# Das Transaktionskonzept

**Prof. Dr. Inga Saatz, FH Dortmund**

**Nov 03, 2023**

**Note:** Nach dieser Vorlesung sollten Sie:

- Das Transaktionskonzept kennen.
- Die Eigenschaften von ACID-Transaktionen anhand von Beispielen erläutern.
- Die Transaktionssteuerungsbefehle (COMMIT, ROLLBACK, AUTOCOMMIT, DEFERRED) anwenden.
- Die Methode der sperrbasierten Synchronisation von Transaktionen kennen.
- Die Rolle des Logs beim Recovery erläutern.

```
%load_ext sql
```

```
```

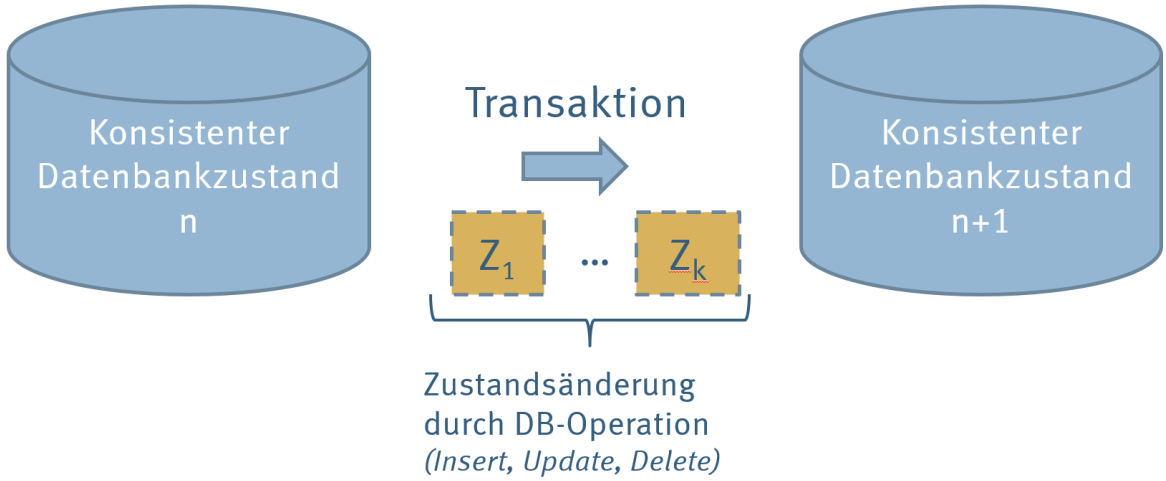
### Beispiel

1. Was passiert, wenn bei der Verarbeitung des Bestellvorgangs ein Fehler bei einer der ausgeführten Datenbankoperationen auftritt?
2. Welche Integritätsbedingungen müssen sichergestellt werden?

Eine **Transaktion** ist eine inhaltlich zusammenhängende Menge von Datenbankoperationen, die ganz oder gar nicht ausgeführt werden.

Eine **Transaktion** überführt einen konsistenten Datenbankzustand in einen wiederum konsistenten Datenbankzustand.

