Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

22. März 2012

Datenbanken II B: DBMS-Implementierung
— Musterlösung zur Klausur —

Name: _____

Matrikelnummer: _____

Studiengang: _____

Aufgabe	Punkte	Max. Punkte	Zeit
1 (Platten-Leistung)		5	10 min
2 (RAID-Systeme)		4	10 min
3 (B-Baum)		5	10 min
4 (Index-Auswahl)		4	10 min
5 (Tupel-Format)		10	20 min
6 (Verzögertes Schreiben)		6	10 min
7 (Data Dictionary Implementierung)		6	10 min
8 (Oracle Data Dictionary Anfrage)		6	10 min
Summe		46	90 min

☐ Ich bin gesundheitlich in der Lage, diese Prüfung abzulegen. (Andernfalls bitte bei Aufsicht melden).

☐ Falls ich zu dieser Prüfung nicht angemeldet sein sollte, stelle ich beim Prüfungsausschuss unwiderruflich den Antrag, nachträglich angemeldet zu werden.

_____

(Unterschrift)

Hinweise:

- Bearbeitungsdauer: 90 Minuten
- Skript, Bücher, Notizen sind erlaubt. Notebooks, PDAs, etc. dürfen nicht verwendet werden. Mobiltelefone bitte ausschalten (bei Bedarf mit Aufsicht besprechen).
- Die Klausur hat 11 Seiten. Bitte prüfen Sie die Vollständigkeit.
- Bitte benutzen Sie den vorgegebenen Platz. Wenn Sie auf die Rückseite ausweichen müssen, markieren Sie bitte klar, daß es eine Fortsetzung gibt.
- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Schreiben Sie bitte deutlich lesbar.
- Fragen Sie, wenn Ihnen eine Aufgabe nicht klar ist!

Aufgabe 1 (Platten-Leistung) 5 Punkte

a) Gegeben sei eine Platte mit 12ms durchschnittlicher Zugriffszeit (“Seek Time”), 6000 Umdrehungen pro Minute, 400 KByte pro Spur, und und einem Ultra-320 SCSI Interface (320 MByte/s). Wie lange dauert der Zugriff auf einen Block von 4 KByte durchschnittlich? Es reicht, ganze Millisekunden anzugeben. Begründen Sie Ihr Ergebnis bitte kurz:

---

12 ms Seek Time werden auf jeden Fall benötigt. Dazu kommt die Latenzzeit, also die Zeit für eine halbe Umdrehung:

6000 Umdrehungen/min sind 6000 Umdrehungen in 60000 ms. Eine Umdrehung dauert also $\frac{60000}{6000}$ ms = 10 ms.

$0.5 * \frac{60000}{6000}$ms = 5 ms

Die Lese- und Übertragungszeiten sind in diesem Fall zu vernachlässigen, so dass auf den Block in durchschnittlich 17 ms zugegriffen werden kann. (Z.B. würde das eigentliche Lesen der Daten 4 KB/400 KB, also eine 1/100-Umdrehung dauern, d.h. 0.1ms. Die Übertragung würde ungefähr eine 1/80 ms benötigen.)

---

b) Wie wird sich das Verhältnis zwischen der Dauer wahlfreier Zugriffe und sequentieller Zugriffe bei Magnetplatten in Zukunft voraussichtlich entwickeln?

- ☒ Die Leistung sequentieller Zugriffe verbessert sich schneller als die wahlfreier Zugriffe. Der Unterschied zwischen diesen Zugriffszeiten wird also wachsen.
- ☐ Die Leistung wahlfreier Zugriffe verbessert sich schneller. Der Unterschied wird geringer werden.
- ☐ Die Leistung beider Parameter verbessert sich ungefähr gleich schnell. Das Verhältnis wird sich nicht wesentlich ändern.

c) Warum kann man bei modernen Magnetplatten außen schneller lesen (mehr MB/s) als innen?

Die äußere Spur ist länger, so dass bei konstanter Umdrehungszeit ein größerer Teil der Spur gelesen werden kann. Bei modernen Festplatten sind auf den äußeren Spuren mehr Sektoren untergebracht, so dass tatsächlich mehr Daten in der gleichen Zeit gelesen werden können.

Aufgabe 2 (RAID-Systeme) 4 Punkte

a) Sie müssen sich zwischen einem Platten-Array mit 4 Platten zu jeweils 500 GB und einem mit 8 Platten zu jeweils 250 GB entscheiden. Abgesehen von der Kapazität haben die Platten die gleichen Daten (Zugriffszeit etc.). Das Array mit den 8 Platten ist 40% teurer. Es verbraucht auch mehr Strom. Gibt es irgendeinen Grund, der für das Array mit 8 Platten sprechen würde?

