

# Entwickelte Trainer-Tools des EILD-Projekts an der H-BRS

Einordnung der Trainer in die Phasen des Datenbankentwurfs



Ein Kooperationsvorhaben empfohlen durch die:



INNOVATION DURCH KOOPERATION

Gefördert durch:

Ministerium für  
Kultur und Wissenschaft  
des Landes Nordrhein-Westfalen



# Projekt EILD.nrw



Im Projekt EILD.nrw werden Open Educational Resources (OER) für die Lehre im Fach Datenbanken entwickelt.

Die Hochschule Bonn-Rhein-Sieg (H-BRS) hat interaktive Trainer entwickelt, die bestimmte Aspekte der Phasen des Datenbankentwurfs trainieren.

Die Trainer werden im Folgenden in diese Phasen eingeordnet und vorgestellt.

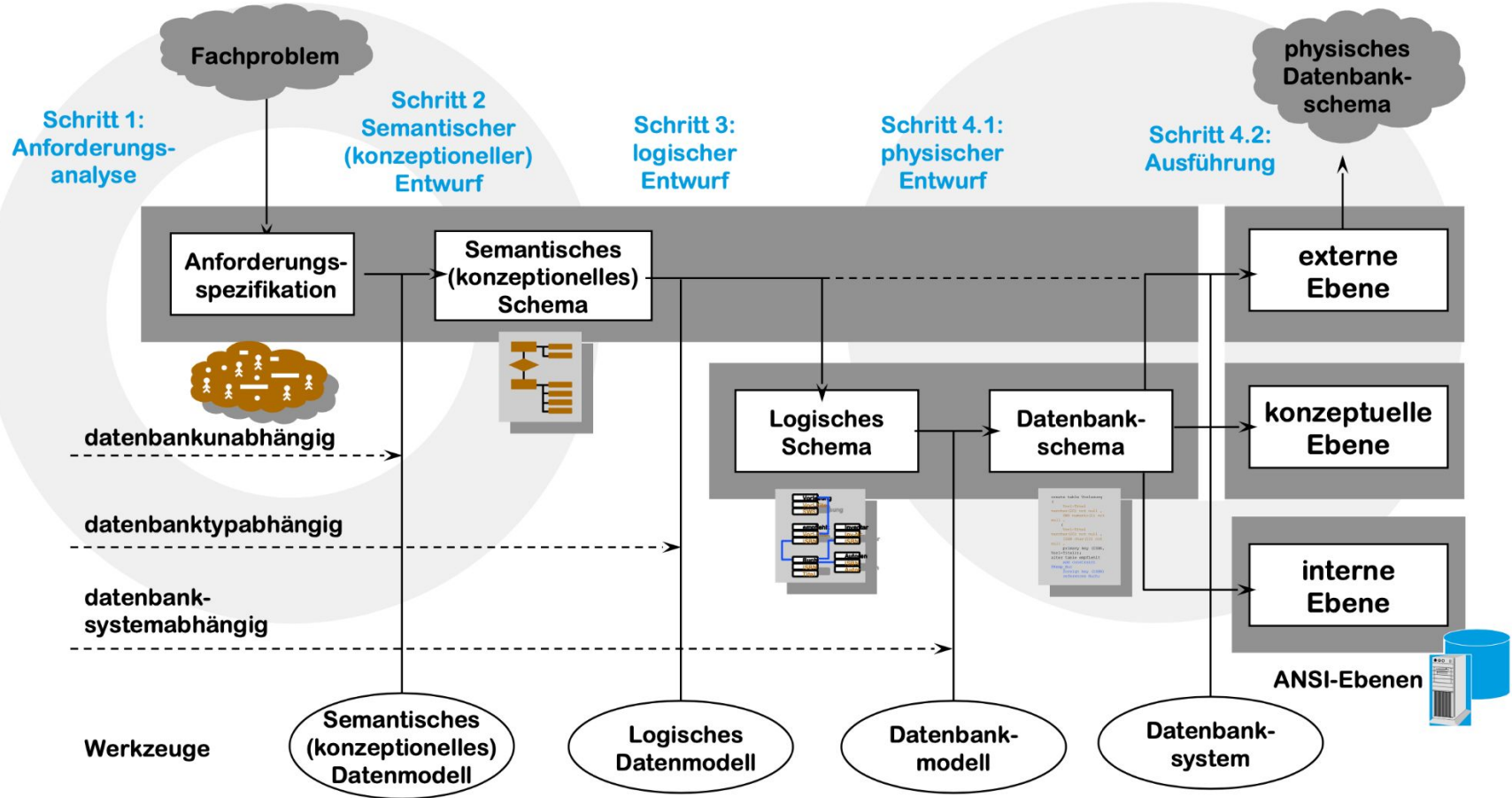
Entwickelten interaktive Trainer:

- [ER-Trainer](#)
- [ER-REL-Trainer](#)
- LOS-Trainer
- Normalisierungstrainer
- Serialisierungstrainer

Weitere entwickelte Tools:

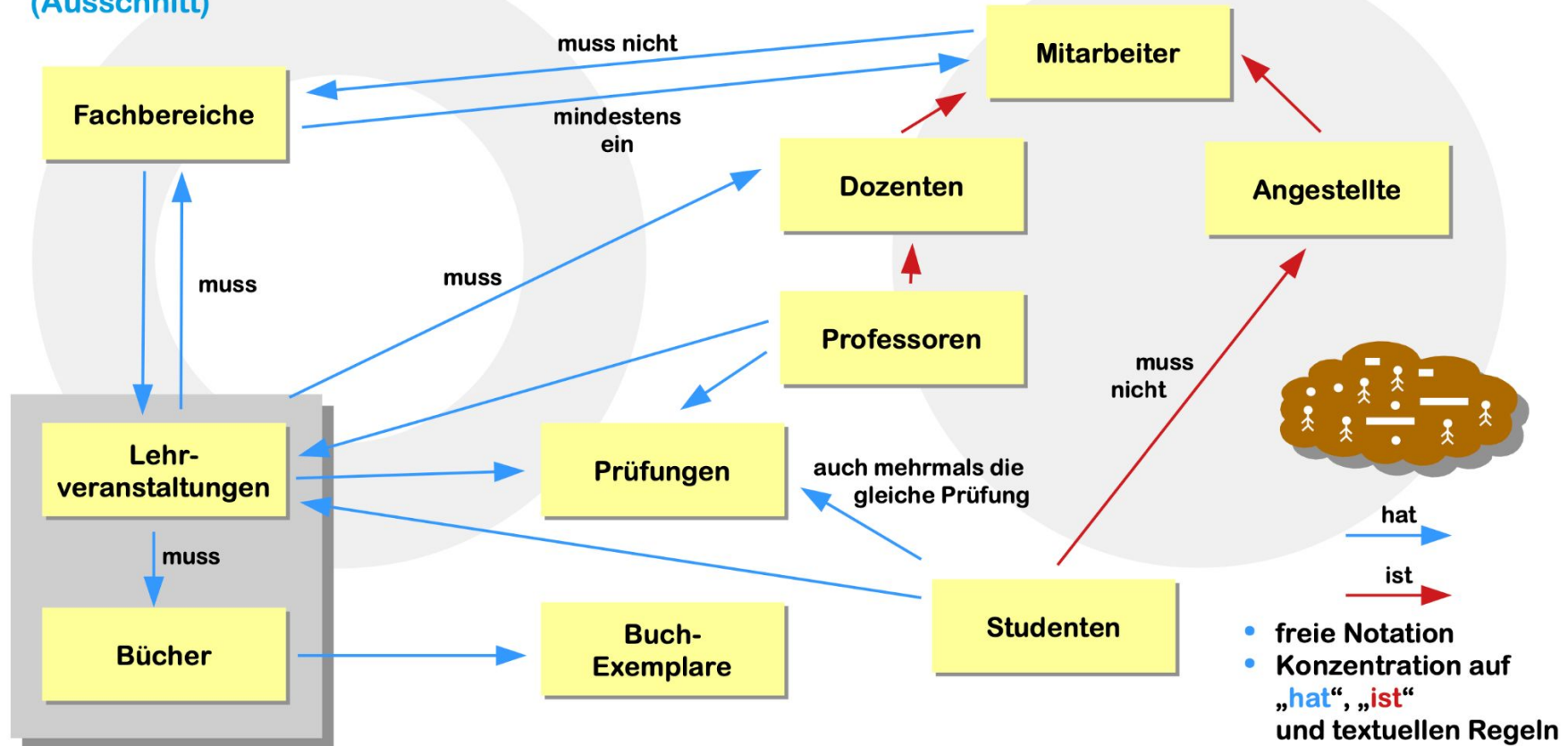
- Interaktives ERD
- Multiple Choice

