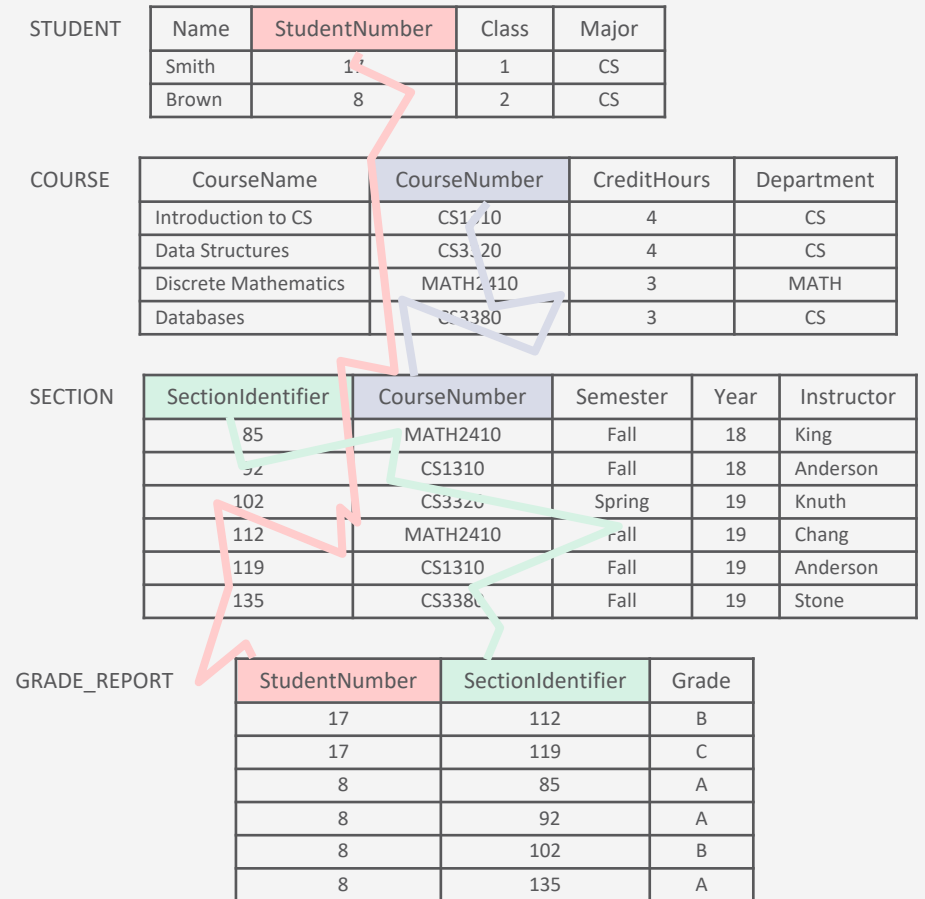


Das relationale Datenmodell

Datenbanken



Inhalte: Datenbanken (DBs)

1. Einführung

- Anwendungen
- Datenbankmanagementsysteme

2. Datenbank-Modellierung

- Entity-Relationship-Modell (ER-Modell)
- Beziehung zwischen ER und UML

3. Das relationale Modell

- Relationales Datenmodell (RM)
- Vom ER-Modell zum RM
- Relationale Algebra als Anfragesprache

4. Datenbank-Entwurf

- Funktionale Abhängigkeiten
- Normalformen

5. Structured Query Language (SQL)

- Datendefinition
- Datenmanipulation

6. Anfrageverarbeitung

- Architektur
- Indexierung
- Anfragepläne, Optimierung

7. Transaktionen

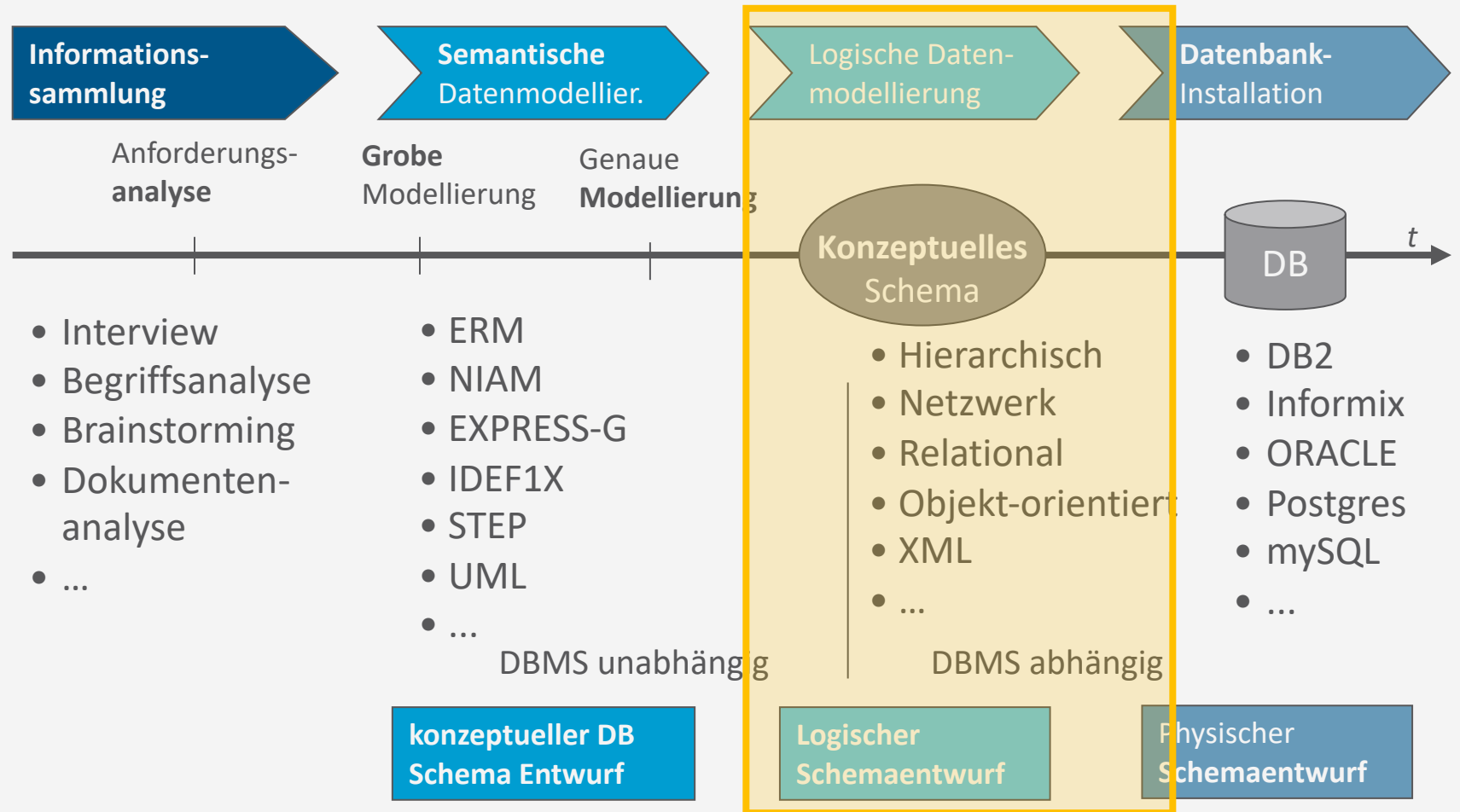
- Transaktionsverarbeitung, Schedules, Sperren
- Wiederherstellung

8. Erweiterung

- Noch offen: verteilte DBs, deduktive DBs (DataLog → Logik-Verbindung), XML, Graph-DBs

Phasen des DB-Entwurfs

- Ausblick: Von der Anwendung her
 - Teil von 2. DB-Modellierung
 - Methode: ERM
 - Teil von 3. Das relationale Datenmodell
 - Methode: relationale Modellierung
 - Teil von 4. DB-Entwurf
 - Teil von 5. SQL & Übergang zu „Hinter den Kulissen“



