

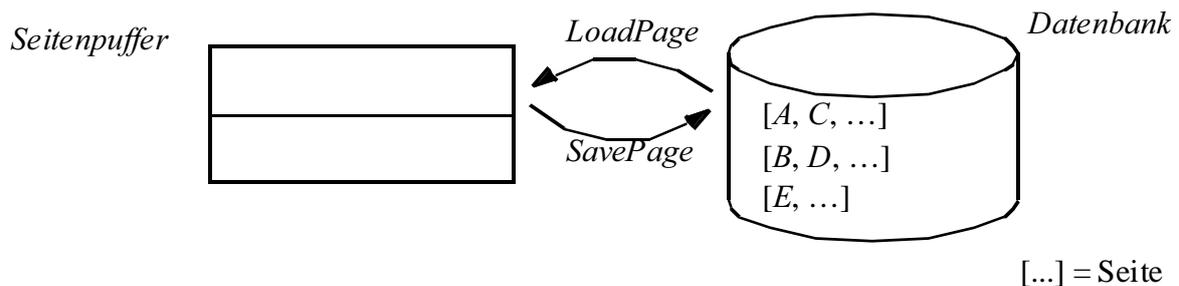
**Datenbanksysteme II**  
 SS 2006

**Übungsblatt 4: Logging**

Besprechung: 29.05. – 30.05.2006

**Aufgabe 4-1** *Verdrängungs- und Ausschreibestrategien*

Betrachten Sie das folgende Beispiel, bei dem die Objekte *A* und *C* sowie *B* und *D* jeweils auf derselben Seite in der Datenbank abgelegt sind. Im folgenden nehmen wir die Einbringstrategie *Update-in-Place* an.



Gegeben sei der folgende Schedule:

$S = r_1(A), r_1(B), r_2(C), w_2(C), w_1(B), r_1(E), w_1(E), EOT_2, w_1(A), EOT_1, r_3(D), w_3(D), EOT_3$

- Erläutern Sie, warum sich die Kombination *Force / No-Steal* für das direkte Einbringen (*Update-in-Place*) nicht realisieren lässt.
- Beschreiben Sie eine mögliche Abfolge von *loadPage*- und *savePage*-Operationen zwischen der Datenbank und dem *unbeschränkten* Seitenpuffer im Hauptspeicher. Dabei soll die Verdrängungsstrategie *No-Steal* benutzt werden. Geben Sie an, welche Ausschreibestrategie Sie verwenden
- Welche Auswirkungen ergeben sich, wenn Sie die Verdrängungsstrategie *Steal* benutzen und der Seitenpuffer eine *beschränkte Kapazität* von zwei Seiten hat? Geben Sie an, welche Ausschreibestrategie Sie verwenden.

**Aufgabe 4-2** *WAL-Prinzip und Commit-Regel*

Erläutern Sie anhand des Beispiels aus Aufgabe 4-1 die Probleme, die bei Verletzung des WAL-Prinzips oder der Commit-Regel auftreten können.

**Aufgabe 4-1 a)**

**Einbringstrategie:**

**Strategie Pufferverwaltung:**

	<i>Aktion</i>	<i>Load/Save</i>	<i>Puffer</i>	<i>DB-Inhalt</i>
0				
1	$r_1(A)$			
2	$r_1(B)$			
3	$r_2(C)$			
4	$w_2(C)$			
5	$w_1(B)$			
6	$r_1(E)$			
7	$w_1(E)$			
8	$EOT_2$			
9	$w_1(A)$			
10	$EOT_1$			
11	$r_3(D)$			
12	$w_3(D)$			
13	$EOT_3$			

**Aufgabe 4-1 b)**

**Einbringstrategie:**

**Strategie Pufferverwaltung:**

	<i>Aktion</i>	<i>Load/Save</i>	<i>Puffer</i>	<i>DB-Inhalt</i>
1-5	...			
6	$r_1(E)$			
7	$w_1(E)$			
8	$EOT_2$			
9	$w_1(A)$			
10	$EOT_1$			
11	$r_3(D)$			
12	$w_3(D)$			
13	$EOT_3$			