# Phasen des Datenbankentwurf



# Phasen des Datenbankentwurf | Schritt 1: Anforderungsanalyse

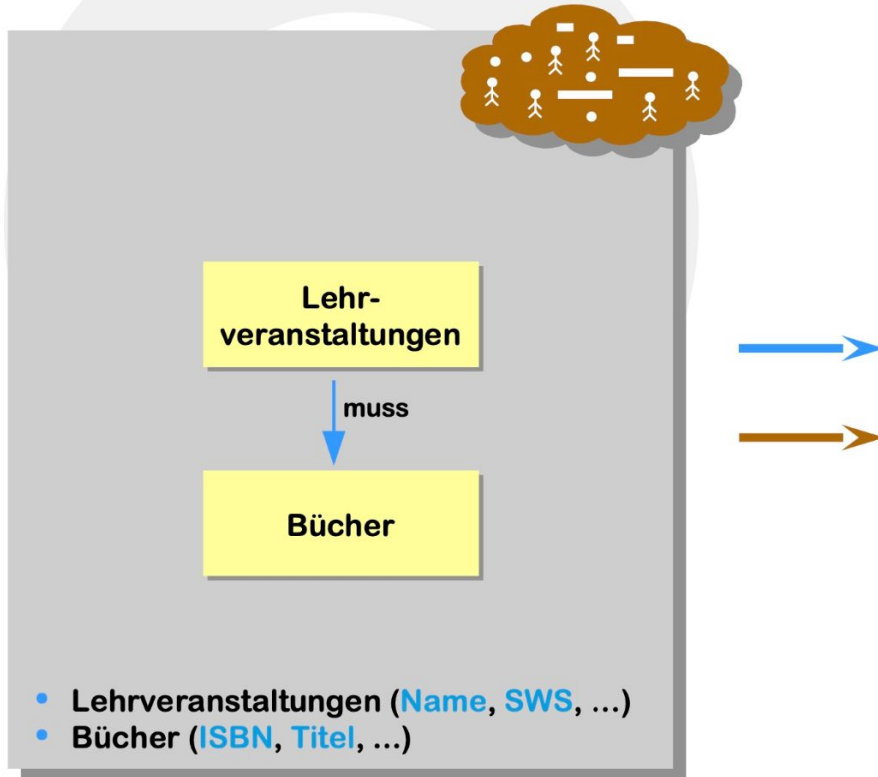
## Skizze der Miniwelt im Beispiel Lehrbetrieb (Ausschnitt)



