
MOBI-DBS-B: Datenbanksysteme Das Relationale Datenmodell

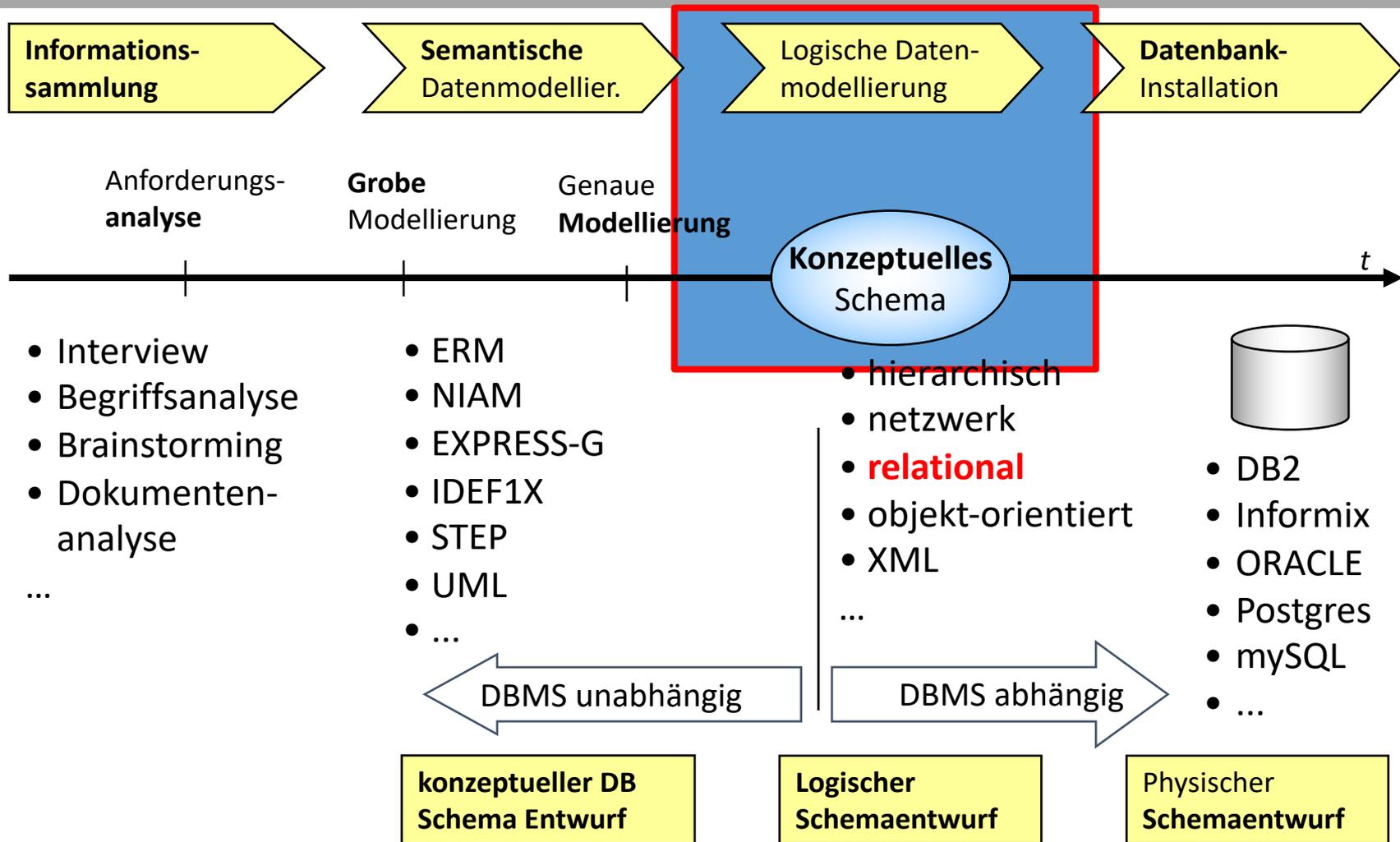
Vorlesung Sommersemester 2019

Tanya Braun, Universität zu Lübeck

Lehrauftrag SoSe 19, Universität Bamberg



Die Phasen des DB-Entwurfs



Das Relationale Datenmodell

Inhalte

- Relationales Datenmodell
 - DB = Sammlung von Relationen
 - Relationen
 - Relationale Datenbanken und -schemata
 - Referentielle Integrität
- Relationale Algebra
 - Insert, delete, update
 - $\pi, \rho, \sigma, \cup, \cap, -, \times, \bowtie$
 - Minimalität
 - Aggregieren, gruppieren
- Vom ER-Modell zum RM

Kompetenzen

- ER-Modelle in ein relationales Schema überführen
- Mengenorientierte Verarbeitung verstehen und anwenden
- Unterschied deklarative und prozedurale Sprachen verstehen
- Publikationen aus der Forschung verstehen

Bezug zu Phasen des DB-Entwurfs

Das relationale Datenmodell

Logische Modellierung

Ein erstes relationales Modell (RM)

- DB einer Universität: Tabellen
 - Studierende
 - Kurse
 - Arbeitsgruppen
 - Noten
 - Voraussetzungen
- Datenelemente von unterschiedlichem Typ
 - String, Integer, etc.
 - Schlüssel
- Logische Zusammenhänge
 - Innerhalb einer Tabelle
 - Zwischen Tabellen
 - Schlüssel!

STUDENT

Name	StudentNumber	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor
85	MATH2410	Fall	18	King
92	CS1310	Fall	18	Anderson
102	CS3320	Spring	19	Knuth
112	MATH2410	Fall	19	Chang
119	CS1310	Fall	19	Anderson
135	CS3380	Fall	19	Stone

GRADE_REPORT

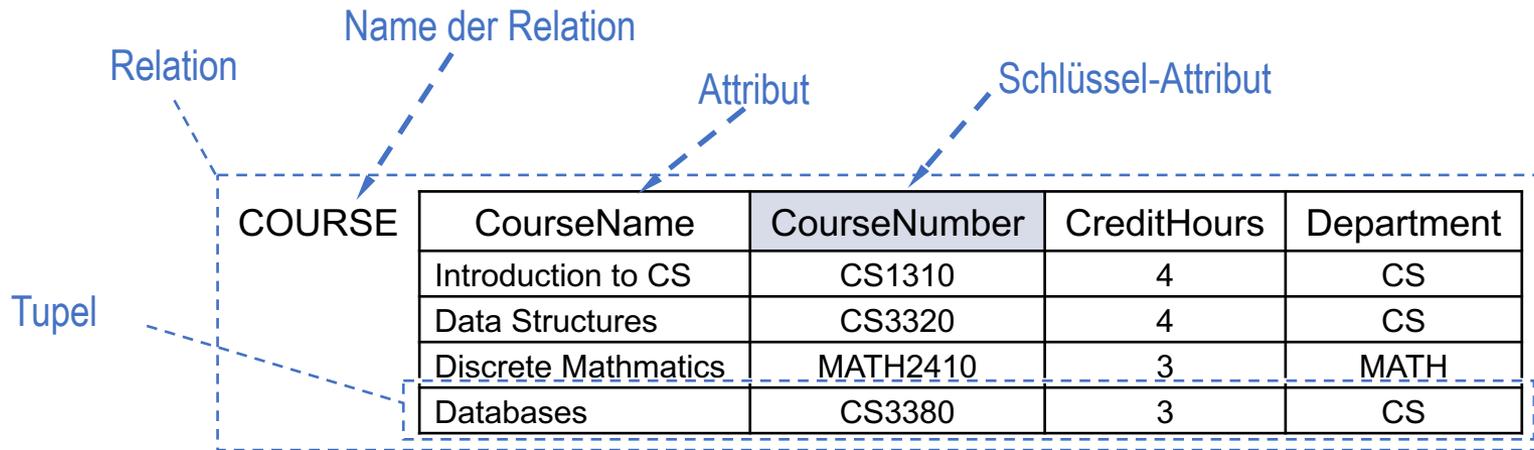
StudentNumber	SectionIdentifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

CourseNumber	PrerequisiteNumber
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Relationen

- Relation, Name der Relation, Attribut und Tupel:



- Namen der Relation und Attribute:
 - Frei wählbar; sinnvoll sind sprechende Bezeichner
- Wertebereich / Domain eines Attributs
 - Datentyp, der die Wertetypen der Attribute beschreibt

Wertebereiche/Domänen von Attributen

- Wertebereich **D** : eine Menge atomarer Werte
 - Jeder einzelne Wert ist aus Sicht des relationalen Modells unteilbar
- Festlegung: Angabe eines Datentyps, dessen Datenwerte den Wertebereich bilden.
 - Angabe eines Namens vereinfacht die Interpretation der Werte
 - Angabe eines Datenformats konkretisiert die Darstellung
 - Ggf. Angabe einer Maßeinheit
- Beispiel

USA_Phone_Numbers:

Der Datentyp ist die Menge der zehnstelligen Telefonnummern, die in den USA gültig sind.

Das Datenformat für USA_Phone_Numbers kann als **Zeichenkette im Format ddd-dd-dddd** deklariert werden, wobei jedes einzelne **d** für eine Ziffer (0|1|2|3|4|5|6|7|8|9) steht

Typische Datentypen für Wertebereiche

- Numerische Standardtypen, z.B.
 - short-integer
 - integer
 - long-integer
 - float
 - double-float
- Zeichenketten, z.B.
 - string
 - char(255)
- Spezielle Datentypen, z.B.
 - date
 - timestamp
- Benutzerdefinierte Datentypen, z.B.
 - PID
 - SSN

Relationenschemata und Relationen

- **Relationenschema** $R(A_1, \dots, A_n)$:
 - (DB-weit eindeutiger) Relationenname R
 - Liste von Attributen A_1, \dots, A_n
 - Attribut A_i :
 - Name einer Rolle, die ein Wertebereich im Relationenschema R spielt
 - Wertebereich D von A_i : $\text{dom}(A_i) = D$
- **Relation** r wird über Relationenschema R identifiziert
 - „Intension“ \rightarrow Relationenschema R
 - „Extension“ \rightarrow „aktuelle“ Relation $r(R)$ oder r
- **Grad** (Degree) eines Relationenschemas R bzw. einer Relation r :
 - die Anzahl n von Attributen, die R bzw. r bilden

Relationenschema R

Relation r

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Grad 4

Relationenschemata und Relationen - Beispiel

- Relationenschema

- COURSE(CourseName, CourseNumber, CreditHours, Department)

- Wertebereiche für Attribute (hier alle benutzerdefiniert)

