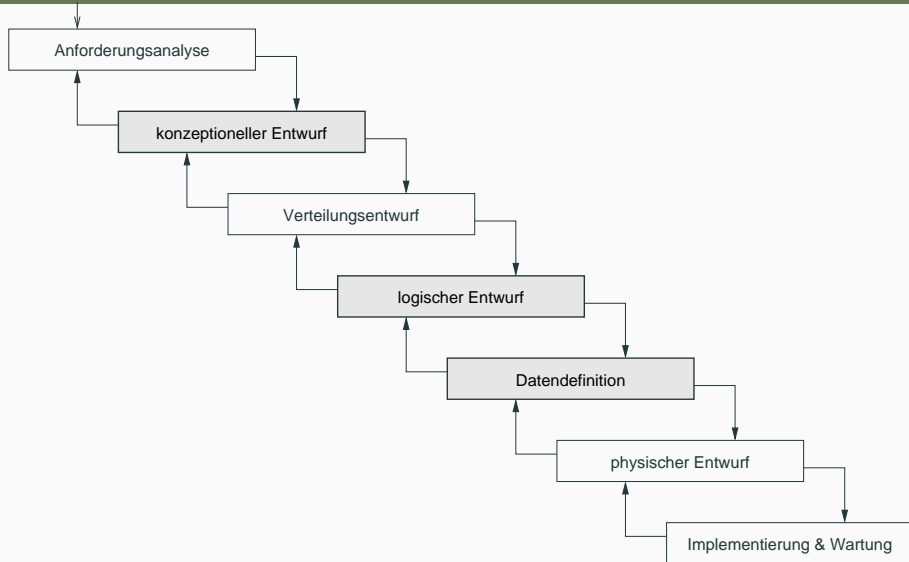


3. Datenbankentwurf

- Entwurfsaufgabe
- Phasenmodell
- Abbildung ER auf relationales Datenbankmodell

- Anforderungen an Entwurfsprozess
 - Informationserhalt
 - Konsistenzerhaltung
 - Redundanzfreiheit
 - Vollständigkeit bezüglich Anforderungsanalyse
 - Konsistenz des Beschreibungsdokuments
 - Ausdruckstärke, Verständlichkeit des benutzten Formalismus
 - formale Semantik der Beschreibungskonstrukte
 - Lesbarkeit der Dokumente
 - weitere Qualitätseigenschaften: Erweiterbarkeit, Modularisierung, Wiederverwendbarkeit, Werkzeugunterstützung etc.

3.2. Phasenmodell



- **Vorgehensweise:**
 - Sammlung des Informationsbedarfs in den Fachabteilungen
- **Ergebnis:**
 - informale Beschreibung (Texte, tabellarische Aufstellungen, Formblätter, usw.) des Fachproblems
 - Trennen der Information über Daten (Datenanalyse) von den Information über Funktionen (Funktionsanalyse)
- **„Klassischer“ DB-Entwurf:**
 - nur Datenanalyse und Folgeschritte
- **Funktionsentwurf:**
 - siehe Methoden des Software Engineering

- erste formale Beschreibung des Fachproblems
Sprachmittel: semantisches Datenmodell, z.B. erweitertes ER-Modell
- *Vorgehensweise:*
 - Modellierung von Sichten z.B. für verschiedene Fachabteilungen
 - Analyse der vorliegenden Sichten in Bezug auf Konflikte
 - Integration der Sichten in ein Gesamtschema
- *Ergebnis:* konzeptionelles Gesamtschema, z.B. (E)ER-Diagramm

- **Namenskonflikte:** Homonyme / Synonyme
 - Homonyme: Schloss; Kunde
 - Synonyme: Auto, KFZ, Fahrzeug
- **Typkonflikte:**
verschiedene Strukturen für das gleiche Element
- **Wertebereichskonflikte:**
verschiedene Wertebereiche für ein Element
- **Bedingungskonflikte:**
z.B. verschiedene Schlüssel für ein Element
- **Strukturkonflikte:** gleicher Sachverhalt durch unterschiedliche Konstrukte ausgedrückt

- sollen Daten auf mehreren Rechnern verteilt vorliegen, muss Art und Weise der **verteilten Speicherung** festgelegt werden

- z.B. bei einer Relation

KUNDE (KNr, Name, Adresse, PLZ, Konto)

- **horizontale** Verteilung:

KUNDE_1 (KNr, Name, Adresse, PLZ, Konto)

where PLZ < 50.000

KUNDE_2 (KNr, Name, Adresse, PLZ, Konto)

where PLZ >= 50.000

- **vertikale** Verteilung (Verbindung über *KNr* Attribut):

KUNDE_Adr (KNr, Name, Adresse, PLZ)

KUNDE_Konto (KNr, Konto)