# Phasen des Datenbankentwurf | Schritt 2: Semantischer (konzeptioneller) Entwurf

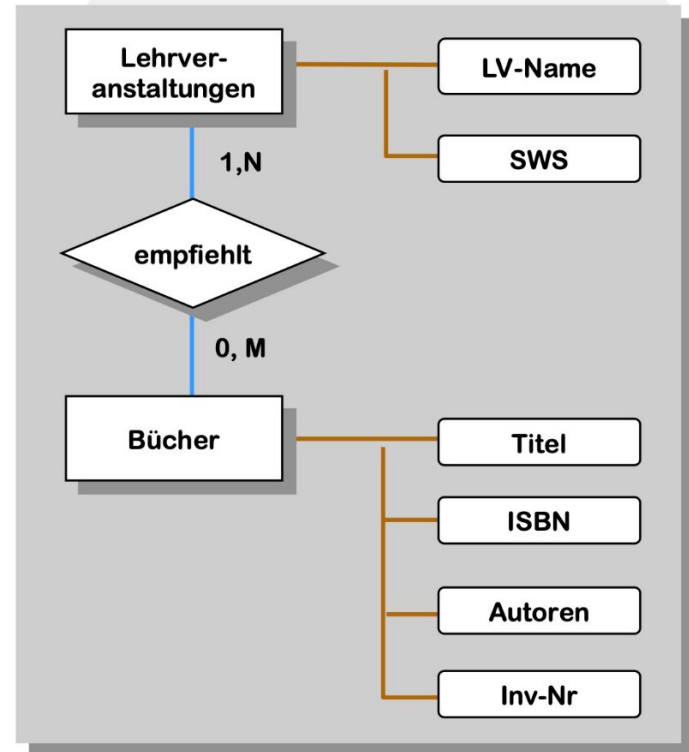
## Miniwelt (Diskursbereich) mittels

- ♦ freier Notation



## Semantisches (konzeptionelles) Schema mittels

- ♦ „Entity Relationship“-Datenmodell z.B. nach Abrial



# ER-Trainer

Mit dem ER-Trainer werden die Kardinalitäten von Beziehungen zwischen Entitäten im ER-Diagramm trainiert. Unterstützt werden neben binären Beziehungen auch ternäre Beziehungen mit bis zu vier Entitäten, (binäre) rekursive Beziehungen und Spezialisierung/Generalisierung mit bis zu drei Sub-Entitäten. Die Beziehungen werden primär in Abrial- bzw. (min-max)-Notation dargestellt, wobei für binäre Beziehungen auch zu anderen Notationen gewechselt werden kann.

Link: [https://github.com/EILD-nrw/er\\_trainer](https://github.com/EILD-nrw/er_trainer)

## ER-Trainer

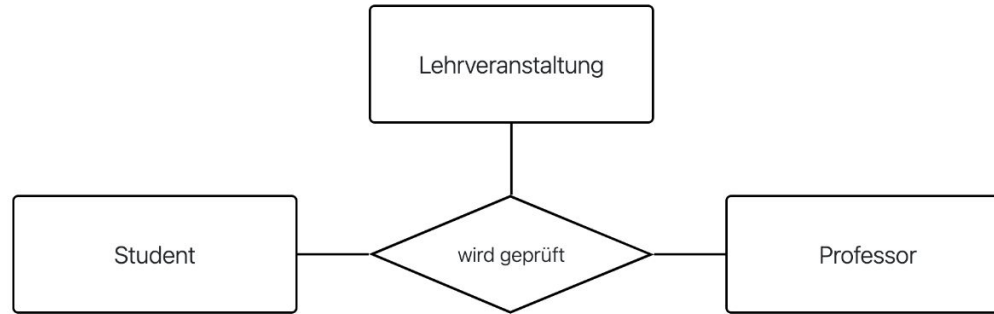
DE ▾

Bitte wählen Sie den zu der Phrase passenden Beziehungstyp in der Auswahlbox aus!

Notation: Abrial ▾

[Legende](#)

Phrase 20/47: In einer neuen Hochschule sollen nun Studenten Lehrveranstaltungen besuchen und am Ende des Semesters von einem Professor geprüft werden.



Student:

Bitte auswählen ▾

Lehrveranstaltung:

Bitte auswählen ▾

Professor:

Bitte auswählen ▾

Die Abrial bzw. (min,max)-Notation gibt für jeden an einer Beziehung beteiligten Entitätstyp an, mit wie vielen Entitäten auf der anderen Seite eine Entität dieses Typs mindestens und höchstens in Beziehung steht.