*Der initiale Zustand (Zustand n=0) der Datenbank ist konsistent.*

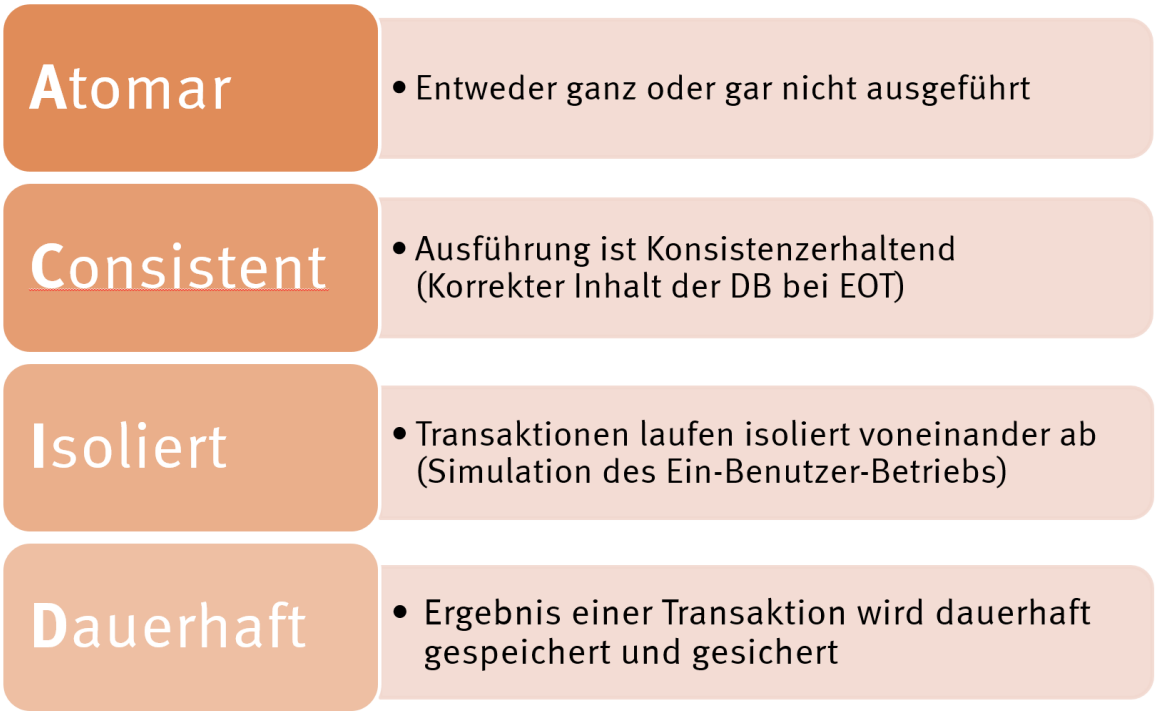



Wird nur die erste Datenbankoperation ($Z_1$) einer Transaktion ausgeführt, ergibt sich ein inkonsistenter Datenbankzustand. Erst wenn **alle** Datenbankoperationen der Transaktion ausgeführt worden sind, ist ein neuer konsistenter Datenbankzustand erreicht.

**Tip:** Video: Was ist eine Transaktion? in ILIAS

## Eigenschaften von Transaktionen

ACID ist ein Akronym und steht für die vier vier Schlüsseleigenschaften einer Transaktion, die eine Transaktion in einem relationalen Datenbankmanagementsystem soll, um einen konsistenten Datenbestand zu bewahren.

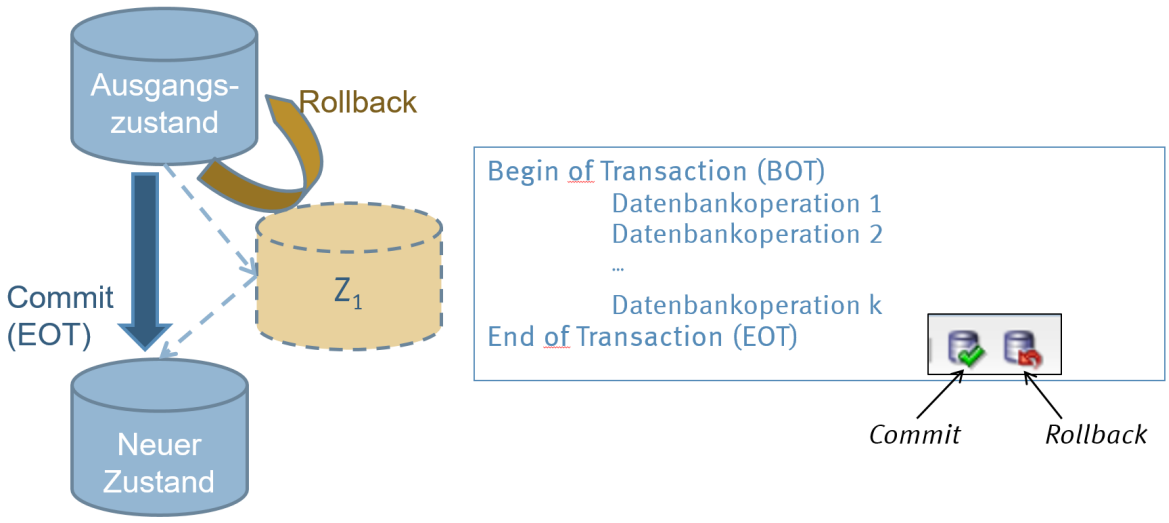



### Atomarität

**Tip:** Video in ILIAS

Eine **Transaktion** fasst Datenbankoperationen zusammen.

- *Begin of Transaction*: Markiert den Beginn der Menge der zusammenhängenden Datenbankoperationen
- *End of Transaction* : Markiert das Ende der Menge der zusammenhängenden Datenbankoperationen
  - `COMMIT`: Erfolgreiches Beenden der Transaktion mit Erreichen eines **neuen** konsistenten Zustands
  - `ROLLBACK`: Abbruch der Transaktion und Rückkehr zum **alten** konsistenten Zustand.




**Note:** Erläuterung Zusammengehörende Datenbankoperationen werden durch eine Transaktion zusammengefasst. Die zusammengefassten Datenbankoperationen können die nur zusammen oder gar nicht ausgeführt werden. Wenn eine Datenbankoperation der Transaktion nicht ausgeführt werden kann oder ein Fehler auftritt, dann erfolgt eine Rückkehr (=Rollback)zu dem Ausgangszustand. Wird am Transaktionsende ein korrekten Datenbankzustand erreicht, d.h. sind die Integritätsbedingungen erfüllt, so wird ein **Commit** ausgeführt. Bei einem Commit werden die Änderungen der Transaktion gesichert.
Wird kein korrekter Datenbankzustand erreicht oder ist ein anderer Fehlerfall oder einen Transaktionsabbruch aufgetreten, dann wird ein **Rollback** ausgeführt. Bei einem Rollback erfolgt die Rücksetzung der DB in einen vorherigen konsistenten Zustand.

