Moduldokumentation

Modul Distributed Data Management (ddm)

Simon Wächter

2019

Inhalt

[1 Einleitung 3](#_Toc1503247)

[1.1 Einleitung 3](#_Toc1503248)

[1.2 Lernziele 3](#_Toc1503249)

[1.3 Prüfungen 3](#_Toc1503250)

[2 Woche 1 4](#_Toc1503251)

[2.1 Gliederung 4](#_Toc1503252)

[2.2 Fachvertiefung 4](#_Toc1503253)

[2.3 Verteiltes Rechnen 4](#_Toc1503254)

[2.3.1 Verteiltes Rechnen (Distributed Computing) 4](#_Toc1503255)

[2.3.2 Was wird verteilt 5](#_Toc1503256)

[2.4 Einordnung der Thematik 5](#_Toc1503257)

[2.4.1 Verteilt, mobil, parallel … 5](#_Toc1503258)

[2.4.2 … Big Data 5](#_Toc1503259)

[2.5 Verteilte Datenbanksysteme 6](#_Toc1503260)

[2.5.1 Verteilte Datenbanksysteme 6](#_Toc1503261)

[2.5.2 Klassifizierung 6](#_Toc1503262)

[2.5.3 DDBS (P2P) Architecture 7](#_Toc1503263)

[2.5.4 MDBS Architecture 8](#_Toc1503264)

[2.5.5 Date’s 12 Regeln 8](#_Toc1503265)

[2.5.6 Aspekte Verteilter Datenbanksysteme 9](#_Toc1503266)

[2.6 Parallele Datenbanksysteme 9](#_Toc1503267)

[2.6.1 Parallele Datenbanksysteme 9](#_Toc1503268)

[2.6.2 Architekturen 9](#_Toc1503269)

[2.6.3 Aspekte paralleler Datenbanksysteme 10](#_Toc1503270)

[2.7 NoSQL Systeme 10](#_Toc1503271)

[2.7.1 NoSQL Systeme 10](#_Toc1503272)

[2.7.2 Konzepte NoSQL Systeme 10](#_Toc1503273)

[2.8 Mobile Datenbanksysteme 11](#_Toc1503274)

[2.9 Semantic Web 11](#_Toc1503275)

[2.10 Gliederung 12](#_Toc1503276)

[2.11 Was sind Trigger? 12](#_Toc1503277)

[2.12 Wozu dienen Trigger? 12](#_Toc1503278)

[2.13 Trigger Konzept 13](#_Toc1503279)

[2.14 Triggertypen 13](#_Toc1503280)

[2.14.1 Triggertypen: Ereignis 13](#_Toc1503281)

[2.14.2 Triggertypen: Timing 13](#_Toc1503282)

[2.14.3 Triggertypen: Granulat 14](#_Toc1503283)

[2.15 DML-Trigger auf Tabellen 14](#_Toc1503284)

[2.16 PL/SQL Block 15](#_Toc1503285)

[2.16.1 Einfacher PL/SQL Block 15](#_Toc1503286)

[2.16.2 Bildschirmausgabe 15](#_Toc1503287)

[2.16.3 Datentypen 16](#_Toc1503288)

[2.16.4 Anweisung 16](#_Toc1503289)

[2.17 1. Erfahrung mit Trigger 17](#_Toc1503290)

[2.17.1 Beispiel-Datenbasis 17](#_Toc1503291)

[2.17.2 1. Erfahrung mit Trigger 18](#_Toc1503292)

[2.18 Konsistent Prüfen 19](#_Toc1503293)

[3 Woche 2 21](#_Toc1503294)

# Einleitung

## Einleitung

Dieses Dokument stellt die Moduldokumentation für das Modul ddm dar. Allfällige Unterlagen sind im Modulordner zu finden.

## Lernziele

Skalierbarkeit, Verfügbarkeit und Ausfallstoleranz bilden Schlüsselanforderungen an heutige verteilte Informationssysteme. In diesem Modul vertiefen sich die Studierenden in die grundlegenden Konzepte und Technologien verteilter Datenhaltung und -verarbeitung. Dabei wird sowohl auf die klassischen Ansätze verteilter Datenbanksysteme eingegangen als auch neuartige Paradigmen der NoSQL Systeme besprochen.

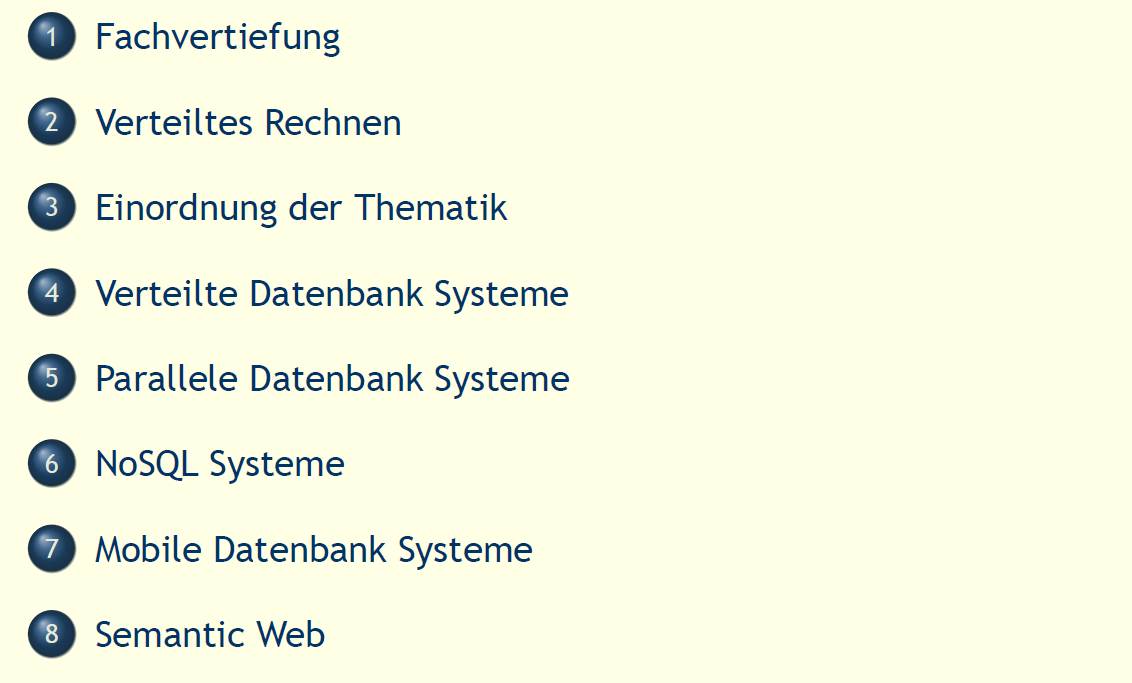
* Aktive Datenbanken:
  + Stored Procedures
  + ECA Prinzip
  + Trigger
* Verteilte Datenbanksysteme:
  + Architektur verteilter Datenbanksysteme
  + verteilter Datenentwurf
  + verteilte Anfrageverarbeitung
  + verteilte Transaktionen
  + Replikation
* NoSQL:
  + Daten-, Speicher- und Anfragemodelle
  + Key/Value, Wide Column, Document, Graph
  + Map/Reduce
  + Partitionierung (Sharding), verteilte Hashtabellen
  + Konsistenz- und Replikationsmodelle
  + CAP Theorem, BASE
* NoSQL Systeme: Cassandra, MongoDB, HBase, Neo4j

## Prüfungen

Die Modulnote setzt sich aus einer Erfahrungsnote zu 50% mit zwei Assessments zu je 25% und einer Modulschlussprüfung zu 50% zusammen.

