

## Datenbanken II B: DBMS-Implementierung

### — Hausaufgabe 9 —

Sie sollen jetzt den Anfang von einem Data Dictionary (System-Katalog) anlegen, nämlich die Liste der Relationen in einer Datei. Entwickeln Sie ein Klasse `rel_c` (Relation/Tabelle) mit folgenden Methoden:

- `rel_c(file_t file, str_t name)` (Konstruktor)
- `bool create(int size)`  
Prüft, dass die Datei nicht bereits eine Relation mit dem Namen enthält, und legt dann so eine Relation an, d.h. es wird ein Segment erzeugt und der Relationenname und die Segment-ID in das Data Dictionary eingetragen.

Sie müssen damit rechnen, dass Sie später noch weitere Informationen über Relationen speichern müssen (z.B. Tupellänge, Anzahl Spalten, und ein Link zur ersten Spalte).

Falls die Erzeugung erfolgreich war (es gab nicht schon eine Relation mit dem Namen, und ein Segment der gewünschten Größe konnte angelegt werden), wird `true` geliefert, sonst `false`.

- `bool open()`  
Prüft, ob die Datendatei eine Relation mit diesem Namen enthält, und lädt ggf. Zugriffsdaten der Relation (d.h. mindestens die Segment-ID). Falls es die Relation gibt, soll `true` geliefert werden, sonst `false`.
- `int seg_id()`  
Liefert die ID des Segmentes, in dem die Relation gespeichert ist. Falls das `rel_c`-Objekt nicht in offenem Zustand ist (d.h. vorher nicht `create()` oder `open()` aufgerufen wurden), soll `-1` geliefert werden.

Diese Schnittstelle ist nur als Vorschlag zu verstehen. Sie könnten natürlich auch einen Konstruktor ohne Parameter anbieten, und dann bei `open` und `create` den Namen der Relation und die Datei angeben. Es wäre vielleicht auch gut, die Schnittstelle für den Benutzer der Mini-DB-Bibliothek von internen Aufrufen klarer zu trennen. Eine mögliche Schnittstelle für den Benutzer wird beschrieben unter:

[http://users.informatik.uni-halle.de/~brass/dbi13/db\\_proj.pdf](http://users.informatik.uni-halle.de/~brass/dbi13/db_proj.pdf)

Wenn Sie wollen, können Sie die Länge von Relationen-Namen einschränken, z.B. auf maximal 18 Zeichen. Dann hätten die Einträge in der Liste der Relationen eine feste Länge (und könnten auch direkt C++-Objekten entsprechen).

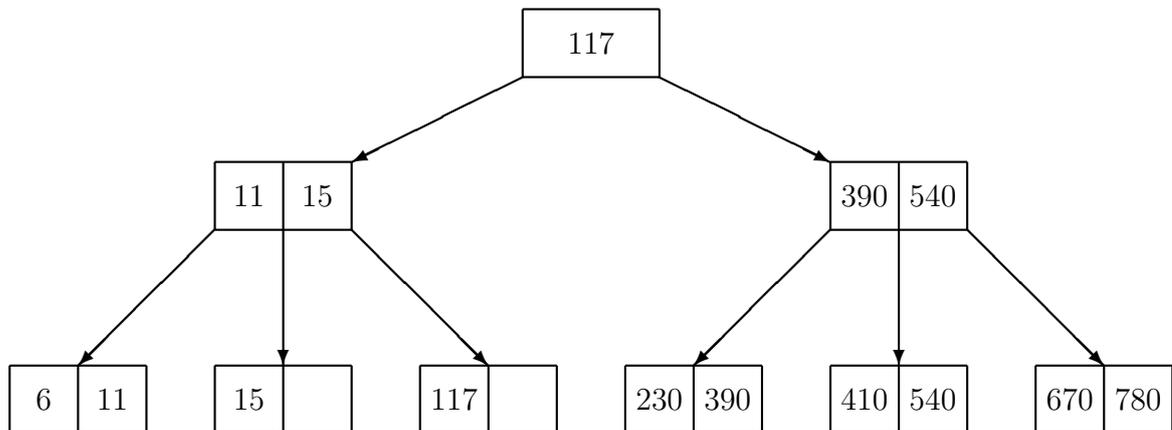
Sie haben in der letzten Hausaufgabe schon einen relationalen Datenblock implementiert. Es würde sich natürlich anbieten, diesen auch für das Data Dictionary zu nutzen. Sie

dürfen die Anzahl Relationen in einer Datei so einschränken, dass die entsprechende Data Dictionary Tabelle in einen Block passt. Es sollten aber mindestens 100 Relationen in einer Datei möglich sein. Normale Relationen (die in Kürze implementiert werden) können natürlich aus vielen Datenblöcken bestehen (sie sollten eine Datei von z.B. 1 GB füllen können).

Sie können aber auch einen ganz eigenen Blocktyp für die Relationsliste verwenden (d.h. Sie brauchen nicht die gleiche Datenstruktur wie für Relationen zu benutzen).

## — Hausaufgabe 9B —

Gegeben sei folgender  $B^+$ -Baum der Höhe 3 und Ordnung 1 (minimal 1 Eintrag pro Knoten, maximal 2):



Geben Sie den resultierenden  $B^+$ -Baum jeweils nach den folgenden 4 Operationen an. Wenden Sie die Operationen immer auf den oben angegebenen  $B^+$ -Baum an, nicht auf das Ergebnis des vorhergehenden Schrittes.

- a) Einfügen von 13.
- b) Einfügen von 9.
- c) Einfügen von 290.
- d) Löschen von 15.