteinander. Ein Navigationspunkt ist durch eine eindeutige ID und dessen Name gegeben, bspw. (1,Mainz).

Die Navigationspunkte sollen in der Tabelle

navpkt (ID, Name)

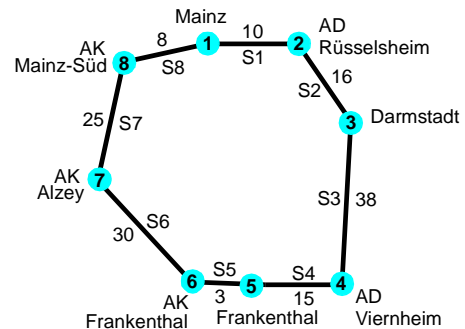
verwaltet werden, die Strecken in der Tabelle

navstr (anfang, ende, name, laenge),

wobei *anfang* und *ende* jeweils die ID eines Navigationspunktes referenzieren und somit zwei Navigationspunkte miteinander verbinden.

- Geben Sie die SQL-DDL-Anweisungen an, mit denen die Tabellen *navpkt* und *navstr* angelegt werden, und fügen sie anschließend mit entsprechenden SQL-DML-Anweisungen die notwendigen Tupel in die Tabellen ein, um die gegebene Karte zu speichern. Beachten Sie dabei, dass eine Strecke zwischen zwei Navigationspunkten für beiden Richtungen gespeichert werden muss.
- Berechnen Sie mit einer rekursionsfreien SQL-Anweisung die Entfernung zwischen Frankenthal und Darmstadt über das Autobahndreieck Viernheim.

Berechnen Sie mit einer rekursiven SQL-Anweisung die kürzeste Verbindung von Frankenthal nach Mainz. Geben Sie dazu die berechnete Route und deren Länge aus.



Aufgabe 3: Lokatoren und Dateireferenzen

Gegeben sei eine DB mit einer Relation „bewerber“, die Stipendienbewerbungen enthält und deren Schema durch folgende SQL/DDDL-Anweisung erzeugt wurde:

```
CREATE TABLE bewerber (  
  name          VARCHAR(30) PRIMARY KEY,  
  eingangsjahr  INTEGER,  
  status        VARCHAR(6),  
  bewerbung    CLOB(100K))
```

Jede Bewerbung besteht aus mehreren Abschnitten. Ein Abschnitt enthält eine Abschrift aus dem Studienbuch, die in einem einheitlichen 1KB-Format erfaßt ist, das mit den Zeichenfolgen '*LEISTUNG*' beginnt. In einem späteren Abschnitt der Bewerbung befindet sich ein Empfehlungsschreiben, das bis zum Ende der Bewerbung geht und mit '*EMPFEHLUNG*' anfängt.

Skizzieren Sie ein C-Programm mit eingebetteten SQL-Anweisungen, das nach allen im Jahr 2000 eingegangenen Bewerbungen mit dem Status 'OK' sucht. Von diesen finden Sie diejenige Bewerbung mit der besten Studienleistung heraus. Schreiben Sie die Studienleistung und das zugehörige Empfehlungsschreiben in die Ausgabedatei „gewinner.txt“. Weiterhin sei eine Funktion gegeben, die zwei Studienleistungen vergleichen und entscheiden kann, welche von den beiden die „bessere“ ist.

Benutzen Sie folgende Funktionen zur Lösung der Aufgabe:

- `int vergleicheStudienleistung (char *s11, char *s12)`
Liefert 1, falls die erste Studienleistung s11 besser als die zweite s12 ist und sonst 2.
- `char *strcpy (char *s1, const char *s2)`
Kopiert die Zeichenfolge s2 in die Zeichenfolge s1. Der Funktionswert ist der Zeiger s1.
- `void *memcpy (void *s1, const void *s2, size_t n)`
Kopiert n Zeichen von dem Speicherbereich, auf den s2 zeigt, nach dem Speicherbereich, auf den s1 zeigt. Der Funktionswert ist der Zeiger s1.
- CLOB-/BLOB-/Dateireferenz-Datentyp und -Funktionen, die in der Vorlesung vorgestellt wurden.