Abschicken

Korrigieren

Zeige Lösung

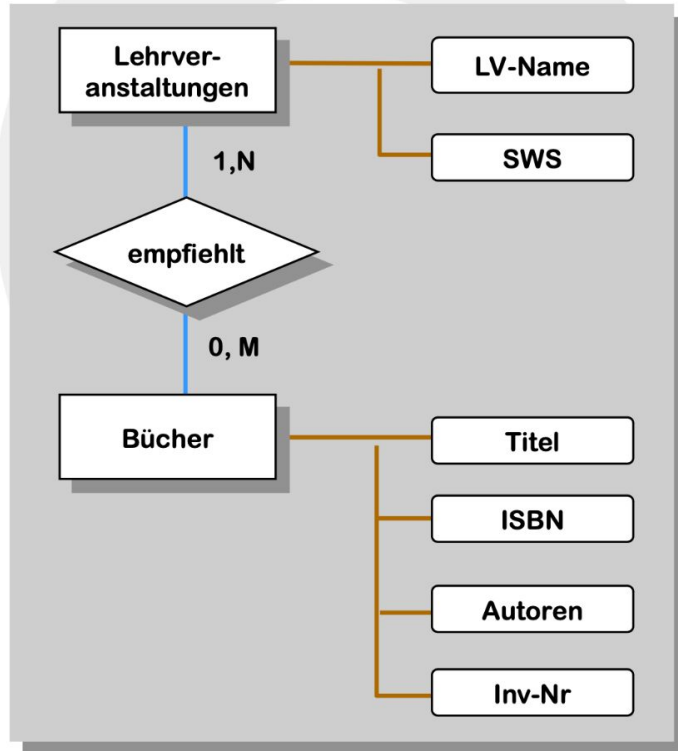
Überspringen

Neustart

# Phasen des Datenbankentwurf | Schritt 3: Logischer Entwurf

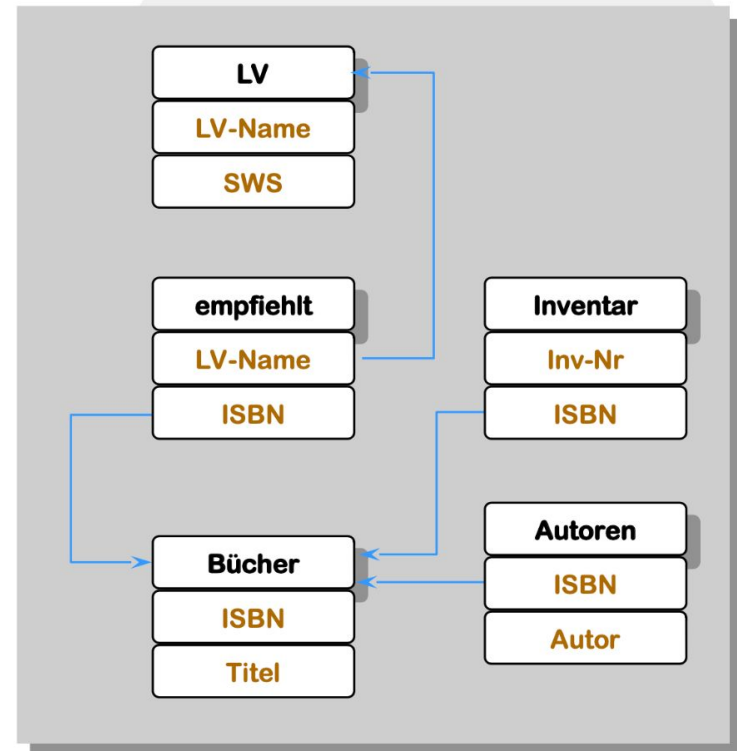
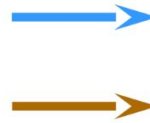
## Semantisches (konzeptionelles) Schema mittels

- „Entity Relationship“-Datenmodell z.B. nach Abrial



## Logisches Schema mittels

- Relationen-Datenmodell





# ER-REL-Trainer

Mit dem ER-REL-Trainer wird, ausgehend von einer Phrase, die Überführung einer Beziehung zwischen Entitäten eines ER-Diagramms in ein logisches relationales Schema trainiert. Es gilt die nötigen Tabellen anzulegen, die erforderlichen Schlüsselattribute zu ergänzen und die Richtung festzulegen, in der die Tabellen miteinander in Beziehung stehen.