### SET AUTO ON

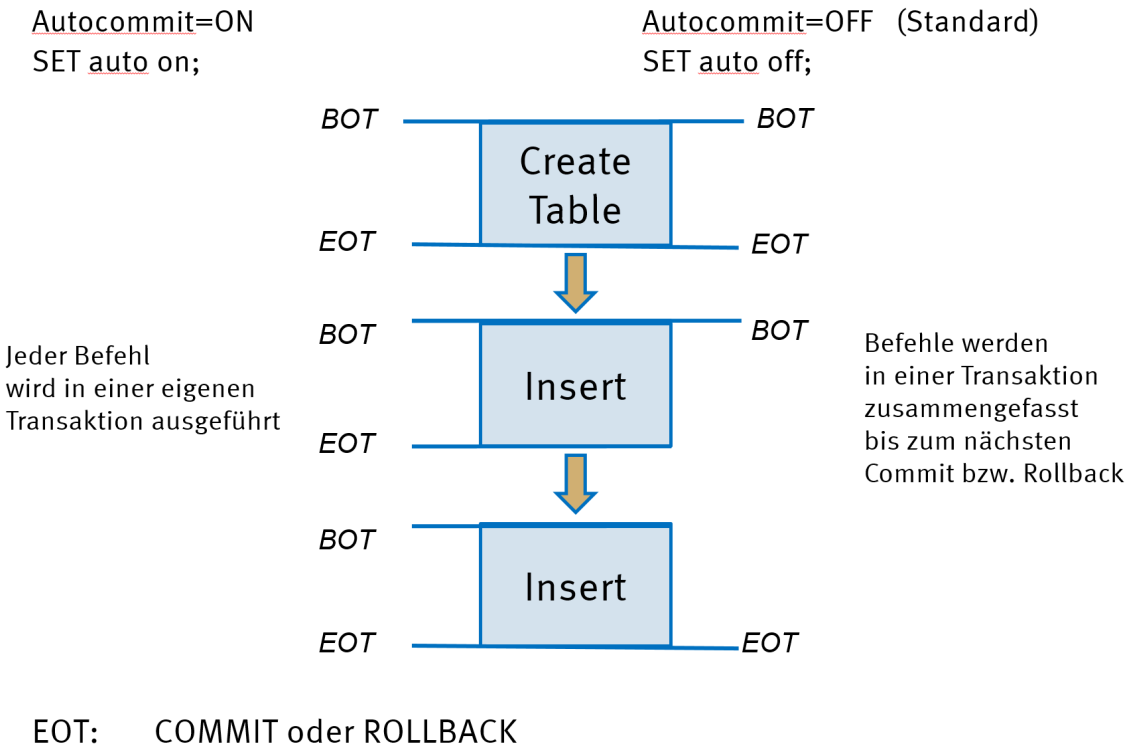
Was beinhaltet die Tabelle `test` nach Ausführung des folgenden Skripts im **Oracle SQLDeveloper**?

```
-- SET AUTO ON;
CREATE TABLE test(
id                INTEGER PRIMARY KEY,
nachname VARCHAR(30)
);
INSERT INTO test VALUES (1,'hallo');
INSERT INTO test VALUES (2,'welt');
SELECT * FROM test;
rollback;
SELECT * FROM test;
```

### SET AUTO OFF

Was ändert sich bei Verwendung von `AUTO OFF`?

```
SET AUTO OFF;
CREATE TABLE test(
id                INTEGER PRIMARY KEY,
nachname VARCHAR(30)
);
INSERT INTO test VALUES (1,'hallo');
INSERT INTO test VALUES (2,'welt');
SELECT * FROM test;
rollback;
SELECT * FROM test;
```




**Warning:** Autocommit bei iPython-sql Autocommit kann bei iPython-sql über den config-Befehl ausgeschaltet werden.

```
%config SqlMagic.autocommit=False
```

Allerdings werden die Einfügeoperationen **nicht** vollständig ausgeführt. Testen Sie dies einmal anhand des obigen Beispiels. Achten Sie darauf, dass dabei die Semikolons nicht zum jeweiligen Statement gehören.

## Konsistenz

**Tip:** Video in ILIAS

Anforderung der Datenintegrität Die Datenbank soll zu jedem Zeitpunkt die in der Realität geltenden Zusammenhänge und Regeln widerspiegeln.