- *Sprachmittel*: Datenmodell des ausgewählten „Realisierungs“-DBMS, z.B. relationales Modell
- *Vorgehensweise*:
 1. (automatische) Transformation des konzeptionellen Schemas z.B. ER \rightarrow relationales Modell
 2. Verbesserung des relationalen Schemas anhand von Gütekriterien (\leadsto Normalisierung):
Entwurfsziele: Redundanzvermeidung, ...
- *Ergebnis*: logisches Schema, z.B. Sammlung von Relationenschemata

- Umsetzung des logischen Schemas in ein konkretes Schema
- *Sprachmittel*: DDL und DML¹ eines DBMS
(z.B. Ingres, Oracle, DB2, ...)
 - Datenbankdeklaration in der DDL des DBMS
 - Realisierung der Integritätssicherung
 - *Definition der Benutzersichten*

¹DDL – Data Definition Language; DML – Data Manipulation Language

- Ergänzen der Datendefinition um Zugriffsunterstützung zur Effizienzverbesserung, z.B. Definition von Indexen
- *Sprachmittel: Speicherstruktursprache SSL*

Phasen

- der Wartung,
- der weiteren Optimierung der physischen Ebene,
- der Anpassung an neue Anforderungen und Systemplattformen,
- der Portierung auf neue Datenbank-Management-Systeme
- etc.

- Integration von Funktions- und Strukturbeschreibung in Objektbeschreibungen
 - Strukturbeschreibung analog OODM
 - abstrakte *Ereignisse* / *Methoden* zur Funktions- / Verhaltensmodellierung

- Validationsmethoden:

Verifikation: Der formale Beweis etwa von Schemaeigenschaften

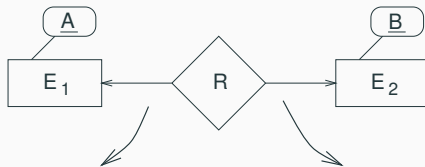
Prototyping: beispielhaftes Arbeiten mit der Datenbank vor der endgültigen Implementierung

Validation mit Testdaten: Überprüfung der Richtigkeit des Entwurfs anhand von realen oder künstlichen Testdaten

- erster Teilschritt des logischen Datenbankentwurfs
- Abbildung von ER-Modell auf
 - Relationenmodell
- Vorgehensweisen:
 - Transformation nach Faustregeln manuell
 - automatische Transformation

Ziel: **kapazitätserhaltende Abbildung**

Kapazitätserhöhende Abbildung



$R = \{A, B\}$

$K = \{\{A\}\}$

A	B
1	3
2	3



A	B
1	3
2	4



Kapazitätserhöhend

$R = \{A, B\}$

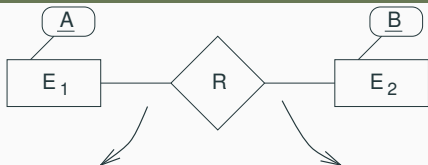
$K = \{\{A\}, \{B\}\}$

A	B
1	3
2	4



Kapazitätserhaltend

Kapazitätsvermindernde Abbildung



$R = \{ A, B \}$

$K = \{ \{ A \} \}$

A	B
1	3
2	3

✓



Kapazitätsvermindernd

$R = \{ A, B \}$

$K = \{ \{ A, B \} \}$

A	B
1	3
2	4

✓

A	B
2	3
2	4
3	4

Kapazitätserhaltend

1. Entity-Typen und Beziehungstypen
→ Relationenschemata
 - Attribute → Attribute des Relationenschemas
 - Schlüssel werden übernommen
2. Kardinalitäten der Beziehungen → Wahl der Schlüssel
3. Relationenschemata von Entity- und Beziehungstypen können eventuell miteinander verschmolzen werden

1. Abbildung von Entity-Typen

- Entity-Typ
 - Relationenschema mit allen Attributen des Entity-Typs
- mehrere Schlüssel vorhanden
 - Auswahl eines Primärschlüssels
- Primärschlüssel wird unterstrichen
- Sonderfälle: Abhängige/Schwache Entity-Typen und Entity-Typen in einer IST-Beziehung

2. Abbildung von Beziehungstypen

- Beziehungstyp \rightarrow Relationenschema mit allen Attributen des Beziehungstyps + Primärschlüssel der beteiligten Entity-Typen (als Fremdschlüssel)
- Auswahl der Schlüssel (hier für binäre Beziehungen)
 - m:n-Beziehung (nicht-funktional!): Beide Primärschlüssel werden gemeinsam Schlüssel
 - 1:n-Beziehung (funktionale Bez. $E_2 \rightarrow E_1$): Der Primärschlüssel der n-Seite ($[0, 1]$ - bzw. $[1, 1]$ -Seite) wird Schlüssel
 - 1:1-Beziehung (wechselseitig funktional): Beide Primärschlüssel werden je ein Schlüssel
- Aus den möglichen Schlüsseln wird ein Primärschlüssel gewählt
- Notation:
 - Primärschlüssel wird unterstrichen
 - Fremdschlüssel werden überstrichen

