



(Bisherige) Gliederung der Vorlesung

0. Vorbemerkungen
1. Einführung in Datenbanken
2. Datenbankmodelle für den Entwurf
3. Datenbankmodelle für die Realisierung
4. Grundlagen von Anfragen und Änderungen
 - 4.1 Kriterien für Anfragesprachen
 - 4.2 Relationenalgebra
 - 4.3 Anfragekalküle
5. Datenbanksprache SQL
6. Datenbankentwurfsprozess



6.1. Entwurfsaufgabe

Anforderungen an Entwurfsprozess

- Informationserhalt
- Konsistenzerhaltung
- Redundanzfreiheit
- Vollständigkeit bezüglich Anforderungsanalyse
- Konsistenz des Beschreibungsdokuments
- Ausdrucksstärke, Verständlichkeit des benutzten Formalismus
- Formale Semantik der Beschreibungskonstrukte
- Lesbarkeit der Dokumente
- Weitere Qualitätseigenschaften:

Unterstützung von Erweiterbarkeit, Modularisierung, Wiederverwendbarkeit, Werkzeugunterstützung, etc.



Phasen des Datenbankentwurfs

- Anforderungsanalyse
- Konzeptioneller Entwurf
- Verteilungsentwurf
- Logischer Entwurf
- Datendefinition
- Physischer Entwurf
- Implementierung und Wartung



Anforderungsanalyse

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 Informationen über Funktionen (Funktionsanalyse)

“Klassischer“ DB-Entwurf:

nur Datenanalyse und Folgeschritte.

Funktionsentwurf

siehe Methoden des Software Engineering



Konzeptioneller Entwurf

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

- **Namenskonflikte** (Homonyme / Synonyme)
 - * Homonyme: Schloß; Kunde
 - * Synonyme: Auto, Kfz, Fahrzeug
- **Typkonflikte**
(verschiedene Strukturen für das gleiche Element)
- **Wertebereichskonflikte**
(verschiedene Wertebereiche für ein Element)



Konzeptioneller Entwurf

- **Bedingungskonflikte**
(z. B. verschiedene Schlüssel für ein Element)
- **Strukturkonflikte**
(gleicher Sachverhalt durch unterschiedliche Konstrukte ausgedrückt)
- Integration der Sichten in ein **Gesamtschema**

Ergebnis: konzeptionelles Gesamtschema, z. B. ER-Diagramm



Verteilungsentwurf

Sollen die 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, Kontostand)

- **horizontale Verteilung:**

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

where PLZ < 50.000

und

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

where PLZ >= 50.000

- **vertikale Verteilung:**

KUNDE_Adr (KNr, Name, Adresse, PLZ)

und

KUNDE_Konto (KNr, Kontostand)

(Verbindung über KNr Attribut)



Logischer Entwurf

Sprachmittel:

Datenmodell des ausgewählten ‘Realisierungs’-DBMS
z. B. relationales Modell

Vorgehensweise:

1. (automatische) Transformation des konzeptionellen Schemas
z. B. ER → relationales Modell
(siehe Abschnitt 6.3)
2. Verbesserung des relationalen Schemas anhand von Gütekriterien
(Normalisierung, siehe Kapitel 7):
Entwurfsziele: Redundanzvermeidung, ...

Ergebnis:

Logisches Schema, z. B. Sammlung von Relationenschemata



Datendefinition

Umsetzung des logischen Schemas in ein konkretes Schema

Sprachmittel:

DDL eines DBMS z. B. Ingres, Oracle, Sybase, Informix

- Datenbankdeklaration in der DDL des DBMS
- Realisierung der Integritätssicherung
- *Definition der Benutzersichten (create view)*



Physischer Entwurf

Ergänzen des physischen Entwurfs um Zugriffsunterstützung bzgl. Effizienzverbesserung, z. B. Definition von Indexen (Indexes)

Sprachmittel: Speicherstruktursprache SSL



Implementierung und Wartung

Phasen

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



6.3. ER-Abbildung auf andere Datenbankmodelle

Erster Teilschritt des logischen Datenbankentwurfs

Abbildung von ER-Modell auf

- Relationenmodell
- Netzwerkmodell oder hierarchisches Modell (*hier nicht behandelt*)