*Lösungsprinzip*:

- Kontrolle der Datenintegrität durch das DBMS statt durch die Anwendungsprogramme

*Vorteile*:

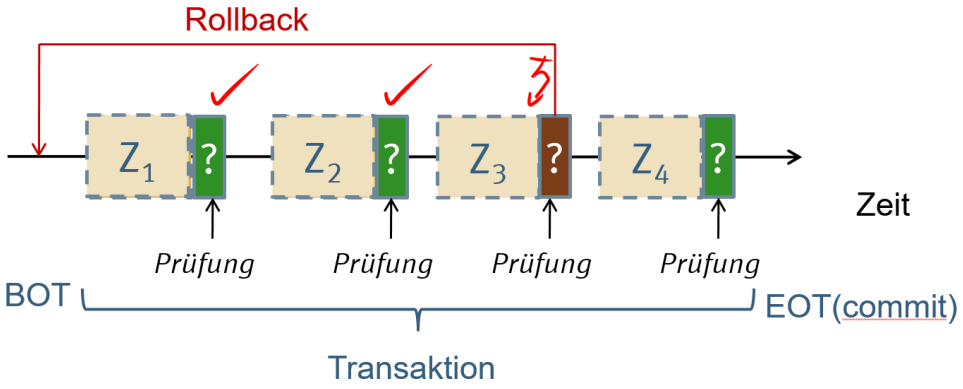
- Effektivere und sichere Kontrolle
- Einfachere Anwendungsprogrammierung
- Schlankere Clients

*Methoden zur Sicherstellung der Datenintegrität*

- **statische** Integritätsbedingungen
  * Referentielle Integrität (Tabellendeklaration)
  * Constraints (Tabellendeklaration)
  * Views (deklarativ bei Definition)
- **dynamische** Integritätsbedingungen
  * **Transaktionen** (Synchronisation der Ausführung von Datenbankoperationen)
  * Trigger (event-gesteuerte Datenbankprogramme)

**Zeitpunkt der Überprüfung**

- Standard:
  Jede DML-Anweisung wird sofort nach der Anweisung geprüft



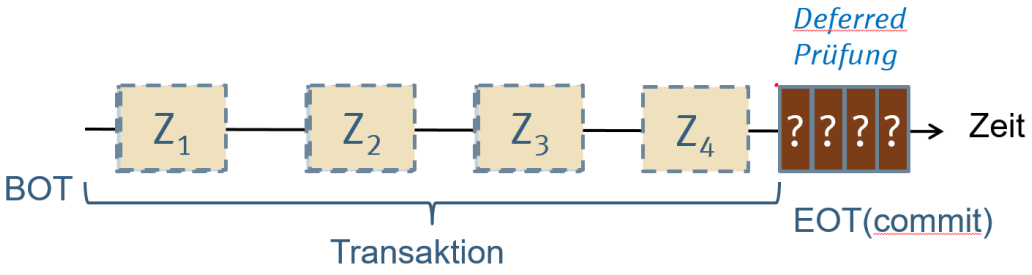

- Wirkung im Fehlerfall:
  - Rücksetzen der verletzten Anweisung
  - Rollback der gesamten Transaktion

### Wo ist der Fehler?

Weshalb wird das Skript nicht vollständig ausgeführt?

```
CREATE TABLE Pferde(
Id INT PRIMARY KEY,
Bezeichnung Varchar(15),
Stute INT,
FOREIGN KEY (Stute) REFERENCES Pferde(Id)
);
INSERT INTO Pferde VALUES (1, 'Marta', NULL);
INSERT INTO Pferde VALUES (2, 'Wildfang', 1);
UPDATE Pferde SET Id=3 WHERE Id=1;
SELECT * from Pferde;
```

Bei Oracle kann die Überprüfung der statischen Integritätsbedingungen an das Ende der Transaktion (deferred) verschoben werden. Im Fehlerfall wird die gesamte Transaktion zurückgesetzt (Rollback)




Anwendung auf das Beispiel:
**1. Schritt**: Verschiebung der Fremdschlüsselprüfung an das Ende der Transaktion

```
CREATE TABLE Pferde(
Id INT PRIMARY KEY,
Bezeichnung Varchar(15),
Stute INT,
FOREIGN KEY (Stute) REFERENCES Pferde(Id)
        INITIALLY DEFERRED DEFERRABLE  -- Verschieben der Integritätsprüfung
);
```

### 2. Schritt

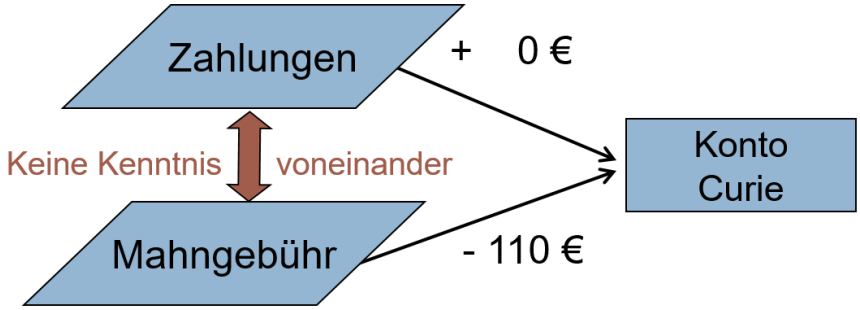
Wie sind die DML-Operationen anzupassen?

```
INSERT INTO Pferde VALUES (1, 'Marta', NULL);
INSERT INTO Pferde VALUES (2, 'Wildfang', 1);
UPDATE Pferde SET Id=3 WHERE Id=1;
```

