

# Datenbanken und Informationssysteme

## Transaktionen

Burkhardt Renz

Fachbereich MNI  
TH Mittelhessen

Sommersemester 2020

# Übersicht

- Motivation
- Transaktionen
  - Definition
  - ACID-Eigenschaften

## Beispiel 1

Wir wollen eine Überweisung durchführen von Konto 1 auf Konto 2.

Die Daten der Konten werden in der Tabelle Konto aufgezeichnet:

KtoNr	Saldo
1	100
2	100

```
update Konto set Saldo = Saldo - 50 where KtoNr = 1;  
update Konto set Saldo = Saldo + 50 where KtoNr = 2;
```

Was passiert, wenn der Rechner mittendrin abstürzt??

## Beispiel 2

Wir wollen eine Überweisung durchführen von Konto 1 auf Konto 2. Sie soll nur möglich sein, wenn das Konto nicht überzogen wird.

Die Daten der Konten werden in der Tabelle Konto aufgezeichnet:

KtoNr	Saldo
1	100
2	100

```
select Saldo from Konto where KtoNr = 1;  
if (Saldo > 60) {  
    update Konto set Saldo = Saldo - 60 where KtoNr = 1;  
    update Konto set Saldo = Saldo + 60 where KtoNr = 2;  
}
```

Was passiert, wenn zwischendrin eine andere Überweisung von Kto 1 stattfindet??

## Beispiel 1 – mit Transaktion

Wir wollen eine Überweisung durchführen von Konto 1 auf Konto 2.

Die Daten der Konten werden in der Tabelle Konto aufgezeichnet:

KtoNr	Saldo
1	100
2	100

```
begin transaction;  
  update Konto set Saldo = Saldo - 50 where KtoNr = 1;  
  update Konto set Saldo = Saldo + 50 where KtoNr = 2;  
commit;
```

Was passiert, wenn der Rechner mittendrin abstürzt??

## Beispiel 2 – mit Transaktion

Wir wollen eine Überweisung durchführen von Konto 1 auf Konto 2. Sie soll nur möglich sein, wenn das Konto nicht überzogen wird.

Die Daten der Konten werden in der Tabelle Konto aufgezeichnet:

KtoNr	Saldo
1	100
2	100

```
begin transaction isolation level serializable;
  select Saldo from Konto where KtoNr = 1;
  if (Saldo > 60) {
    update Konto set Saldo = Saldo - 60 where KtoNr = 1;
    update Konto set Saldo = Saldo + 60 where KtoNr = 2;
  }
commit;
```

Was passiert, wenn zwischendrin eine andere Überweisung von Kto 1 stattfindet??

# Übersicht

- Motivation
- Transaktionen
  - Definition
  - ACID-Eigenschaften

# Transaktion

## Definition

Eine **Transaktion** ist eine **logische Verarbeitungseinheit** auf einer Datenbank, die eine oder mehrere Datenbankoperationen (Einfügen, Löschen, Ändern und/oder Suchen) umfasst.

Eine Transaktion wird mit dem Befehl `commit` als gültig erklärt oder mit dem Befehl `rollback` rückgängig gemacht.



# Eigenschaften von Transaktionen

<b>A</b>	Atomarität ( <i>atomicity</i> )
<b>C</b>	Konsistenz ( <i>consistency</i> )
<b>I</b>	Isolation ( <i>isolation</i> )
<b>D</b>	Dauerhaftigkeit ( <i>durability</i> )

# Atomarität

## Definition (Atomarität)

Die Teilschritte einer Transaktion werden vom DBMS als **eine unteilbare, atomare** Einheit durchgeführt, d.h. alle oder gar keiner.

Merke: „Alles oder nichts“

# Konsistenz

## Definition (Konsistenz)

Die Datenbank hat vor und nach einer Transaktion stets einen **konsistenten Zustand**, d.h. alle Integritätsbedingungen des Datenbankschemas sind erfüllt.

Merke: „Daten bleiben konsistent“

# Isolation

## Definition (Isolation)

Eine Transaktion läuft **isoliert** gegenüber dem Einfluss anderer Transaktionen ab, so als ob sie exklusiven Zugriff auf die Daten hätte.

Eventuell wird die Transaktion vom DBMS abgebrochen, weil andernfalls ein unerlaubter Einfluss anderer Transaktionen erfolgt wäre.

Merke: „Eine Transaktion hat die Daten quasi allein“

# Dauerhaftigkeit

## Definition (Dauerhaftigkeit)

Die Ergebnisse einer bestätigten Transaktion (nach dem Commit) sind **dauerhaft gesichert**, d.h. das DBMS garantiert, dass bei einem Fehler der bestätigte Zustand wiederhergestellt werden kann.

Merke: „Nichts geht verloren“