# Woche 1

## Gliederung

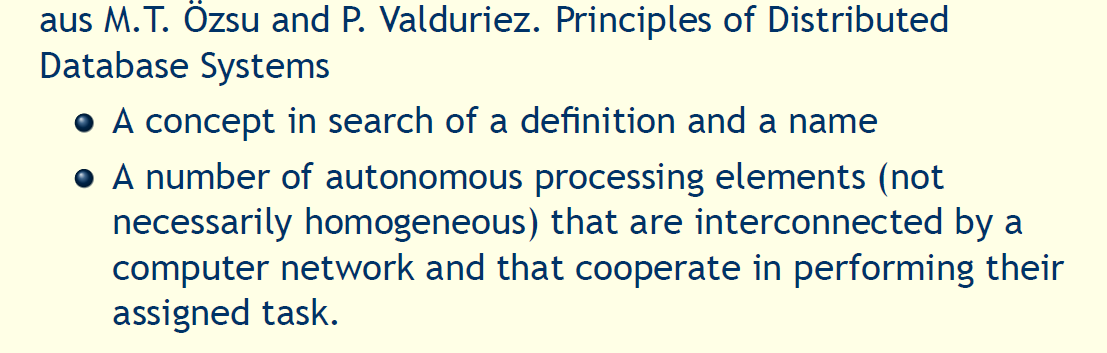


## Fachvertiefung



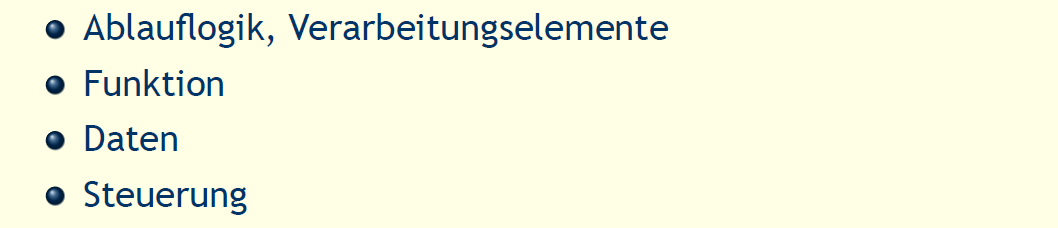
## Verteiltes Rechnen

### Verteiltes Rechnen (Distributed Computing)



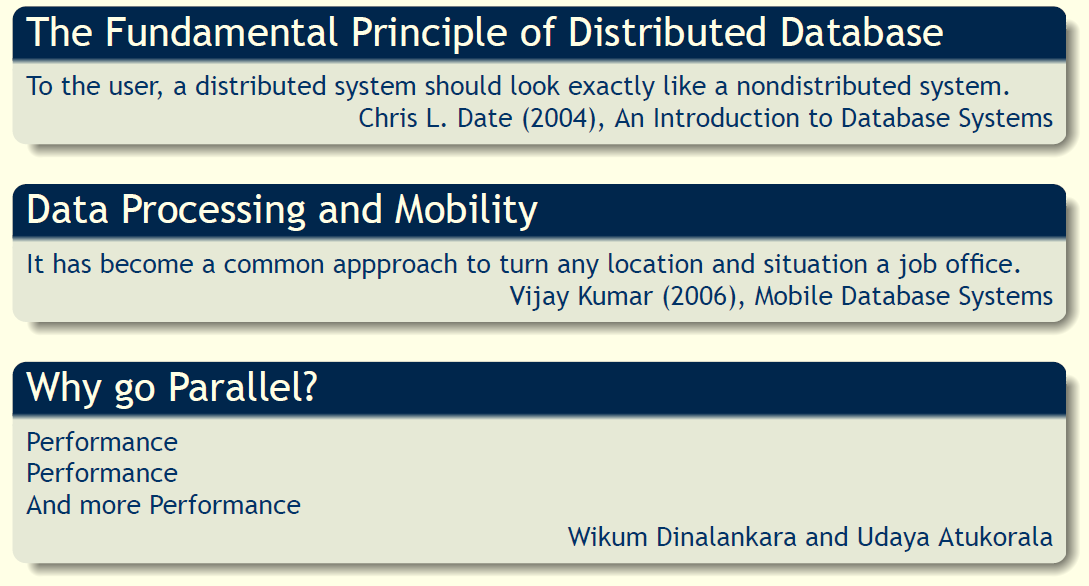
In verteilten Systemen kann die Ablauflogik, Daten, Steuerung verteilt sein. Für einen Benutzer verhält sich ein verteiltes System wie ein nicht verteiltes System. Warum sollte man Parallelisieren? -> Performance, Performance, Performance