- Platten können parallel arbeiten, insbesondere das Seek.
- Wenn Speicherplatz in der Größe einer Platte für das Speichern von Paritäten verwendet werden soll, erhält man mit 8 Platten der halben Kapazität insgesamt mehr Speicherplatz.

---

b) Sie haben sich für das Array mit 4 Platten entschieden. Sie wollen den Ausfall einer einzelnen Platte ohne Datenverlust überstehen, und ansonsten die Speicherkapazität des Gesamtsystems maximieren. Bei gleicher Speicherkapazität soll die Leistung möglichst gut sein. Welchen RAID-Level würden Sie wählen? Welche Gesamt-Speicherkapazität bekommen Sie damit?

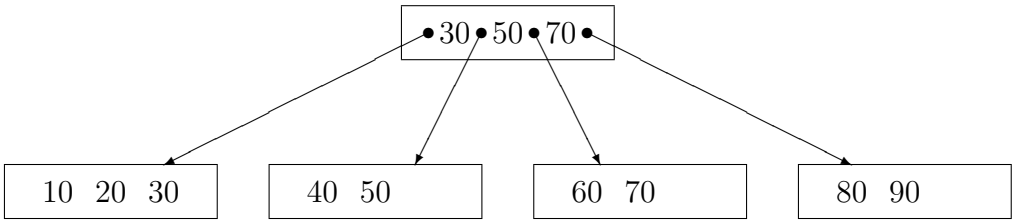
Ideal ist RAID 5:

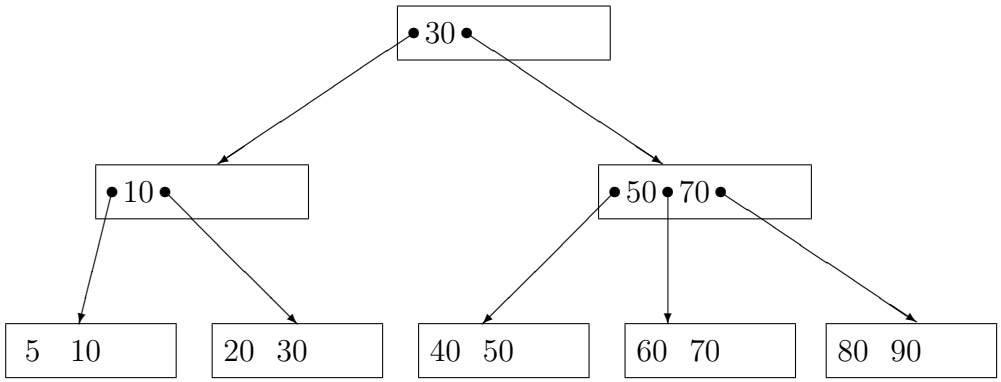
- Durch Verwendung von Paritätsblöcken statt Spiegeln erhöht sich die Speicherkapazität.
- Verglichen mit Raid 4 werden die Paritätsblöcke auf alle Platten verteilt, so dass besser parallel gearbeitet werden kann.

Man erhält so insgesamt $(4-1) * 500$ GB $= 1.5$ TB Gesamt-Speicherkapazität.

Aufgabe 3 (B-Baum) 5 Punkte

Angenommen, in einen Knoten passen maximal 3 Attributwerte und die zugehörigen Zeiger. Fügen Sie in den folgenden B^+-Baum den Wert 5 ein:





Aufgabe 4 (Index-Auswahl) 4 Punkte

Gegeben sei eine Tabelle in einem Data Warehouse, in das täglich die Verkaufszahlen für verschiedene Waren für eine Menge von Filialen eingespeichert werden:

VERKAUF(DATUM, FILIAL_NR, WAREN_NR, STUECKZAHL).

