

# Datenbanken II

## Übungsblatt 2 – WiSe 2014/15

1. Betrachten Sie die Datei in Abbildung 1. Die Datei ist als Heap von Datensätzen fester Länge gespeichert und verwendet eine Free List zur Verwaltung des freien Speichers.
  - a) Wie sieht die Datei in Abbildung 1 nach folgenden Operationen aus:
    - i. Einfügen von (A-210,Downtown,700)
    - ii. Löschen von Datensatz 2
    - iii. Einfügen von (A-117,Perryridge,300)
  - b) Wie sieht die Datei in Abbildung 1 nach folgenden Operationen aus:
    - i. Löschen von Datensatz 5
  - c) Wie sieht die Datei in Abbildung 1 nach folgenden Operationen aus:
    - i. Löschen von Datensatz 7
    - ii. Löschen von Datensatz 8

The diagram shows a table representing a heap file. The table has 9 rows labeled 'header' through 'record 8'. Each row has 5 columns. The first three columns contain data, and the fourth column contains a pointer. Arrows on the right side of the table indicate the free list: record 0 points to record 1, record 1 points to record 2, record 2 points to record 3, record 3 points to record 4, record 4 points to record 5, record 5 points to record 6, record 6 points to record 7, and record 7 points to record 8. Record 8 has a null pointer. The entire table is enclosed in a red border.

header				
record 0	A-102	Perryridge	400	
record 1				
record 2	A-215	Mianus	700	
record 3	A-101	Downtown	500	
record 4				
record 5	A-201	Perryridge	900	
record 6				
record 7	A-110	Downtown	600	
record 8	A-218	Perryridge	700	

Abbildung 1: Heap-Datei mit Datensätzen fester Länge und einer Free List.

2. Eine Slotted Page der Größe 512 Byte soll dimensioniert werden, d.h., die Größe der Felder im Kopfteil der Seite und die Adressierungsart sollen bestimmt werden. Der Kopfteil hat die Form  $(a, f, g_1, p_1, g_2, p_2, \dots, p_n, g_n)$ , wobei  $a$  die Anzahl der Datensätze im Block ( $= n$ ) speichert,  $f$  den Free Space Pointer,  $g_i$  die Größe des  $i$ -ten Datensatzes und  $p_i$  den Pointer zum  $i$ -ten Datensatz,  $1 \leq i \leq n$ . Es werden zwei Optionen in Betracht gezogen.
  - *Byte-Adressierung*: Gibt ein Byte innerhalb des Blocks an, d.h., die kleinste Adresse ist 0 und die größte Adresse ist 511. Ein Datensatz kann bei jedem Byte beginnen.

- *Wort-Adressierung*: Gibt ein 2-Byte Speicherwort innerhalb des Blocks an. Aus der Wort-Adresse  $w$  ergibt sich die Byte-Adresse  $b$  als  $b = 2w$ . Ein Datensatz kann nur an einer Wort-Adresse beginnen.

Beantworten Sie folgende Fragen:

- a) Für die Byte-Adressierung werden für  $a$ ,  $f$ ,  $g_i$  und  $p_i$  jeweils 9 Bit benötigt (maximaler Wert 511). Um Platz zu sparen, werden Byte-Grenzen im Kopfteil ignoriert und die Bits werden dicht gepackt, d.h., für jedes Feld werden nur 9 Bit gebraucht. Wieviele Datensätze der Größe 1 Byte, 2 Byte, oder 20 Byte können so auf eine Slotted Page gespeichert werden?
- b) Wieviele Datensätze der Größe 1 Byte, 2 Byte, 20 Byte können auf eine Slotted Page mit Wort-Adressierung gespeichert werden?
- c) Welche Variante halten Sie für vorteilhafter? Warum?