12 Transaktionen

- Problematik: Logisch zusammengehörige Änderungen von Daten

- Beispiele:
  - Buchungen: Eine Buchung wird unter Angabe von Datum, Betrag, Konto und Gegenkonto gemacht. Es muss bei beiden Konten ein Buchungssatz aufgenommen werden.
  - Personalverwaltung: Die Bezeichnung einer Abteilung wird geändert. In allen Personalsätzen, in denen diese Abteilung vorkommt, muss die Abteilungsbezeichnung geändert werden.
  - Auftragsverwaltung: Eine Auftragsposition wird storniert. Der Auftragspositionssatz muss gelöscht und die Menge der reservierten Einheiten im Artikelsatz vermindert werden.

- Hinweis: Transaktionen sind auch notwendig, wenn keine Redundanzen vorliegen, z.B. Löschen eines Auftrags, der aus einem Kopfsatz und mehreren Positionssätzen besteht.

- Werden nicht alle Änderungen durchgeführt, sind die Daten in einem logisch inkonsistenten Zustand.

- Transaktion (Transaction, Commit Unit, LUW - Logical Unit of Work, Unit of Recovery):
  - Arbeitseinheit (Folge von Anweisungen), die (bezüglich der Auswirkungen auf Dateien) entweder ganz durchgeführt werden muss oder überhaupt nicht durchgeführt werden darf.
  - Prinzip: Alles oder Nichts.
  - Programmabschnitt, in dem die Daten von einem konsistenten Zustand wieder in einen konsistenten Zustand übergeführt werden.
  - Transaktion muss 'atomar' (unteilbar) sein.

- Transaktion wird ordnungsgemäß beendet: Commit
- Transaktion wird nicht ordnungsgemäß beendet: Rollback, Abort, Reset, Backout

- Im Anwendungsprogramm werden Beginn und Ende von Transaktionen
  - entweder explizit angegeben
    (z.B. BEGIN / COMMIT TRANSACTION, COMMIT WORK für Transaktionsende)
  - oder implizit angenommen (z.B. Prozeduren, bestimmte Blöcke)

- Länge von Transaktionen:
  - so kurz als möglich:
    - geringer Systemoverhead
    - wenige Funktionen sind im Fehlerfall zu wiederholen
    - kurze Sperreinheiten für andere Anwender (siehe 'Synchronisation paralleler Transaktionen')
  - so lang als notwendig:
    - konsistenter Datenzustand muss gewährleistet sein

- Üblicherweise kann immer höchstens eine Transaktion (pro Benutzer / Task) aktiv sein, Transaktionen können nicht geschachtelt sein (flache Transaktionen).
  Es gibt auch Konzepte mit geschachtelten Transaktionen, sogenannten Subtransaktionen. Damit ist flexiblere Recovery möglich, denn es kann entweder die ganze Transaktion, oder nur eine bestimmte Subtransaktion zurückgenommen werden.

- Synchronisationspunkt (Sync-Point): Grenze zwischen zwei aufeinanderfolgenden Transaktionen. Alle Datenänderungen müssen permanent gemacht (fixiert) werden, gesperrte Daten freigegeben werden (siehe 'Synchronisation paralleler Transaktionen')

- Ein Programm(system), der Transaktionsmonitor (Transaktionsmanager, Transaction Processing Monitor, TP-Monitor), muss dafür sorgen, dass Datenveränderungen von Transaktionen, die nicht ordnungsgemäß beendet wurden, wieder zurückgenommen werden.

- Der Transaktionsmonitor ist entweder eine Komponente des Datenbanksystems oder ein eigenes Programm (z.B. UTM im BS2000 / SIEMENS, CICS / IBM, MS Transaction Server / Microsoft, Tuxedo / BEA)

- Transaktionen müssen nach dem ACID-Prinzip ablaufen
  - Atomicity (Atomarität, Unteilbarkeit):
    - Veränderung der Daten erfolgt vollständig oder gar nicht (alles oder nichts)
    - wird ein Einzelschritt nicht erfolgreich beendet, so wird der gesamte Vorgang zurückgesetzt
  - Consistency (Konsistenz):
    - Daten sind immer in einem logisch konsistenten Zustand
    - Verarbeitung erfolgt von einem konsistenten Zustand zum nächsten
  - Isolation:
    - Datenänderungen innerhalb der Transaktion sind nur ihr selbst bekannt
    - Andere Transaktionen haben gleichzeitig keinen oder nur beschränkten Zugriff auf diese Daten
    - Transaktion darf keine andere, gleichzeitig mit ihr laufende, beeinflussen
    - Problembereich 'Synchronisation paralleler Transaktionen' (Concurrency)
  - Durability (Persistenz, Dauerhaftigkeit):
    - Nach Transaktionsabschluss müssen die Datenänderungen dauerhaft (gegen Fehler jeglicher Art) bestehen bleiben
    - Problembereich 'Recovery'