

## Wiederholung (4): Abhängigkeiten des Logging von anderen Systemkomponenten

### (a) Sperrverwaltung

Loggranulat  $\leq$  Sperrgranulat, sonst Lost Updates möglich!

Z.B. "Satzsperrungen bei Seitenlogging"  $\Rightarrow$  UNDO/REDO einer Änderung kann parallel durchgeführte Änderungen derselben Seite überschreiben

### (b) Einbringstrategien

- **Direktes Einbringen** (*Update-in-Place*)

- Geänderte Werte werden auf ihre ursprüngliche Position zurück geschrieben, d.h. Schreiben ist gleichzeitig auch Einbringen in DB
- Unterbrechungsfreiheit kann nicht garantiert werden: *non-atomic*  $\Rightarrow$  UNDO nötig

- **Indirektes Einbringen**

- Geänderte Seiten werden in einem separaten Block auf Platte geschrieben, Einbringung in die DB erfolgt zu einem späteren Zeitpunkt
- alte Version bleibt erhalten  $\Rightarrow$  UNDO nicht nötig
- Einbringen in DB erfolgt atomar (Umschalten der Seitentabellen): *atomic*

### (c) Pufferverwaltung

- Verdrängungsstrategien

- **Steal**: Seiten dürfen jederzeit ersetzt und in die DB eingebracht werden  $\Rightarrow$  DB kann unbestätigte Änderungen enthalten  $\Rightarrow$  UNDO nötig
- **No-Steal**: Seiten dürfen **frühestens** bei EOT aus dem Puffer entfernt und in DB eingebracht werden  $\Rightarrow$  DB enthält nur Änderungen erfolgreicher TAs  $\Rightarrow$  UNDO nicht nötig

- Ausschreibestrategien (EOT-Behandlung)

- **Force**: geänderte Seiten werden **spätestens** bei EOT in DB geschrieben  $\Rightarrow$  Änderungen aller erfolgreichen TAs sind in DB enthalten  $\Rightarrow$  REDO nicht nötig
- **No-Force**: Änderungen können auch erst nach EOT in DB geschrieben werden  $\Rightarrow$  Änderungen erfolgreicher TAs sind evtl. nicht in DB enthalten  $\Rightarrow$  REDO nötig

**Praxis:** Steal / No-Force  $\rightarrow$  erfordert zwar UNDO und REDO, zeigt aber unter Normalbetrieb beste Leistungsmerkmale.