- $\text{dom}(\text{CourseName}) = \text{CourseNames}$
- $\text{dom}(\text{CourseNumber}) = \text{CourseNumbers}$
- $\text{dom}(\text{CreditHours}) = \text{CreditHours}$
- $\text{dom}(\text{Department}) = \text{Departments}$

- Allgemeinere Alternativen: String, String, Integer, String

Attribute

Relationenschema R

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Relationen/Relationenzustände

- Eine Relation (oder ein **Relationenzustand**) r bzw. $r(R)$ eines Relationenschemas $R(A_1, \dots, A_n)$ ist eine Menge von n -Tupeln t_j mit $j=1, \dots, m$:
 - $r = \{t_1, \dots, t_m\}$.
 - $m =$ **Kardinalität** von r (Anzahl an Tupeln/Zeilen in r)
- Jedes n -Tupel t_j ist eine geordnete Liste von n Werten:
 - $t_j = \langle v_1, \dots, v_n \rangle$
- Jeder Wert v_i :
 - entweder ein Element aus $\text{dom}(A_i)$
 - oder **NULL** (spezieller Wert, bedeutet „undefiniert“ oder „unbekannt“)

$m = 4$

COURSE	CourseName	CourseNumber	CreditHours	Department
t_1	Introduction to CS	CS1310	4	CS
t_2	Data Structures	CS3320	4	CS
t_3	Discrete Mathematics	MATH2410	3	MATH
t_4	Databases	CS3380	3	CS

Relationen/Qualifizierung

- Gegeben
 - $R(A_1, \dots, A_n)$ ist ein Relationenschema n-ten Grades
 - $t = \langle v_1, \dots, v_n \rangle$: ist ein Tupel der Relation $r(R)$
 - v_i ist der Wert, der im Tupel t dem Attribut A_i entspricht.
- $R.A$: qualifiziert ein Attribut A mit dem Relationennamen R
- Für einzelne Komponentenwerte von einem Tupel t gilt:
 - $t[A_i]$ bzw. $t.A_i$ beziehen sich auf den Wert v_i in t für Attribut A_i
 - $t[A_u, \dots, A_z]$ und $t.(A_u, \dots, A_z)$ beziehen sich auf Werte $\langle v_u, \dots, v_z \rangle$ von Subtupeln von t , die den Attributen A_u, \dots, A_z von R entsprechen.

Course.CourseName

$t_1[\text{CreditHours}] = t_1.\text{CreditHours}$

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Mathematische Sichtweise von Relationen

- Eine Relation $r(R)$ ist eine mathematische Relation n -ten Grades auf die Wertebereiche $\text{dom}(A_1), \dots, \text{dom}(A_n)$.
- $r(R)$: Teilmenge des kartesischen Produkts der Wertebereiche, die R definieren:
 - $r(R) \subseteq (\text{dom}(A_1) \times \dots \times \text{dom}(A_n))$
 - Das kartesische Produkt spezifiziert alle Wertekombinationen der zugrundeliegenden Wertebereiche, d.h. alle möglichen Tupel
 - $r(R)$ alle tatsächlichen Tupel
- Beispiel
 - $\text{course}(\text{COURSE})$ hat vier tatsächliche Tupel
 - Alle möglichen Tupel?

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Umfang

- Anzahl der möglichen Werte / Kardinalität eines Wertebereichs:
 - $|\text{dom}(A_i)|$
- Damit die Anzahl möglicher unterschiedlicher Tupel:
 - $|\text{dom}(A_1)| \cdot \dots \cdot |\text{dom}(A_n)|$
- Davon enthält der aktuelle Relationszustand i.d.R. nur einen Ausschnitt (zum Glück)
 - Anzahl tatsächlicher Tupel = Kardinalität der Relation m
 $m \ll |\text{dom}(A_1)| \cdot \dots \cdot |\text{dom}(A_n)|$
- Beispiel
 - Wertebereich kann einschränkend wirken (String vs. Menge von Kursnamen)
 - Logische Zusammenhänge zwischen Attributen schränken die möglichen Tupel weiter ein

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Keine Ordnung im relationalen Modell!

- Erinnerung: r bzw. $r(R)$ eines Relationenschemas $R(A_1, \dots, A_n)$ ist eine **Menge** von n -Tupeln t_j mit $j=1, \dots, m$:
 - $r = \{t_1, \dots, t_m\}$.
- Eine Menge hat keine Ordnung \rightarrow keine Reihenfolge im rel. Modell

COURSE

CourseName	CourseNumber	CreditHours	Department
Discrete Mathematics	MATH2410	3	MATH
Introduction to CS	CS1310	4	CS
Databases	CS3380	3	CS
Data Structures	CS3320	4	CS



COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

„Dynamik“

- Relation $r(R)$ variiert über die Zeit
 - Änderungen an Tupelwerten
 - Löschen und Hinzufügen von Tupeln
 - Beispiele
 - Die Stunden des Kurses „Database“ steigen auf 4
 - Neuer Kurs „Algorithms“
- Relationenschema R ändert sich hingegen selten
 - Schema-Evolution: Redesign des DB-Schemas und damit der darin enthaltenen Relationenschemata
 - Bestehende Daten müssen entsprechend angepasst werden

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	4	CS
	Algorithms	CS3390	4	CS

Spezielle Werte in Tupeln: NULL-Werte

- **NULL**-Wert kann eingetragen werden, wenn richtiger Wert unbekannt oder nicht zutreffend ist
- Problem: unterschiedliche Interpretation möglich, z.B.:
 - Wert ist nicht vorhanden.
Bspw. liegt bei Erstimmatrikulation noch kein Prüfungsergebnis vor.
 - Wert ist im Kontext des Tupels nicht zutreffend.
Bspw. ist keine Angabe zu einem Studienschwerpunkt vorhanden, wenn das Studienfach dies nicht zulässt.
 - Es ist unbekannt, ob der Wert nicht vorhanden oder nicht zutreffend ist.
Bspw. ist kein Eintrag unter „Fachschaftsvertreter“ vorhanden
(bricht mit der *Closed World Assumption*)
 - NULL \neq NULL (in einer Spalte)
Unbekannte, aber möglicherweise unterschiedliche Werte

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS
	Algorithms	CS3390	4	NULL

Noch nicht
bekannt

Schlüssel

Spezielle Attribute

Schlüssel

- Jede Relation besitzt einen **Primärschlüssel**, der ein einzelnes Attribut oder eine Kombination von Attributen ist, so dass eine eindeutige Identifikation jedes Tupels innerhalb der Relation ermöglicht wird
 - Ein Schlüsselwert v identifiziert ein Tupel t
- Beispiel
 - Das Attribut CourseNumber ist Schlüssel für die Relation COURSE
 - CourseNumber identifiziert jedes Tupel in COURSE eindeutig, da die CourseNumber einzigartig für jeden Kurs ist
 - Beispiel: Schlüsselwert CS1310 identifiziert Tupel t_1

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Schlüssel: Definitionen

- Für Relationenschemata gilt i.d.R.:
 - Es gibt Teilmengen SK der Attribute von R , für die nie zwei Tupel eines Relationenzustands $r(R)$ die gleiche Wertkombination besitzen dürfen; d.h.:
 - Für alle Tupel t_1 und t_2 gilt: $t_1[SK] \neq t_2[SK]$
 - alle Tupel sind bzgl. der Attributmenge SK eindeutig
- Alle solchen Teilmengen SK heißen **Superschlüssel**
 - Trivialer Superschlüssel: alle Attribute von R
- Minimaler Superschlüssel = Schlüssel:
 - **Minimal**: kein Attribut kann weggelassen werden, ohne dass die Schlüsseleigenschaft verloren geht

trivialer Superschlüssel

minimaler Schlüssel

COURSE	CourseName	CourseNumber	CreditHours	Department
	Introduction to CS	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Databases	CS3380	3	CS

Schlüssel und Primärschlüssel

- Relationen können mehrere Schlüssel (minimale Superschlüssel) enthalten
 - Genannt **Schlüsselkandidaten**
 - Attribute eines Schlüsselkandidaten: Prime-Attribute
- Einer der Schlüsselkandidaten wird gewählt als **Primärschlüssel**
 - Attribute des Primärschlüssels (**dürfen nicht NULL sein**): Primärschlüssel-Attribute
- Oft ist die Einführung eines künstlichen Schlüssels sinnvoll
 - z.B. eine eindeutige Nummer (ID)

Schlüsselkandidaten

gewählter Primärschlüssel

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Schlüssel

- Eine Relation mit Schlüssel repräsentiert eine Funktion von den Primärschlüsselattributen zu den Nicht-Schlüsselattributen
 - Ein Schlüsselwert v identifiziert ein Tupel t
 - Damit erhält man die Attributwerte von t
- Beispiel
 - Schlüsselwert CS1310 identifiziert Tupel t_1
 - CS1310 \rightarrow (Introduction to CS, CS1310, 4, CS) bzw.
 - CS1310 \rightarrow Introduction to CS
 - CS1310 \rightarrow CS1310
 - CS1310 \rightarrow 4
 - CS1310 \rightarrow CS

Schlüsselwert

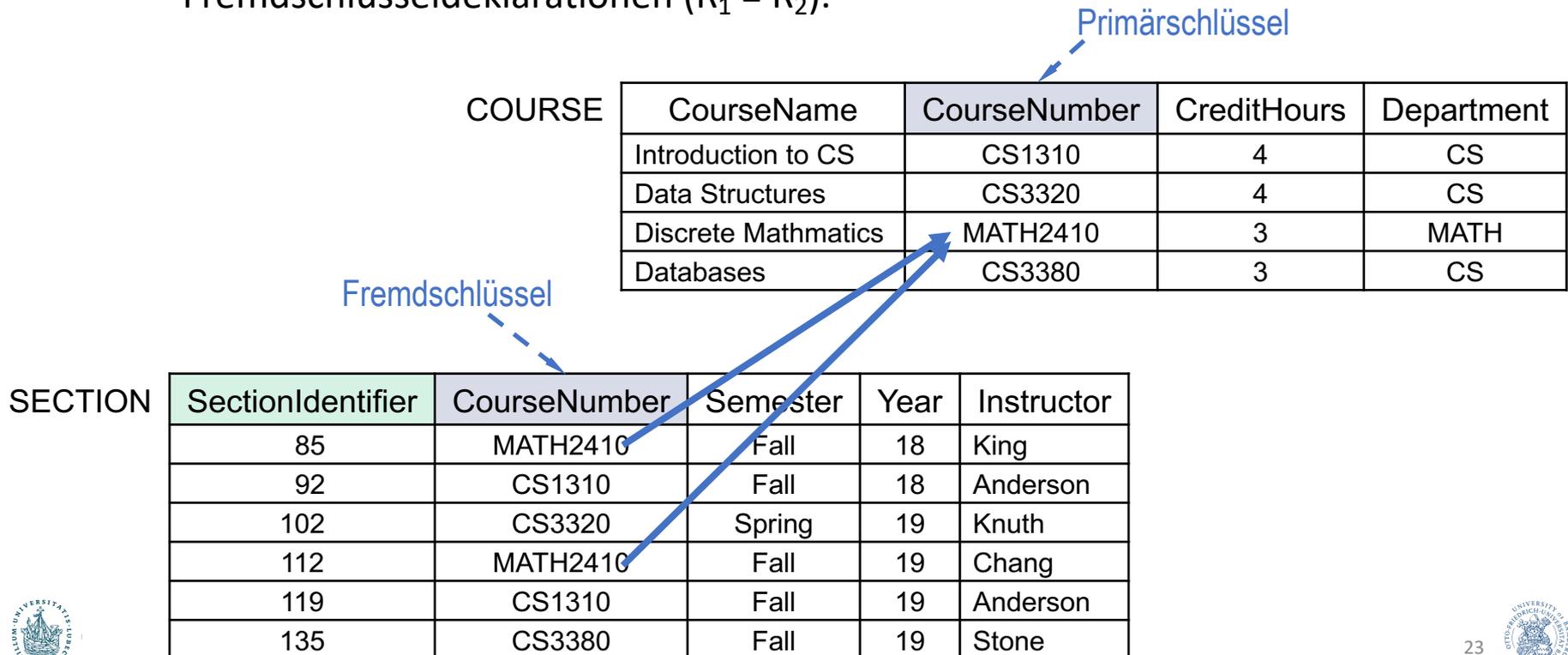
t_1 COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

