Aufgabenblatt 5
Optimierung und Recovery

- Abgabetermin: **Dienstag, 26.1.2010 (23:59 Uhr)**
- Zur Prüfungszulassung muss ein Aufgabenblatt mit mind. 25% der Punkte bewertet werden und alle weiteren Aufgabenblätter mit mindestens 50% der Punkte.
- Die Aufgaben sollen in Zweiergruppen bearbeitet werden.
- Abgabe:
  - per E-Mail an dbs2-200910@hpi.uni-potsdam.de mit Subject
    Abgabe DBS II: Aufgabenblatt <n> Namen
  - ausschließlich pdf-Dateien
  - eine Datei pro Aufgabe mit folgendem Dateinamen:
    blatt<aufgabenblattNr>aufgabe<aufgabenNr><Nachnamen>.pdf
    Bitte **keine Leerzeichen, Unterstriche, Umlaute, Sonderzeichen**, ... im Dateinamen!
  - ***jedes* Blatt beschriftet mit Namen**
  - Wir korrigieren die Abgaben aufgabenweise. Das beschriebene Verfahren vereinfacht uns die Arbeit erheblich!

Aufgabe 1: Join-Kardinalität

Schätze die Kardinalität des Joins über $R(A,B) \bowtie S(B,C)$ ab unter Verwendung der Histogramme für $R(B)$ und $S(B)$, die die Häufigkeit der 4 häufigsten Werte angeben. Beide Relationen enthalten 20 unterschiedliche Werte im Attribut B ($V(R,B) = V(S,B) = 20$). **5 P**

	0	1	2	3	4	andere Werte
$R.B$	5	6	4	5		32
$S.B$	10	8	5		7	48

Vergleiche die berechnete Abschätzung mit der einfacheren Abschätzung, die man unter der Annahme erhält, dass alle 20 Attributwerte gleichverteilt sind und $T(R) = 52$ und $T(S) = 78$ gilt.

Aufgabe 2: Join-Reihenfolge

Gegeben seien die folgenden Relationen und deren Statistiken:

$W(A,B)$	$X(B,C)$	$Y(C,D)$	$Z(D,A)$
$T(W) = 100$	$T(X) = 200$	$T(Y) = 300$	$T(Z) = 400$
$V(W,A) = 20$			$V(Z,A) = 100$
$V(W,B) = 60$	$V(X,B) = 50$		
	$V(X,C) = 100$	$V(Y,C) = 50$	
		$V(Y,D) = 50$	$V(Z,D) = 40$

Bestimme die Join-Reihenfolge als Left Deep Tree. Gib dazu alle Tabellen des Algorithmus der Dynamischen Programmierung an. Was ist die optimale Join-Reihenfolge? Was sind deren Kosten? **10 P**

Aufgabe 3: Hash-Join

Nimm an, dass für das Bewegen des Schreib-Lesekopfes auf einen beliebigen Block durchschnittlich 100 ms und für das Lesen bzw. Schreiben eines Blocks 0,5 ms benötigt werden. Somit beträgt die Zeit für das Übertragen (in beide Richtungen) eines beliebigen Blocks 100,5 ms und für das Übertragen von k aufeinanderfolgenden Blöcken $100 + k * 0,5$ ms.

Sei $B(R) = 1000$, $B(S) = 500$ und $M = 101$. Es soll ein Hash-basierter Join-Algorithmus für die Berechnung von $R \bowtie S$ verwendet werden, wobei die Bucketanzahl so gering wie möglich sein soll. Nimm hierbei an, dass die Tupel gleichmäßig auf die Buckets verteilt sind.

a) Wie lange dauert die Berechnung von $R \bowtie S$, wenn ein zweiphasiger Hash-Join-Algorithmus verwendet wird? **4 P**

b) Wie lange dauert die Berechnung von $R \bowtie S$, wenn ein hybrider Hash-Join-Algorithmus verwendet wird? **4 P**

c) Wie lange dauert die Berechnung von $R \bowtie S$, wenn ein Sort-Join-Algorithmus verwendet wird und sortierte Teillisten auf aufeinanderfolgenden Blöcken der Festplatte gespeichert werden? Nimm dabei an, dass alle Tupel mit gleichem Join-Wert in einen Block passen. **4 P**

Aufgabe 4: Undo-Logging

Gegeben sei die folgende Folge von Undo-Log-Einträgen, die von zwei T und U erzeugt wurden:

```
<START T>
<T, A, 10>
<START U>
<U, B, 20>
<T, C, 30>
<U, D, 40>
<COMMIT U>
<T, E, 50>
<COMMIT T>
```

Beschreibe und begründe die vom Recovery-Manager auszuführenden Aktionen, wenn sich ein Fehler ereignet hat und der letzte auf der Festplatte geschriebene Log-Eintrag

a) `<START U>`

b) `<COMMIT U>`

c) `<T, E, 50>`

d) `<COMMIT T>`

ist.

Gib zusätzlich für jeden der vier Fälle an, welche von T bzw. U geschriebenen Werte bereits auf die Festplatte geschrieben sein *müssen*. Welche Werte *könnten* bereits auf die Festplatte geschrieben worden sein? **12 P**