3. Verschmelzen von Relationenschemata

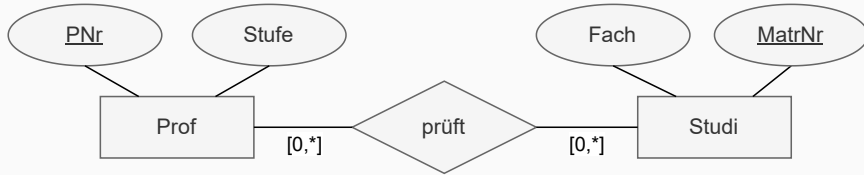
bei zwingenden Beziehungen mit $[1,1]$ -Kardinalität

- 1:n- oder 1:1-Beziehung mit einer $[1,1]$ -Kardinalität:
das Entity-Relationenschema der $[1,1]$ -Seite und das Relationenschema der Beziehung werden verschmolzen
- 1:1-Beziehung mit zwei $[1,1]$ -Kardinalitäten:
beide Entity-Relationenschemata werden mit dem Relationenschema der Beziehung verschmolzen

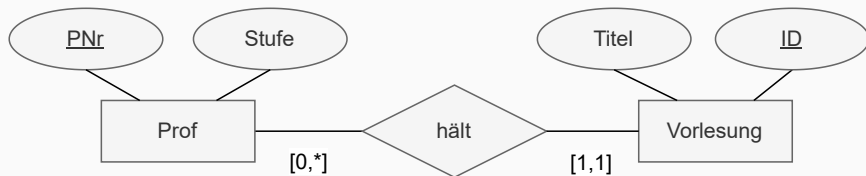
Abbildung ER-Schema nach RDM (Zusammenfassung)

ER-Konzept	wird abgebildet auf relationales Konzept
Entity-Typ E_i Attribute von E_i Primärschlüssel P_i	Relationenschema R_i Attribute von R_i Primärschlüssel P_i
Beziehungstyp dessen Attribute $1 : n$ $1 : 1$ $m : n$	Relationenschema Attribute: P_1, P_2 (der beteiligten Entitytypen) weitere Attribute P_2 wird Primärschlüssel der Beziehung P_1 und P_2 werden Schlüssel der Beziehung $P_1 \cup P_2$ wird Primärschlüssel der Beziehung
IST-Beziehung	R_1 erhält zusätzlichen Schlüssel P_2

E_1, E_2 : an Beziehung beteiligte Entity-Typen, P_1, P_2 : deren Primärschlüssel,
 $1 : n$ -Beziehung: E_2 ist n -Seite; IST-Beziehung: E_1 ist speziellerer Entity-Typ
 $1 : n$ funktionale Bez. $E_2 \rightarrow E_1$; $1 : 1$ wechsseitig funktional; $n : m$ nicht funktional



- Prof(PNr, Stufe)
- Studi(MatrNr, Fach)
- prüft(PNr, MatrNr)



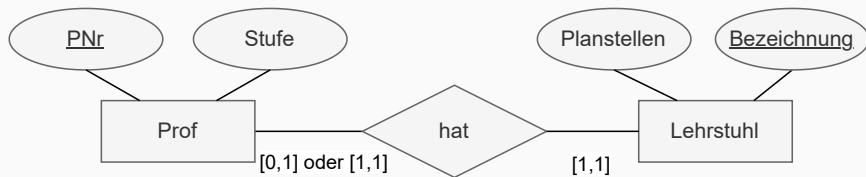
Ohne Verschmelzungen:

- Prof(PNr, *Stufe*)
- Vorlesung(*ID*, Titel)
- hält(ID, PNr)

Mit Verschmelzung von *Vorlesung* und *hält*:

- Vorlesung(ID, *Titel*, PNr)

1:1-Beziehung



Ohne Verschmelzungen:

- Prof(PNr, Stufe)
- Lehrstuhl(Bezeichnung, Planstellen)
- hat(Bezeichnung, PNr) oder
hat(Bezeichnung, PNr)

Mit Verschmelzung von *Lehrstuhl* und *hat*:

- Lehrstuhl(Bezeichnung, Planstellen, PNr)

→ sowohl *PNr* als auch *Bezeichnung* sind Schlüssel der Beziehung, einer wird als Primärschlüssel der Relation gewählt

Auswirkung von [1,1]-Kardinalitäten

[1,1]:[1,1]-Beziehung

Prof	PNr	Lehrstuhlbezeichnung	Stufe	Planstellen
	4711	Datenbanksysteme	W3	3
	5588	Rechnernetze	W3	4