Detailed description: A table representing a 'COURSE' relation. The first row is the header with columns 'CourseName', 'CourseNumber', 'CreditHours', and 'Department'. The second row is highlighted with a blue dashed box and labeled 't1' on the left. A red dashed line points from the text 'Schlüsselwert' above to the 'CourseNumber' cell of the first row. The table contains four rows of data: (Introduction to CS, CS1310, 4, CS), (Data Structures, CS3320, 4, CS), (Discrete Mathematics, MATH2410, 3, MATH), and (Databases, CS3380, 3, CS).

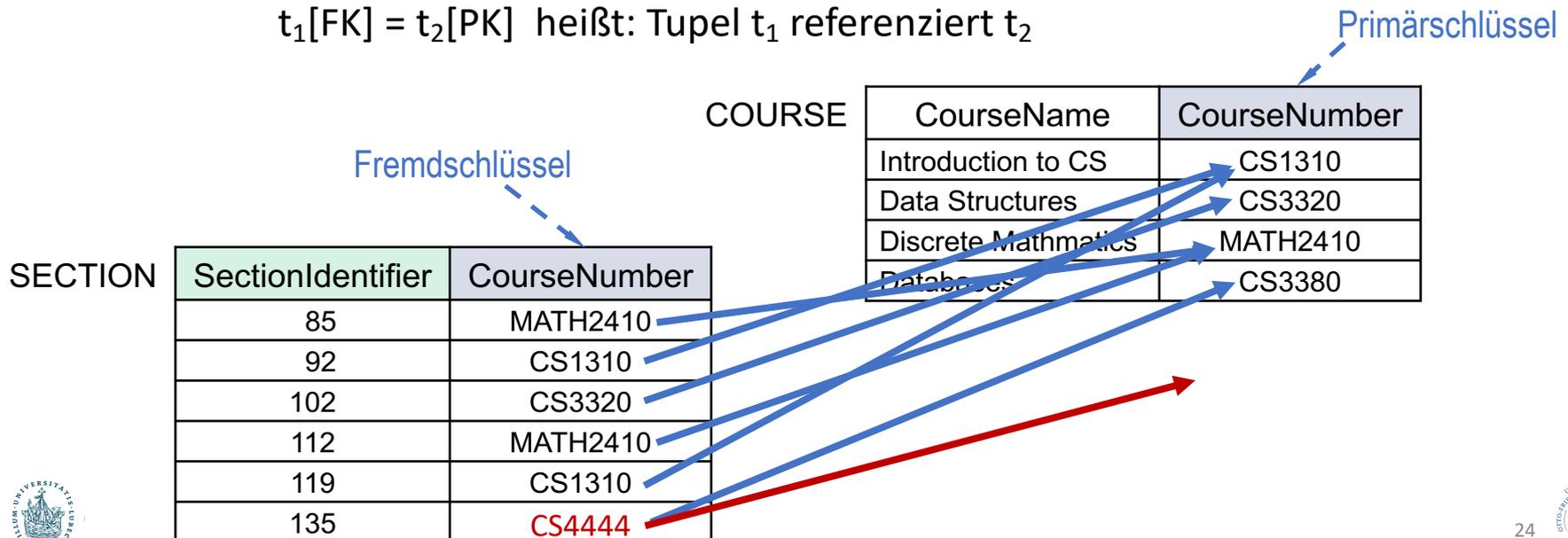
Beziehungen zwischen Tupeln

- Durch Identifikation des referenzierten Objektes über seinen Primärschlüssel (→ assoziative Identifikation)
 - Einen Schlüssel, der in Relation R_1 zur Identifikation eines Tupels in Relation R_2 benutzt wird, bezeichnet man als **Fremdschlüssel**
 - Rekursive Beziehungen (z.B. Angestellte : Vorgesetzte) führen zu reflexiven Fremdschlüsseldeklarationen ($R_1 = R_2$).



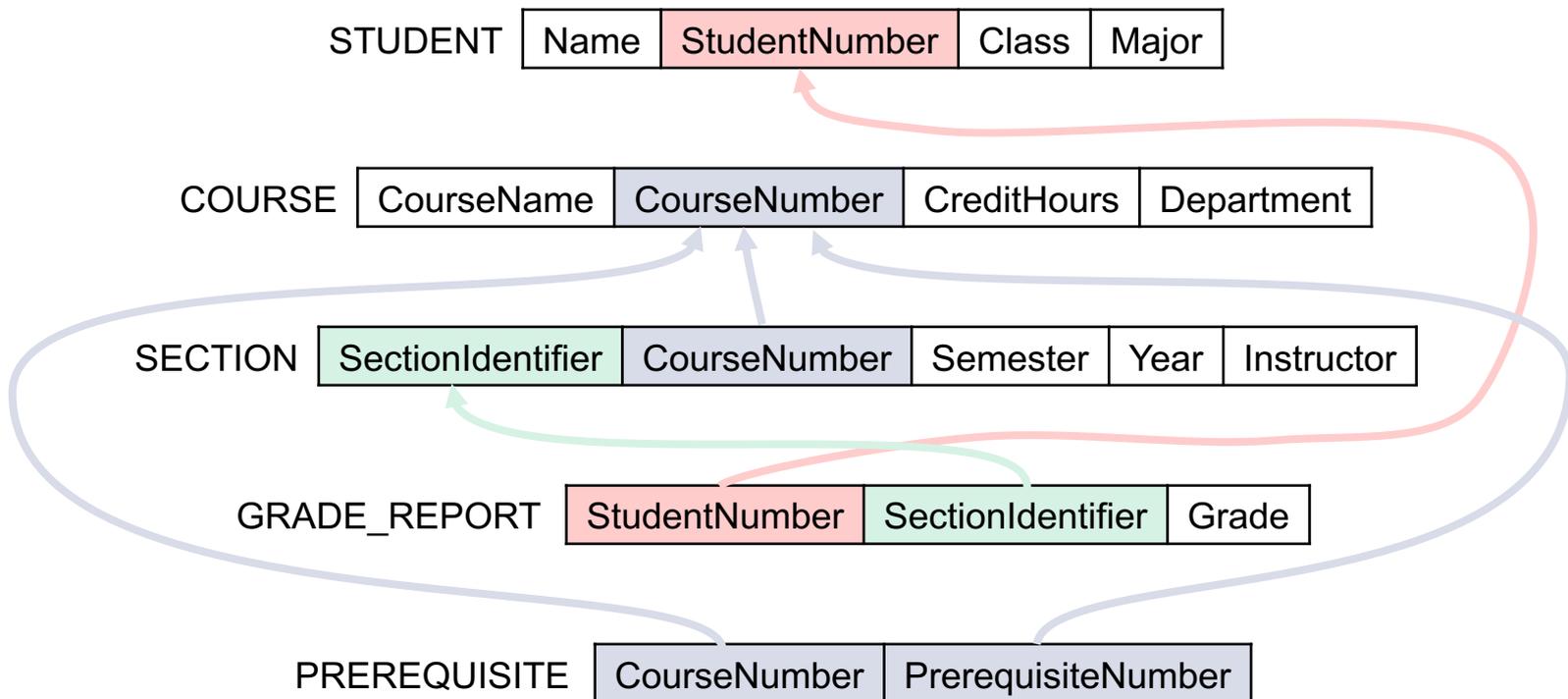
Fremdschlüssel: Definition

- Eine Attributmenge FK im Relationenschema R_1 ist ein Fremdschlüssel von R_1 , der auf Relation R_2 referenziert, falls gilt:
 1. Die Attribute in FK haben den gleichen Wertebereich bzw. die gleichen Wertebereiche wie die Primärschlüssel-Attribute PK von R_2
 - Die Attribute FK gelten als Referenz bzw. referenzieren auf R_2
 2. Ein Wert von FK in einem Tupel t_1 des aktuellen Zustands $r_1(R_1)$ ist
 - NULL (keine Beziehung)
 - Er kommt als Wert von PK in einem Tupel t_2 im Zustand $r_2(R_2)$ vor:
 $t_1[FK] = t_2[PK]$ heißt: Tupel t_1 referenziert t_2



