

# ACID-Prinzipien für Transaktionen

Atomicity – Consistency – Isolation – Durability

# Was ist eine Transaktion?

- Eine Transaktion ist eine **Einheit von Operationen**, die vollständig oder gar nicht ausgeführt wird.
- Beispiel: Banküberweisung
  - Konto A – 100€ wird abgezogen
  - Konto B – 100€ Betrag wird gutgeschrieben
- Wichtig: Entweder **beides geschieht** oder **nichts davon**.

# Bedeutung von ACID

- ACID ist ein Akronym für die **vier Eigenschaften einer zuverlässigen Transaktion**:
- **Atomicity** (Atomarität)
- **Consistency** (Konsistenz)
- **Isolation** (Isolation)
- **Durability** (Dauerhaftigkeit)

# Atomicity (Atomarität)

- Transaktion ist **untrennbar**: Alles oder nichts.
- Wenn ein Teil fehlschlägt, wird die gesamte Transaktion rückgängig gemacht.
- Beispiel:
  - Fehler bei der Gutschrift → Abbuchung wird zurückgenommen

# Consistency (Konsistenz)

- Datenbank bleibt **in einem gültigen Zustand** vor und nach der Transaktion.
- Alle definierten Regeln und Einschränkungen bleiben gültig.
- Beispiel:
  - Kontostand darf nicht negativ sein → Transaktion wird abgelehnt, falls Regel verletzt würde.

# Isolation (Isolation)

- Gleichzeitige Transaktionen **beeinflussen sich nicht gegenseitig**.
- Jede Transaktion sieht die Daten so, als wäre sie **allein aktiv**.
- Verhindert u.a. „dirty reads“:
  - Transaktion A ändert den Kontostand auf 0 €.
  - Transaktion B liest den neuen Wert (0 €).
  - Transaktion A wird **abgebrochen** → Kontostand ist wieder 100 €.
  - → Transaktion B hat mit einem **falschen Zwischenwert** gearbeitet.
- Verhindert u.a. „Non-Repeatable Read“ (nicht wiederholbares Lesen)
  - Eine Transaktion liest einen Datensatz **zweimal**, bekommt aber **unterschiedliche Werte**, weil eine andere Transaktion **dazwischen den Wert geändert** hat.
  - Beispiel:
    - Transaktion A liest Artikelanzahl = 10.
    - Transaktion B verkauft 2 Artikel → Artikelanzahl = 8.
    - Transaktion A liest erneut → sieht jetzt 8.

# Durability (Dauerhaftigkeit)

- Sobald eine Transaktion abgeschlossen ist (COMMIT), sind die Änderungen **dauerhaft gespeichert**.
- Daten bleiben auch bei Systemabstürzen erhalten (z. B. durch Write-Ahead Logging).
- Beispiel: Gutschrift ist auch nach Serverneustart noch vorhanden

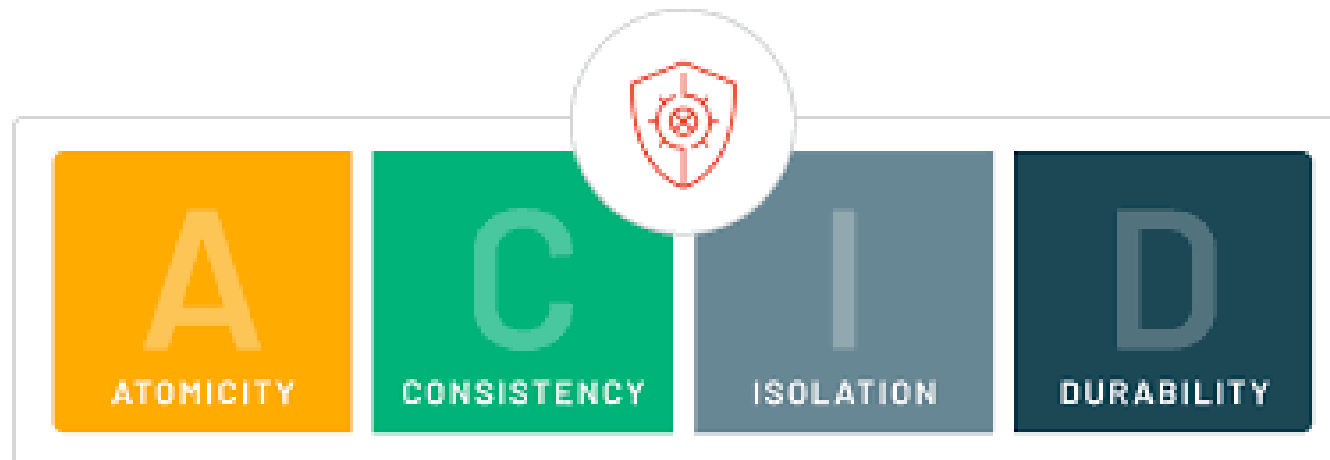
# Warum ist ACID wichtig?

- Gewährleistet **Datensicherheit** und **Integrität**
- Kritisch in:
  - Banken
  - Gesundheitswesen
  - E-Commerce
- Ohne ACID könnten Fehler zu **Inkonsistenzen, Datenverlust oder Sicherheitslücken** führen



# Zusammenfassung

- ACID = Grundlage für zuverlässige Transaktionen
- **Atomicity**: Alles oder nichts
- **Consistency**: Regeln bleiben gültig
- **Isolation**: Keine Beeinflussung zwischen Transaktionen
- **Durability**: Dauerhafte Speicherung



# Weiterführende Ressourcen

- YouTube: "ACID Explained in 5 Minutes"
- PostgreSQL Transaktionsdoku:  
<https://www.postgresql.org/docs/current/tutorial-transactions.html>