

Modul Datenbanken

Vorlesung 3

Vom Datenbankentwurf zur Implementierung

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/).

Beim letzten Mal besprochen

- [Was sind Datenbanksysteme](#)
- Entwicklungsumgebung:
 - Installation von PostgreSQL (lokal) (**nicht** erfolgreich)
 - Verbindung mit PostgreSQL Service auf IFI Datenbank-Server (Virtual Machine)
 - mit PgAdminIII
 - mit psql auf der Kommando-Zeile

Beim letzten Mal nicht besprochen

- [Datenmodelle und relationales Datenmodell](#)
- [Entity Relationship Modellierung](#)

Beim letzten Mal nicht besprochen

- [Datenmodelle und relationales Datenmodell](#)
- [Entity Relationship Modellierung](#)

Aber jetzt mal endlich...

Teilnehmerinnen-Datenbank

1. Runde (semi-strukturiert)

Schritte im Projekt

1. Problem in der realen Welt

- Wer sind die Teilnehmerinnen in meinem Kurs?

2. Datenerfassung

- Liste der Teilnehmerinnen
- Implizites Model (Tabelle)

3. System zur Verarbeitung und Speicherung

- CSV Datei

Teilnehmerinnen-Datenbank

2. Runde (strukturiert)

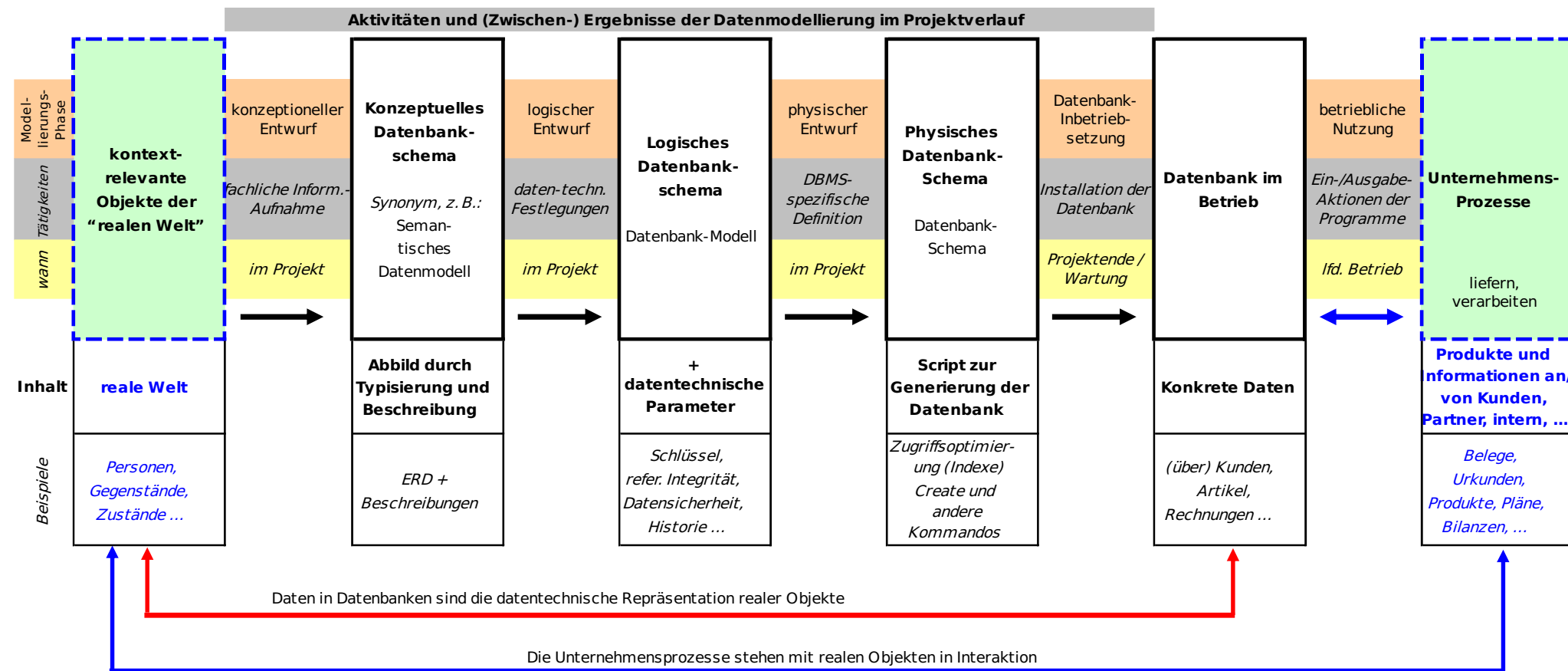
Mit dem relationalem Datenbank Ziel vorm Auge