### Was wird verteilt

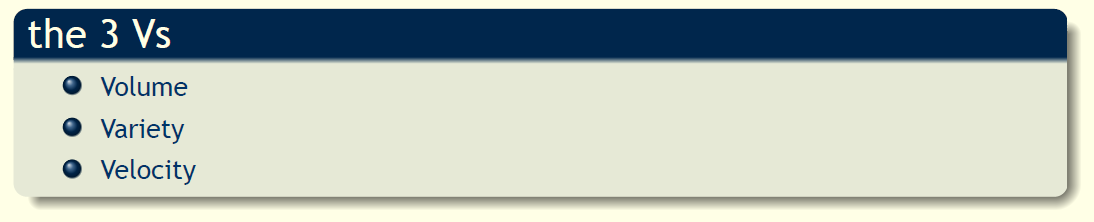


## Einordnung der Thematik

### Verteilt, mobil, parallel …

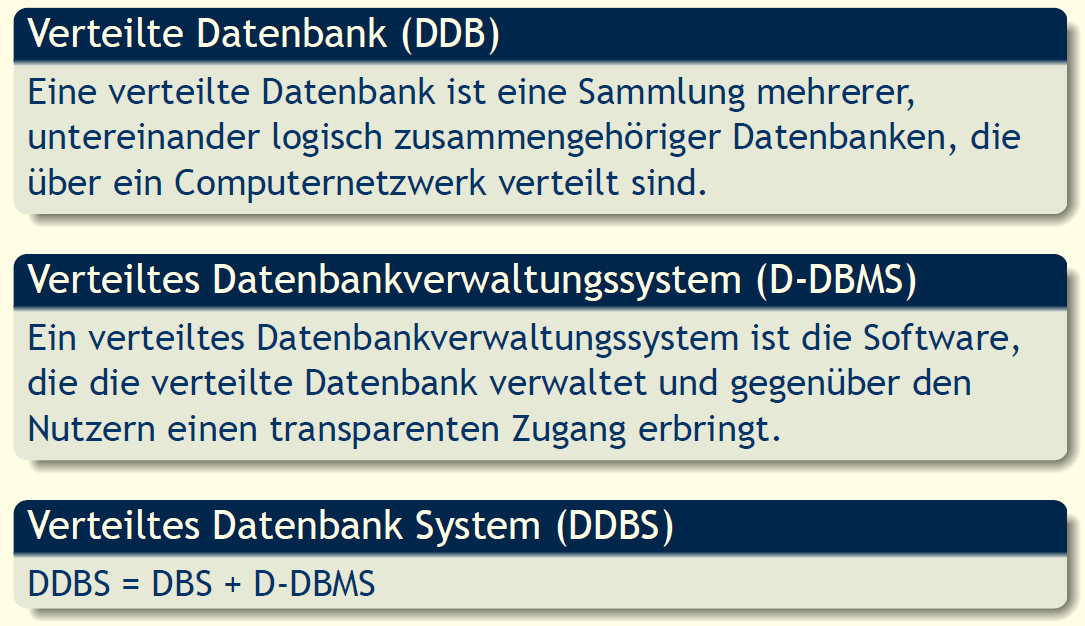


### … Big Data

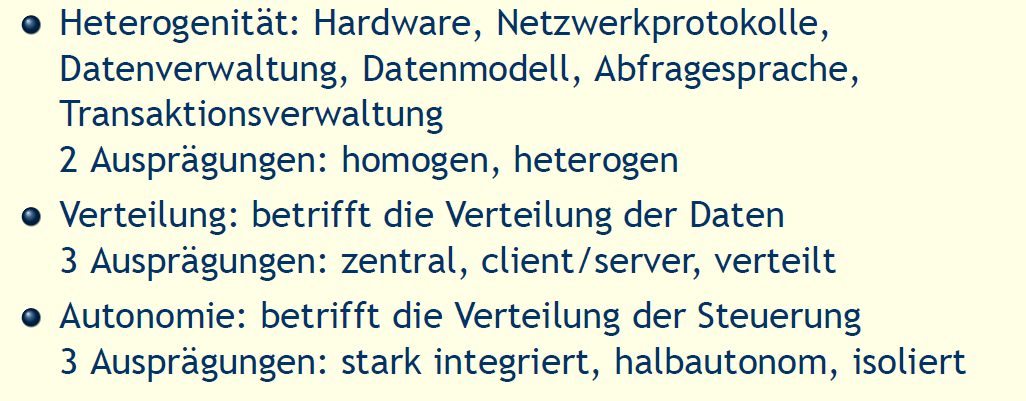


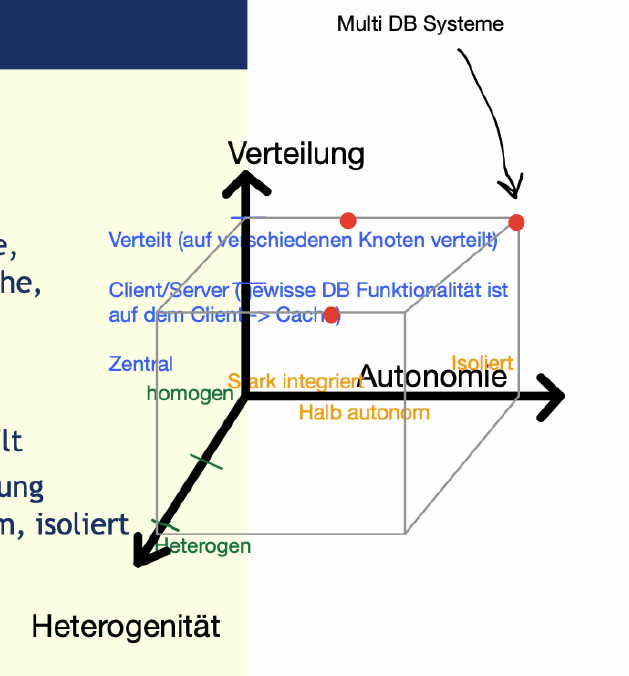
## Verteilte Datenbanksysteme

### Verteilte Datenbanksysteme

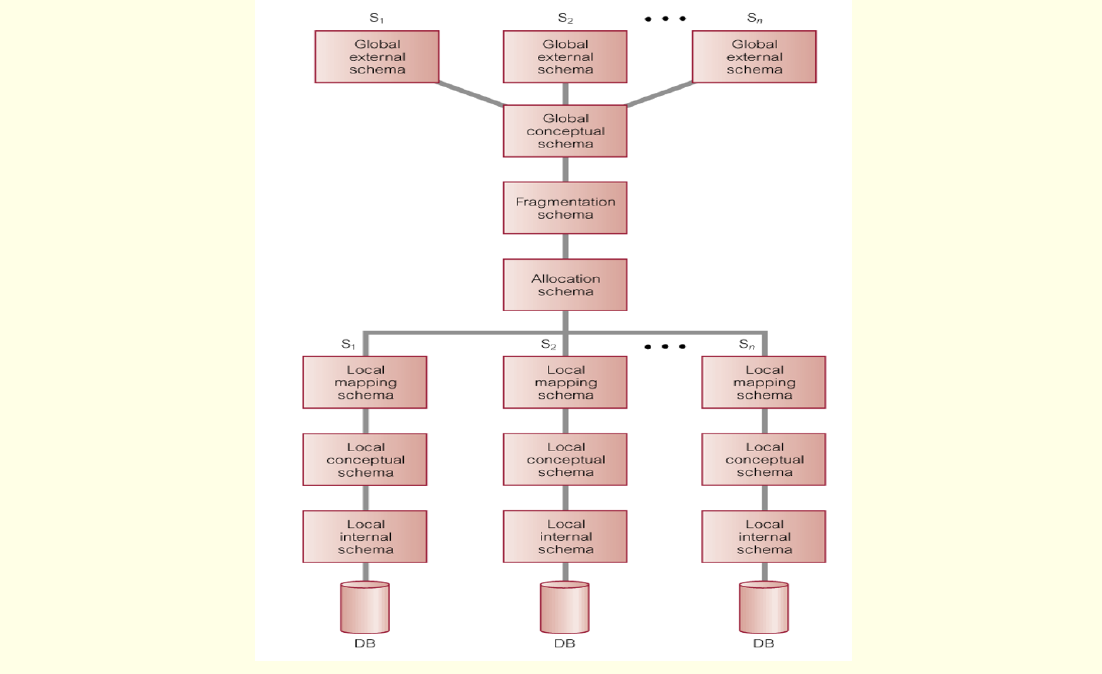


### Klassifizierung

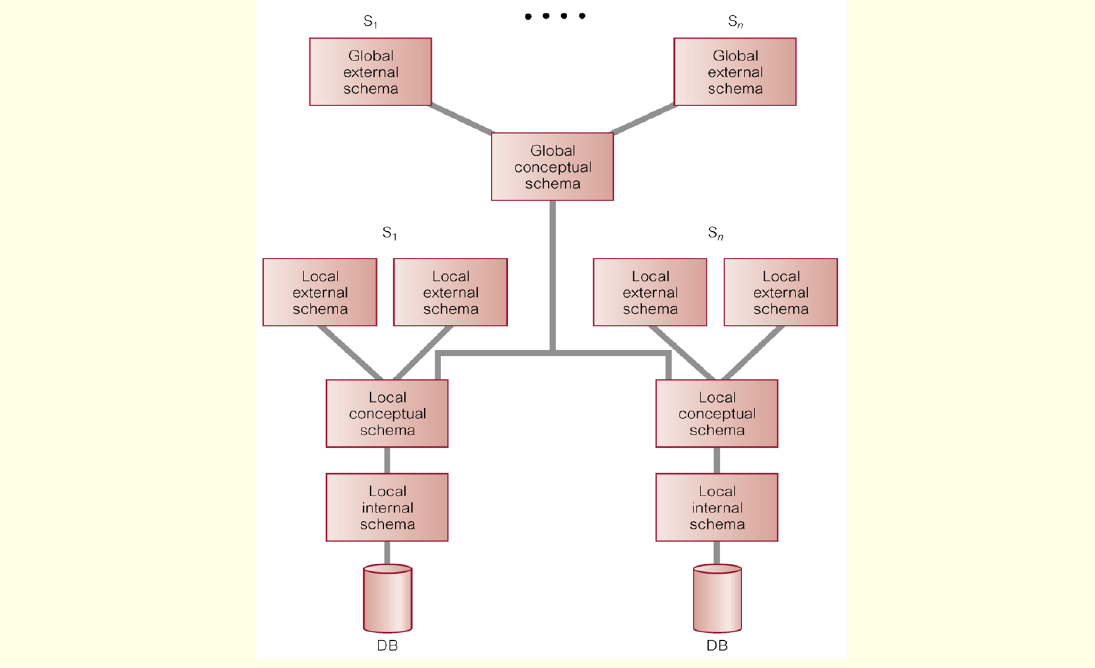