## Isoliertheit

Video in ILIAS

Es zu einer Anomalie (=Fehlverhalten), wenn die beiden Programme *Zahlungen* und *Mahngebühr* unabhängig voneinander auf den Kontostand der Kundin *Curie* ändernd zugreifen.



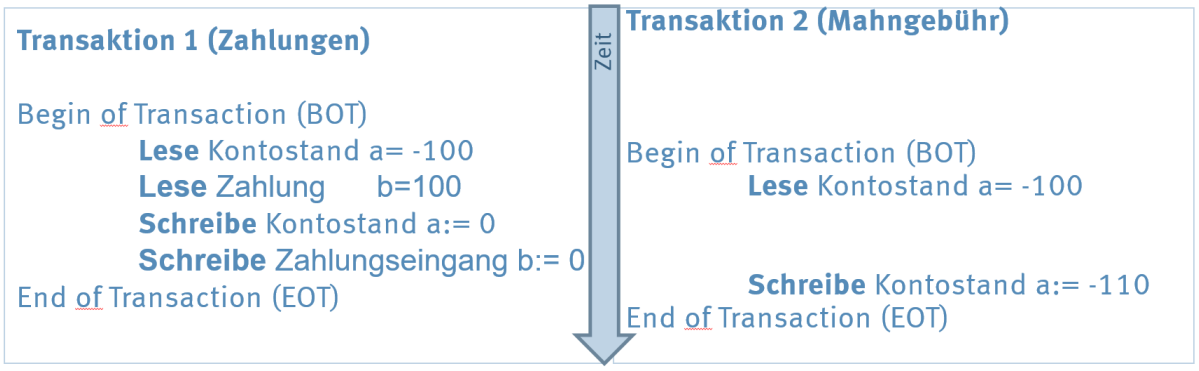

| **Programm Zahlungen:** | |
|---|---|
| Alter Kontostand: | -100 € |
| Zahlungseingang | 100 € |
| Neuer Kontostand: | 0 € |

| **Programm Mahngebühr** | |
|---|---|
| Alter Kontostand: | -100 € |
| Mahngebühr: | - 10 € |
| Neuer Kontostand: | -110 € |

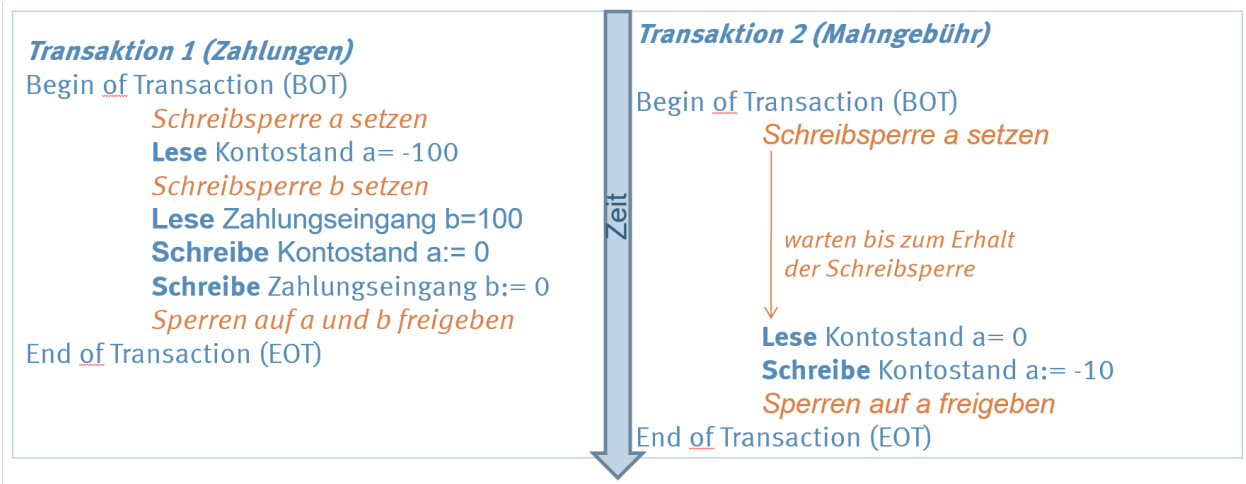
### Aufgabe

Wovon hängt der resultierende Kontostand nach der Ausführung der beiden Programme ab?