Schritte im Projekt

1. Problem in der realen Welt → Anwendungsdefinition
 - Wer sind die Teilnehmerinnen in meinem Kurs?
2. Explizite Datenbankmodellierung → Konzeptioneller Entwurf
3. Datenbank Realisierung → Structured Query Language (SQL)
 1. Implementierung (source code) des Modells
 2. Datenerfassung (Daten laden und manipulieren)
4. System zur Verarbeitung und Speicherung
 - Datenbankmanagementsystem im Betrieb: PostgreSQL

Alternativer Blick: Datenbankentwicklungszyklus

illieren: Entwicklung von der fachlichen, implementierungsunabhängigen Konzeption bis zur Datenbank



„[DatMod v semMod zur DBK](#)“ von [VÖRBY](#), Konvertierung zu SVG [Perhelion](#) - eigene Erstellung, aus Wikipedia-Text abgeleitet. Lizenziert unter Gemeinfrei über [Wikimedia Commons](#).

Anwendungsdefinition

Welchen Zwecken soll die Erfassung aller Kursteilnehmerinnen dienen?

Die Teilnehmerinnen-Datenbank soll folgendes Ermöglichen:

- Informationsweitergabe an alle Teilnehmerinnen
- Namen lernen
- Notenvergabe
- Ermittlung des Wissensstands
- Wer macht welches Datenbankprojekt

Konzeptioneller Entwurf

Entity Relationship Modellierung (ERM)

- Semantischer Datenbankentwurf
 - unabhängig von konkreten Datenbank-spezifischen Modellen
 - dienen zum Entwurf von relationalen, Netzwerk- oder Objekt-Datenbanken
- Graphisch
- Idee simpel
 - Leider sehr viele inkonsistente Varianten

ERM: Entity (Entität)

- Ein Entity-Relationship-Model (ERM) geht von Entitäten (\sim = Objekten) aus.

"Eine Entität ist eine **eigenständige Einheit**, die im Rahmen des betrachteten Modells **eindeutig identifiziert** werden kann."

- Ein Entitätstyp wird durch Attribute genauer beschrieben und stellt somit eine abstrakte Beschreibung oder Charakterisierung von Entitäten da.

- Beispiel:

Lehrer = Entitätstyp

Renzo \sim = Entität (ein spezieller Lehrender)

ERM: Attribute

- Eigenschaften von Entitäten bzw. Entitätstypen werden durch Attribute beschrieben
- Attribute haben einen Namen und eine Domaine (= Bestimmung der Wertmenge).
 - Domainen werden in ERM oft nicht explizit angegeben

ERM: Keys (Schlüssel)

Da die Definition einer Entität beinhaltet, dass diese zumindest im Rahmen eines Modells eindeutig identifiziert werden kann, braucht jeder Entitätstyp eine Menge von Attributen als Schlüssel.

Die Auswahl eines oder mehrerer Attribute als Schlüssel legt fest, dass es keine zwei Entitäten eines Entitätstyp geben kann die identische Attributwerte haben.

- Wichtige Eigenschaften:
 - Eindeutigkeit
 - Zuteilbarkeit

ERM: Relationship (Beziehung)

Verschiedene Entitäten können zueinander in Beziehung gesetzt werden.