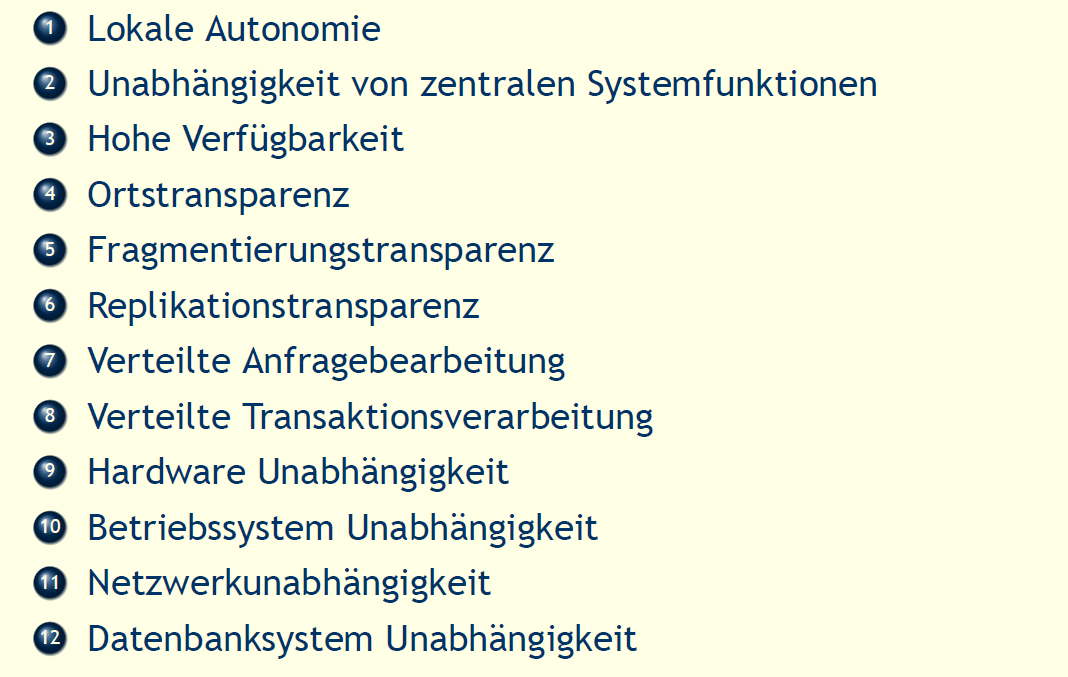
### DDBS (P2P) Architecture



### MDBS Architecture



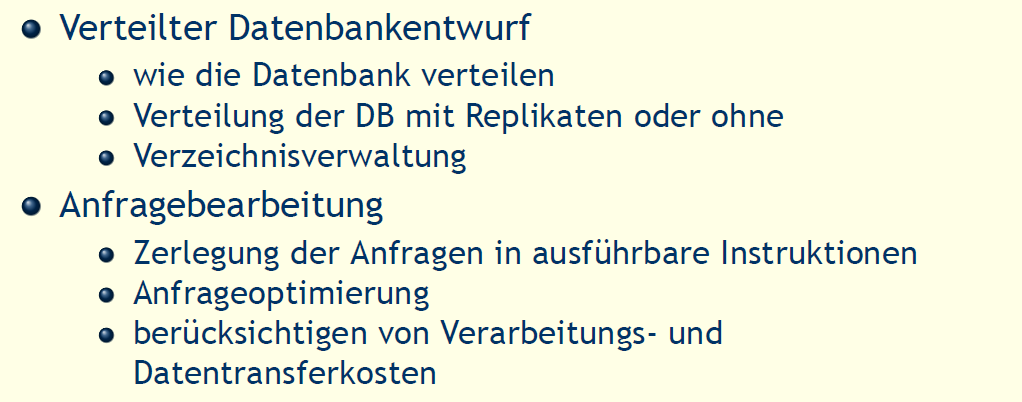
### Date’s 12 Regeln

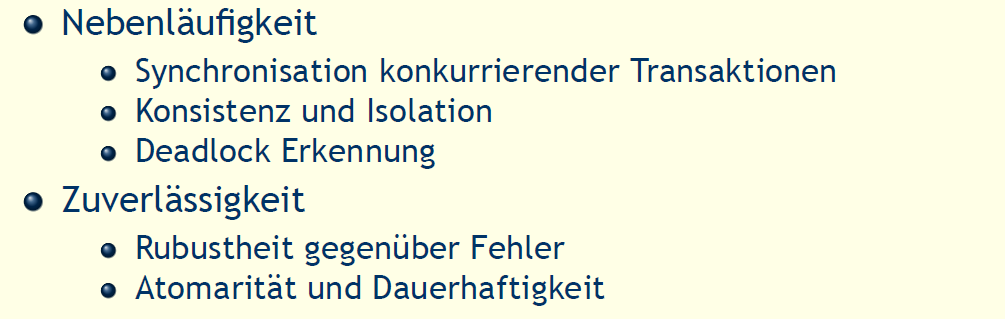


Ortstransparenz und Namenstransparenz:

* Kann mittels View oder Synonymen erstellt werden.
* Damit die Orts- und Namenstransparenz auf jeder DB genutzt werden kann müssen die Links
* symmetrisch sein. Als von jeder DB ein Link auf die anderen erstellt werden.