Diese Anomalie wird als **Lost Update** bezeichnet, da die Änderung eines der beiden Programms überschrieben wird.



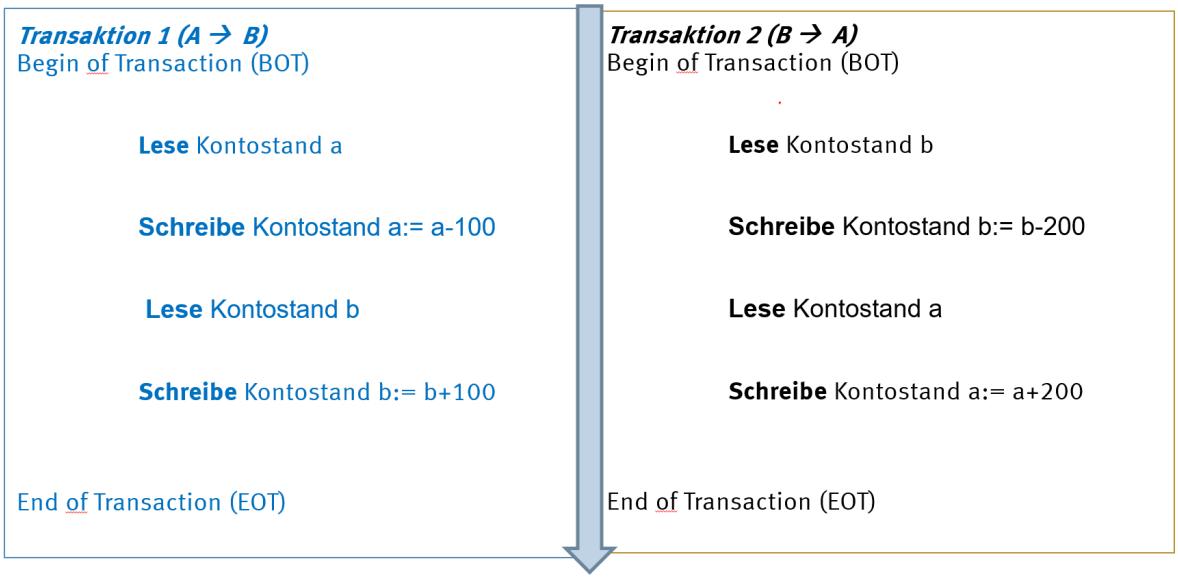

Anforderung: Ablaufintegrität im Mehrbenutzerbetrieb Es muss die Korrektheit von Daten bei gleichzeitigem ändernden Zugriff durch mehrerer Benutzer auf eine Dateneinheit sichergestellt werden.



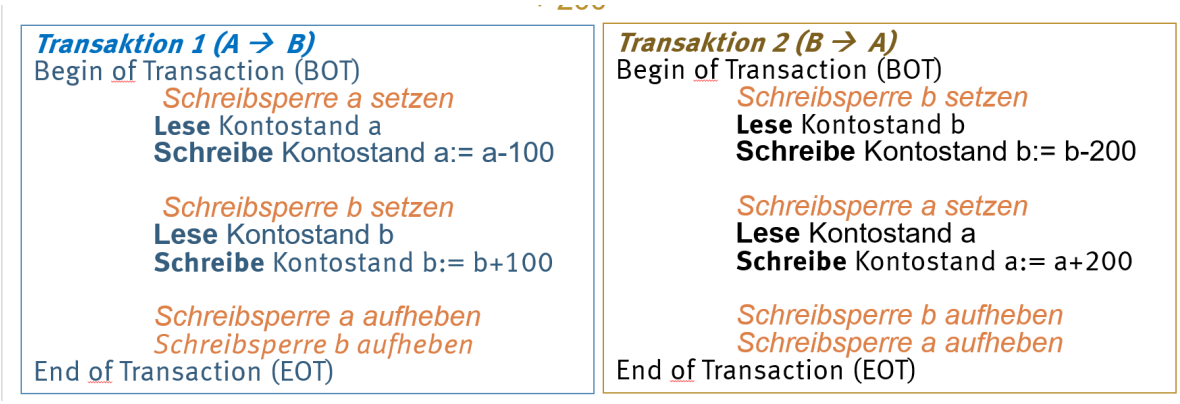

**Note:** Erläuterung Die Synchronisation von Transaktion durch ein Sperrverfahren ist eine Möglichkeit die Ablaufintegrität sicherzustellen.
Wenn eine Transaktion ein Datenobjekt zugreift, dann wird dieses gesperrt. Eine andere Transaktion muss warten und kann erst dann auf das Datenobjekt zugreifen, wenn die erste Transaktion die Sperre wieder freigegeben hat. Hierdurch wird ein Ein-Benutzer-Betriebs auf dem Datenobjekt simuliert, wodurch die Transaktionen unabhängig (isoliert) voneinander ablaufen.