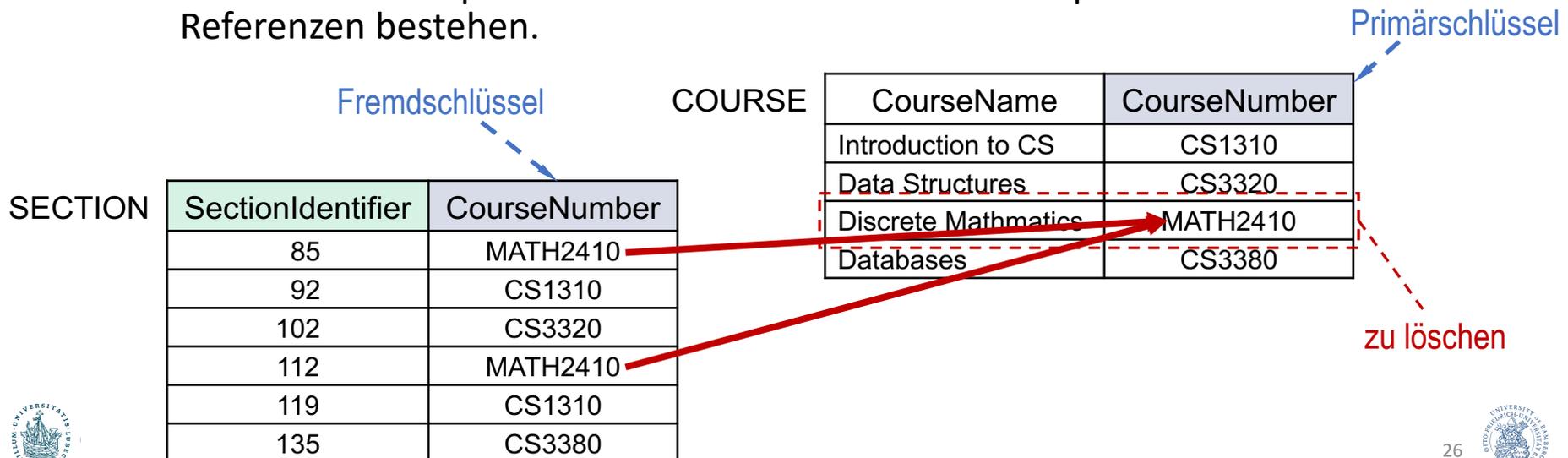
Referentielle Integrität - Fremdschlüssel

- Problem: **Inkonsistenz**
- Lösung: referentielle Integritätsbedingungen auf Fremdschlüsseln
 - **Referentielle Integrität**: zu jedem benutzten Fremdschlüssel existiert ein Tupel mit einem entsprechenden Primärschlüsselwert in der referenzierten Tabelle.



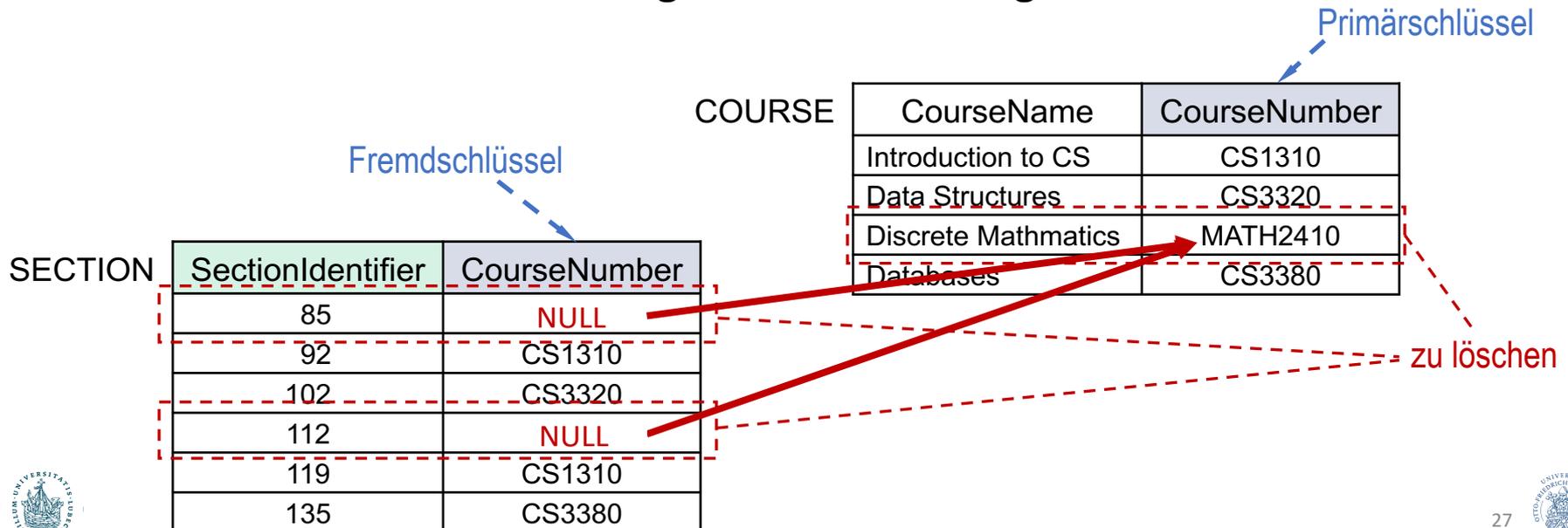
Referentielle Integrität: Konflikte

- Bedingungen der referentiellen Integrität werden mögl.weise verletzt
 - Durch Änderungen im Relationenzustand
 - Einhaltung der referentiellen Integrität ist Teil von ACID! (Welcher?)
- Überprüfung der referentiellen Integrität ist notwendig beim
 - Einfügen eines neuen Fremdschlüsselwertes in eine Beziehungstabelle. Das referenzierte Objekt mit diesem Wert als Primärschlüssel muss existieren.
 - Aktualisierung eines Primärschlüsselwertes. Alle Referenzen müssen aktualisiert werden.
 - Löschen eines Tupels aus einer Relation. Auf dieses Tupel dürfen keine Referenzen bestehen.



Referentielle Integrität: Konflikte – Löschen

- Gibt es noch Referenzen, bieten sich mehrere Möglichkeiten an:
 - Eine Fehlermeldung wird erzeugt.
 - Propagierung der Löschoperation, das referenzierende Tupel wird ebenfalls gelöscht (→ kaskadiertes Löschen).
 - Die Referenzen können durch Setzen des Fremdschlüssels auf einen Nullwert ungültig gemacht werden, sofern dieser nicht Bestandteil des Schlüssels ist
 - Weitere Möglichkeiten (mehr nächstes Mal)
- DBMS bieten Unterstützung zur Verhinderung solcher Konflikte



Zusammenfassung/Überblick

- Ein relationales Datenmodell ist eine Menge benannter **Relationen**
- Eine Relation ist eine Menge von Elementen (**Tupeln**)
 - deren Struktur durch **Attribute** definiert,
 - deren Identität durch **Schlüssel** realisiert und
 - deren Werte durch **Domänen** kontrolliert werden
- Beziehungen zwischen Relationen werden über **Fremdschlüssel** realisiert
 - **Referentielle Integrität** muss gewahrt bleiben!
- Relationen werden meist durch **Tabellen** dargestellt
 - **Zeilen**
 - Jede Zeile repräsentiert ein Element der Relation (ein Tupel)
 - Die Zahl der Zeilen ist variabel und wird **Kardinalität** der Relation genannt
 - **Spalten**
 - Die Spalten der Tabellen enthalten die Attribute der Relation
 - Die Zahl der Spalten einer Tabelle wird im Schema festgelegt
 - Jeder Spalte ist eine Domäne zugeordnet, welche die zulässigen Werte für das Attribut in allen Zeilen festlegt

Relationenschemata

Und was haben DBs damit zu tun?



DB-Schema und DB-Zustand

- Die meisten Datenbanken sind relationale Datenbanken
 - Ihnen unterliegt ein relationales Datenbankschema
- Ein relationales **Datenbankschema** DS umfasst:
 - Menge von *Relationenschemata*: $DS = \{R_1, \dots, R_m\}$
 - Menge von *Integritätsbedingungen* IC (integrity constraints)
 - „Intension“
- Ein **Datenbankzustand** DB (zum Zeitpunkt x) von DS umfasst die Menge der zum Zeitpunkt x aktuellen Relationszustände, d.h.
 - $DB = \{r_1, \dots, r_m\}$
 - „Extension“
- Für $DB = \{r_1, \dots, r_m\}$ gilt:
 - Jedes r_i ist ein zulässiger Zustand für R_i
 - Die Gesamtheit der Relationszustände r_i erfüllen die in IC spezifizierten Integritätsbedingungen.

Beispiel eines DB-Schemas

STUDENT	Name	StudentNumber	Class	Major
---------	------	---------------	-------	-------

COURSE	CourseName	CourseNumber	CreditHours	Department
--------	------------	--------------	-------------	------------

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
---------	-------------------	--------------	----------	------	------------

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
--------------	---------------	-------------------	-------

PREREQUISITE	CourseNumber	PrerequisiteNumber
--------------	--------------	--------------------

Beispiel eines DB-Zustands

STUDENT

Name	StudentNumber	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Databases	CS3380	3	CS

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor
85	MATH2410	Fall	18	King
92	CS1310	Fall	18	Anderson
102	CS3320	Spring	19	Knuth
112	MATH2410	Fall	19	Chang
119	CS1310	Fall	19	Anderson
135	CS3380	Fall	19	Stone

GRADE_REPORT

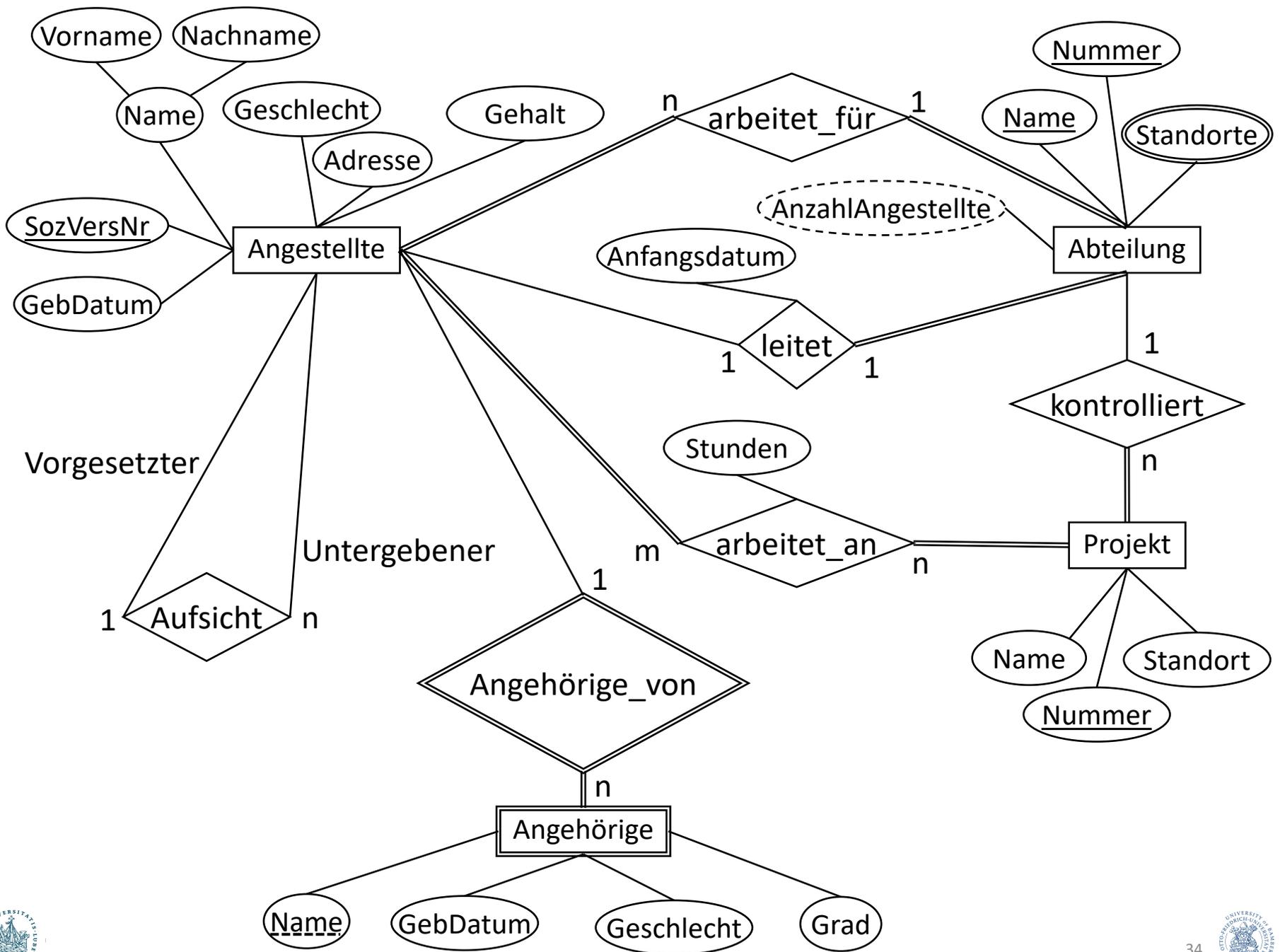
StudentNumber	SectionIdentifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