Es finden nur Einfügungen statt und keine Löschungen oder Updates. Es werden immer alle Daten einer Filiale für den aktuellen Tag eingefügt, bevor die Daten der nächsten Filiale eingefügt werden. Es gibt viele verschiedene Waren und viele Filialen.

a) Angenommen, Sie sollen einen optimalen Index wählen zur Beschleunigung von Anfragen der folgenden Art (wobei für die Konstanten beliebige Werte eingesetzt werden können):

```
SELECT SUM(STUECKZAHL)
FROM   VERKAUF
WHERE  WAREN_NR = 12345
AND    FILIAL_NR = 678
AND    DATUM BETWEEN '01.01.2012' AND '31.01.2012'
```

Welchen Index würden Sie wählen? Geben Sie den `CREATE INDEX` Befehl an. Begründen Sie außerdem Ihre Entscheidung. Es kann mehrere korrekte Antworten geben.

```
CREATE INDEX I1 ON VERKAUF(WAREN_NR, FILIAL_NR, DATUM, STUECKZAHL)
```

Die Sortierreihenfolge auf Blattebene muss so sein, dass für das Datum ein möglichst kleines Intervall durchlaufen werden muss. Wenn man die Stückzahl mit indiziert, müssen keine wahlfreien Zugriffe auf die Datenblöcke erfolgen. Die Reihenfolge der Attribute `WAREN_NR` und `FILIAL_NR` im Index ist egal.

---

b) Betrachten Sie nun folgende Anfrage:

```
SELECT FILIALE, DATUM
FROM   VERKAUF
WHERE  WAREN_NR = 12345
AND    STUECKZAHL > 1000
```

Welcher der folgenden Indexe ist am besten geeignet, um diese Anfrage auszuwerten?

- ☒ `IND1` über `(WAREN_NR, STUECKZAHL)`
- ☐ `IND2` über `(STUECKZAHL, WARENNR)`
- ☐ `IND3` über `(FILIALE, STUECKZAHL, WARENNR)`

Aufgabe 5 (Tupel-Format) 10 Punkte

Im Benchmark am Ende der Übungen sollte eine Relation `R` mit drei Attributen angelegt werden:

- `A` vom Typ `NUMERIC(5)`,
- `B` vom Typ `VARCHAR(40)`,
- `C` vom Typ `NUMERIC(5)`.

Es sollten in die Relation 10000 Tupel eingefügt werden, wobei `A` von `1` bis `10000` läuft, `C` in umgekehrter Richtung von `9999` bis `0`, und `B` den Wert `ABCDEFGHIJKLMNOPQRSTUVWXYZ` hat (in allen Tupeln, die Länge sind 26 Zeichen). Der eigentliche Benchmark bestand aus dem Einfügungen der 10000 Tupel und einem (mehrfach wiederholten) Full Table Scan, wobei auf die Werte von `A` und `C` zugegriffen wird (nicht `B`).

a) Wie lang wäre ein Tupel im Oracle-Format? Um es konkret zu machen, berechnen Sie bitte die Länge des ersten Tupels, d.h. `A=1`, `C=9999`. Erläutern Sie kurz, wie Sie das Ergebnis berechnet haben.

Bei Zahlen speichert Oracle zwei Ziffern in einem Byte. Der Exponent braucht 1 Byte.

Row Header	Länge A	Wert A	Länge B	Wert B	Länge C	Wert C
3	1	2	1	26	1	3

Insgesamt 37 Byte.

---

b) Wie würde man bei Oracle `PCTFREE` wählen, wenn man für den Benchmark optimale Voraussetzungen haben will?

Die Daten sollen möglichst stark gepackt werden, und es finden keine Updates statt, also wählt man `PCTFREE` $= 0$.

c) Könnte man bei `PCTUSED` etwas falsch machen, oder ist das in diesem Fall völlig egal? Welchen Wert würden Sie wählen? Was halten Sie von `PCTUSED = 100-PCTFREE`?

`PCTUSED` wird dann relevant, wenn beim Einfügen einer neuen Zeile ein Block in der Freispeicherliste angetroffen wird, der nicht mehr genügend Speicherplatz für diese Zeile hat. Dieser Block wird entfernt, es sei denn, es sind noch nicht `PCTUSED` Prozent des Blocks gefüllt. Ein Wert von 100 würde dazu führen, dass Blöcke nur dann aus der Freispeicherliste entfernt werden, wenn sie tatsächlich zu 100% (also exakt bis auf das letzte Byte) gefüllt sind. Das ist unwahrscheinlich: Wenn alle Tupel 37 Byte lang wären, müßte der freie Speicherplatz eines Blockes durch 37 teilbar sein. Im Endeffekt würde die Freispeicherliste immer länger, und Einfügungen würden unnötig verzögert. Der Abstand zwischen `PCTUSED` und `100-PCTFREE` muss mindestens so groß sein, das ein Tupel (minus ein Byte) dazwischen passt. Er darf aber auch viel größer sein. Da in der Aufgabe die Blockgröße nicht angegeben ist, kann man den maximalen Wert für `PCTUSED` nicht berechnen. Typische Werte liegen bei ca. 50–95.