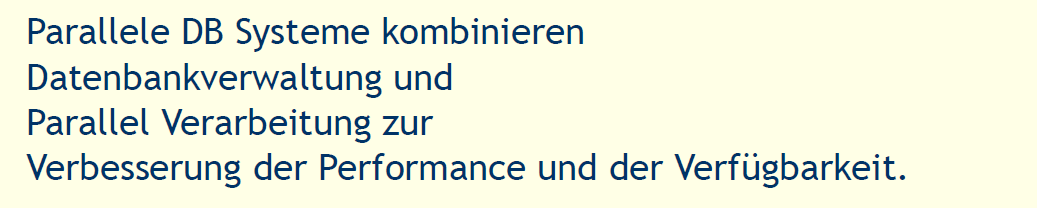
### Aspekte Verteilter Datenbanksysteme



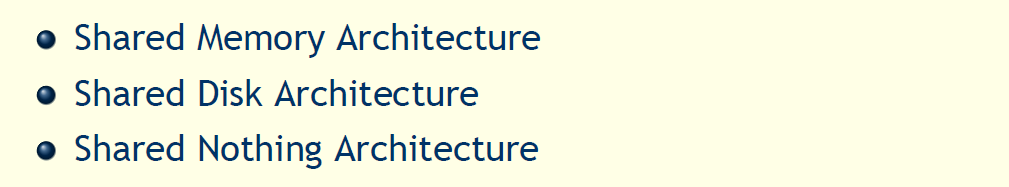


## Parallele Datenbanksysteme

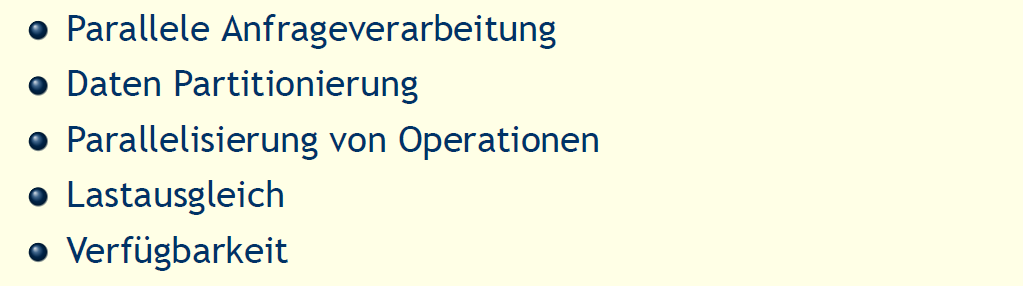
### Parallele Datenbanksysteme



### Architekturen

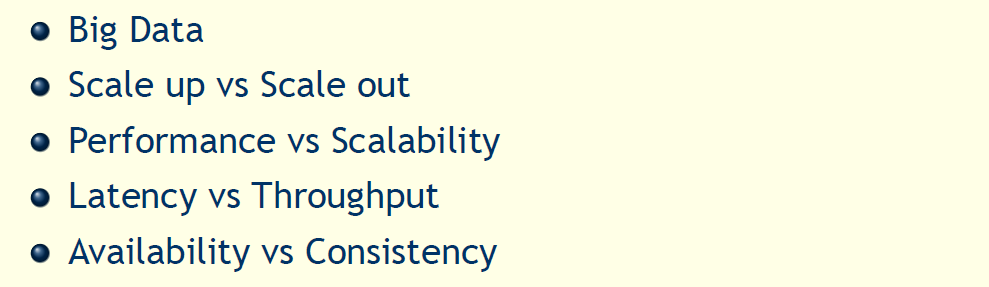


### Aspekte paralleler Datenbanksysteme



## NoSQL Systeme

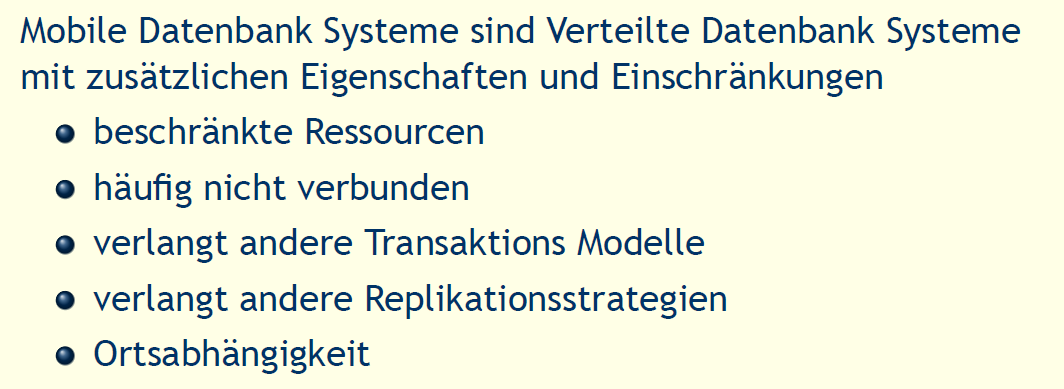
### NoSQL Systeme



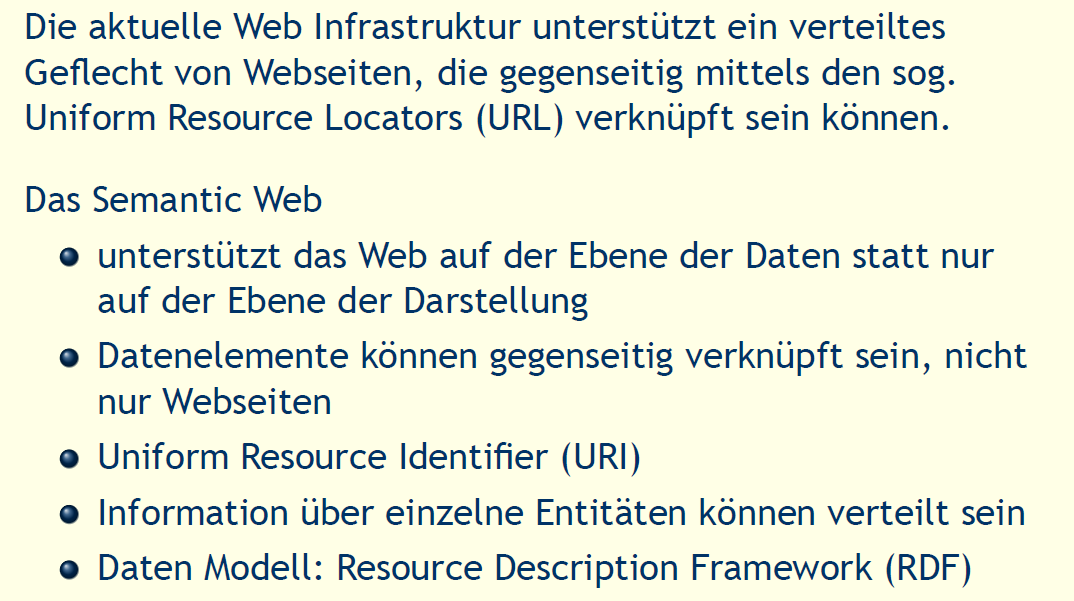
### Konzepte NoSQL Systeme



## Mobile Datenbanksysteme



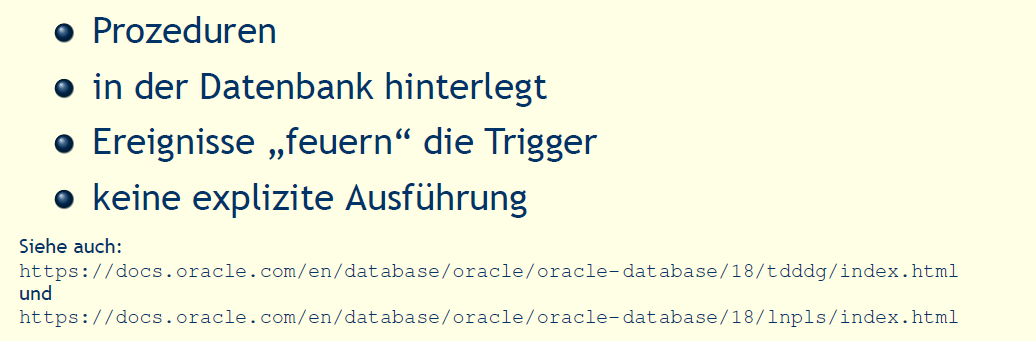
## Semantic Web



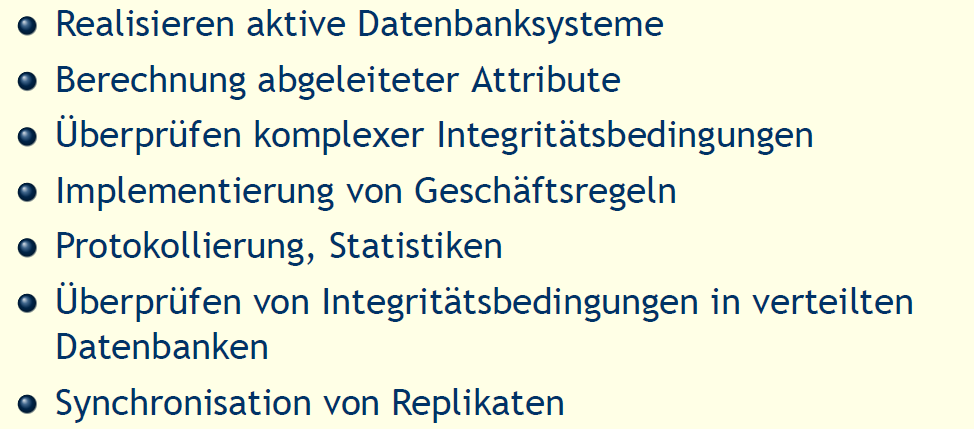
## Gliederung



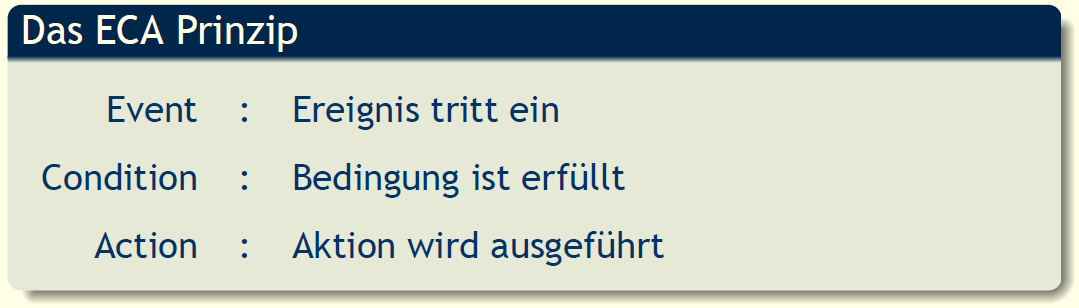
## Was sind Trigger?