CourseNumber	PrerequisiteNumber
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

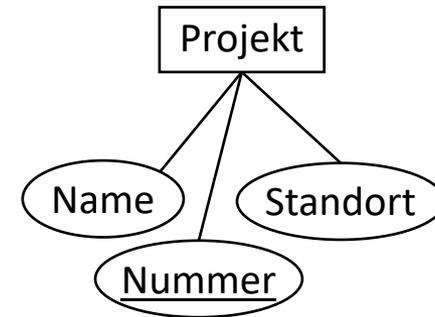
Entwurf relationaler Schemata

Vom ER-Diagramm zum relationalen Schema



Relationale Darstellung von Entitätstypen

- Übersetzung von Entitäten mit Attributen in Relationen
 - Entität E mit Attributen A_1, \dots, A_n wird zu Relation $E(A_1, \dots, A_n)$
 - Wertebereiche für A_1, \dots, A_n festlegen
- Was ist mit Schlüsseln?



PROJEKT	Nummer	Name	Standort
---------	--------	------	----------

Wertebereiche

- Nummer: Integer
- Name: String
- Standort: String (oder benutzerdefinierte Standorte)

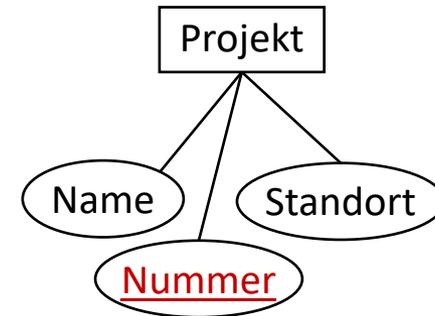
Relationale Darstellung: Entitäten + Schlüssel

- Übersetzung von Entitäten mit Attributen in Relationen

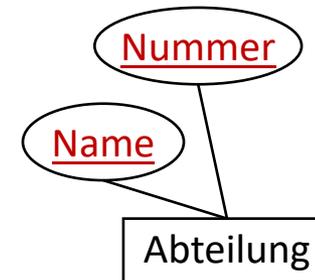
- Entität E mit Attributen A_1, \dots, A_n wird zu Relation $E(A_1, \dots, A_n)$
- In ER-Diagramm identifizierende Attribute angeben
 - Vorsicht geboten!
Nicht immer direkt nutzbar
 - Schlüssel im relationalen Modell **minimale** Superschlüssel
 - Interpretation der ER Schlüssel teilweise nicht eindeutig
 - Name und Nummer zusammen eindeutig? Allein eindeutig?

- Was ist mit besonderen Attributen?

- Welche gab es?



PROJEKT	<u>Nummer</u>	Name	Standort
---------	---------------	------	----------



Entscheidung (evtl. nach Rücksprache mit Kunden); wir nehmen Nummer als Schlüssel, da

- Nummer kleiner als Name
- Vergleiche weniger aufwendig (Int vs. String Vergleich)

ABTEILUNG	Name	<u>Nummer</u>
-----------	------	---------------

Relationale Darstellung von Entitätstypen

- Mehrwertige Attribute

- Nicht direkt ins relationale Datenmodell übersetzbar (warum?)
 - Relationenzustand für ABTEILUNG (Name, Nummer, Standort)

Nummer nicht mehr eindeutig!

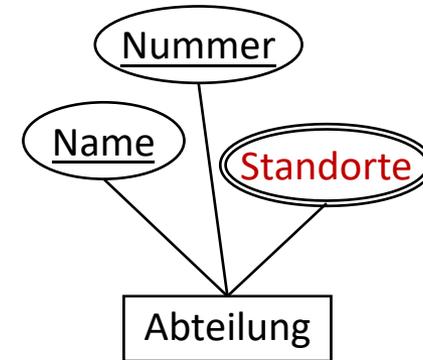
ABTEILUNG	Name	<u>Nummer</u>	Standort
	XYZ	42	A
	XYZ	42	B
	XYZ	42	C

- Lösung: eigene Relation bauen

- Übersetzter Relationenzustand

ABTEILUNG	Name	<u>Nummer</u>
	XYZ	42

ABT_STNDRT	<u>AbtNr</u>	Standort
	42	A
	42	B
	42	C



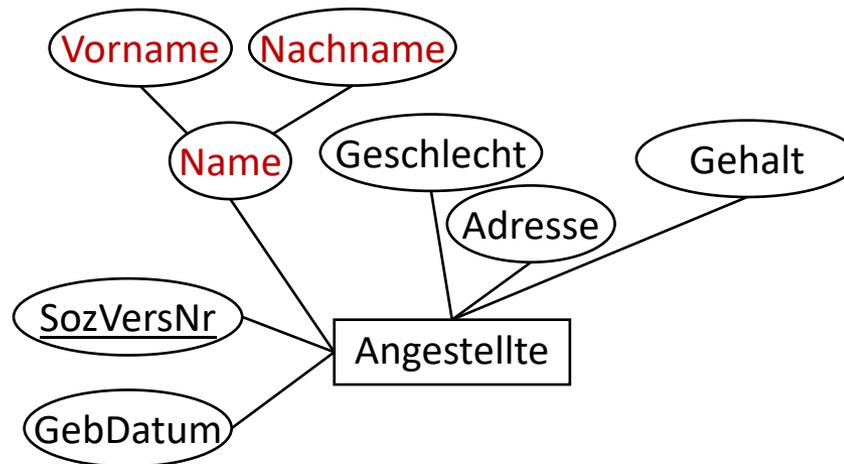
ABTEILUNG	Name	<u>Nummer</u>
-----------	------	---------------

ABT_STNDRT	<u>AbtNr</u>	Standort
------------	--------------	----------

Fremdschlüssel

Relationale Darstellung von Entitätstypen

- Zusammengesetzte Attribute



- Als ein Attribut in Relation aufnehmen (gibt Teilung auf)

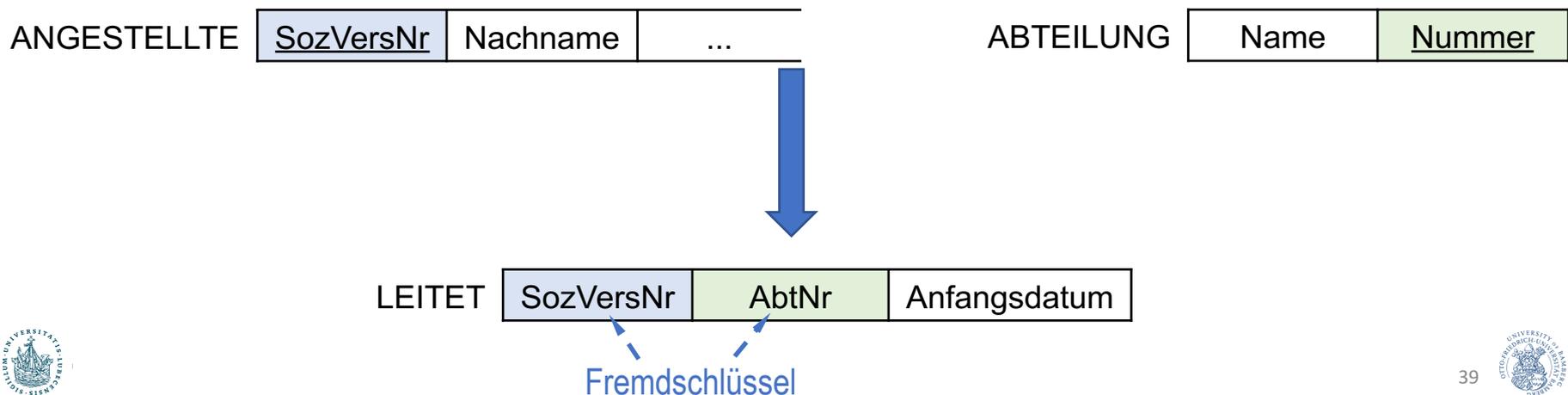
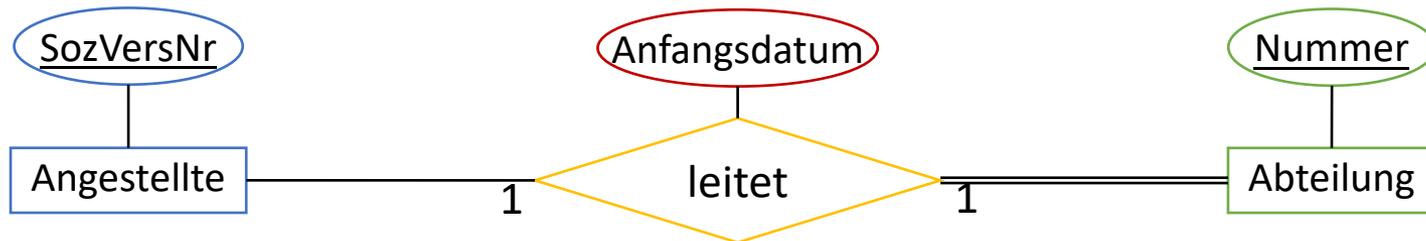
ANGESTELLTE	<u>SozVersNr</u>	Name	Geschlecht	Adresse	Gehalt	GebDatum
-------------	------------------	------	------------	---------	--------	----------

