

# 8 Transaktionen, Integrität und Trigger

8.1 Grundbegriffe

8.2 Transaktionsbegriff

8.3 Transaktionen in SQL

8.4 Integritätsbedingungen in SQL

8.5 Trigger



---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Teil VIII  
Transaktionen, Integrität und Trigger

Transaktionen, Integrität und Trigger

- 1 Grundbegriffe
- 2 Transaktionsbegriff
- 3 Transaktionen in SQL
- 4 Integritätsbedingungen in SQL
- 5 Trigger

Sattler / Saake      Datenbanksysteme      Letzte Änderung: 11.10.2011      8-1

Transaktionen, Integrität und Trigger      Grundbegriffe

### Integrität

- **Integritätsbedingung** (engl. *integrity constraint* oder *assertion*):  
Bedingung für die „Zulässigkeit“ oder „Korrektheit“
- in Bezug auf Datenbanken:
  - ▶ (einzelne) Datenbankzustände,
  - ▶ Zustandsübergänge vom alten in den neuen Datenbankzustand,
  - ▶ langfristige Datenbankentwicklungen

Sattler / Saake      Datenbanksysteme      Letzte Änderung: 11.10.2011      8-2

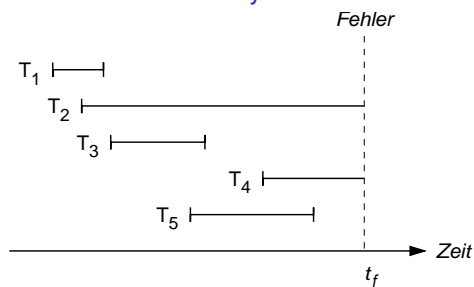




### Transaktion: Integritätsverletzung

- Beispiel:
  - ▶ Übertragung eines Betrages  $B$  von einem Haushaltsposten  $K1$  auf einen anderen Posten  $K2$
  - ▶ Bedingung: Summe der Kontostände der Haushaltsposten bleibt konstant
- vereinfachte Notation
 
$$\text{Transfer} = \langle K1:=K1-B; K2:=K2+B \rangle;$$
- Realisierung in SQL: als Sequenz zweier elementarer Änderungen  $\rightsquigarrow$  Bedingung ist zwischen den einzelnen Änderungsschritten nicht unbedingt erfüllt!

### Transaktion: Verhalten bei Systemabsturz



### Transaktion: Verhalten bei Systemabsturz /2

- Folgen:
  - ▶ Inhalt des flüchtigen Speichers zum Zeitpunkt  $t_f$  ist unbrauchbar  $\rightarrow$  Transaktionen in unterschiedlicher Weise davon betroffen
- Transaktionszustände:
  - ▶ zum Fehlerzeitpunkt noch aktive Transaktionen ( $T_2$  und  $T_4$ )
  - ▶ bereits vor dem Fehlerzeitpunkt beendete Transaktionen ( $T_1$ ,  $T_3$  und  $T_5$ )



















## Integritätssicherung durch Trigger

1. Bestimme Objekt  $o_i$ , für das die Bedingung  $\phi$  überwacht werden soll
  - ▶ i.d.R. mehrere  $o_i$  betrachten, wenn Bedingung relationsübergreifend ist
  - ▶ Kandidaten für  $o_i$  sind Tupel der Relationsnamen, die in  $\phi$  auftauchen
2. Bestimme die elementaren Datenbankänderungen  $u_{ij}$  auf Objekten  $o_i$ , die  $\phi$  verletzen können
  - ▶ Regeln: z.B. Existenzforderungen beim Löschen und Ändern prüfen, jedoch nicht beim Einfügen etc.

## Integritätssicherung durch Trigger /2

3. Bestimme je nach Anwendung die Reaktion  $r_i$  auf Integritätsverletzung
  - ▶ Rücksetzen der Transaktion (**rollback**)
  - ▶ korrigierende Datenbankänderungen
4. Formuliere folgende Trigger:

```
create trigger t-phi-ij after  $u_{ij}$  on  $o_i$ 
when  $\neg\phi$ 
begin  $r_i$  end
```

5. Wenn möglich, vereinfache entstandenen Trigger

## Trigger in Oracle

- Implementierung in PL/SQL
- Notation

```
create [ or replace ] trigger trigger-name
before | after
insert or update [ of spalten ]
or delete on tabelle
[ for each row
[ when ( prädikat ) ] ]
PL/SQL-Block
```





