

Lernzettel

Transaktionssysteme und ACID-Eigenschaften:
Konsistenz, Isolation, Fehlertoleranz

Universität:	Technische Universität Berlin
Kurs/Modul:	Informationssysteme und Datenanalyse
Erstellungsdatum:	September 19, 2025



Zielorientierte Lerninhalte, kostenlos!
Entdecke zugeschnittene Materialien für deine Kurse:

<https://study.AllWeCanLearn.com>

Informationssysteme und Datenanalyse

Lernzettel: Transaktionssysteme und ACID-Eigenschaften: Konsistenz, Isolation, Fehlertoleranz

(1) Grundbegriffe und Ziel.

Transaktionssysteme wickeln mehrere Operationen in einer Transaktion ab. Eine Transaktion hat die Eigenschaft, dass sie entweder vollständig abgeschlossen wird oder keinerlei Änderung sichtbar macht. Formal:

$$\text{Transaktion } T : \text{Start} \rightarrow \{\text{Commit}, \text{Abort}\}$$

(2) ACID-Eigenschaften im Überblick.

Die vier grundlegenden Eigenschaften einer Transaktion sind:

- **Atomicität.** Die Transaktion wird als unteilbare Einheit ausgeführt. Entweder alle Operationen werden erfolgreich ausgeführt oder keine davon.
- **Konsistenz.** Vor und nach einer Transaktion gilt das Integritätsprinzip des Systems: alle Integritäts-Constraints bleiben wahr.
- **Isolation.** Gleichzeitige Transaktionen beeinflussen sich so wenig wie möglich gegenseitig; das resultierende Zustandsspektrum entspricht einer seriellen Ausführung.
- **Dauerhaftigkeit (Durability) / Fehlertoleranz.** Nach einem erfolgreichen Commit bleiben die Auswirkungen auch bei Systemausfällen erhalten.

(3) Formaler Blick auf die ACID-Eigenschaften.

$$\text{ACID} = \{A, C, I, D\}$$

Atomicität, Konsistenz, Isolation und Dauerhaftigkeit arbeiten zusammen, um Zuverlässigkeit von Transaktionen zu gewährleisten.

(4) Atomicität und Konsistenz genauer.

- **Atomicität.** Eine Transaktion gilt als abgeschlossen, wenn alle sich darauf beziehenden Operationen erfolgreich gelaufen sind; sonst bleibt das System unverändert.
- **Konsistenz.** Transaktionen dürfen keine Integritätsbedingungen verletzen. Typische Constraints sind Primärschlüssel, Fremdschlüssel, Domain-Constraints.

(5) Isolationsebenen (Isolation Levels).

Isolation bestimmt, wie stark Transaktionen voneinander abgeschirmt sind. Übliche Ebenen in relationalen Systemen:

- **Read Uncommitted.** Es sind auch uncommittete Änderungen sichtbar (Dirty Reads).
- **Read Committed.** Nur committete Werte sichtbar; Dirty Reads vermieden.
- **Repeatable Read.** Gleiche Abfragewerte bei erneutem Lesen innerhalb derselben Transaktion garantiert; Non-Repeatable Reads vermieden, Phantom Reads möglich.

- **Serializable.** Simuliert sequentielle Ausführung; höchste Isolation.

(6) Fehlertoleranz und Durability.

Nach einem Commit müssen die Änderungen persistent bleiben, auch bei Absturz. Typische Mechanismen:

- **Write-Ahead Logging (WAL).** Vor jeder Datenänderung wird ein Log-Eintrag geschrieben.
- **Checkpoints.** Regelmäßige Sicherung des aktuellen Zustands, um Recovery zu beschleunigen.
- **Crash-Recovery.** Nach einem Absturz werden Transaktionen anhand Logs rekonstruiert (Redo/Undo).

(7) Verhältnis von Transaktions- zu Data-Warehousing-Systemen.

Transaktionssysteme (OLTP) fokussieren auf schnelle, konsistente Updates kürzerer Transaktionen. Data Warehousing (OLAP) zielt auf analytische Abfragen über große Datenmengen; hier kommen oft azyklische Ladeprozesse und zeitversetzte Konsistenzmodelle zum Einsatz. Wichtige Brücken-Themen sind *Data Staging*, *ETL* und *Data Mutures* mit zeitlichen Konsistenzanforderungen.

(8) Beispiel: Guthaben-Transfer zwischen zwei Konten.

Ziel: 100 Guthaben von Konto A auf Konto B übertragen. Die Transaktion muss atomic sein.

```
BEGIN;  
UPDATE Konto SET Saldo = Saldo - 100 WHERE KontoID = 'A';  
UPDATE Konto SET Saldo = Saldo + 100 WHERE KontoID = 'B';  
COMMIT;
```

(9) Häufige Anomalien, und wie Isolation sie verhindert.

- Dirty Read: Lesen uncommiteter Werte.
- Non-Repeatable Read: Gleiche Abfrage liefert bei erneutem Lesen unterschiedliche Ergebnisse.
- Phantom Read: Neue Zeilen erscheinen bei erneuter Abfrage.

Serializable schützt davor, gefolgt von Repeatable Read und weiterführenden Strategien in der Implementierung.

(10) Kurzfazit.

Transaktionssysteme ermöglichen konsistente, isolierte und fehlertolerante Abläufe über mehrere Operationen hinweg. ACID-Eigenschaften definieren die Mindestanforderungen, um Zuverlässigkeit in betrieblichen Informationssystemen sicherzustellen.