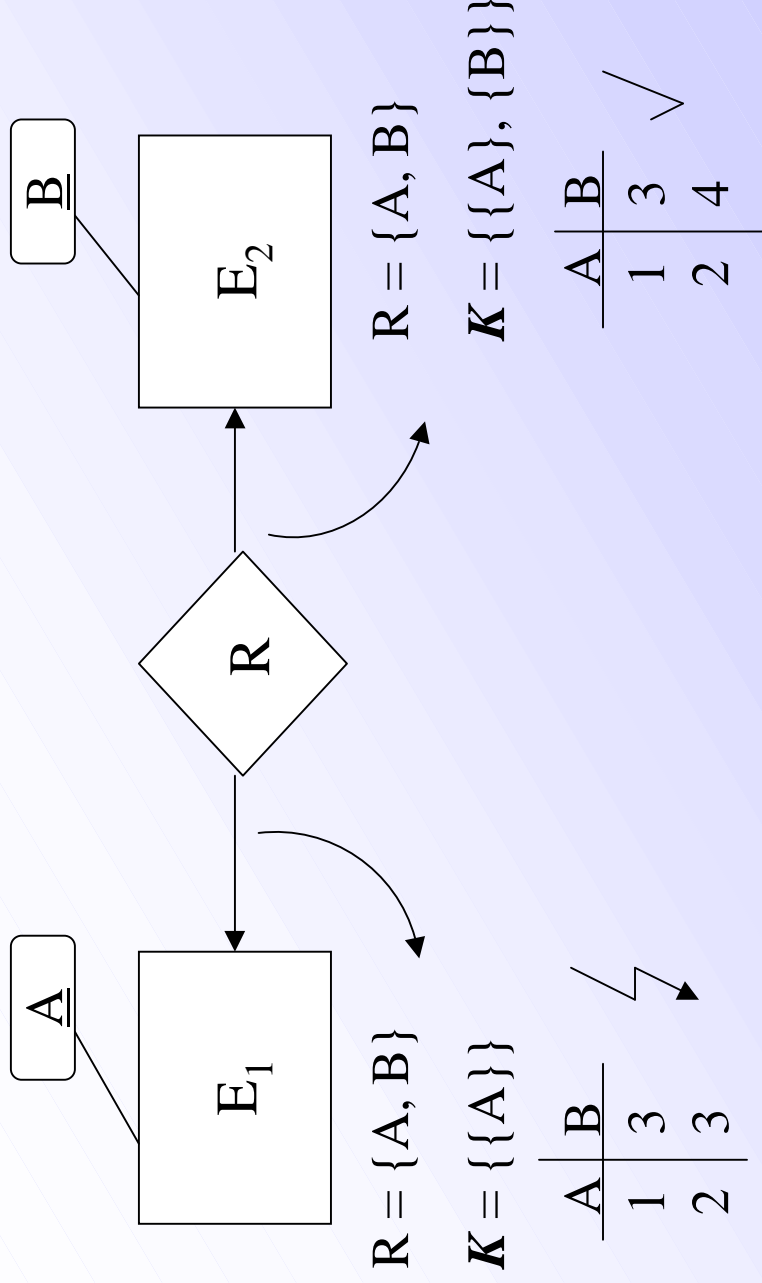
Vorgehensweisen:

- Transformation nach Faustregeln manuell
- automatische Transformation

Ziel: kapazitätserhaltende Abbildung



Kapazitätserhöhende Abbildung von ER-Diagrammen auf relationale Schemata

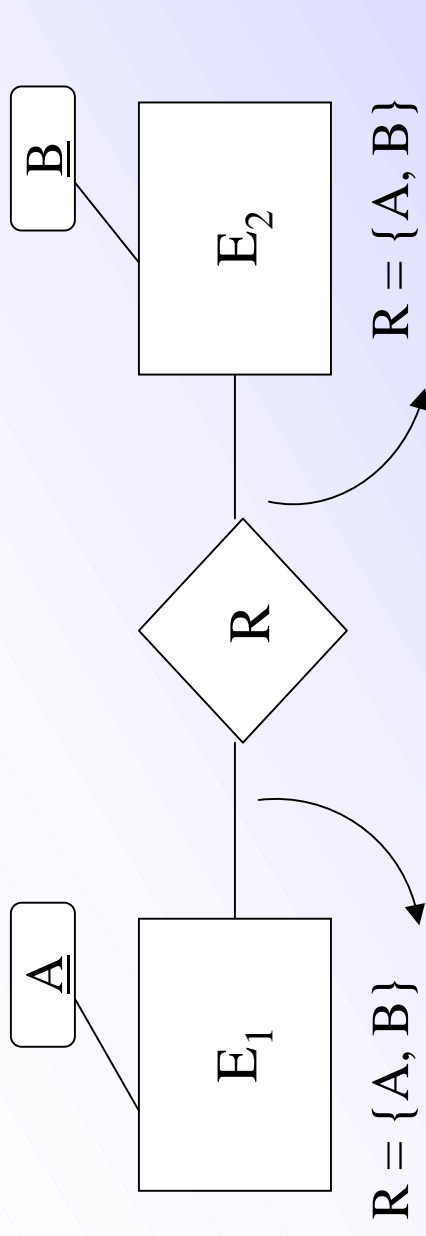


Kapazitätserhöhend

Kapazitätserhaltend



Kapazitätsvermindernde Abbildung von ER-Diagrammen auf relationale Schemata



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

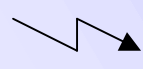
A	B
1	3
2	3

✓

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

A	B
1	3
2	4

✓



Kapazitätsvermindernd

Kapazitätserhaltend



Abbildung auf das relationale Modell

- Entity-Typen und Beziehungstypen → Relationenschemata
 - Attribute → Attribute des Relationenschemas
 - Schlüssel werden übernommen
- Kardinalitäten der Beziehungen → Wahl der Schlüssel
- Relationenschemata von Entity- und Beziehungstypen können eventuell miteinander verschmolzen werden
- Einführung diverser Fremdschlüsselbedingungen



Abbildung ER-Diagramm nach relationalen Schemata

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 IST-Beziehung	Relationenschema Attribute: P_1, P_2 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 R_1 erhält zusätzlichen Schlüssel P_2

Bezeichnungen:

E_1, E_2 : an Beziehung beteiligte Entity-Typen

P_1, P_2 : deren Primärschlüssel

bei 1 : n -Beziehung: E_2 ist die n -Seite

bei IST-Beziehung: E_1 ist speziellerer Entity-Typ



Abbildung von Entity-Typen

- Entity-Typ → Relationenschema mit allen Attributen des Entity-Typs
- mehrere Schlüssel vorhanden → Auswahl eines Primärschlüssels



Abbildung von Beziehungstypen

- Beziehungstyp → Relationenschema mit allen Attributen des Beziehungstyps + Primärschlüssel der beteiligten Entity-Typen
- Auswahl der Schlüssel (hier für binäre Beziehungen)
 - m:n-Beziehung: Beide Primärschlüssel werden Schlüssel
 - 1:n-Beziehung: Der Primärschlüssel der n-Seite (bei der funktionalen Notation die Seite ohne Pfeilspitze) wird Schlüssel
 - 1:1-Beziehung: Beide Primärschlüssel werden je ein Schlüssel, einer wird Primärschlüssel

Dies gilt bei optionalen Beziehungen ([0, _])

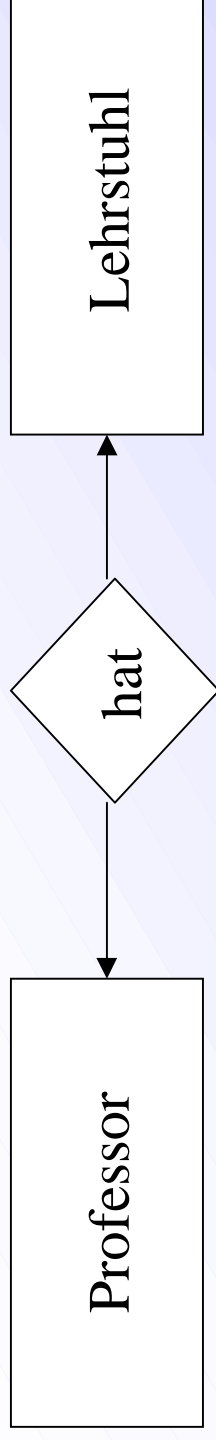


Verschmelzen von Relationenschemata

Bei zwingenden Beziehungen ([1, _])