Link: [https://github.com/EILD-nrw/er\\_rel\\_trainer](https://github.com/EILD-nrw/er_rel_trainer)

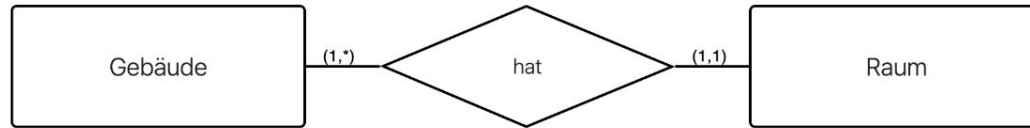
## ER-REL-Trainer

DE ▾

Gegeben ist ein ER-Diagramm, das eine Beziehung zwischen Entitäten zeigt. Ihre Aufgabe ist es das ER-Diagramm in ein logisches relationales Schema zu überführen und dafür die nötigen Tabellen anzulegen, darin die erforderlichen Schlüsselattribute zu ergänzen und die Richtung festzulegen, in der die Tabellen miteinander in Beziehung stehen.

ER-Notation: Abrial ▾ [Legende](#)

Phrase 12/50: Ein Architekt möchte wichtige Eckdaten zu den einzelnen individuellen Räumen seiner Gebäude verwalten.



+ Tabelle: "Gebäude"

+ Tabelle: "Raum"

+ Tabelle: "hat"

Hinweis: Über die Buttons unter dem ER-Diagramm können neue Tabellen angelegt werden. Entscheide, welche Tabellen benötigt werden.

Abschicken

Korrigieren

Zeige Lösung

Überspringen

Neustart

## Bindegliedfunktion

- ♦ **Datenbanksystemhersteller orientieren sich zur Definition und Manipulation der Daten in der Datenbank an bestimmten Datenmodellen (, halten sich aber nicht immer vollständig daran)**
- ♦ **logische Datenmodelle verstehen sich somit als Bindeglied und formale Rahmen für die Umsetzung eines semantischen Schemas in ein konkretes Datenbankschema**
- ♦ **logische Datenmodelle sind daher “idealisierte” Formen bestimmter Datenbankmodelle bestimmter Hersteller**
- ♦ **Verzicht auf systemspezifische Feinheiten, Beschränkung auf grundsätzliche Modellierungskonzepte bestimmter Zielsysteme**
- ♦ **derzeit werden von den kommerziell angebotenen Datenbanksystemen vier grundsätzlich unterschiedliche Stoßrichtungen unterstützt**

## Hierarchische Datenmodelle

- ♦ **älteste Datenmodell mit stark abnehmender Bedeutung aber noch im Betrieb**

## Netzwerk Datenmodelle

- ♦ **in den 70er Jahren stark favorisiert, aber heute nahezu bedeutungslos**

## Relationale Datenmodell

- ♦ **derzeit größte praktische Bedeutung**

## Objekt-orientierte Datenmodelle

- ♦ **wurde vielfach als Nachfolger der relationalen Datenmodelle gehandelt**
- ♦ **praktische Bedeutung für Anwendungen mit sehr speziellen Anforderungen der Modellierung**
- ♦ **erfolgreich sind objekt-relationale Datenmodelle**

## NoSQL Datenmodelle

- ♦ **oft schemalos**
- ♦ **starke Nähe zur Implementierung d. Anwendung**

# LOS-Trainer



# Schritte der Normalisierung

## Erste Normalform (1NF)

- ◆ keine Wiederholungsgruppen

## Zweite Normalform (2NF)

- ◆ 1NF
- ◆ Eliminierung von Attributen, die bereits von Teilen des Schlüssels abhängig sind