### Ohne Sperren

Welche Sperren sind zu setzen und freizugeben?
Transaktion 1: Konto A überweist 100€ auf Konto B.
Transaktion 2: Konto B überweist 200€ auf Konto A.

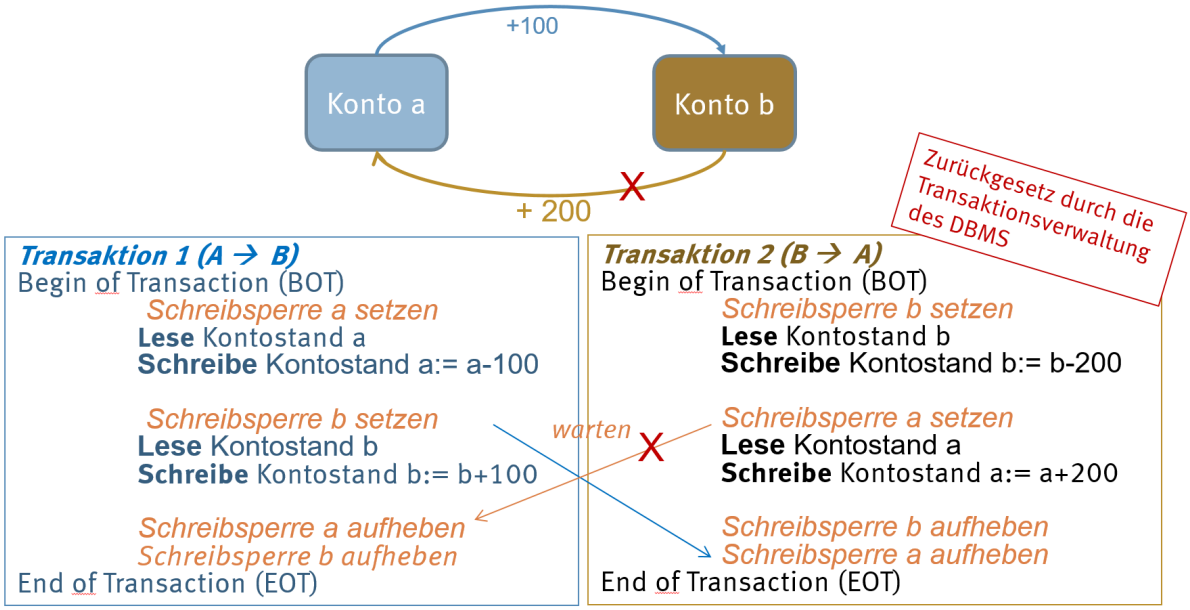


### Mit Sperren




Welches Problem tritt hier auf?

### Ein Nachteil



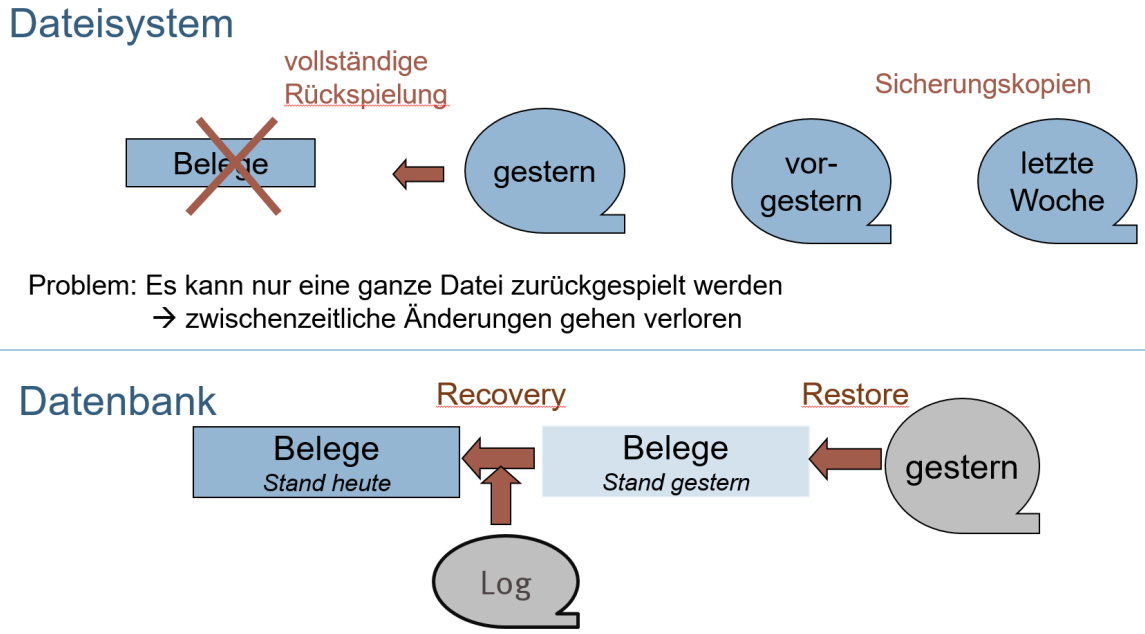

**Nachteil**
Ein Dead-Lock tritt auf, wenn zwei oder mehrere Transaktionen gegenseitig auf die Freigabe von Sperren warten. Eine solche Situation wird durch manche DBMS aufgelöst, in dem einzelne Transaktionen zurückgesetzt (Rollback) werden, so dass deren Sperren freigegeben werden.