Übersicht: 3. Das relationale Datenmodell

A. *Relationales Datenmodell*

- Relationen, Attribute, relationale Datenbanken und –schemata
- Schlüssel: Primärschlüssel, Fremdschlüssel, referentielle Integrität

B. *Entwurf relationaler Schemata*

- Vom ER-Diagramm zum relationalen Modell

C. *Relationale Algebra*

- $\pi, \rho, \sigma, \cup, \cap, -, \times, \bowtie$
- Minimalität
- Aggregieren, gruppieren
- Einfügen, löschen, aktualisieren

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 (über Schlüssel)

STUDENT

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

COURSE

CourseName	CourseNumber	CreditHours	Department
Introduction to CS	CS1110	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

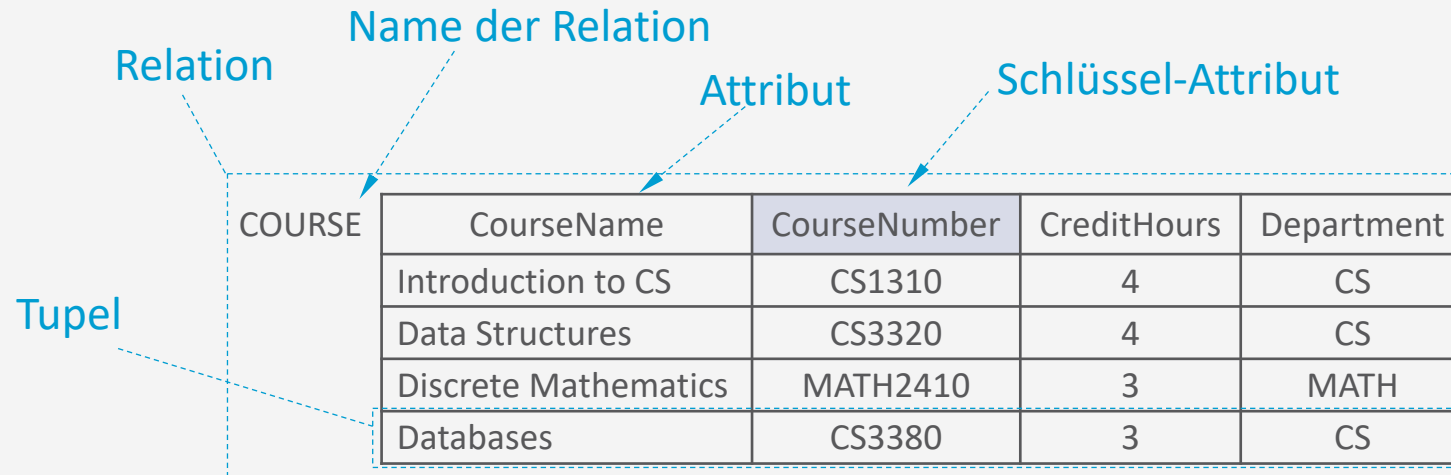
StudentNumber	SectionIdentifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

CourseNumber	PrerequisiteNumber
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Relationen

- Relation, Name der Relation, Attribut und Tupel:



The diagram illustrates a table representing a relation. The table has a header row with columns: CourseName, CourseNumber, CreditHours, and Department. The first column is labeled 'COURSE'. The 'CourseNumber' column is highlighted as the 'Schlüssel-Attribut' (key attribute). The entire table is enclosed in a dashed box labeled 'Relation'. A specific row is highlighted with a solid box and labeled 'Tupel'. Arrows point from the labels to the corresponding parts of the table.

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

- 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-ddddd** 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.
 - Persistent identifier
 - Sozialversicherungsnummer

Relationenschemata und Relationen

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

Relation r Attribute Relationenschema R Grad 4

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

Relationenschemata und Relationen

- Beispiel:
 - Relationenschema: COURSE(CourseName, CourseNumber, CreditHours, Department)
 - Wertebereiche für Attribute (hier alle benutzerdefiniert)
 - $\text{dom}(\text{CourseName}) = \text{CourseNames}$ (set of course names)
 - $\text{dom}(\text{CourseNumber}) = \text{CourseNumbers}$ (set of course numbers)
 - $\text{dom}(\text{CreditHours}) = \text{CreditHours}$ (set of credit hours)
 - $\text{dom}(\text{Department}) = \text{Departments}$ (set of departments)
 - Allgemeinere Alternativen: String, String, Integer, String

Relationenschema *R*
Grad 4

Relation *r* Attribute

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
 - $R(A_1, \dots, A_n)$ ein Relationenschema n -ten Grades
 - $t = \langle v_1, \dots, v_n \rangle$ 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

$t_1[CreditHours] = t_1.CreditHours$

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
 - course(COURSE) hat vier tatsächliche Tupel

Wie sehen alle möglichen Tupel aus?

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 m der Relation

$$m \ll |\text{dom}(A_1)| \cdot \dots \cdot |\text{dom}(A_n)|$$

- Wertebereich kann einschränkend wirken

(String vs. Menge von Kursnamen)

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

- Logische Zusammenhänge zwischen Attributen schränken die möglichen Tupel weiter ein

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
 → keine Reihenfolge im relationalen 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: Re-Design 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	3 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
 - Wert ist nicht vorhanden
 - Wert ist im Kontext des Tupels nicht zutreffend
 - Es ist unbekannt, ob der Wert nicht vorhanden oder nicht zutreffend ist
 - Kann mit der *Closed World Assumption* brechen
 - **NULL** \neq **NULL** (in einer Spalte)
 - Unbekannte, aber möglicherweise unterschiedliche Werte

Welche Probleme mit NULL Ihnen ein?

Noch nicht bekannt

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

Das relationale Datenmodell und relationale DBs

- 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

GRADE_REPORT

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

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

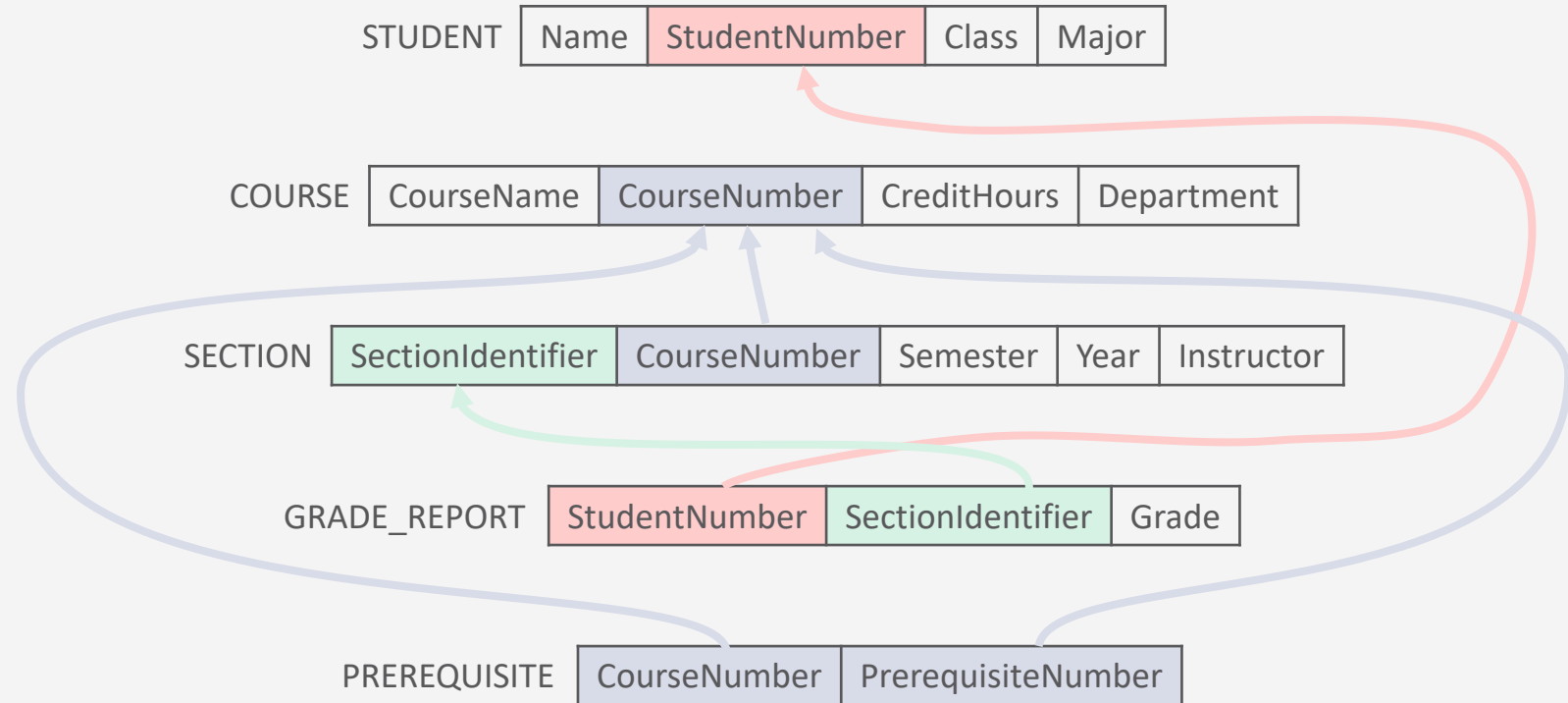
SectionIdentifier	CourseNumber	Semester	Year	Instructor
85	MATH2410	Fall	18	King
92	CS1310	Fall	18	Anderson
102	CS3320	Spring	19	Knuth
112	MATH2410	Fall	19	Chang
119	CS1310	Fall	19	Anderson
135	CS3380	Fall	19	Stone