## Dritte Normalform (3NF)

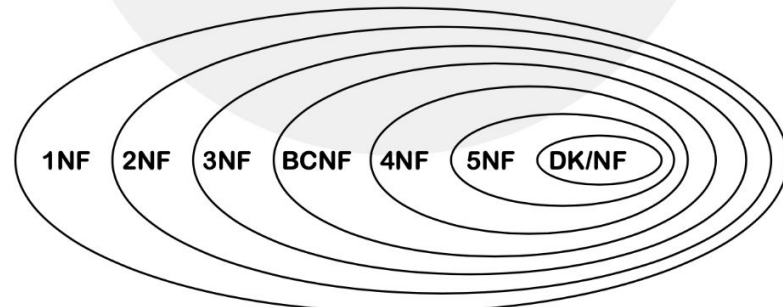
- ◆ 2NF
- ◆ Eliminierung von Attributen, die zusätzlich von nicht-Schlüsselattributen abhängig sind

## Boyce Codd Normalform (BCNF)

- ◆ 3NF
- ◆ zusätzlich Eliminierung von Abhängigkeiten zwischen Schlüsselattributen

## Praktische Bedeutung

- ◆ 3NF oder BCNF sollte bei jedem Datenbankentwurf hergestellt werden
- ◆ die vollständige Redundanzfreiheit wird mit der Domain / Key Normalform (DK/NF) erzielt
- ◆ 4NF, 5NF und DK/NF haben eher theoretische Bedeutung, da noch keine einfachen Techniken zur Unterstützung der Entwickler gefunden wurden
- ◆ häufig wird beim physischen Entwurf wieder denormalisiert (Laufzeitgründe)



Teilmengen von Relationen

# Normalisierungstrainer

Normalisierung

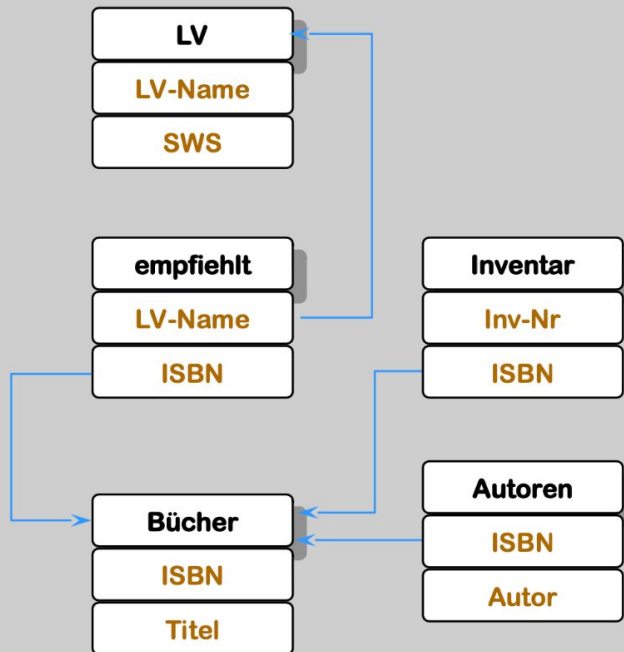
# Screenshot zum Normalisierungstrainer



# Phasen des Datenbankentwurf | Schritt 4: Physischer Entwurf + Ausführung

## Logisches Schema mittels

### ♦ Relationen Datenmodell



## Datenbankschema mittels

### ♦ DDL der SQL

```
create table Lehrveranstaltung (  
    LV-Name varchar(20) not null ,  
    SWS numeric(1) not null ,  
    primary key (Vorl-Titel));  
  
create table Buecher (  
    Titel varchar(20) not null ,  
    ISBN char(10) not null ,  
    primary key (ISBN));  
  
create table empfiehl (  
    LV-Name varchar(20) not null ,  
    ISBN char(10) not null ,  
    primary key (ISBN, Vorl-Titel));  
  
alter table empfiehl  
    add constraint RK_emp_Buc  
    foreign key (ISBN)  
    references Buecher;  
  
alter table empfiehl  
    add constraint RK_Vor_emp  
    foreign key (LV-Name)  
    references Lehrveranstaltung;  
  
...
```

# Serialisierungstrainer

Serialisierungstrainer



Weitere entwickelte Tools

# Interaktives ERD

## Screenshot zum interaktiven ERD

# Multiple Choice

## Screenshot zum Multiple Choice