Einige Nachteile von Sperrverfahren

- Dead Locks werden nicht immer durch das DBMS erkannt.
- Performanceeinbußen durch häufig auftretende Dead Locks.
- Lange laufende (Lese-)Transaktionen blockieren viele kurze (Schreib-)Transaktionen.
- Sperrverfahren sind nicht geeignet für mobile Datenbankanwendungen. (Weshalb?)

## Dauerhaftigkeit

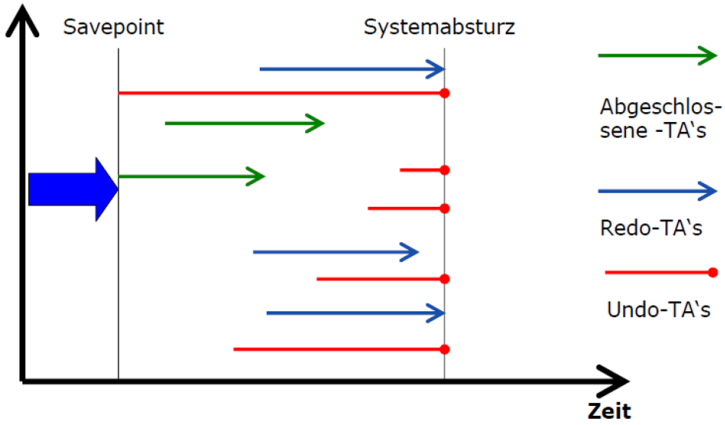
Video in ILIAS




Begriffe

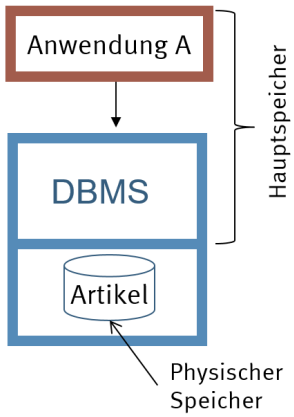
- Datensicherung : Sicherung der gesamten Datenbank (Backup)
- Restore: Wiedereinspielung eines Backups
- Recovery: Wiederherstellung eines korrekten Datenbank-Zustandes beispielsweise nach Transaktionsfehlern, Plattenfehlern und Systemfehlern.

**Note:** Situation beim Recovery




| Transaktionsart | Recovery-Maßnahme |
|---|---|
| Abgeschlossen → | Nicht erforderlich |
| Redo → | Erneutes Einspielen der Änderungen (Redo) |
| Undo — | Zurücksetzung der Transaktionen (Undo) in umgekehrter Reihenfolge |

- Abgeschlossene Transaktionen →
  - Die Transaktion ist abgeschlossen **und**
  - die erfolgten Änderungen sind physikalisch gespeichert
  - Recovery: keine Fehlerbehandlung erforderlich




- Redo-Transaktionen →
  - Die Transaktion ist abgeschlossen **und**
  - die erfolgten Änderungen sind **nur** im (flüchtigen) Hauptspeicher durchgeführt worden, so dass
  - die erfolgten Änderungen noch **nicht** physikalisch gespeichert sind
    - Recovery: Wiederholung (Redo) der Änderungen

- Undo-Transaktionen —
  - Die Transaktion ist nicht abgeschlossen worden.
  - Erfolgte Änderungen können bereits im (flüchtigen) Hauptspeicher als auch physikalisch umgesetzt worden sein.
    - Recovery: Rückgängigmachen (Undo) der Änderungen → Rollback

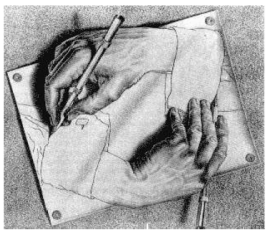
## Selbsttest

1. Was versteht man unter einer Transaktion?
2. Wodurch wird die Ablaufintegrität sichergestellt?
3. Wodurch wird der Mehrbenutzerbetrieb ermöglicht?
4. Wie unterscheiden sich Restore und Recovery?
5. Wie werden Transaktionen in Java-Programmen verwendet?

### Selbsttestaufgabe 1

Testen Sie, ob bei einer direkten Rekursion in Oracle eine Verschiebung der Prüfung der Fremdschlüsselbedingungen erforderlich ist oder nicht.

Direkte Rekursion



M.C. Escher, zeichnede Hände