- Als einzelne Attribute in Relation aufnehmen (gibt Zusammengehörigkeit auf)

ANGESTELLTE	<u>SozVersNr</u>	Nachname	Vorname	Geschlecht	Adresse	Gehalt	GebDatum
-------------	------------------	----------	---------	------------	---------	--------	----------

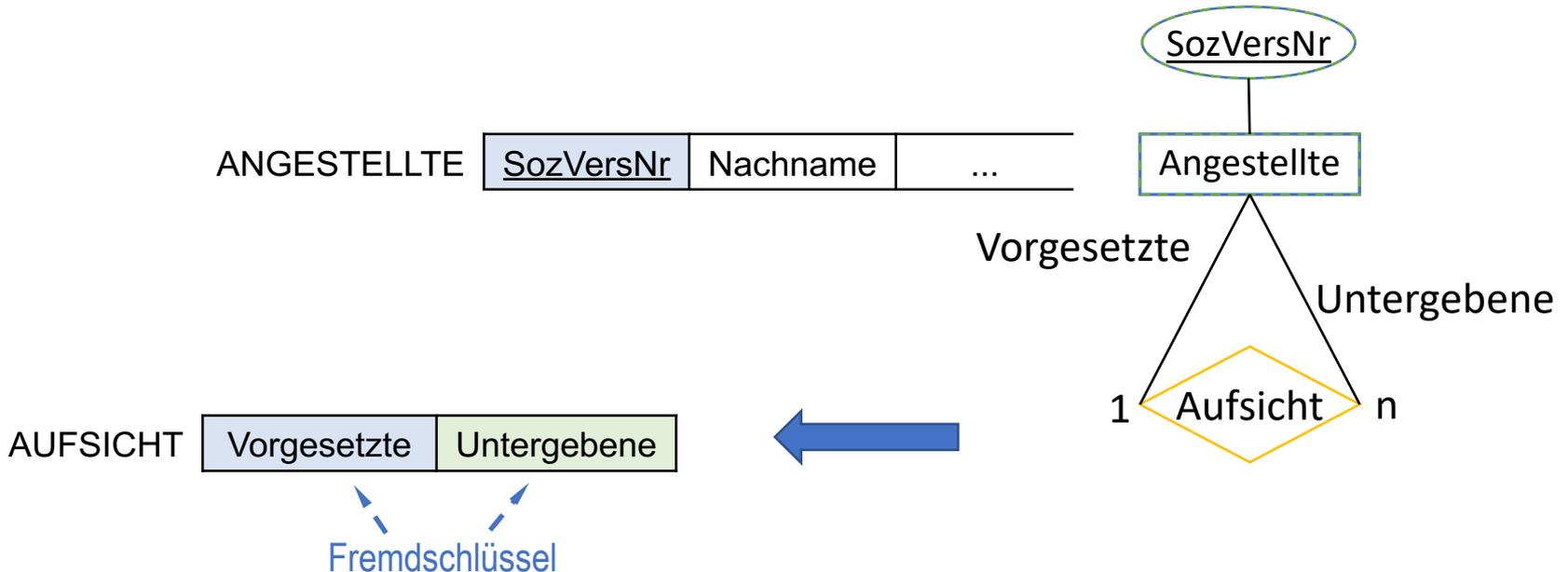
Relationale Darstellung von Assoziationen

- Übersetzung von Assoziationen/Beziehungen mit Attributen in Relationen
 - Beziehung R mit Attributen A_1, \dots, A_n
 - Zwischen Entitäten E_1 und E_2 mit Schlüsseln $P_{1,1}, \dots, P_{1,l}$ und $P_{2,1}, \dots, P_{2,m}$
 - wird zu Relation $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$
 - Auch bei rekursiven Relationen (Vorgesetzter/Untergebener), dann $E_1 = E_2$



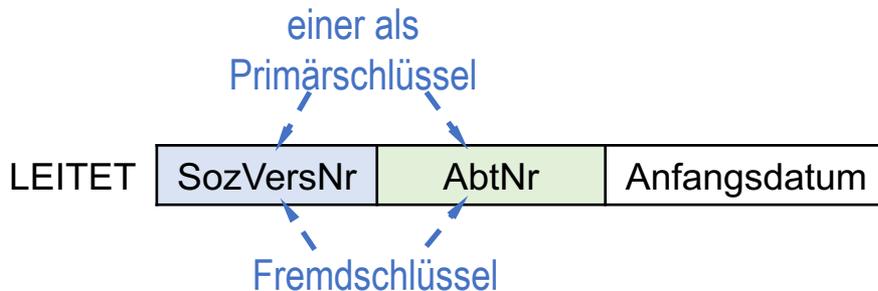
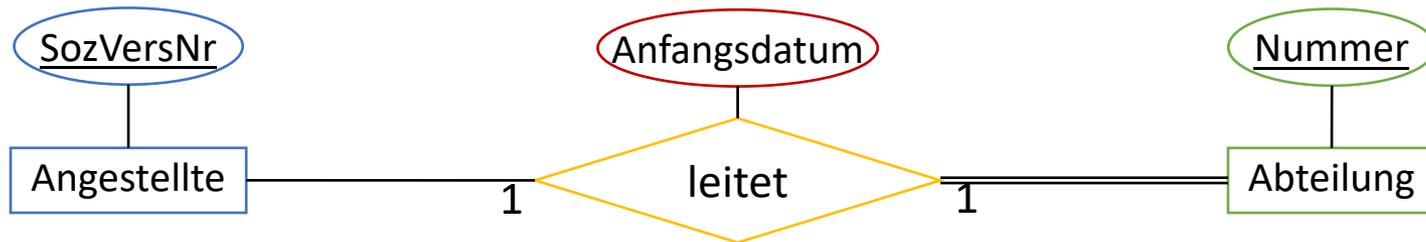
Relationale Darstellung von Assoziationen

- Übersetzung von Assoziationen/Beziehungen mit Attributen in Relationen
 - Beziehung R mit Attributen A_1, \dots, A_n
 - Zwischen Entitäten E_1 und E_2 mit Schlüsseln $P_{1,1}, \dots, P_{1,l}$ und $P_{2,1}, \dots, P_{2,m}$
 - wird zu Relation $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$
 - Auch bei rekursiven Relationen (Vorgesetzter/Untergebener), dann $E_1 = E_2$



Relationale Darstellung: Assoziation + Schlüssel

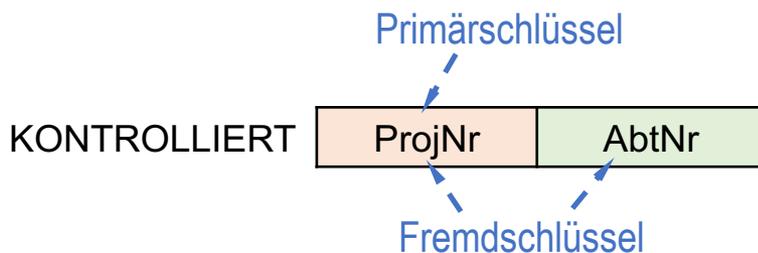
- Häufig abhängig von Fremdschlüsseln
 - Fremdschlüssel $P_{1,1}, \dots, P_{1,l}$ und $P_{2,1}, \dots, P_{2,m}$
 - in Relation $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$
- Kardinalitäten betrachten!
 - 1:1 Beziehung



LEITET	SozVersNr	AbtNr
	A1	42
	B2	21
	C4	54

Relationale Darstellung: Assoziation + Schlüssel

- Häufig abhängig von Fremdschlüsseln
 - Fremdschlüssel $P_{1,1}, \dots, P_{1,l}$ und $P_{2,1}, \dots, P_{2,m}$
 - in Relation $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$
- Kardinalitäten betrachten!
 - 1:n Beziehung

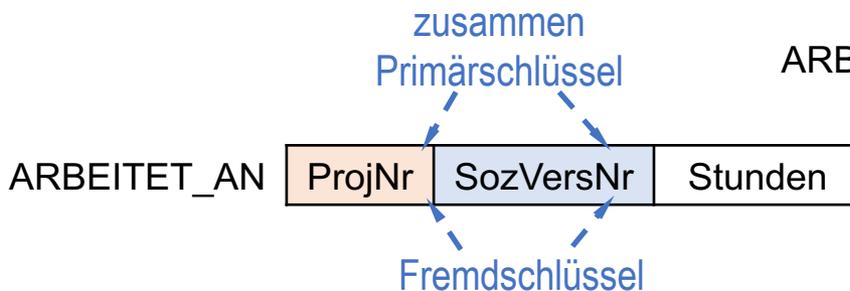
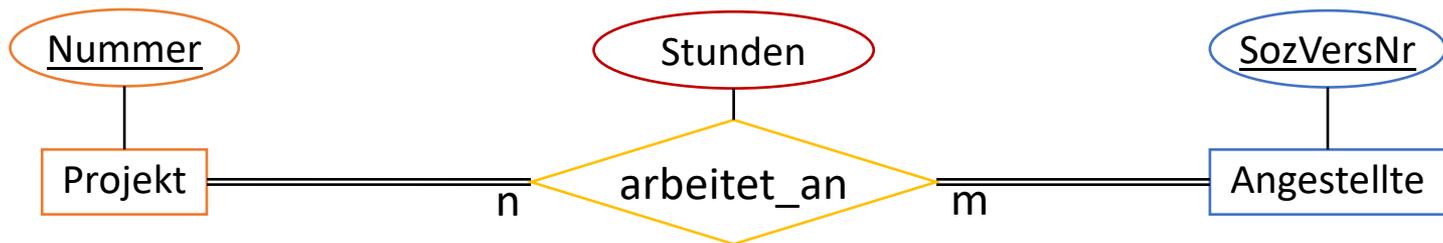


KONTROLLIERT

ProjNr	AbtNr
1	42
2	42
3	21
4	54

Relationale Darstellung: Assoziation + Schlüssel

- Häufig abhängig von Fremdschlüsseln
 - Fremdschlüssel $P_{1,1}, \dots, P_{1,l}$ und $P_{2,1}, \dots, P_{2,m}$
 - in Relation $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$
- Kardinalitäten betrachten!
 - n:m Beziehung