PREREQUISITE

CourseNumber	PrerequisiteNumber
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

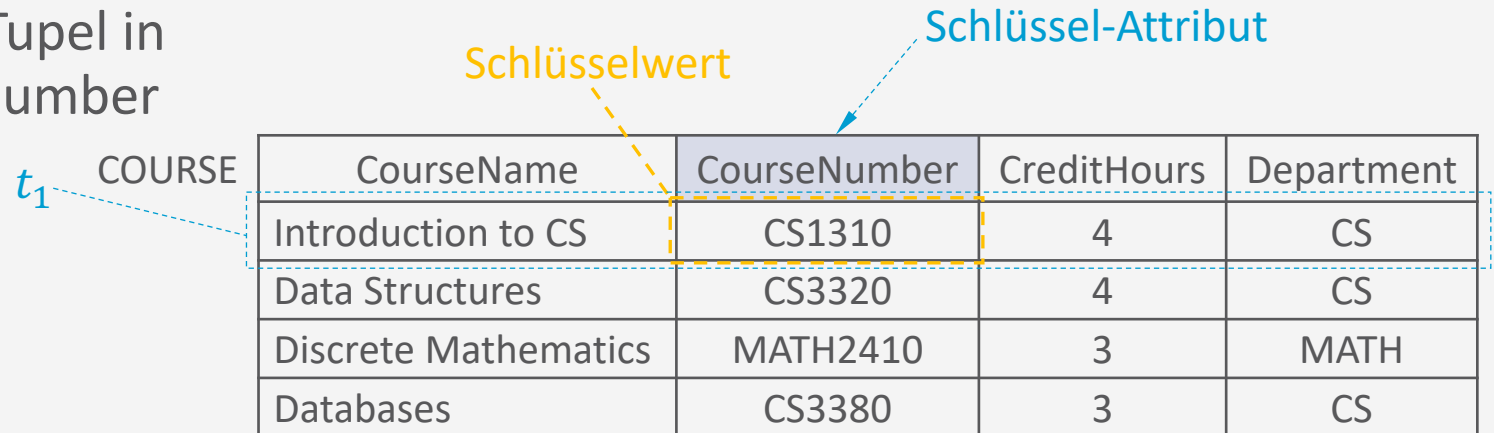
Schlüssel

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
t_1	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 zwei Tupel eines Relationenzustands $r(R)$ nie 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)

gewählter Primärschlüssel

Schlüsselkandidaten

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

Beziehungen zwischen Tupeln

- 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 führen zu reflexiven Fremdschlüsseldeklarationen ($R_1 = R_2$)
 - Beispiel: Angestellte : Vorgesetzte
 - Beispiel: CourseNumber : PrerequisiteNumber

Fremdschlüssel

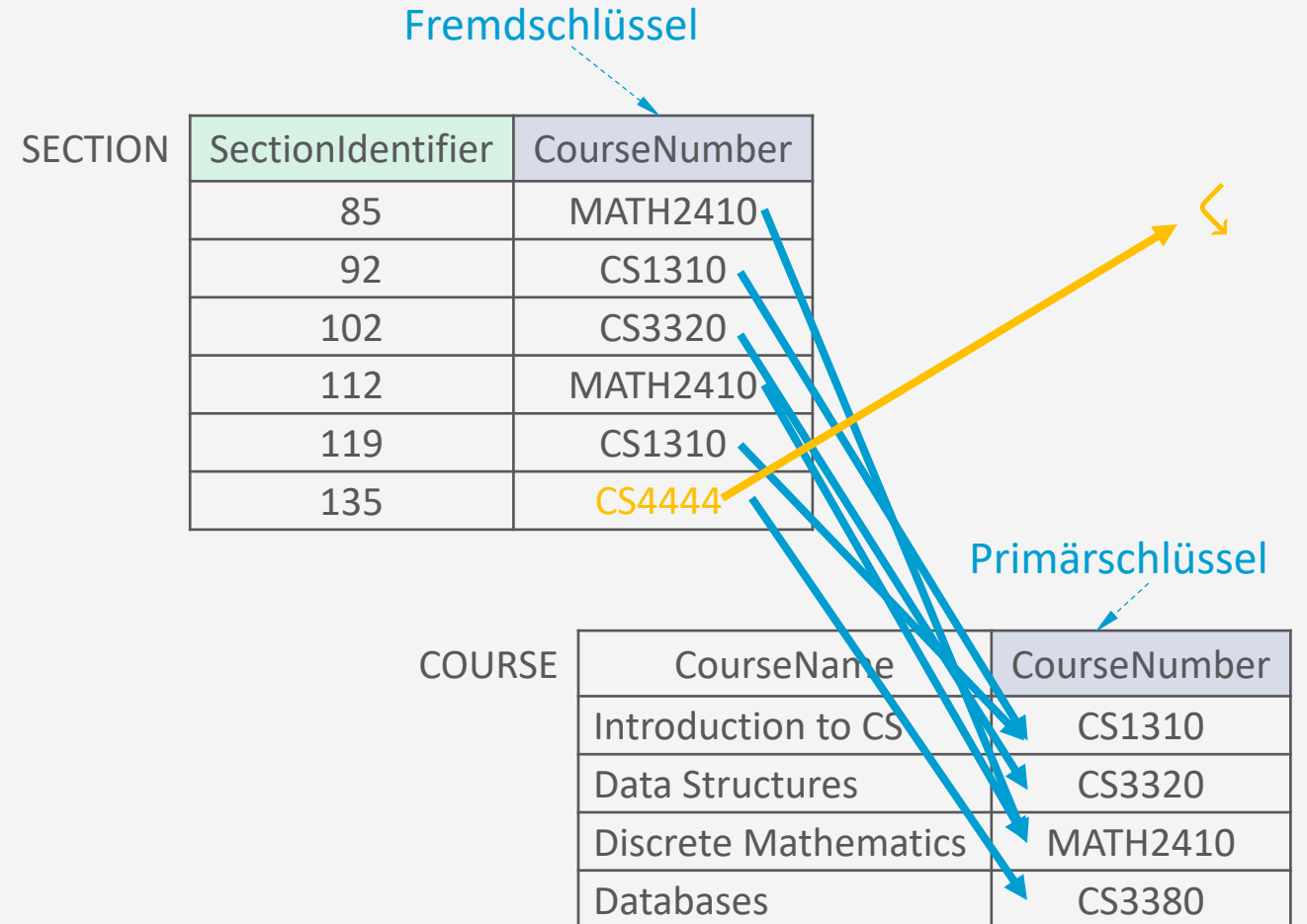
SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	23	King
	92	CS1310	Fall	23	Anderson
	102	CS3320	Spring	24	Knuth
	112	MATH2410	Fall	24	Chang
	119	CS1310	Fall	24	Anderson
	135	CS3380	Fall	24	Stone