- In jeder Beziehung haben Entitäten gewisse Rollen
- Beziehungen können Eigenschaften (Attribute) haben
- Beziehungen haben Kardinalitäten

Notationen

Es gibt verschiedene Formen ERM zu notieren (textuell und/oder graphisch):

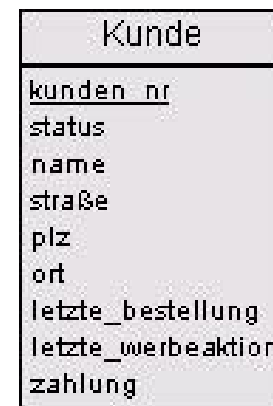
- [Chen Notation](#)
- [Crow Foot's](#)
- [Unified Modelling Language](#) (UML)

Crow Foot's (Martin) Notation

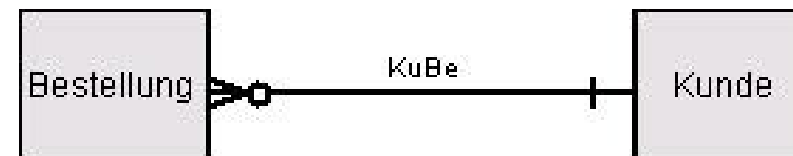
Entity



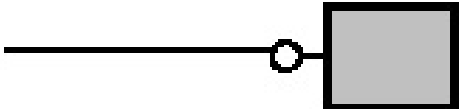
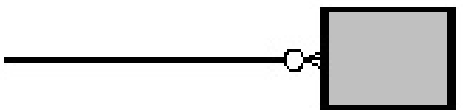


Attribute



Beziehungen/Relationship



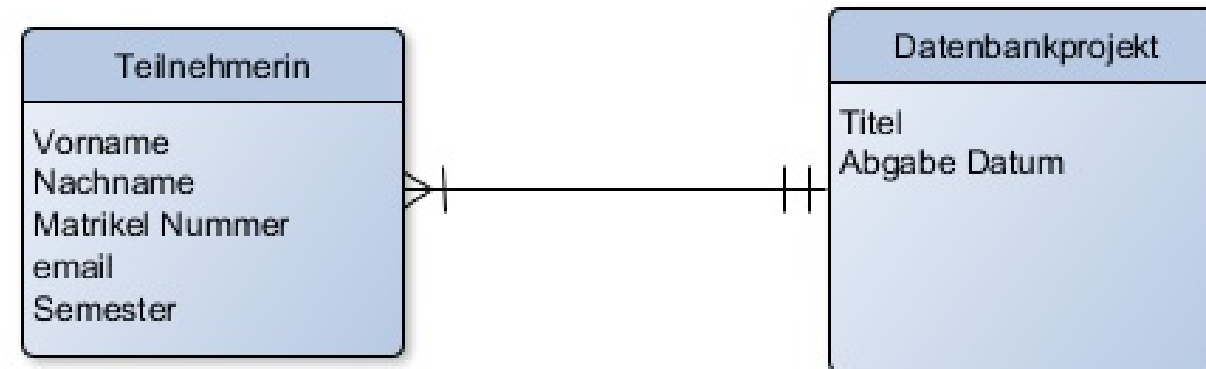
Kardinalitäten

(1)	(2)	(3)	(4)
0..1		1	c
0..*		N	mc
1..1		1	1
1..*		N	m

1. UML Notation
2. Crow Foot
3. Darstellung nach Chen
4. Darstellung nach Zehn

[Abb. 10.6 aus Matthiessen](#)

ERM Teilnehmerinnen



Structured Query Language (SQL)

SQL ist eine Datenbanksprache

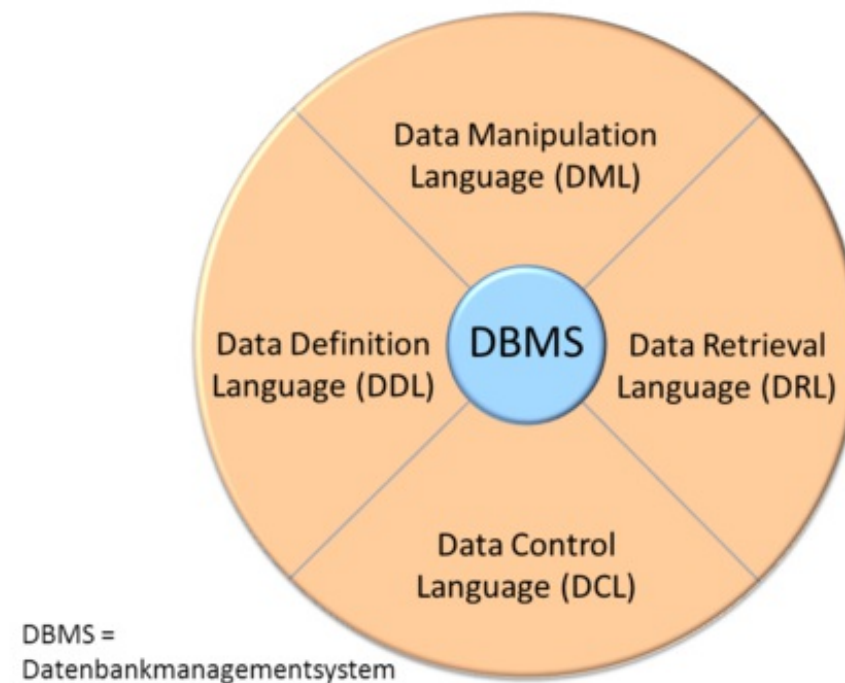
1. zur Definition von Datenstrukturen/Modellen
2. zum Bearbeiten (Einfügen, Verändern, Löschen)
3. zum Abfragen von darauf basierenden Datenbeständen
4. zur Rechtevergabe