## Wozu dienen Trigger?

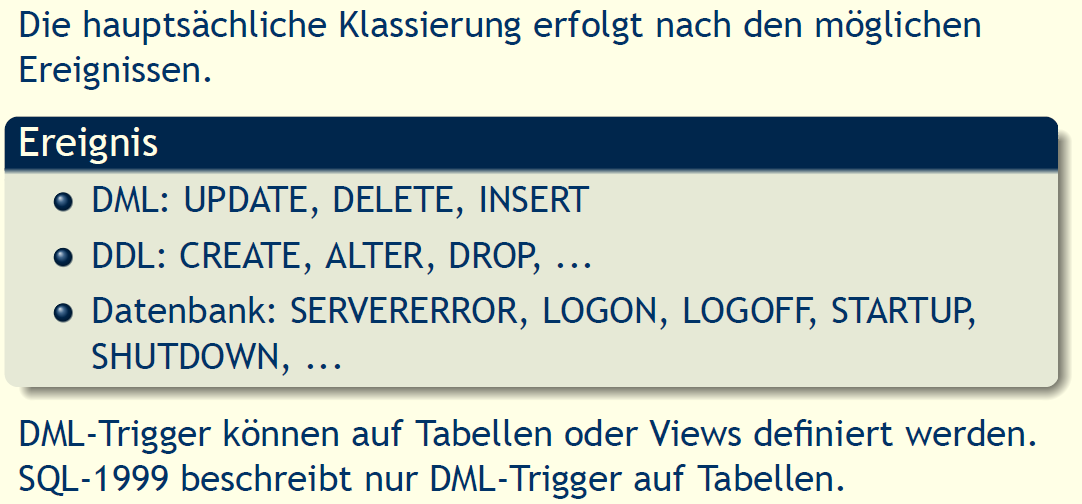


## Trigger Konzept

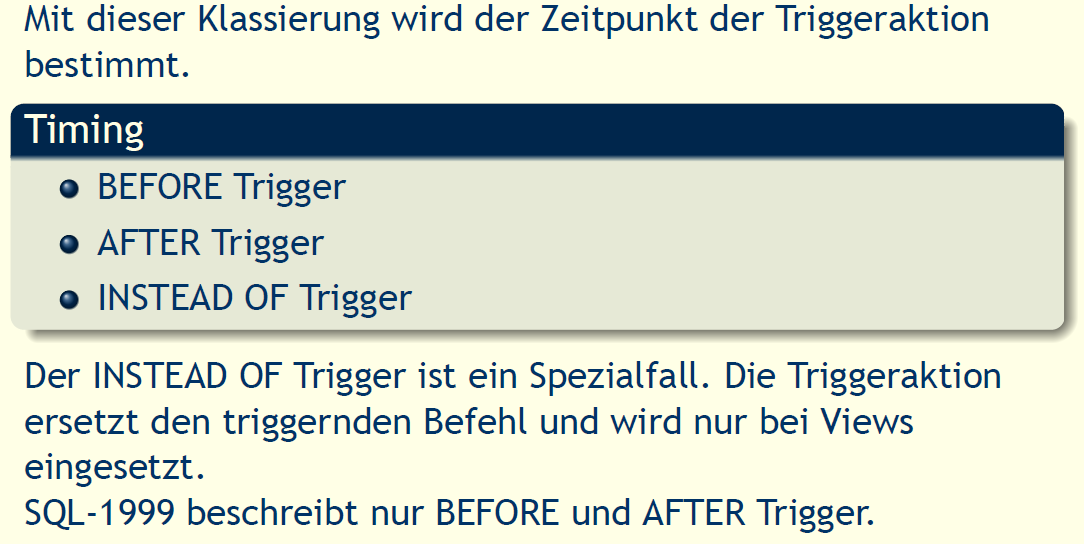


## Triggertypen

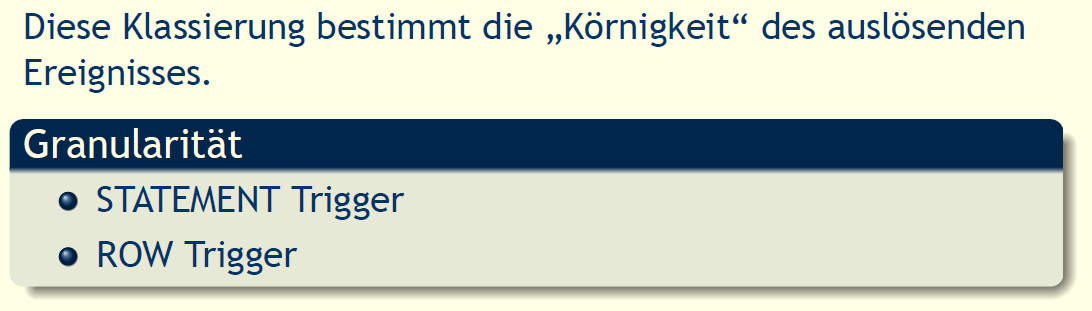
### Triggertypen: Ereignis



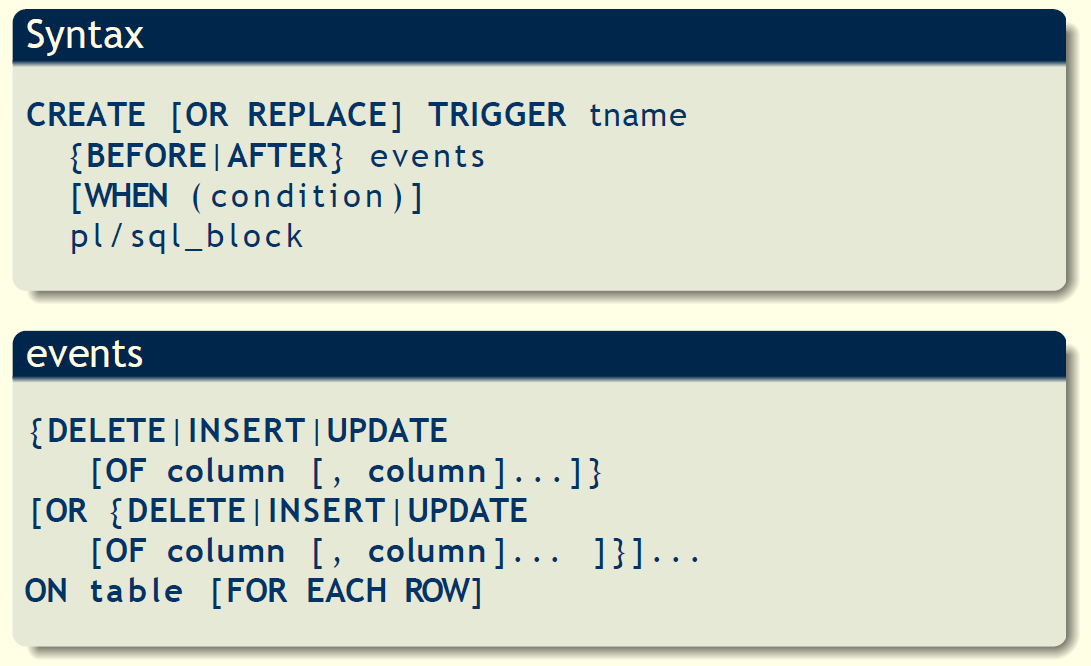
### Triggertypen: Timing



### Triggertypen: Granulat



## DML-Trigger auf Tabellen

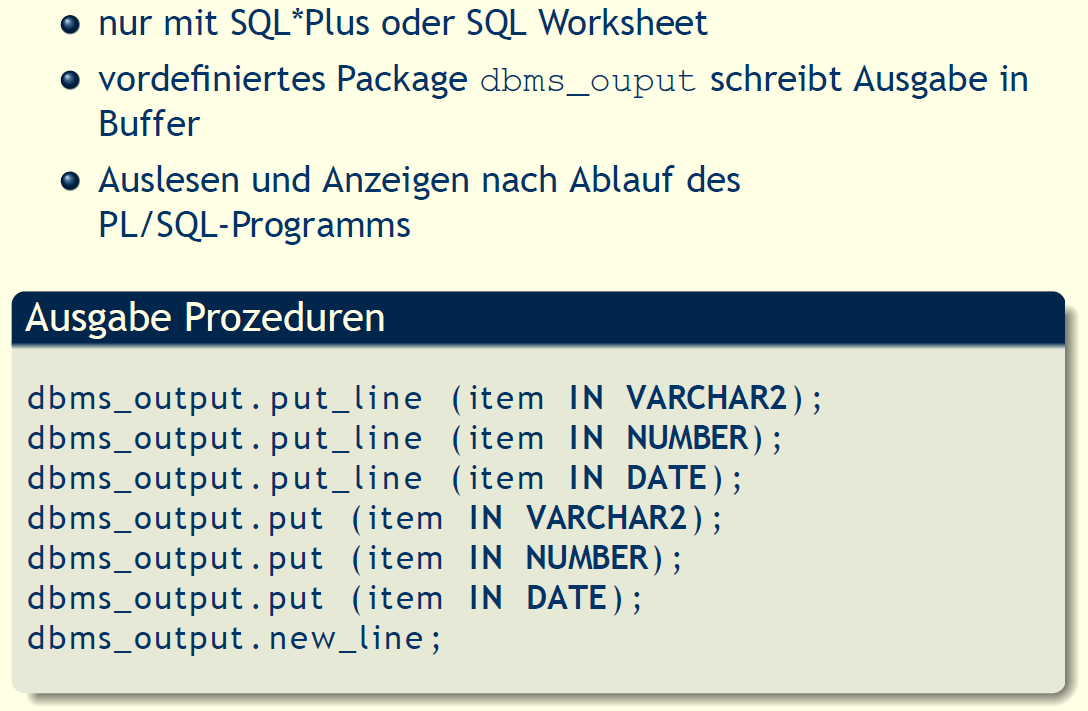