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

Primärschlüssel

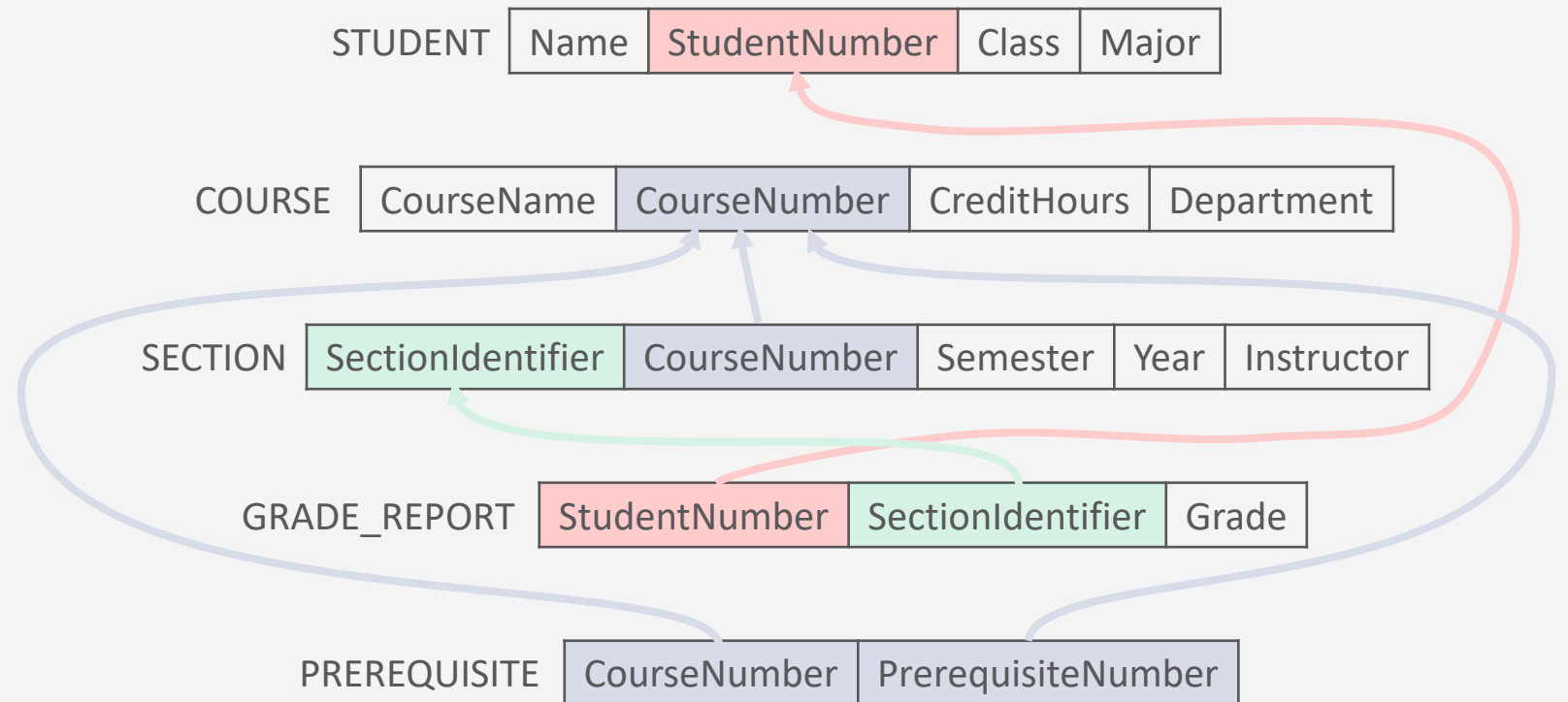
Fremdschlüssel: Definition

- Eine Attributmengung FK im Schema R_1 ist ein Fremdschlüssel von R_1 , der auf Schema R_2 referenziert, falls gilt:
 1. Die Attribute in FK haben die gleichen Wertebereiche wie die Primärschlüssel-Attribute PK von R_2
 - Die Attribute FK gelten als Referenz auf R_2
 2. Ein Wert von FK in einem Tupel t_1 des aktuellen Zustands $r_1(R_1)$ ist
 - NULL (keine Beziehung)
 - 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

- Durch Änderungen im Relationenzustand Bedingungen referentieller Integrität möglicherweise verletzt
- Integrität überprüfen notwendig beim
 - Einfügen eines neuen Fremdschlüsselwertes in eine Beziehungstabelle
 - Referenziertes Objekt mit diesem Wert als Primärschlüssel muss existieren
 - Aktualisieren eines Primärschlüsselwertes
 - Alle Referenzen müssen aktualisiert werden
 - Löschen eines Tupels aus einer Relation
 - Auf dieses Tupel dürfen keine Referenzen bestehen

SECTION

SectionIdentifier	CourseNumber
85	MATH2410
92	CS1310
102	CS3320
112	MATH2410
119	CS1310
135	CS3380

Fremdschlüssel

Was können wir machen?

