Aufgabenblatt 6
Transaktionsmanagement und Anfragebearbeitung

- Abgabetermin: **Dienstag, 5.7.11**
- 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 dbs1-2011@hpi.uni-potsdam.de
  - 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: Konfliktserialisierbarkeit

Sind die folgenden Schedules konfliktserialisierbar?

Begründe deine Entscheidung jeweils auf zwei Wegen:

- mittels des graphbasierten Tests und
- durch Angabe eines konfliktäquivalenten seriellen Schedules bzw. durch Zeigen eines möglichen Konflikts (Angabe der entsprechenden Aktionen). Bei der Angabe eines konfliktäquivalenten seriellen Schedules ist es ausreichend, die Reihenfolge der Transaktionen anzugeben.

a) $S_1 = \langle r_1(X), r_2(X), w_1(X), r_2(Y), w_2(Y), r_1(Y) \rangle$ **3 P**

b) $S_2 = \langle r_1(X), r_2(X), w_1(X), r_2(Y), w_1(Y), r_2(Y) \rangle$ **3 P**

c) $S_3 = \langle r_3(X), r_1(Y), w_3(Y), w_2(X), w_2(Y) \rangle$ **3 P**

d) $S_4 = \langle w_1(X), w_2(X), w_2(Y), w_1(Y), r_2(X), w_3(Y) \rangle$ **3 P**

Aufgabe 2: Konsistenz, 2PL-Bedingung, Legalität

Betrachte den folgenden Schedule (entspricht S_1 aus Aufgabe 1 mit ergänzten $*$lock-Operationen):

$sl_1(X), r_1(X), sl_2(X), r_2(X), u_2(X), w_1(X), xl_2(Y), r_2(Y), w_2(Y), sl_1(Y), u_1(X), r_1(Y), u_1(Y), u_2(Y)$

a) Sind die beiden Transaktionen konsistent? Warum (nicht)? **2 P**

b) Erfüllen die beiden Transaktionen die 2PL-Bedingung? Warum (nicht)? **2 P**

c) Ist der Schedule legal? Warum (nicht)? (Hinweis: Betrachte nur die gegebene Reihenfolge. Es sollen keine Aktionen zurückgestellt oder verschoben werden.) **1 P**

Aufgabe 3: Scheduler

Betrachte den folgenden Schedule:

$r_1(A), r_2(B), r_3(C), r_1(B), r_2(C), r_3(D), w_1(C), w_2(D), w_3(E)$

Hinweis zum Einfügen von $*$lock-Operationen:

- Lock-, shared-lock- und exclusive-lock Operationen so eng wie möglich an den Aktionen;
- Die Transaktionen sollen konsistent sein und die 2PL-Eigenschaft erfüllen.
- Unlocks ganz ans Ende.

a) Füge lock- und unlock-Operationen ein.

Stelle den Ablauf in einem DBMS-Scheduler dar. Welcher Effekt ist zu beobachten? Welchem seriellen Schedule entspricht die Ausführung, d. h. welcher serielle Schedule hat den gleichen Effekt? **5 P**

b) Füge shared-lock-, exclusive-lock- und unlock-Operationen ein.

Stelle den Ablauf in einem DBMS-Scheduler dar. Ist eine Veränderung zur vorigen Teilaufgabe zu beobachten? Welchem seriellen Schedule entspricht diese Ausführung, d. h. welcher serielle Schedule hat den gleichen Effekt? **5 P**

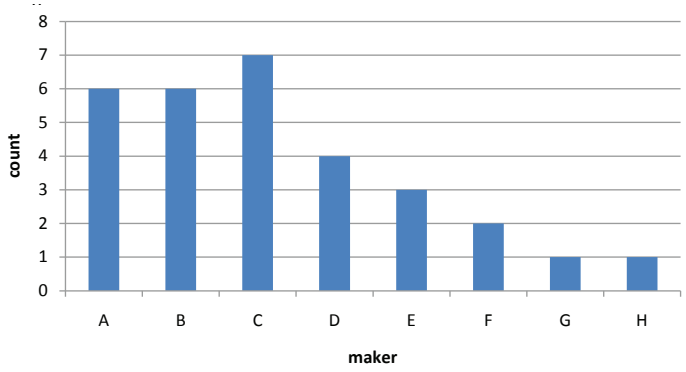
Aufgabe 4: Anfragebearbeitung

Gegeben sei das aus der Übung bekannte Produkt-Schema:

- Product(maker, model, type)
- PC(model, speed, ram, hd, rd, price) mit model $\rightarrow$ Product.model
- Laptop(model, speed, ram, hd, screen, price) mit model $\rightarrow$ Product.model
- Printer(model, color, type, price) mit model $\rightarrow$ Product.model

Nimm die folgenden Kardinalitäten/ Werteverteilungen an:

- Product: 30 Tupel
- PC: 13 Tupel
- Laptop: 10 Tupel
- Printer: 7 Tupel
- maker:



Bestimme die Ergebniskardinalität (also die Anzahl der Tupel im Ergebnis) der folgenden Anfrage. Skizziere dazu den Operatorbaum und gib die Anzahl der Tupel an jeder Kante an. **5 P**

$\pi_{model,price}(\sigma_{maker='A' \vee maker='B'}(Product \bowtie (\pi_{model,price}(PC) \cup \pi_{model,price}(Laptop) \cup \pi_{model,price}(Printer))))$