SQL Eigenschaften

- basiert auf relationaler Algebra
- an English angelehnt
- Deklarativ und funktional
- Fast alle Datenbanken verstehen SQL
- Standardisiert
 - PostgreSQL hat einer der besten Umsetzungen

SQL Überblick

Structured Query Language (SQL)



- DML = Data Manipulation Language: Ändern, Einfügen, Löschen und lesender Zugriff
- DDL = Data Definition Language: Definition des Datenbankschemas
- DCL = Data Control Language: Rechteverwaltung und Transaktionskontrolle

DDL: Create Table

Teilnehmerin
Vorname
Nachname
Matrikel Nummer
email
Semester

```
CREATE TABLE teilnehmer (  
  --Spalten Name Datentyp,  
  vorname text,  
  nachname text,  
  matrikel_nr integer,  
  email text,  
  semester integer  
);
```

s.

<http://www.postgresql.org/docs/9.3/interactive/ddl-basics.html> und
<http://www.postgresql.org/docs/9.3/interactive/sql-createtable.html>

DDL: Create Table Primary Key

Teilnehmerin
Vorname
Nachname
Matrikel Nummer
email
Semester

```
CREATE TABLE teilnehmer (  
  --Spalten Name Datentyp,  
  vorname text,  
  nachname text,  
  matrikel_nr integer,  
  email text,  
  semester integer  
);
```


DDL: Create Table Primary Key

Teilnehmerin
Vorname
Nachname
Matrikel Nummer
email
Semester

```
CREATE TABLE teilnehmer (  
  --Spalten Name Datentyp,  
  vorname text,  
  nachname text,  
  -- Simpler (nicht bester Primary Key)  
  matrikel_nr integer PRIMARY KEY,  
  email text,  
  semester integer  
);
```

DML: Daten Einfügen

Teilnehmerin
Vorname
Nachname
Matrikel Nummer
email
Semester

```
INSERT INTO teilnehmer  
  (vorname, nachname, matrikel_nr, email, semester)  
VALUES  
  ('renzo', 'kottmann', 007, 'renzo@007.bond', 0);
```

s. <http://www.postgresql.org/docs/9.3/interactive/dml-insert.html> und
<http://www.postgresql.org/docs/9.3/interactive/sql-insert.html>

Teilnehmerinnen Datenbank

- Erarbeitetes [Ergebnis dieser Vorlesung/Uebung](#)

Weiterfuehrende Fragen:

1. Wie ändert sich das ERM und die implementierung wenn folgende Anforderng hinzukommt:
 - Die Datenbank soll für alle vergangenen und zukünftigen Datenbankkurse informationen speichern können
2. Welche Datentypen gibt es schon in PostgreSQL?
3. Kann man eigene Datentypen definieren?
 - Wenn ja, welche Möglichkeiten gibt es?
4. Welche weiteren SQL-Befehle für Datenmodell-Management (DDL) gibt es noch?

Uebung: Datenbank Einrichtung

1. Installiere auf einem Rechner PostgreSQL version > 9.3
2. Erstelle eine neue Datenbank
3. Erstelle zwei User (role) A und B (mit Namen Deiner Wahl)
4. Erstelle eine Datenbank die dem User A gehört
5. Erstelle Tabellen die User A gehören und gib User B Leserechte (SELECT permission) auf diesen Tabellen

Schoene Woche noch :)

Referenzen:

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