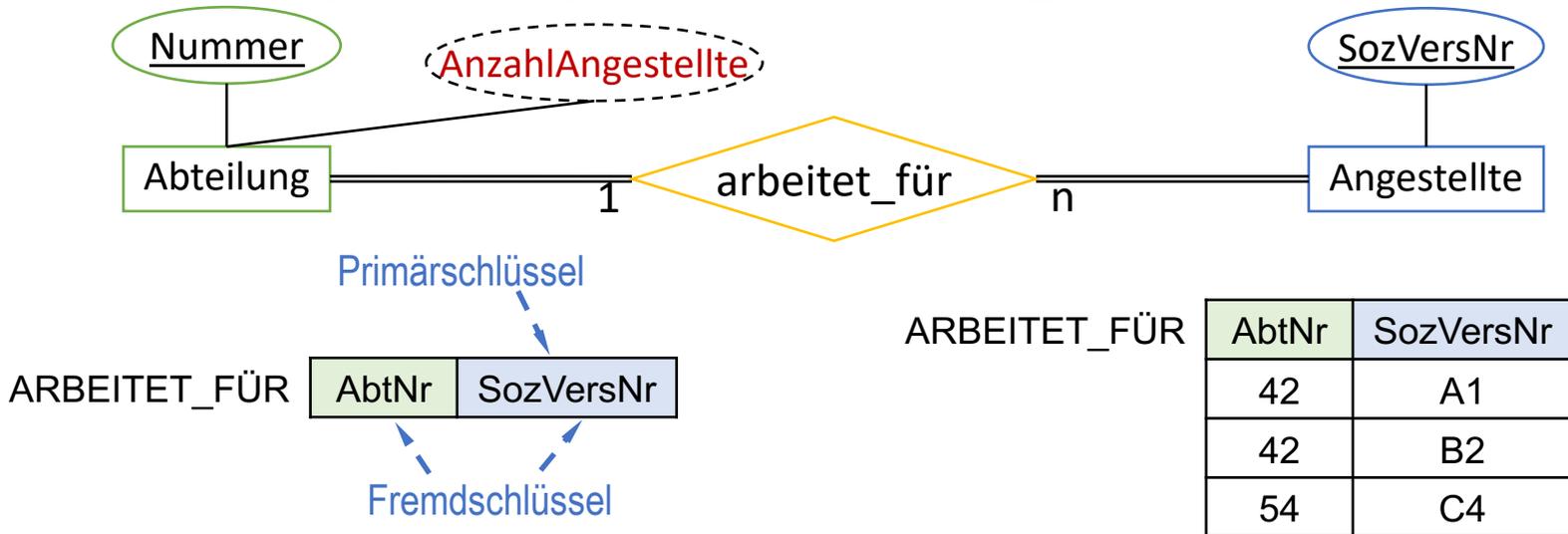


ARBEITET_AN

ProjNr	SozVersNr	Stunden
1	A1	29.02.2000
1	B2	29.02.2000
2	A1	23.07.2004
3	A1	23.07.2004
3	C4	03.05.2018

Relationale Darstellung: Abgeleitete Attribute

- Häufig abhängig von Fremdschlüsseln
 - Fremdschlüssel $P_{1,1}, \dots, P_{1,l}$ und $P_{2,1}, \dots, P_{2,m}$
 - in Relation $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$
- **Abgeleitete Attribute** ergeben sich aus Zuständen anderer Teile
 - „AnzahlAngestellte“ ergibt sich durch „arbeitet_für“



- Klassische Anfrage an DB (Wie? - nächste Vorlesung)
 - Anzahl an Angestellten, die für eine Abteilung (z.B. 42) arbeiten (Ergebnis: 2)

Relationale Darstellung: Schwache Entitäten

- Schwache Entität E_1 mit partiellen Schlüsseln $P_{1,1}, \dots, P_{1,l}$
 - mit starker Entität E_2 mit Schlüsseln $P_{2,1}, \dots, P_{2,m}$
 - identifiziert über Beziehung R mit Attributen A_1, \dots, A_n
 - wird zu Relation $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$



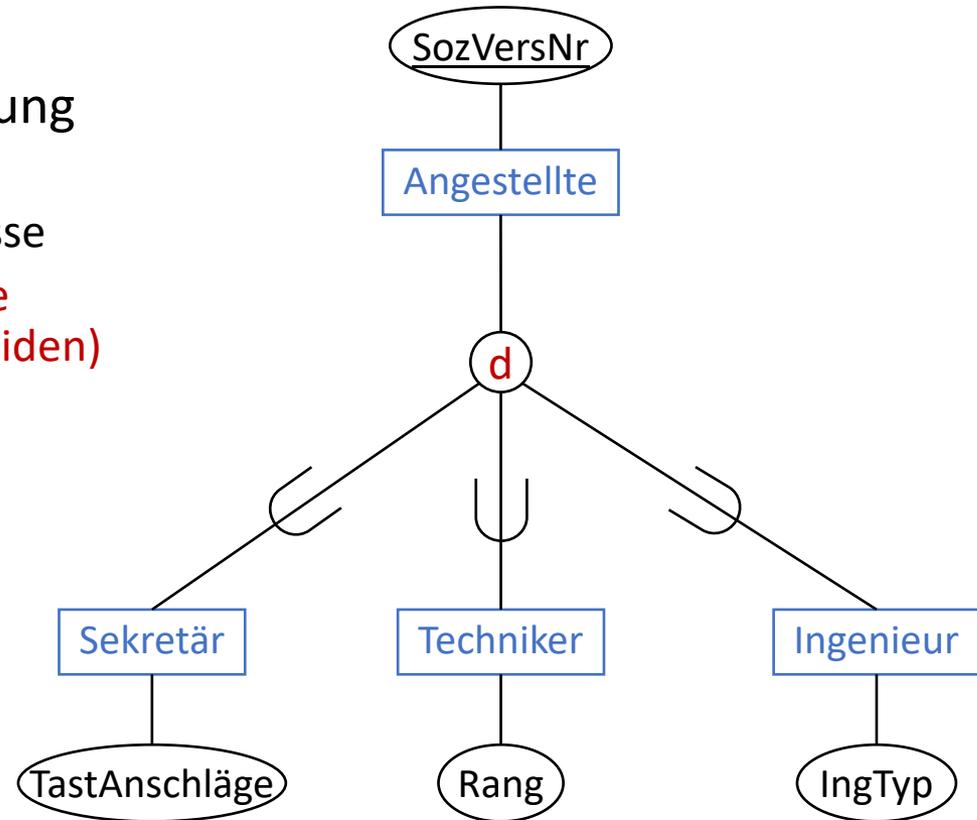
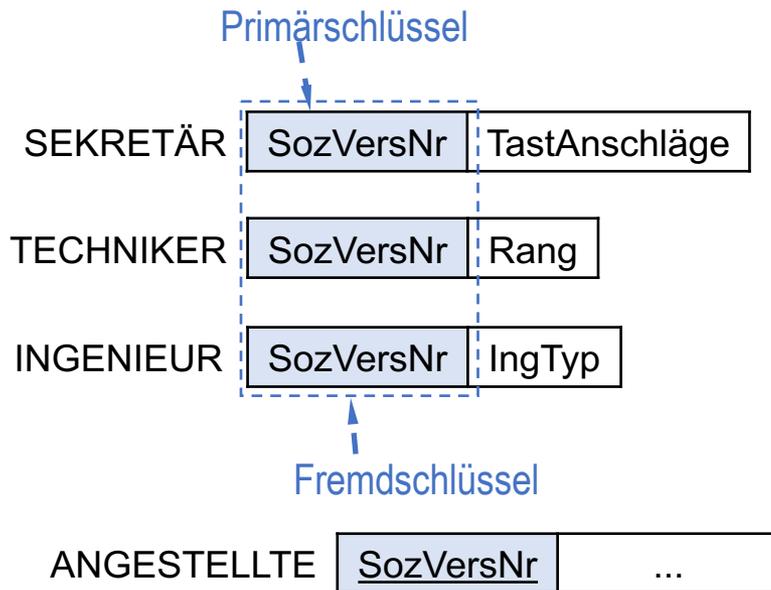
ANGEHÖRIGE	Name	GebDatum	Geschlecht	Grad	SozVersNr
------------	------	----------	------------	------	-----------

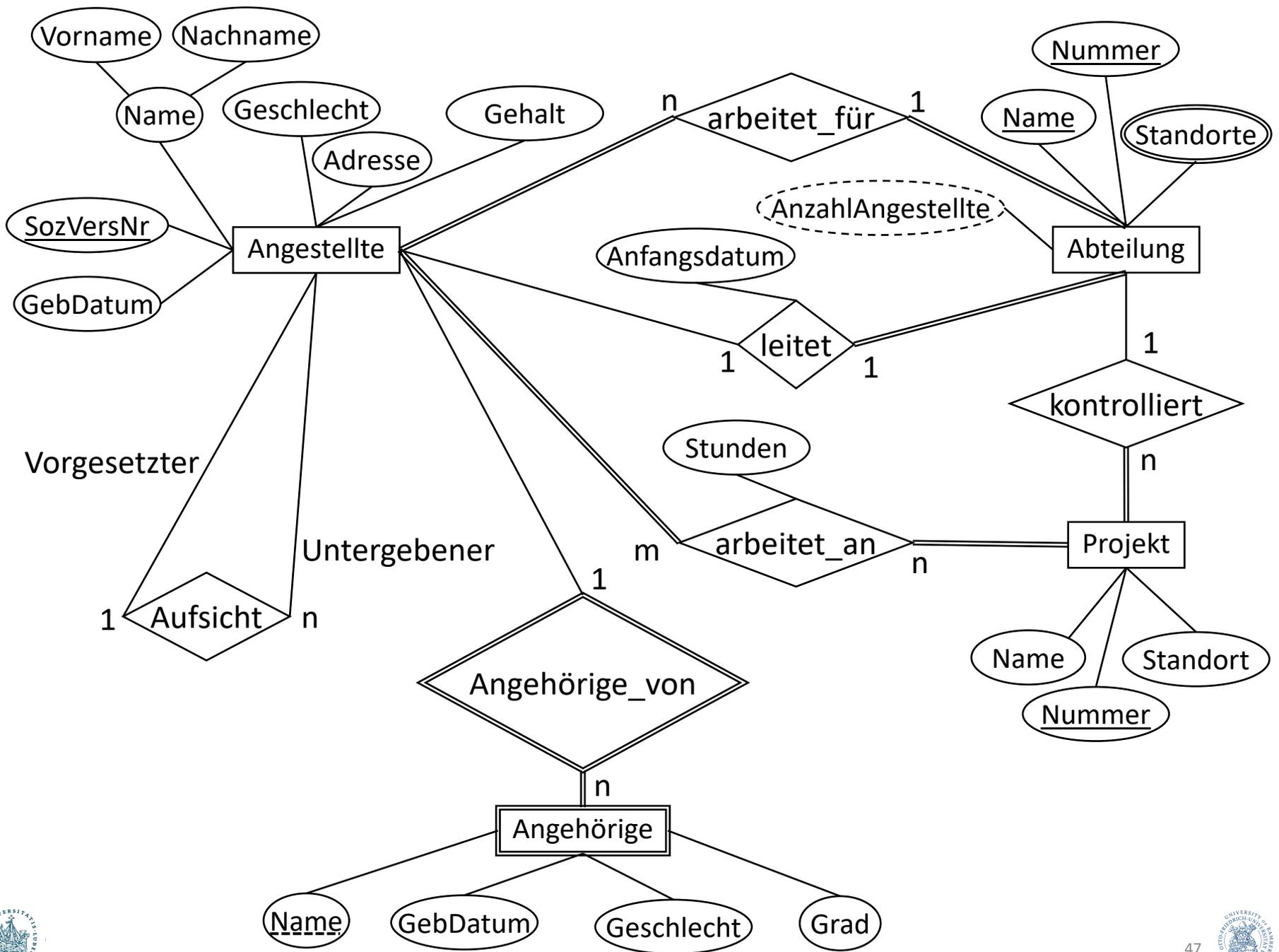
← zusammen Primärschlüssel (dashed arrow from Name to SozVersNr)
 ← Fremdschlüssel (dashed arrow from SozVersNr to the label)