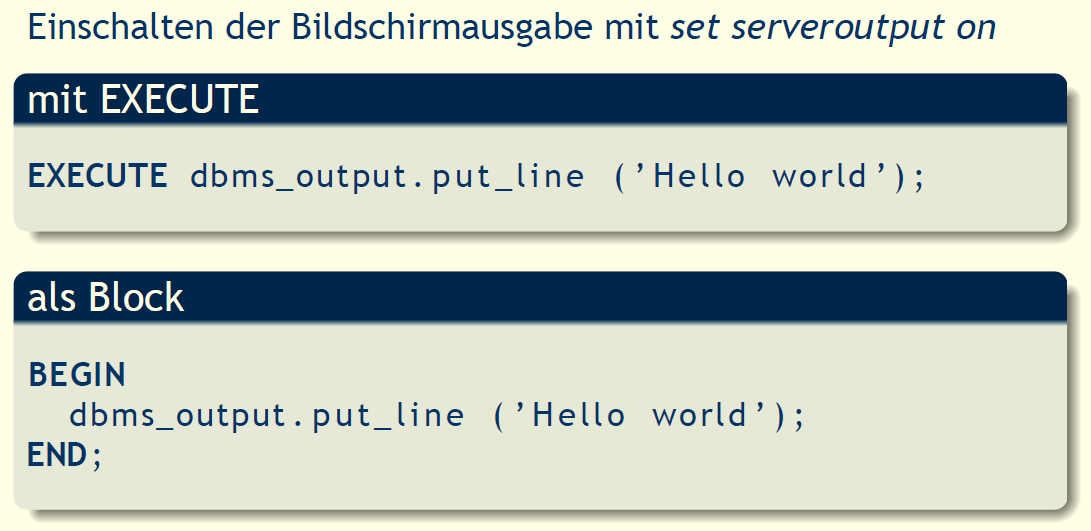
## PL/SQL Block

### Einfacher PL/SQL Block

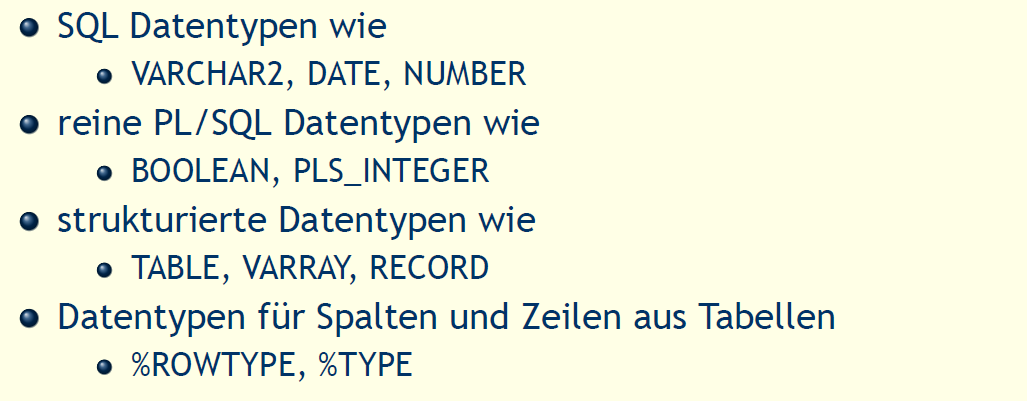


### Bildschirmausgabe

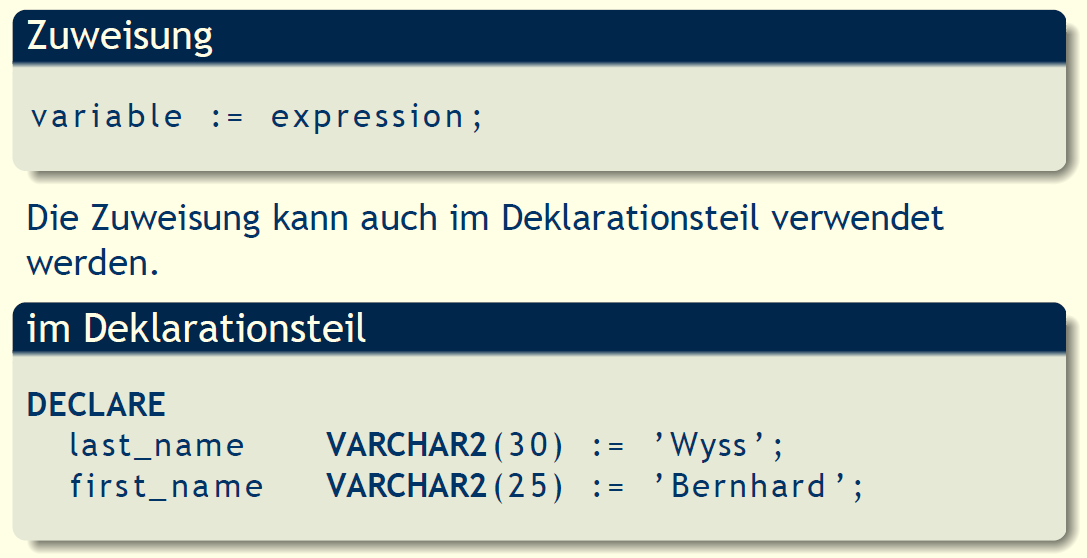


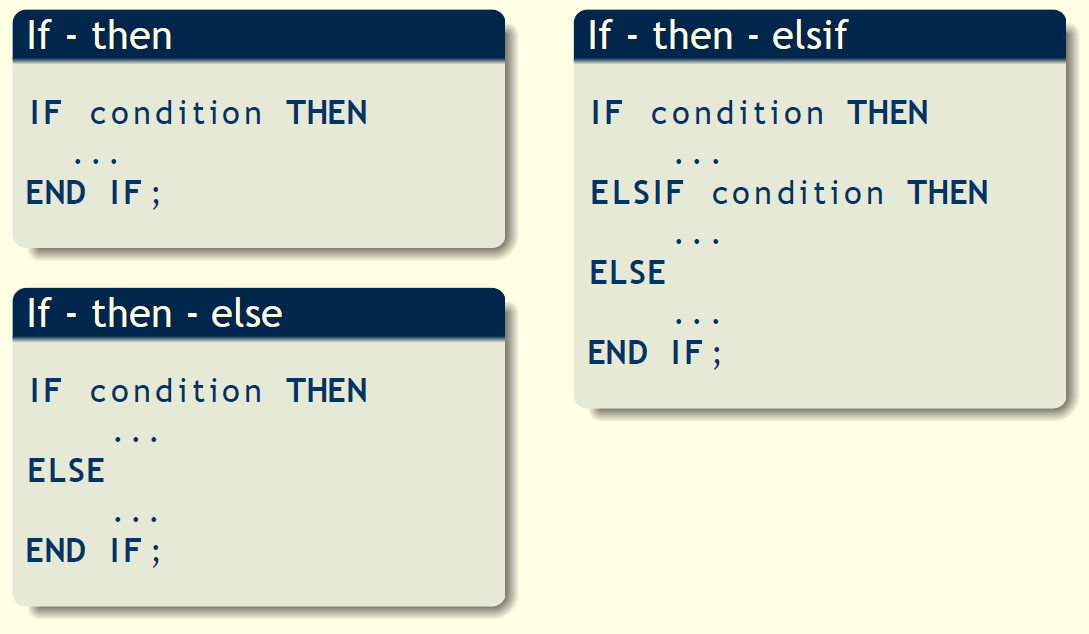


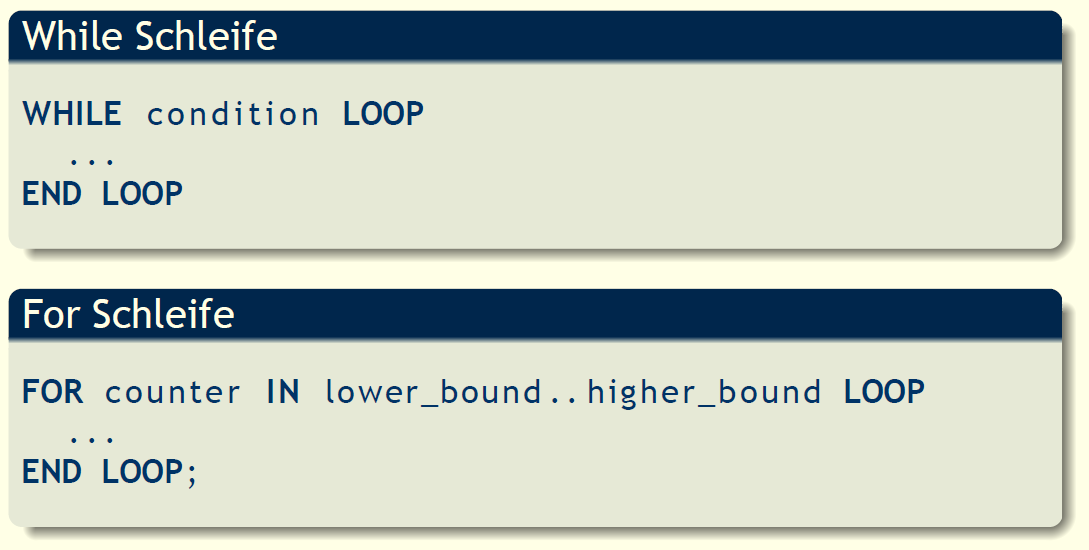
### Datentypen



### Anweisung

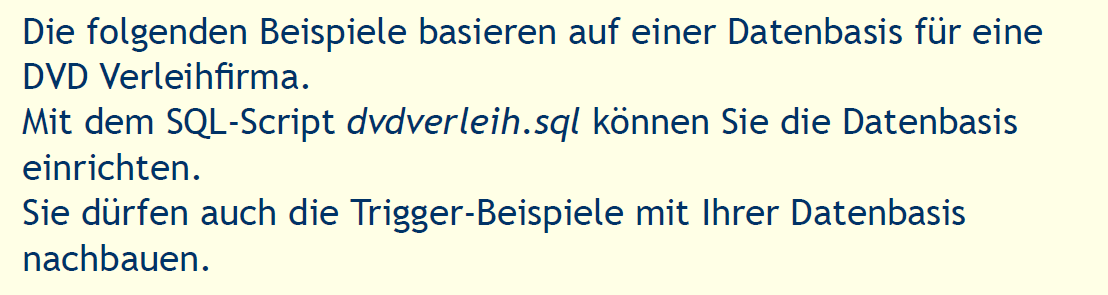




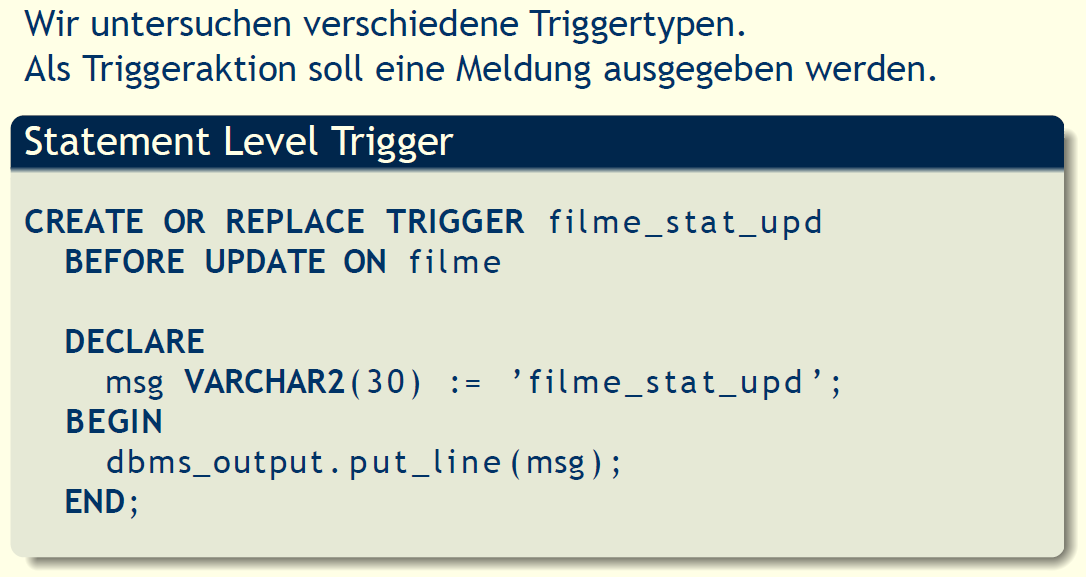