---

d) Die meisten Teilnehmer der Übungen haben ein Tupelformat mit Feldern fester Länge gewählt. Erläutern Sie kurz die Vor- und Nachteile gegenüber dem Oracle-Format. Wenn Sie ein anderes Format gewählt haben, können Sie auch dieses Format beschreiben, und das dann mit dem Oracle-Format vergleichen. Nennen Sie mindestens drei verschiedene Aspekte.

- Platz: Tupel sollen möglichst kompakt gespeichert werden, so dass ein Full Table Scan möglichst schnell ist. Hier ist das Oracle-Format besser.
- Zugriff auf bestimmtes Attribut: Bei einem festen Format steht der Offset bereits im Voraus fest. Oracle dagegen muss Längenbytes der vorangehenden Spalten lesen.
- Schemaänderungen: Mit dem Oracle-Format können Spalten zu einer bestehenden Tabelle leicht mit `ALTER TABLE` hinzugefügt werden, ohne die Daten umzukopieren. Das gilt auch für Vergrößerungen der Maximallänge bei `VARCHAR`.

---

e) Wie würden Sie die Tupel speichern, um beim Benchmark möglichst gut abzuschneiden? Gibt es ein Tupelformat, das in diesem Benchmark sowohl das Oracle-Format als auch das Format mit fester Länge schlagen könnte?

Man speichert zuerst Spalten mit fester Länge und dann Spalten mit variabler Länge (es wird hier angenommen, das `A` und `C` als 32-Bit Integers gespeichert werden):

Länge Tupel oder Zeiger auf nächstes Tupel	Wert A	Wert C	Länge B	Wert B
	4	4	1	26

Aufgabe 6 (Verzögertes Schreiben) 6 Punkte

Oracle und viele andere Systeme schreiben veränderte Blöcke nicht sofort auf die Platte.

a) Wie können Sie dennoch die Dauerhaftigkeit der Änderungen beim “`COMMIT`” sichern?

Dafür gibt es die Log-Dateien. Diese enthalten im Prinzip ein Protokoll aller Änderungen. Beim Neustart nach einem Absturz werden die Logdateien gelesen und alle Änderungen, die ihren Weg noch nicht in die Datenbank-Dateien gefunden haben, neu ausgeführt (REDO-Phase). Dazu hat jede Änderung eine fortlaufende Nummer (Log Sequence Number). Die LSN der letzten Änderung steht in jedem DB-Block. So kann das System leicht sehen, ob die Änderung schon im Block enthalten ist. Danach kommt die UNDO-Phase: Änderungen nicht abgeschlossener Transaktionen werden zurückgenommen. (Natürlich muss man in der Klausur nicht so viel schreiben.)

---

b) Wenn doch auf jeden Fall beim `COMMIT` etwas auf die Platte geschrieben werden muß, warum kann man auf diese Art das `COMMIT` dennoch deutlich schneller bestätigen, als wenn man die veränderten Blöcke sichern würde? Nennen Sie mindestens zwei Gründe:

Die Änderungen sind kleiner als die ganzen Blöcke, die die Änderungen enthalten. Z.B. 10 kleine Tupel in verschiedene Relationen eingefügt passen in einen Block der Log-Datei, aber es wären 10 verschiedene Blöcke der Datenbank-Dateien betroffen. Außerdem ist die Log-Datei “Write only”. Wenn sie die einzige Datei auf der Platte ist, steht der Schreib/Lesekopf immer auf der richtigen Spur.

---

c) Gibt es eine Situation, in der man auf diese Art insgesamt weniger Schreiboperationen für Blöcke hat, als wenn man veränderte Blöcke sofort schreiben würde?

Ja, ein Block kann mehrfach von verschiedenen Transaktionen geändert und nur einmal am Ende geschrieben werden.

Aufgabe 7 (Implementierung von Data Dictionaries) 6 Punkte

a) Bei einem sehr einfachen DBMS, wie dem, was wir in den Übungen entwickelt haben, welche Daten müßte man im Data Dictionary über Tabellen/Relationen speichern? D.h. welche Spalten müßte eine Katalog-Table `SYS_TABLES` haben?

- Name der Tabelle
- Segment-Nummer oder Start-Block
- Anzahl Blöcke, die bei einem Full Table Scan gelesen werden müssen
- Beginn einer Liste von Blöcken mit freiem Speicherplatz (für Einfügung neuer Tupel)
- Länge der Tupel
- Eventuell Verkettung, Link zu erster Spalte
  (Nummer des Eintrags in `SYS_COLUMNS`)

(Manche Daten können eventuell auch anders repräsentiert werden.)