Primärschlüssel

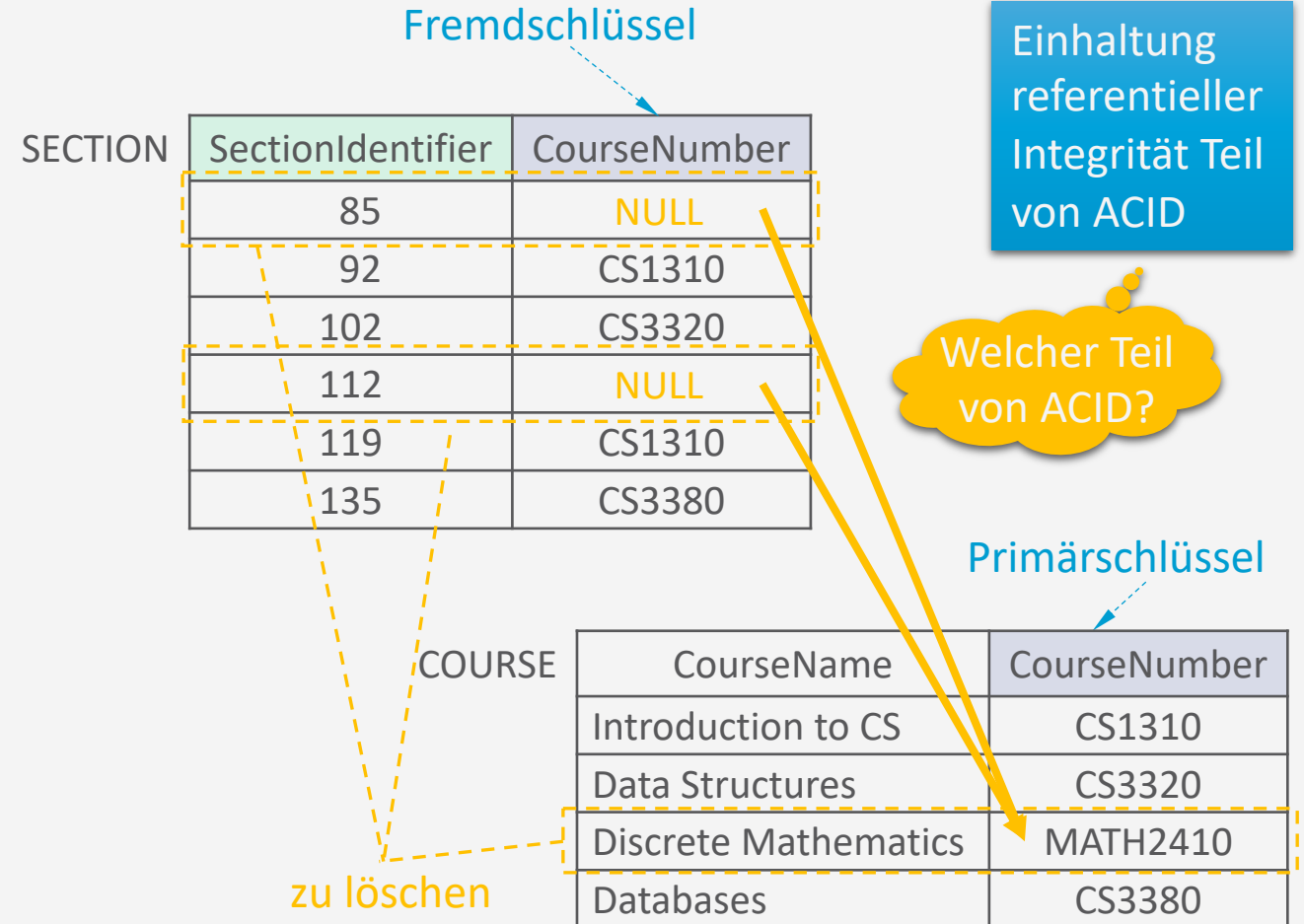
COURSE

CourseName	CourseNumber
Introduction to CS	CS1310
Data Structures	CS3320
Discrete Mathematics	MATH2410
Databases	CS3380

zu löschen

Referentielle Integrität: Konflikte – Löschen

- Gibt es noch Referenzen, bieten sich mehrere Möglichkeiten an, u.a.:
 - Fehlermeldung erzeugen
 - Löschoperation propagieren, so dass das referenzierende Tupel ebenfalls gelöscht wird (→ kaskadiertes Löschen)
 - Referenzen durch Setzen des Fremdschlüssels auf einen Nullwert ungültig machen, sofern dieser nicht Bestandteil des Schlüssels ist
- DBMS bieten Unterstützung zur Verhinderung solcher Konflikte



Zwischenzusammenfassung

- 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: Elemente der Relation (ein Tupel)
 - Die Zahl der Zeilen ist variabel und wird **Kardinalität** der Relation genannt
 - Spalten: Attribute der Relation

Übersicht: 3. Das relationale Datenmodell

A. *Relationales Datenmodell*

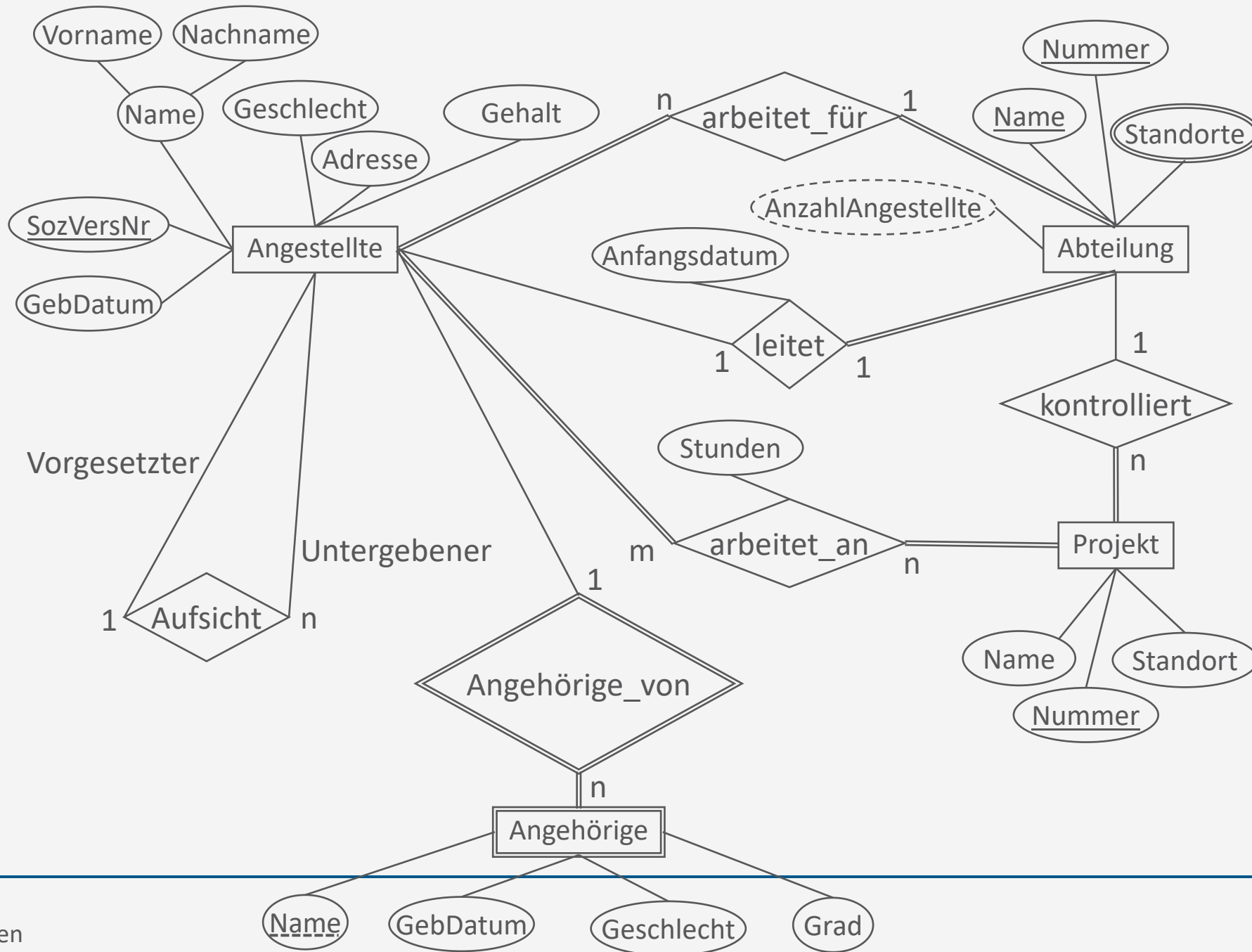
- Relationen, Attribute, relationale Datenbanken und –schemata
- Schlüssel: Primärschlüssel, Fremdschlüssel, referentielle Integrität

B. *Entwurf relationaler Schemata*

- Vom ER-Diagramm zum relationalen Modell

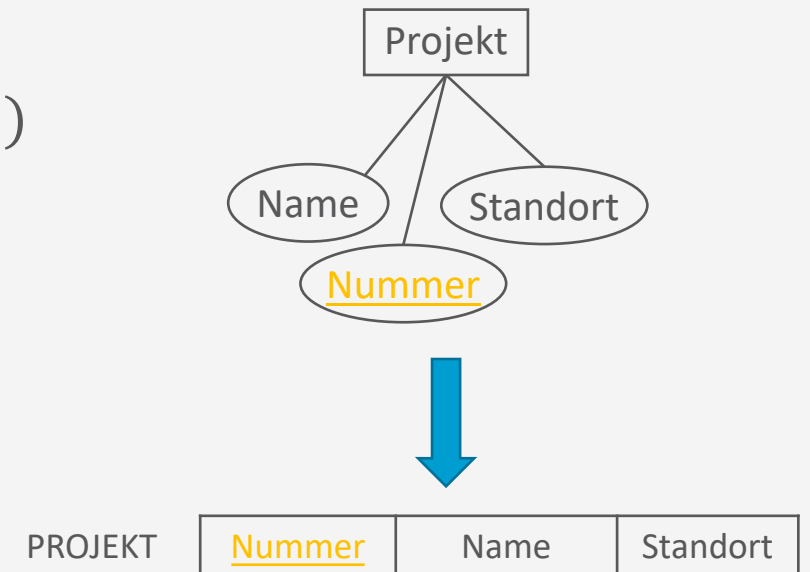
C. *Relationale Algebra*

- $\pi, \rho, \sigma, \cup, \cap, -, \times, \bowtie$
- Minimalität
- Aggregieren, gruppieren
- Einfügen, löschen, aktualisieren