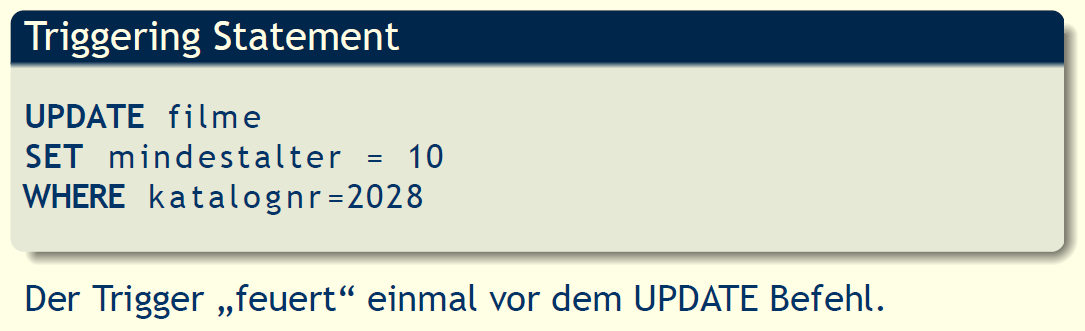
## 1. Erfahrung mit Trigger

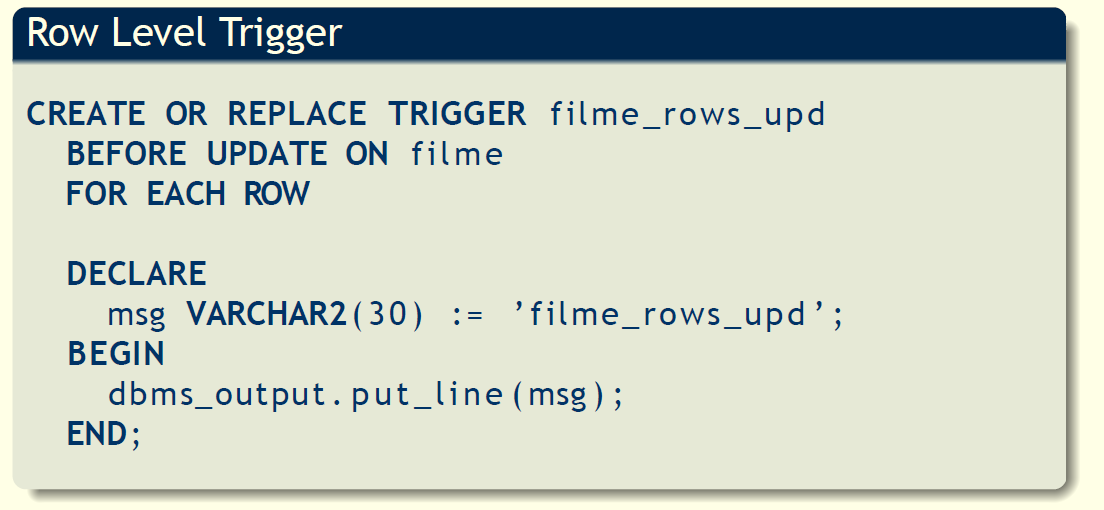
### Beispiel-Datenbasis

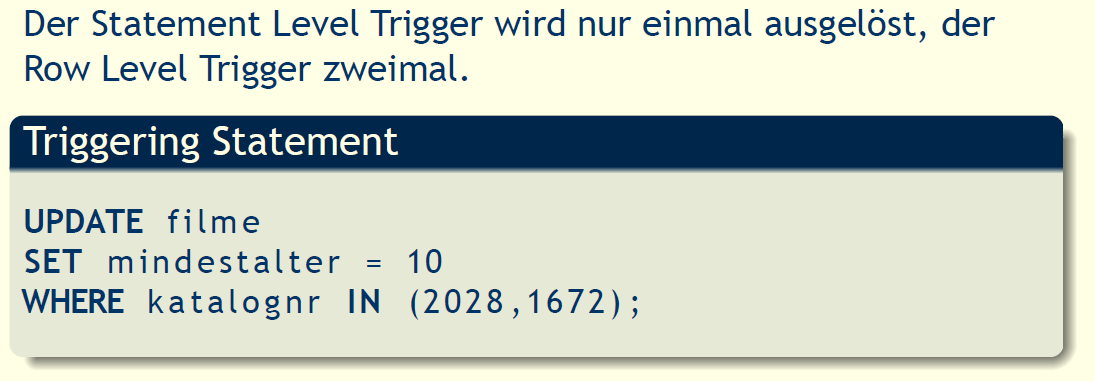


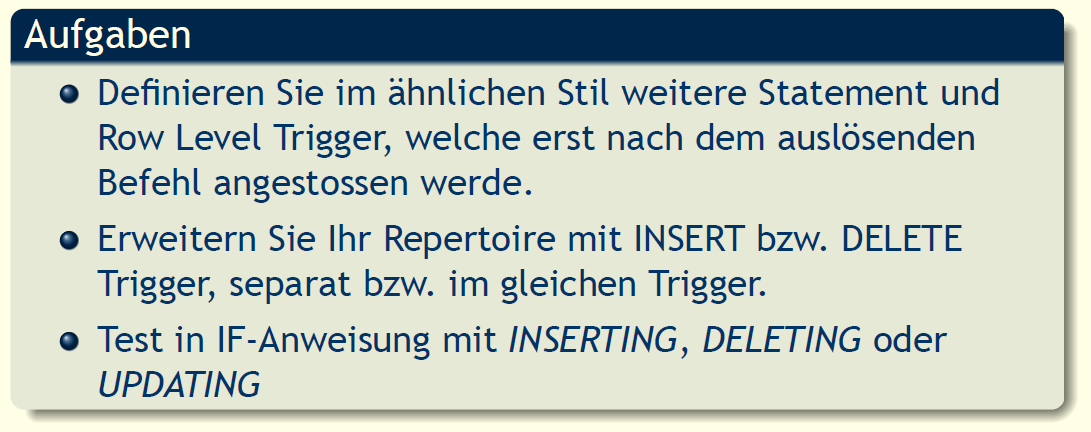
### 1. Erfahrung mit Trigger



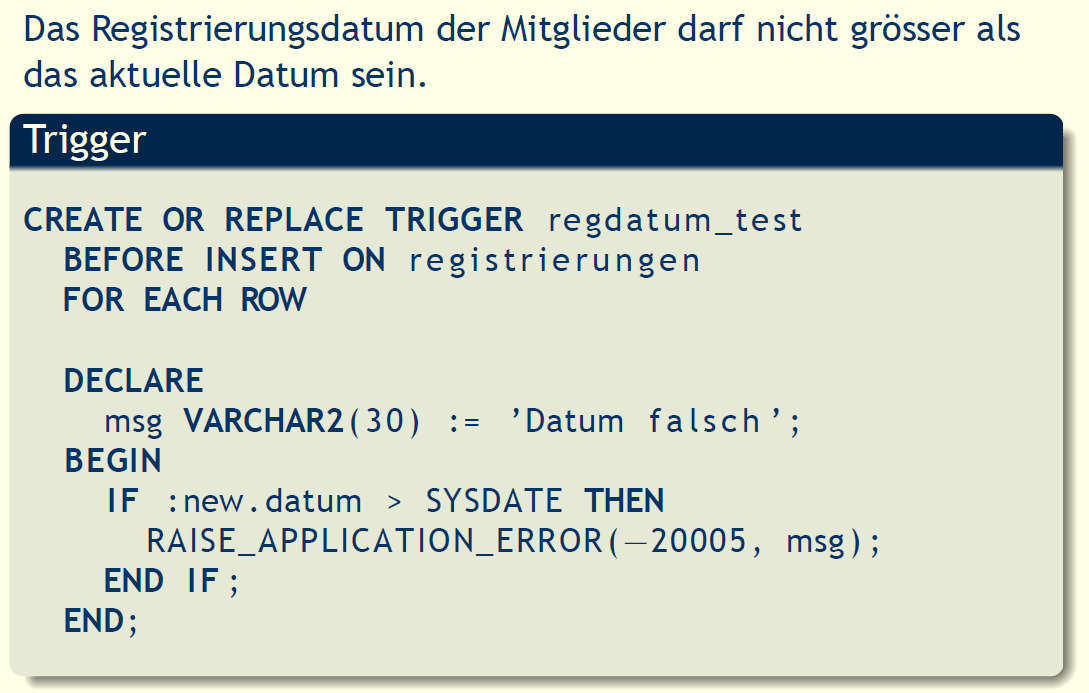


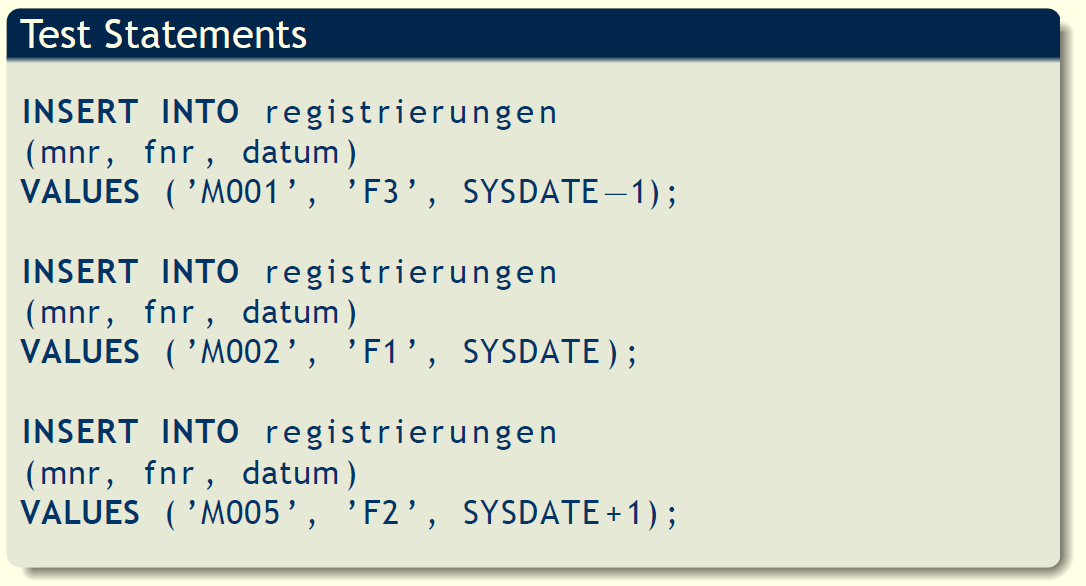


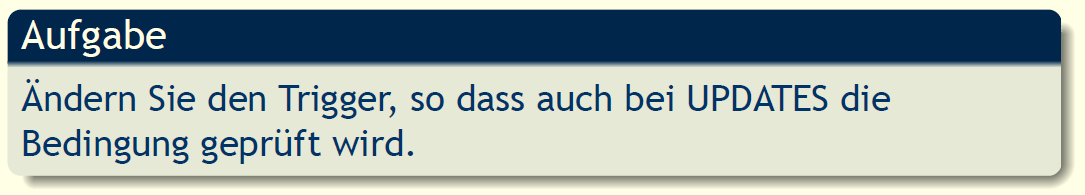




## Konsistent Prüfen







# Woche 2