- 1:n-Beziehung: das Entity-Relationenschema der n-Seite kann in das Relationenschema der Beziehung integriert werden
- 1:1-Beziehung: beide Entity-Relationenschemata können in das Relationenschema der Beziehung integriert werden

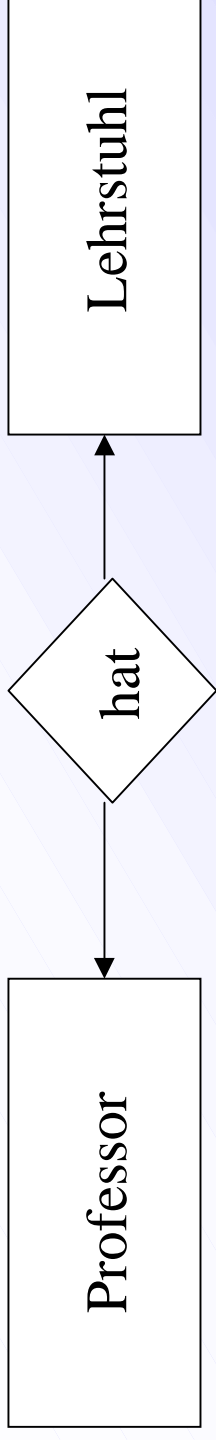
1:1 - Beziehung



Ergebnis der Transformation: 3 Relationenschemata

- **Professoren** mit den Attributen PANr und Stufe
- **Lehrstühle** mit den beiden Attributen Lehrstuhlbezeichnung und Anzahl_Planstellen
- **Hat_Lehrstuhl** mit den Primärschlüsseln der beiden beteiligten Entity-Typen jeweils als Schlüssel dieses Schemas, also PANr und Lehrstuhlbezeichnung.

1:1 - Beziehung



Ergebnis der Transformation: 3 Relationenschemata

- **Professoren** (PANr, Stufe)
- **Lehrstühle** (Lehrstuhlbezeichnung, Anzahl_Planstellen)
- **Hat_Lehrstuhl** (PANr, Lehrstuhlbezeichnung)

Hinzu kommen Fremdschlüsselbedingungen:

- **Hat_Lehrstuhl** (PANr) → **Professoren** (PANr)
- **Hat_Lehrstuhl** (Lehrstuhlbezeichnung) → **Lehrstühle** (Lehrstuhlbezeichnung)



1:1 - Beziehung: Auswirkung von [1,1] - Kardinalitäten

auch denkbar alles in einem Relationenschemata

Professoren (PANr, Stufe, Lehrstuhlbezeichnung, Anzahl_Planstellen)

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

Professoren

PANr	Lehrstuhlbezeichnung	Stufe	Anzahl_Planstellen
4711	Datenbank- und Informationssysteme	C4	4
5588	Datenbanken und Informationssysteme	C4	5

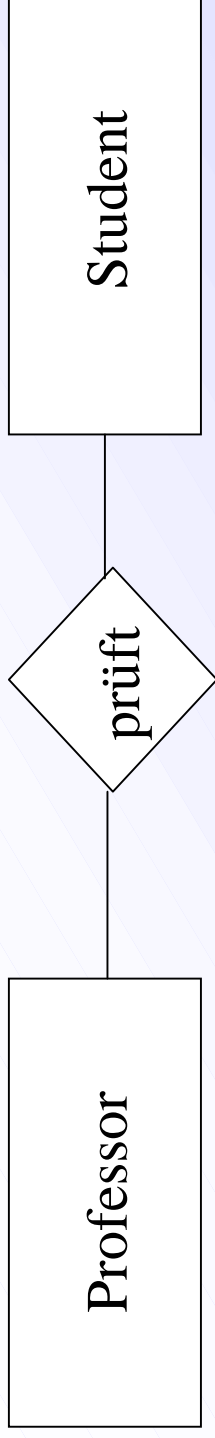
[0,1]:[1,1]-Beziehung: Lehrstühle können unbesetzt bleiben

