



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS22/23
Michael Jungmair, Stefan Lehner, Moritz Sichert, Lukas Vogel (gdb@in.tum.de)
<https://db.in.tum.de/teaching/ws2223/grundlagen/>

Blatt Nr. 13

Hausaufgabe 1

Gegeben sei die Anfrage:

```
select *  
  from R, S, T  
 where R.A = S.A and S.B = T.B and T.C = R.A
```

Des Weiteren soll gelten:

- S.A und T.C seien Fremdschlüssel auf R
- S.B sei Fremdschlüssel auf T
- R.A, T.B seien Primärschlüssel von R respektive T
- Ihre Query-Engine unterstützt nur Nested-Loop-Joins
- Kardinalitäten: $|R| = 100$, $|S| = 1000$, $|T| = 10$
- Es gibt keine Indexe

- a) Für Equijoins zwischen R_i mit Fremdschlüssel auf den Primärschlüssel in R_j gilt die Abschätzung:

$$f_{i,j} = \frac{1}{|R_j|}$$

Warum?

- b) Bestimmen Sie, wie in der Vorlesung gezeigt, den optimalen Ausführungsplan als Baum mit Kosten-/Kardinalitätsabschätzungen mit Hilfe von Dynamischem Programmieren. Verwenden Sie die Kostenfunktion C_{out} .

Hausaufgabe 2

Wofür stehen die vier Buchstaben ACID? Erklären Sie für jeden der vier Konzepte, warum es für eine Datenbank wichtig ist. Geben Sie dazu jeweils ein Beispiel an, was passieren könnte, wenn dieses Konzept nicht gelten würde.

Hausaufgabe 3

Sie verwenden ein Datenbanksystem mit Write-Ahead-Logging und der Strategie $\neg force$ und *steal*. Die Datenbank verwaltet lediglich zwei Datenobjekte, X mit dem Anfangswert 10 und Y mit dem Anfangswert 100.

Sie starten die 3 Transaktionen T_1 , T_2 und T_3 zum gleichen Zeitpunkt:

T_1	T_2	T_3
BOT	BOT	BOT
$r(X, x_1)$	$r(Y, y_2)$	$r(X, x_3)$
$x_1 := x_1 + 1$	$r(X, x_2)$	$x_3 := x_3 \cdot 10$
$w(X, x_1)$	$y_2 := y_2 \cdot 2$	$w(X, x_3)$
COMMIT	$x_2 := x_2 + 5$	COMMIT
	$w(Y, y_2)$	
	$w(X, x_2)$	
	COMMIT	

Während der Ausführung stürzt Ihre Datenbank ab. Sie wissen nicht, ob - und wenn ja, welche - Transaktionen festgeschrieben wurden. Sie wissen nur, dass die Datenbank ausschließlich *serielle Historien* erzeugt, also dass Transaktionen immer atomar ausgeführt werden und somit keine Verzahnung möglich ist. Bevor Sie die Datenbank neu starten, durchsuchen Sie die Festplatte und stellen fest, dass Y dort den Wert 200 hat. Nachdem die Datenbank neu gestartet wurde und der Recovery-Prozess abgeschlossen ist, liefert sie für X den Wert 110.

Sie wollen nun dem Fehler auf den Grund gehen:

- Finden Sie zunächst anhand der Zwischenwerte für X und Y heraus, welche Transaktionen *winner* sind, und welche *loser*.
- Geben Sie das Log an, wie es zum Zeitpunkt des Absturzes auf der Platte stand (verwenden Sie logische Protokollierung).
- Geben Sie das Log nach Beendigung des Recovery-Prozesses an.

Hausaufgabe 4

Demonstrieren Sie anhand eines Beispiels, dass man die Strategien *force* und \neg *steal* nicht kombinieren kann, wenn parallele Transaktionen gleichzeitig Änderungen an Datenobjekten innerhalb einer Seite durchführen. Betrachten Sie dazu z.B. die unten dargestellte Seitenbelegung, bei der die Seite P_A die beiden Datensätze A und D enthält. Entwerfen Sie eine verzahnte Ausführung zweier Transaktionen, bei der eine Kombination aus *force* und \neg *steal* ausgeschlossen ist.