---

b) Welche Daten müßte man über Spalten speichern? Sie können ein Tupelformat fester Länge voraussetzen (falls Sie ein anderes Format verwendet haben, oder hier besprechen wollen, beschreiben Sie ganz kurz das Format). Wir hatten die Datentypen `INTEGER` und `CHAR(`n`)`. D.h. welche Spalten müßte eine Katalog-Table `SYS_COLUMNS` haben?

- Name der Spalte;
- Datentyp (int oder String)
- Maximale Länge des Strings
- Offset (Position des Spaltenwertes im Tupel, wie viel Bytes Abstand zum Beginn des Tupels)
- Zugehörige Relation;
- Position der Spalte von links nach rechts in der Relation (kann man auch über Offset bekommen)
- Ggf. Verkettung mit nächster Spalte der Relation

---

c) Wenn man das Data Dictionary mit den gleichen Datenstrukturen wie die Benutzertabellen implementiert, warum kann das überhaupt funktionieren? Man braucht doch das Data Dictionary für den Tabellenzugriff.

Die Zugriffsdaten (Start-Block, Tupel-Länge, Offsets der Spalten) müssen in das DBMS eincompiliert sein.

Aufgabe 8 (Oracle Data Dictionary Anfrage) 6 Punkte

a) Schreiben Sie eine SQL-Abfrage an das Oracle Data Dictionary, die Non-Unique Indexe über kleinen Tabellen findet (maximal 8 Blöcke unter der “High Water Mark”, das Segment kann größer sein). Solche Indexe sind ja normalerweise nicht empfohlen (Unique Indexe sind nötig zur Überwachung von Schlüsseln). Sie dürfen davon ausgehen, das alle relevanten Tabellen und Indexe Ihnen gehören. Es reicht, wenn Sie den Namen des Indexes und der Tabelle ausgeben. Folgende Tabellen des Data Dictionaries sind möglicherweise nützlich:

- `IND(TABLE_NAME, TABLE_OWNER, INDEX_NAME, UNIQUENESS` (Werte `UNIQUE` und `NONUNIQUE`), `TABLESPACE_NAME, INITIAL_EXTENT, NEXT_EXTENT, PCT_INCREASE, PCT_FREE, BLEVEL, LEAF_BLOCKS, DISTINCT_KEYS, AVG_LEAF_BLOCKS_PER_KEY, AVG_DATA_BLOCKS_PER_KEY, CLUSTERING_FACTOR, ...)`
- `USER_IND_COLUMNS(INDEX_NAME, TABLE_NAME, COLUMN_NAME, COLUMN_POSITION, COLUMN_LENGTH, ...)`
- `TABS(TABLE_NAME, TABLESPACE_NAME, PCT_FREE, PCT_USED, INITIAL_EXTENT, NEXT_EXTENT, MIN_EXTENTS, MAX_EXTENTS, PCT_INCREASE, FREELISTS, NUM_ROWS, BLOCKS, EMPTY_BLOCKS, CHAIN_CNT, AVG_ROW_LEN, AVG_SPACE, AVG_SPACE_FREELIST_BLOCKS, NUM_FREELIST_BLOCKS, LAST_ANALYZED, ...)`
- `DBA_SEGMENTS(OWNER, SEGMENT_NAME, PARTITION_NAME, SEGMENT_TYPE, TABLESPACE_NAME, HEADER_FILE, HEADER_BLOCK, BYTES, BLOCKS, EXTENTS, INITIAL_EXTENT, NEXT_EXTENT, MIN_EXTENTS, MAX_EXTENTS, PCT_INCREASE, FREELISTS, FREELIST_GROUPS, RELATIVE_FNO, BUFFER_POOL, ...)`

```
SELECT I.TABLE_NAME, I.INDEX_NAME
FROM IND I, TABS T
WHERE I.UNIQUENESS = 'NONUNIQUE'
    AND I.TABLE_NAME = T.TABLE_NAME
    AND T.BLOCKS < 8
```

---

b) Was müssen Sie tun, damit die Anfrage auch aktuell zutreffende Ergebnisse liefert? (Denken Sie allgemein an die statistischen Daten im Data Dictionary.)

Der ANALYZE TABLE Befehl muss ausgeführt werden für alle Tabellen, die möglicherweise betroffen sind. Nur damit wird die Angabe BLOCKS aktualisiert.