Professoren

PANr	Lehrstuhlbezeichnung	Stufe	Anzahl_Planstellen
4711	Datenbank- und Informationssysteme	C4	4
5588	Datenbanken und Informationssysteme	C4	5
⊥	Rechnernetze	⊥	2

dann besser zwei Relationenschemata

n:m-Beziehung

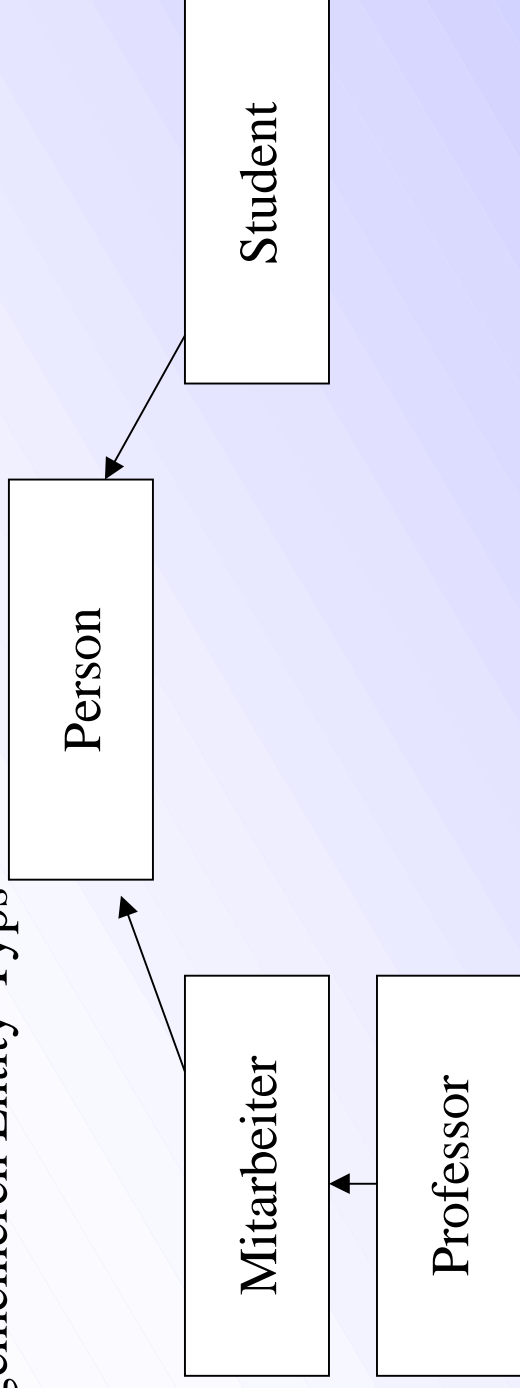


- Professoren mit den Attributen PANr und Stufe
- Studenten unter anderem mit den Attributen Matrikelnummer und Studienfach
- Prüft mit den Primärschlüsseln der beteiligten Entity-Typen zusammen als Primärschlüssel dieses Schemas, also {PANr, Matrikelnummer}

Fremdschlüssel?

IST-Beziehung

- kein eigenes Relationenschema für den Beziehungstyp
- im Relationenschema des spezielleren Entity-Typs zusätzlich der Primärschlüssel des allgemeineren Entity-Typs



- Mitarbeiter mit AngNr als Schlüssel. Zusätzlich Primärschlüssel PANr von Personen geerbt. Entscheidung für PANr als Primärschlüssel
- Professoren: PANr wird von Mitarbeiter vererbt
- Studenten mit Attribut Matrikelnummer (Schlüssel). Auswahl zwischen „lokalem“ Schlüssel und geerbtem Schlüssel PANr