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

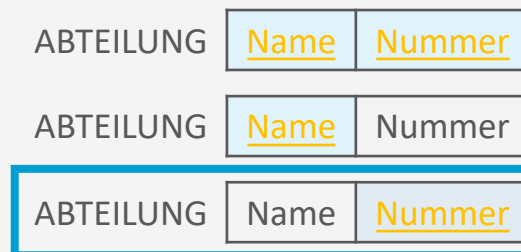
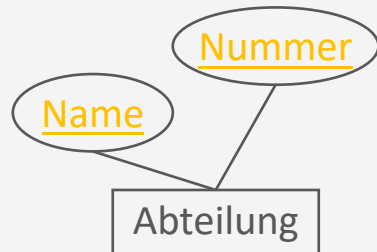


Wertebereiche

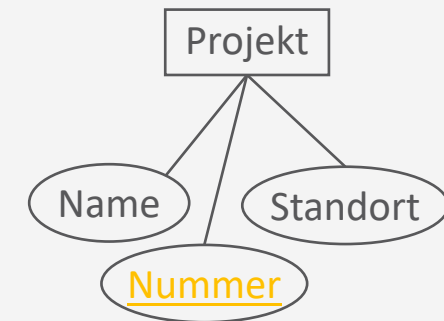
- Nummer: Integer
- Name: String
- Standort: String
- Oder benutzerdefiniert

Relationale Darstellung von Entitätstypen

- *Schlüssel-Attribute*
 - In ER-Diagramm identifizierende Attribute angegeben
 - Vorsicht geboten! Nicht immer direkt nutzbar
 - Schlüssel im relationalen Modell **minimale** Superschlüssel
 - Interpretation der ER-Schlüssel teilweise nicht eindeutig
 - Beispiel: Name und Nummer zusammen eindeutig? Allein eindeutig?



- Entscheidung (evtl. nach Rücksprache mit Kunden);
 Nummer als Schlüssel von Vorteil, da
- Nummer kleiner als Name
 - Vergleiche weniger aufwendig



Wertebereiche

- Nummer: Integer
- Name: String
- Standort: String
- Oder benutzerdefiniert

Relationale Darstellung von Entitätstypen

- *Mehrwertige Attribute*
 - Nicht direkt ins relationale Datenmodell übersetzbar
 - Relationenzustand für ABTEILUNG (Name, Nummer, Standort)

Warum?

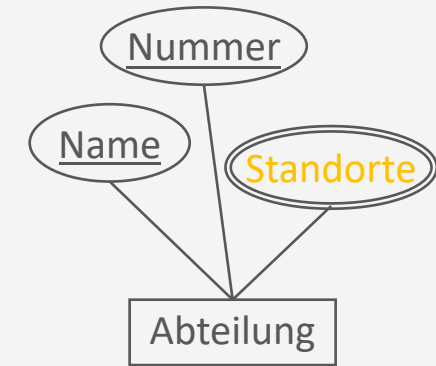
Schlüssel nicht mehr eindeutig!

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	AbtNr	Standort
	42	A
	42	B
	42	C



ABTEILUNG	Name	<u>Nummer</u>

ABT_STNDRT	<u>AbtNr</u>	Standort

Fremdschlüssel

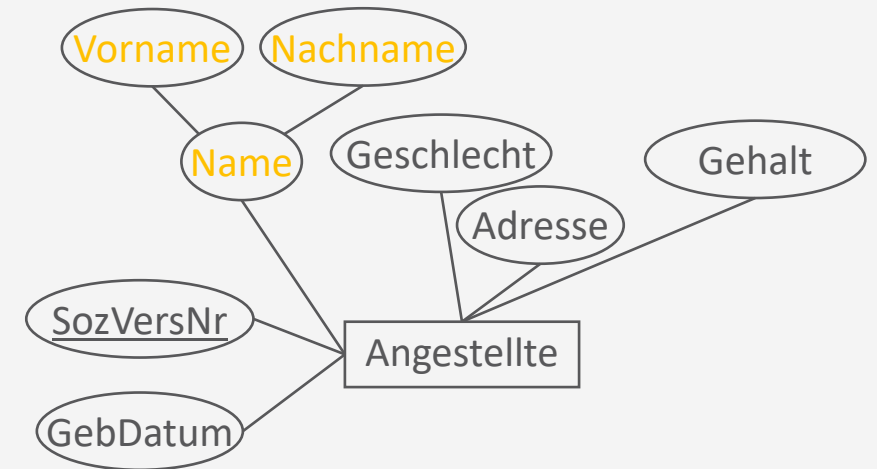
Relationale Darstellung von Entitätstypen

- *Zusammengesetzte Attribute*
 - Als ein Attribut in Relation aufnehmen
 - Gibt Teilung auf
 - Als einzelne Attribute in Relation aufnehmen
 - Gibt Zusammengehörigkeit auf
- Beispiel:
 - Entweder