→ Verschmelzung von *Prof*, *hat* und *Lehrstuhl*

Prof(PNr, Stufe, Lehrstuhlbezeichnung, Planstellen)

[0,1]:[1,1]-Beziehung:

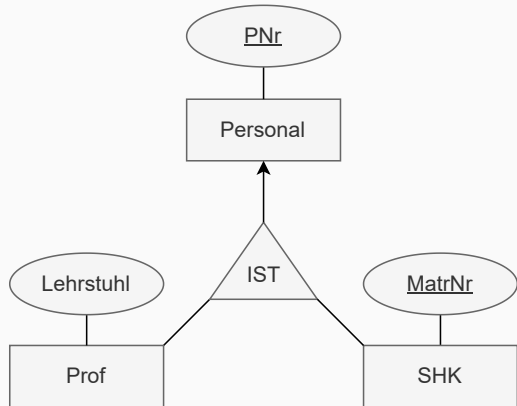
Prof	PNr	Lehrstuhlbezeichnung	Stufe	Planstellen
	4711	Datenbanksysteme	W3	3
	5588	Rechnernetze	W3	4
	⊥	Bioinformatik	⊥	2

Lehrstühle können unbesetzt bleiben

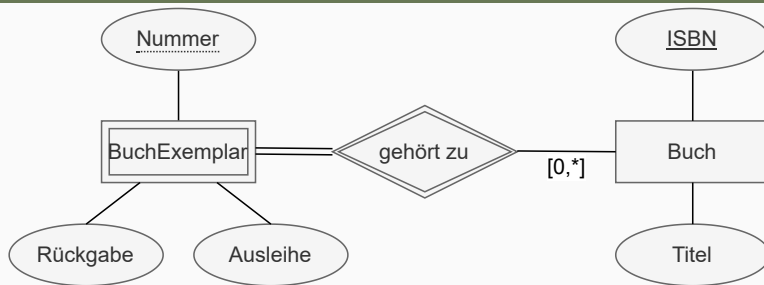
dann besser zwei Relationenschemata → keine Verschmelzung mit *Prof*

- IST-Beziehung hat kein eigenes Relationenschema
- im Relationenschema des unteren Entity-Typs zusätzlich der Primärschlüssel des oberen Entity-Typs als Fremdschlüssel-Attribut

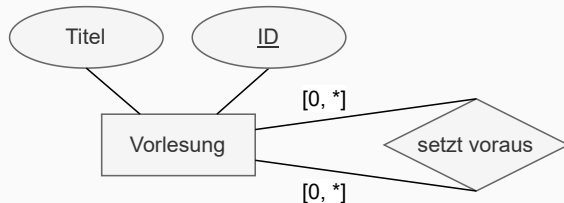
- Obertyp: Personal(PNr)
- Untertyp: Prof(PNr, Lehrstuhl)
- Untertyp: SHK(PNr, MatrNr)
oder SHK(MatrnNr, PNr)
→ Wahl zwischen „lokalem“ Schlüssel und geerbtem Schlüssel



Abhängige Entity-Typen



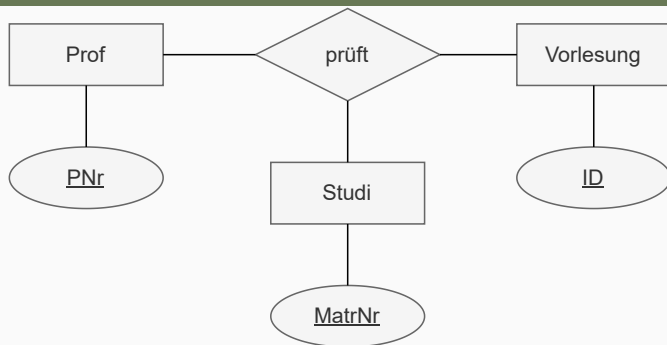
- **Buch**(ISBN,Titel)
- Abhängiger Entity-Typ erhält Primärschlüssel des identifizierenden Entity-Typs als Fremdschlüssel-Attribut.
→ wird zusammen mit partiellem Schlüssel der Primärschlüssel der Relation
- **BuchExemplar**(ISBN,Nummer, Rückgabe, Ausleihe)



Umbenennung der übernommenen Primärschlüssel

- Vorlesung(ID, Titel)
- z.B. setztVoraus(VorgängerID, NachfolgerID)

Mehrstellige Beziehungen



- prüft(PNr, ID, MatrNr)
- Primärschlüssel der Relation abhängig davon, welcher Beziehungstyp vorliegt (1:1:1, 1:1:n, 1:n:m oder n:m:k)
→ Übungsaufgabe