

# Modul Datenbanken

## Vorlesung 10

### Transaktionen

IFI Wintersemester 2016/17

by Renzo Kottmann



This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/).

# Transaktion

Eine Transaktion ist eine Menge von Operationen die atomic, consistent, isoliert und dauerhaft sind (ACID).

# ACID

- Alle Änderungen mit  
Chirurgischer Integrität in einem  
Isoliertem Vorgang zusammengefasst und mit  
Dauerhaftem Ergebnis.

Die wichtigsten Eigenschaften im Einzelnen:

- Atomic

Eine Transaktion gruppiert mehrere Anweisungen in eine einzige atomare Operation in der "Alles oder Nichts" stattfindet.

- Consistent

Eine Transaktion bringt eine Datenbank von einem konsistenten Zustand in den nächsten.

- Isolated

Transaktion ist unabhängig, d.h. sie wird nicht durch konkurrierende Transaktionen beeinflusst.

- Durable

Bei der erfolgreichen Beendigung einer Transaktion sind alle Änderungen dauerhaft gespeichert.

# Konsistenz bei sehr vielen gleichzeitigen Änderungen

- Transaktionen garantieren, dass trotz gleichzeitiger Änderungen von vielen verschiedenen Benutzern, alle Änderungen konsistent sind.
- Tatsächlich ist jede einzelne SQL-Anweisung bei den meisten Datenbank implizit eine einzelne Transaktion, d.h. eine Gruppe von Anweisungen mit nur einer Anweisung

# Transaktion mit mehreren Anweisungen

[PostgreSQL Documentation fuer syntaktische Details und fortgeschrittene Eigenschaften](#)

Ausgangslage: Datenbank ist in einem super gutem Zustand: Z1

```
-- Transaktion wird angefangen:  
BEGIN;  
SQL Anweisung 1;  
SQL Anweisung 2;  
...  
SQL Anweisung N;  
-- Bleibt offen bis:  
COMMIT;
```

Wenn Transaktion **gut laeuft**, d.h. alle Anweisungen korrekt sind:

Datenbank ist in einem super gutem Zustand: Z2

Wenn Transaktion **fehlschlaegt**, d.h. nur eine der Anweisungen zum Fehler fuehrt:

Datenbank wird in den vorherigen Zustand zurueck gesetzt (rollback): Z1

[PostgreSQL Documentation fuer syntaktische Details und fortgeschrittene](#)

# Gewollter Transaktionsabbruch

```
-- Transaktion wird angefangen:  
BEGIN;  
SQL Anweisung 1;  
SQL Anweisung 2;  
...  
SQL Anweisung N;  
-- Bleibt offen bis:  
ROLLBACK;
```

Wenn Transaktion **fehlschlaegt** oder **erfolgreich** ist:

Datenbank wird in den vorherigen Zustand zurueck gesetzt (rollback): Z1

# DDL Entwicklung

- PostgreSQL ist einer der wenigen DBMS, die auch DDL Anweisungen in Transaktionen ausführen kann

```
-- Transaktion wird angefangen:  
BEGIN;  
CREATE TABLE test (id integer PRIMARY KEY);  
INSERT INTO test VALUES (1),(2),(3);  
SELECT * from test;  
UPDATE test SET id = id + 3;  
SELECT * from test;  
ROLLBACK;
```

- Sehr nuetzlich fuer iterative, evolutionaere Datenbankentwicklung



# Referenzen:

- M. Unterstein and G. Matthiessen, Relationale Datenbanken und SQL in Theorie und Praxis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.