

# AG Datenbanken und Informationssysteme

Wintersemester 2006 / 2007

Prof. Dr.-Ing. Dr. h. c. Theo Härder  
Fachbereich Informatik  
Technische Universität Kaiserslautern



<http://www.dvs.informatik.uni-kl.de>

## 12. Übungsblatt

Für die Übung am Donnerstag, **01. Februar 2007**,  
von 15:30 bis 17:00 Uhr in 13/222.

### Aufgabe 1: Redo Lauf

Weisen Sie nach, daß beim ARIES-Ansatz (Non-Atomic, Steal, NoForce, Fuzzy Checkpoint) die Idempotenz des Wiederanlaufs das REDO aller protokollierten Änderungen – also auch der von Verlierertransaktionen durchgeführten Änderungen – verlangt. Betrachten Sie dazu zwei Transaktionen  $T_1$  und  $T_2$ , die parallel ablaufen und die Datensätze  $A(T_1)$  resp.  $B(T_2)$  der Seite  $P$  modifizieren. Hierbei soll  $T_1$  die Verlierer- und  $T_2$  die Gewinnertransaktion sein. Diskutieren Sie die Zustände der Seite  $P$  auf der Platte:

- Vor der Modifikation von  $A$ .
- Nach der Modifikation von  $A$ , aber vor der Modifikation von  $B$ .
- Nach der Modifikation von  $B$ .

Was passiert bei Wiederanlauf in Bezug auf diese drei möglichen Zustände der Seite  $P$ ?

### Aufgabe 2: Logging und Recovery mit Rollback und Sicherungspunkten

In der folgenden Tabelle sei eine Historie gegeben, in der ein Fuzzy Checkpoint erzeugt wurde. Sie enthält außerdem zwei Rollback-Operationen, die mit  $a_i$  begonnen wurden, wobei die eine Rollback-Operation vor dem Systemabsturz erfolgreich ausgeführt wurde und die andere nicht. Das DBMS stürzt nach der Ausführung der letzten Operation ab und verwendet vollständiges Redo bei der Recovery.

a) Vervollständigen Sie die folgende Tabelle:

Zeit	Aktion	Änderung im DB-Puffer (Seite, LSN)	Änderung in der DB (Seite, LSN)	Log-Eintrag im Log-Puffer (LSN, TAID, Log-Info, PrevLSN, UndoNxtLSN)	Zur Log-Datei hinzugefügte Log-Einträge (LSNs)
z <sub>1</sub>	b <sub>1</sub>				
z <sub>2</sub>	w <sub>1</sub> (A)				
z <sub>3</sub>	w <sub>1</sub> (B)				
z <sub>4</sub>	c <sub>1</sub>				
z <sub>5</sub>	flush(P <sub>A</sub> )				
z <sub>6</sub>	b <sub>2</sub>				
z <sub>7</sub>	w <sub>2</sub> (A)				
z <sub>8</sub>	w <sub>2</sub> (C)				
z <sub>9</sub>	cp				
z <sub>10</sub>	a <sub>2</sub>				
z <sub>11</sub>	comp <sub>2</sub> (C)				
z <sub>12</sub>	comp <sub>2</sub> (A)				
z <sub>13</sub>	r <sub>2</sub>				
z <sub>14</sub>	b <sub>3</sub>				
z <sub>15</sub>	flush(P <sub>A</sub> )				
z <sub>16</sub>	w <sub>3</sub> (A)				
z <sub>17</sub>	w <sub>3</sub> (B)				
z <sub>18</sub>	flush(P <sub>B</sub> )				
z <sub>19</sub>	w <sub>3</sub> (C)				
z <sub>20</sub>	a <sub>3</sub>				
z <sub>21</sub>	comp <sub>3</sub> (C)				
z <sub>22</sub>	b <sub>4</sub>				
z <sub>23</sub>	comp <sub>3</sub> (B)				
z <sub>24</sub>	w <sub>4</sub> (D)				
z <sub>25</sub>	flush(P <sub>B</sub> )				
z <sub>26</sub>	c <sub>4</sub>				

b) Führen Sie den Restart durch und ergänzen Sie anschließend die Tabelle aus a).

**Aufgabe 3: Integritätsbedingungen in SQL am Beispiel „Lieferung“**

Gegeben seien folgende Relationen:

LIEFERANT (LNR, LNAME, STATUS, ORT)

PROJEKT (PNR, PNAME, ORT)

LIEFERUNG (LNR, TNR, PNR, MENGE)

TEIL (TNR, TNAME, FARBE, GEWICHT)

Formulieren Sie folgende Integritätsbedingungen in SQL:

- a) Jedes Teil hat einen positiven Gewichtswert.
- b) Alle roten Teile wiegen weniger als 50 (Gewichtseinheiten).
- c) Keine zwei Projekte dürfen sich an demselben Ort befinden.
- d) Jedes Projekt muss sich an einem Ort befinden, an dem es mindestens einen Lieferanten gibt.
- e) Die Lieferanten aus 'MZ' liefern insgesamt mehr Teile als die aus 'SB'.