ANGEHÖRIGE	Name	GebDatum	Geschlecht	Grad	SozVersNr
	Alice	A1
	Eve	A1
	Bob	A1
	Charlie	B2
	Judy	C4

Relationale Darstellung: EER-Modell

- EER-Konstrukte:
Spezialisierung, Generalisierung
 - Subklasse wird zur Relation
 - Fremdschlüssel von Superklasse
 - **Keine Vererbung der Attribute (doppelte Speicherung vermeiden)**





Entsprechendes relationales Schema

ANGESTELLTE	<u>SozVersNr</u>	Nachname	Vorname	Geschlecht	Adresse	Gehalt	GebDatum
-------------	------------------	----------	---------	------------	---------	--------	----------

AUFSICHT	Vorgesetzte	<u>Untergebene</u>
----------	-------------	--------------------

ABTEILUNG	Name	<u>Nummer</u>
-----------	------	---------------

ARBEITET_FÜR	AbtNr	<u>SozVersNr</u>
--------------	-------	------------------

LEITET	<u>SozVersNr</u>	AbtNr	AnfDatum
--------	------------------	-------	----------

ARBEITET_AN	<u>ProjNr</u>	<u>SozVersNr</u>	Stunden
-------------	---------------	------------------	---------

PROJEKT	<u>Nummer</u>	Name	Standort
---------	---------------	------	----------

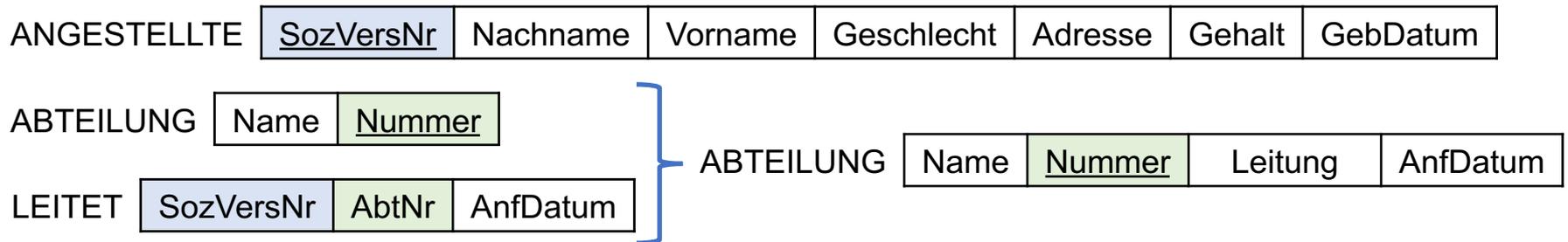
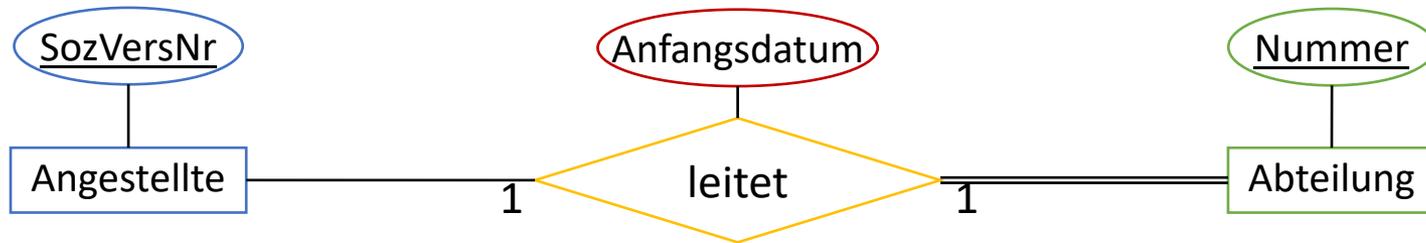
KONTROLLIERT	<u>ProjNr</u>	AbtNr
--------------	---------------	-------

ANGEHÖRIGE	<u>Name</u>	GebDatum	Geschlecht	Grad	<u>SozVersNr</u>
------------	-------------	----------	------------	------	------------------

Neun Relationen

Verfeinerung des relationalen Schemas

- Verfeinerung durch Zusammenfassung
 - Relationen mit gleichem Schlüssel kann man zusammenfassen
 - Aber nur diese und keine anderen!



1. Wenn „SozVersNr“ Schlüssel → LEITET mit ANGESTELLTE zusammenfassen
 2. Wenn „AbtNr“ Schlüssel → LEITET mit ABTEILUNG zusammenfassen
- Da jede Abteilung eine Leitung hat, aber nicht jeder Angestellte leitet, ist 2. hier besser

Verfeinerung des relationalen Schemas

- Relationen mit gleichem Schlüssel kann man zusammenfassen
- **Aber nur diese und keine anderen!**
- Vor allem sinnvoll bei totaler Partizipation (auch der Fall auf vorheriger Folie)
 - Jede angestellte Person arbeitet für genau eine Abteilung



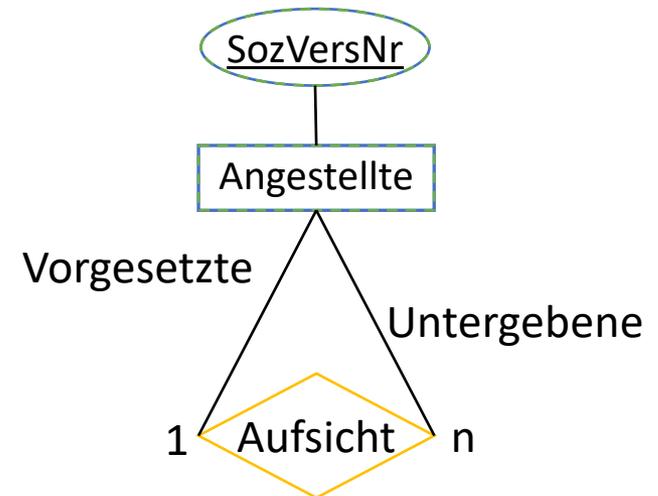
ANGESTELLTE	<u>SozVersNr</u>	Nachname	Vorname	Geschlecht	Adresse	Gehalt	GebDatum
-------------	------------------	----------	---------	------------	---------	--------	----------

ARBEITET_FÜR	AbtNr	<u>SozVersNr</u>
--------------	-------	------------------

ANGESTELLTE	<u>SozVersNr</u>	Nachname	Vorname	Geschlecht	Adresse	Gehalt	GebDatum	AbtNr
-------------	------------------	----------	---------	------------	---------	--------	----------	-------

Verfeinerung des relationalen Schemas

- Relationen mit gleichem Schlüssel kann man zusammenfassen
- **Aber nur diese und keine anderen!**
- Ohne totale Partizipation können NULL Werte erforderlich werden
 - Wenn fast jeder eine vorgesetzte Person hat, dann Zusammenschluss sinnvoll
 - Andernfalls viele Zellen mit NULL Werten; Beziehung nicht so schnell ersichtlich



ANGESTELLTE	<u>SozVersNr</u>	Nachname	Vorname	Geschlecht	Adresse	Gehalt	GebDatum
-------------	------------------	----------	---------	------------	---------	--------	----------

AUFSICHT	Vorgesetzte	<u>Untergebene</u>
----------	-------------	--------------------

ANGESTELLTE	<u>SozVersNr</u>	Nachname	Vorname	Geschlecht	Adresse	Gehalt	GebDatum	Vorges.
-------------	------------------	----------	---------	------------	---------	--------	----------	---------

Verfeinertes relationales Schema

ANGESTELLTE	<u>SozVersNr</u>	Nachn.	Vorn.	Geschlecht	Adresse	Gehalt	GebDatum	AbtNr	Vorges.
-------------	------------------	--------	-------	------------	---------	--------	----------	-------	---------

ABTEILUNG	Name	<u>Nummer</u>	Leitung	AnfDatum
-----------	------	---------------	---------	----------

ARBEITET_AN	<u>ProjNr</u>	<u>SozVersNr</u>	Stunden
-------------	---------------	------------------	---------

PROJEKT	<u>Nummer</u>	Name	Standort	AbtNr
---------	---------------	------	----------	-------

ANGEHÖRIGE	<u>Name</u>	GebDatum	Geschlecht	Grad	<u>SozVersNr</u>
------------	-------------	----------	------------	------	------------------

Entwurf relationaler Schemata

- Zwei alternative Methoden

1. Entwickle zunächst ein ER-Diagramm, leite daraus ein relationales Schema mit Entitäten- und Beziehungstabellen ab

- (vgl. C. Batini, S. Ceri, S.B. Navathe. Conceptual Database Design - An Entity Relationship Approach, Benjamin/Cummings, Redwood City, Kalifornien, 1992)

2. Sammle so genannte funktionale Abhängigkeiten aus der Anforderungsdefinition und erzeuge daraus ein relationales Schema in Normalform

- (Im Trend 1970...80).
- Ausführlich in der Literatur beschrieben (vgl. S.M. Lang, P.C. Lockemann. Datenbankeinsatz. Springer, Berlin u.a., 1995).

Rückblick

- Konzepte des relationalen Modells
 - Entitäten und Beziehungen als Relationen
 - Formale Betrachtung von Relationen – Wertebereiche, Attribute, Tupel
 - Attribute mit speziellen Rollen – Schlüssel, referentielle Integrität
- Relationale Datenbanken und Datenbankschemata
- Vom ER-Diagramm zum relationalen Datenmodell