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

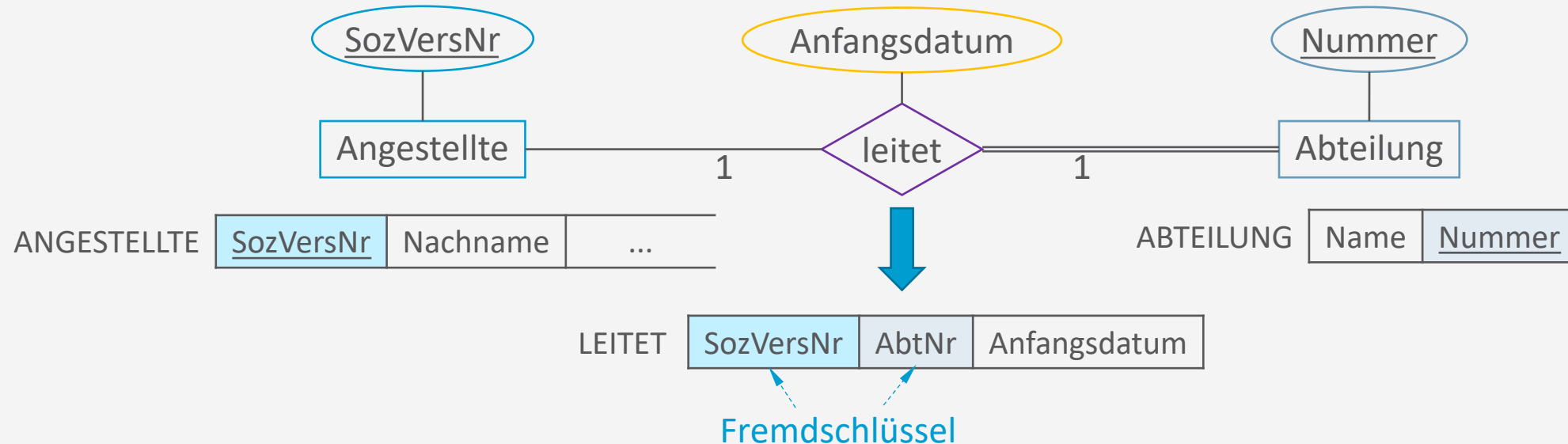
- Oder

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



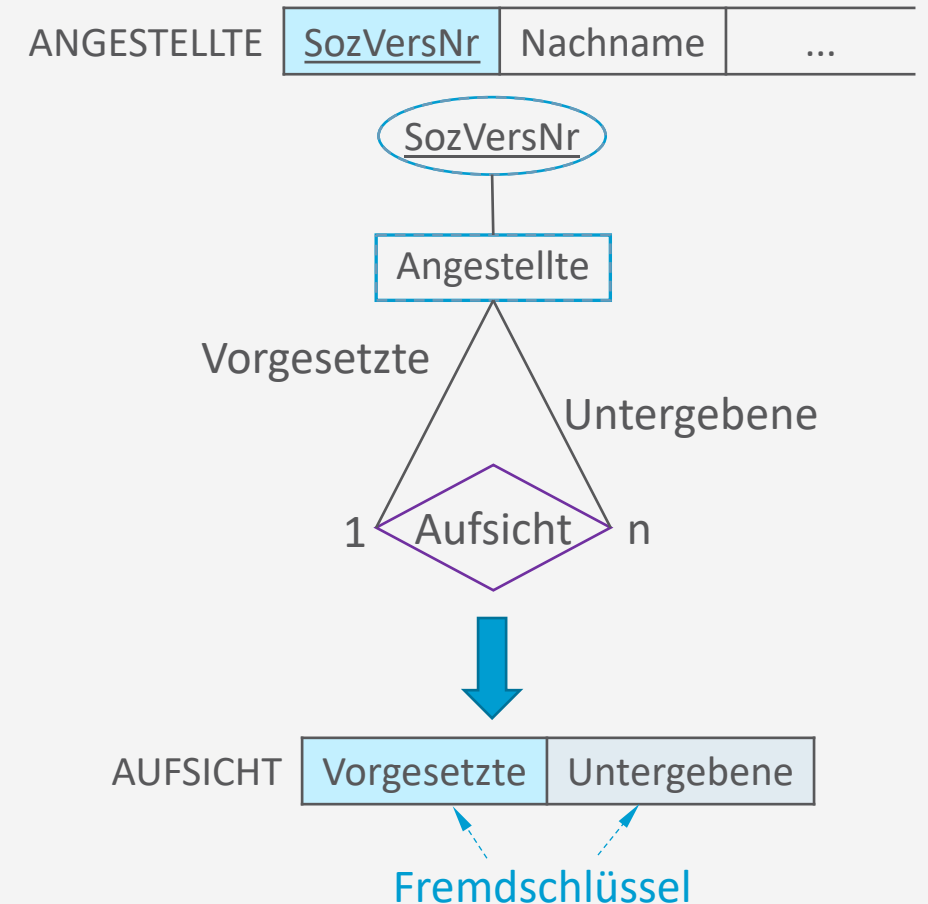
Relationale Darstellung von Assoziationen

- 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 $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$



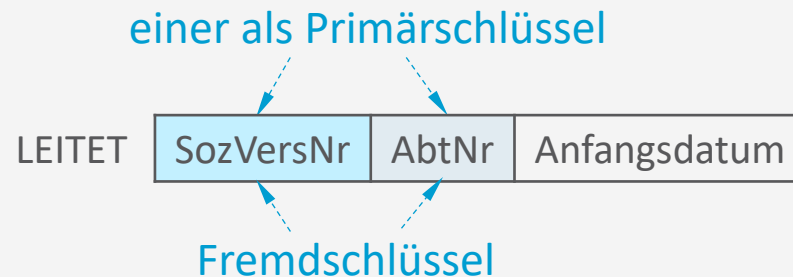
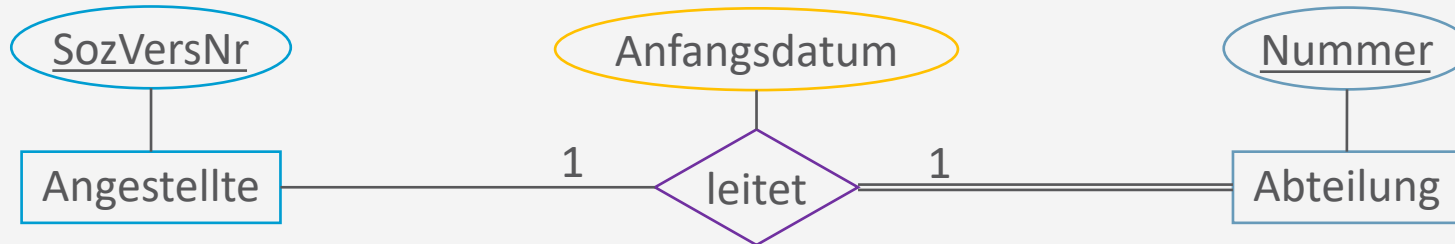
Relationale Darstellung von Assoziationen

- Auch bei rekursiven Relationen, dann $E_1 = E_2$
 - Beispiel: Vorgesetzte / Untergebene
- Beziehung R mit Attributen A_1, \dots, A_n
- Zwischen Entitäten E_1 und $E_2 = E_2$ mit Schlüsseln $P_{1,1}, \dots, P_{1,l}$ und $P_{1,1}, \dots, P_{1,l}$
- Wird zu Relation $R(P_{1,1}, \dots, P_{1,l}, P_{1,1}, \dots, P_{1,l}, A_1, \dots, A_n)$



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 $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$
- Primärschlüssel bestimmen über Kardinalitäten
 - **1:1 Beziehung**

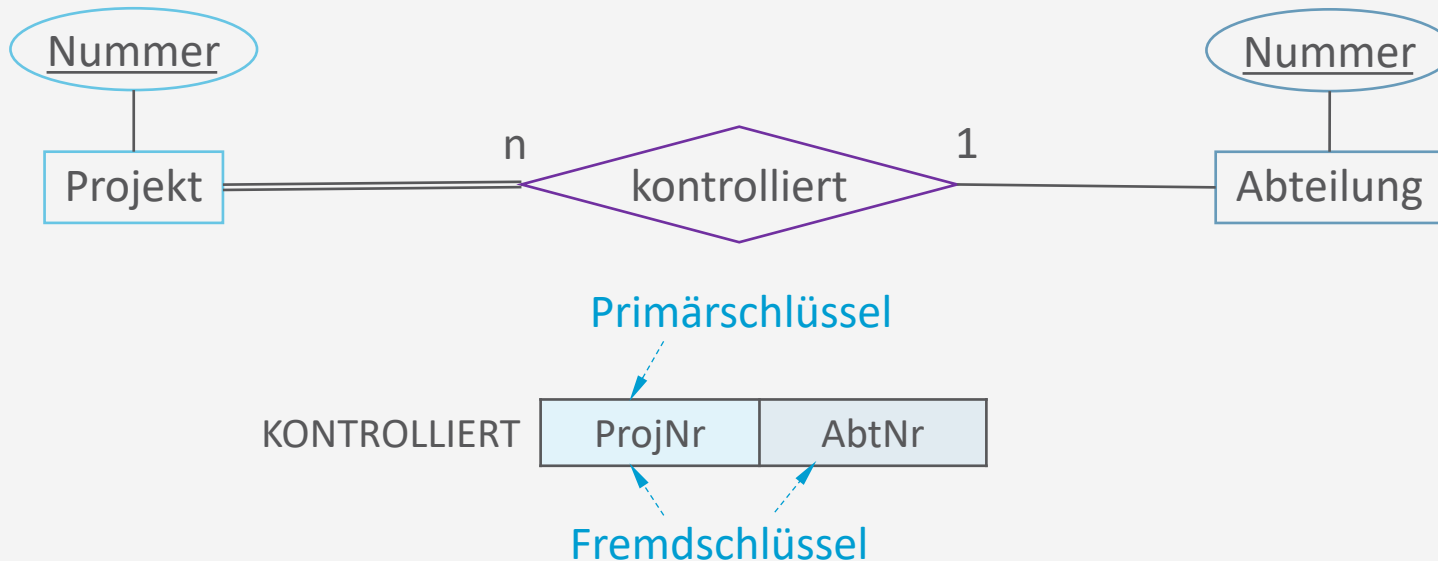


Bei 1:1 Beziehungen kommen beide Fremdschlüssel maximal einmal vor, können also als Schlüssel verwendet werden.

LEITET	SozVersNr	AbtNr	Anfangsdatum
	A1	42	01.11.2020
	B2	21	15.02.2010
	C4	54	01.01.1999

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 $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$
- Primärschlüssel bestimmen über Kardinalitäten
 - 1:n Beziehung

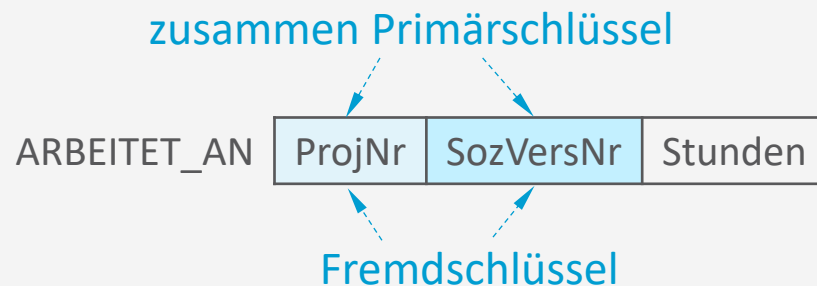
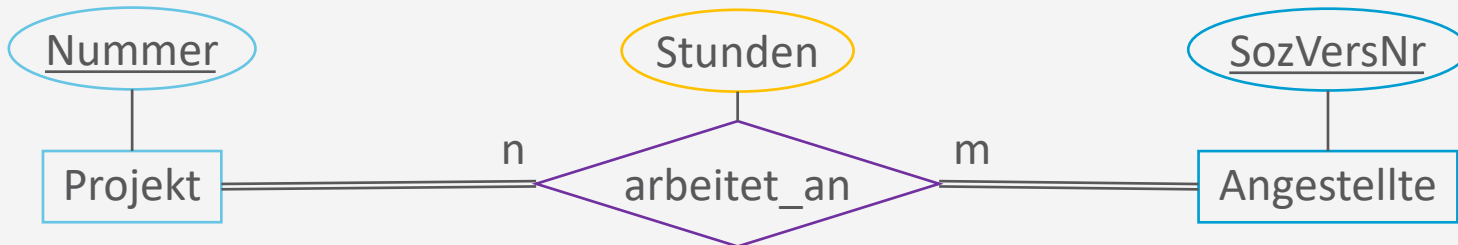


Bei 1:n Beziehungen kommt der Fremdschlüssel der Entität E_1 maximal einmal vor, kann also als Schlüssel verwendet werden.

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 $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$
- Primärschlüssel bestimmen über Kardinalitäten
 - **n:m Beziehung**

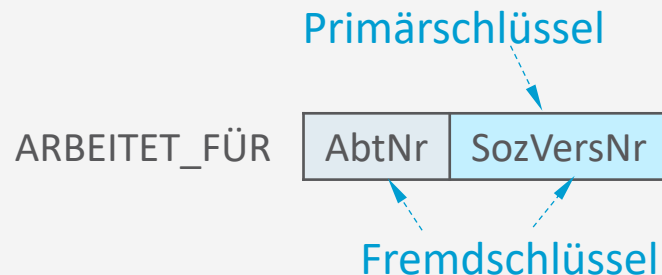
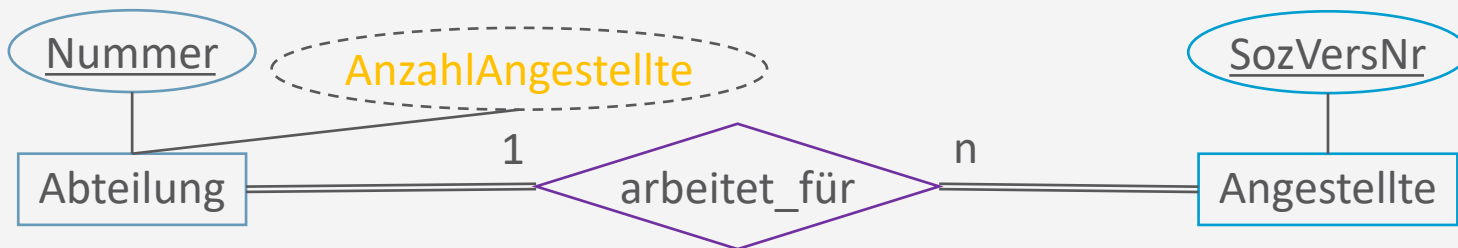


Bei n:m Beziehungen kommt die Kombination der Fremdschlüssel der Entitäten E_1, E_2 maximal einmal vor, können zusammen als Schlüssel verwendet werden.

ARBEITET_AN	ProjNr	SozVersNr	Stunden
	1	A1	20
	1	B2	10
	2	A1	10
	3	A1	10
	3	C4	30

Relationale Darstellung: Abgeleitete Attribute

- **Abgeleitete Attribute** ergeben sich aus Zuständen anderer Teile
 - „AnzahlAngestellte“ ergibt sich durch „arbeitet_für“



Klassische Anfrage an DB (Wie?
- nächster Block zu relationaler Algebra)

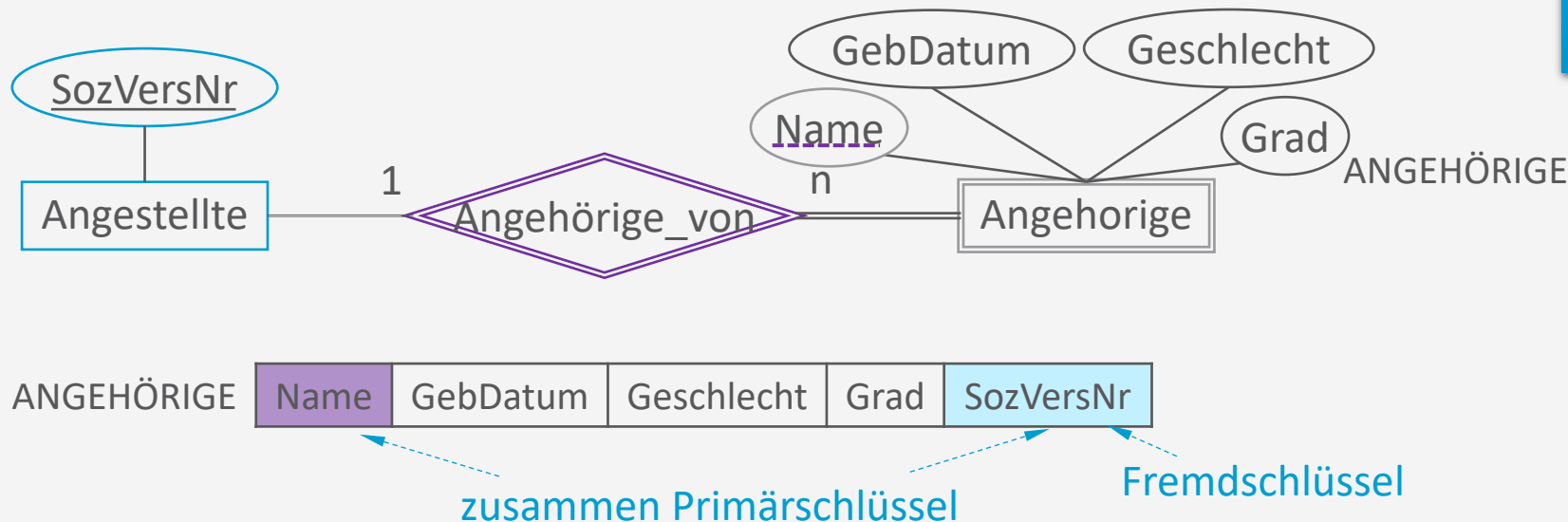
- Anzahl an Angestellten, die für eine Abteilung (z.B. 42) arbeiten (Ergebnis: 2)

ARBEITET_FÜR	AbtNr	SozVersNr
	42	A1
	42	B2
	54	C4

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 $R(P_{1,1}, \dots, P_{1,l}, P_{2,1}, \dots, P_{2,m}, A_1, \dots, A_n)$

Partielle Schlüssel der schwachen Entität werden zusammen mit den Fremdschlüsseln der starken Entität zum Schlüssel.

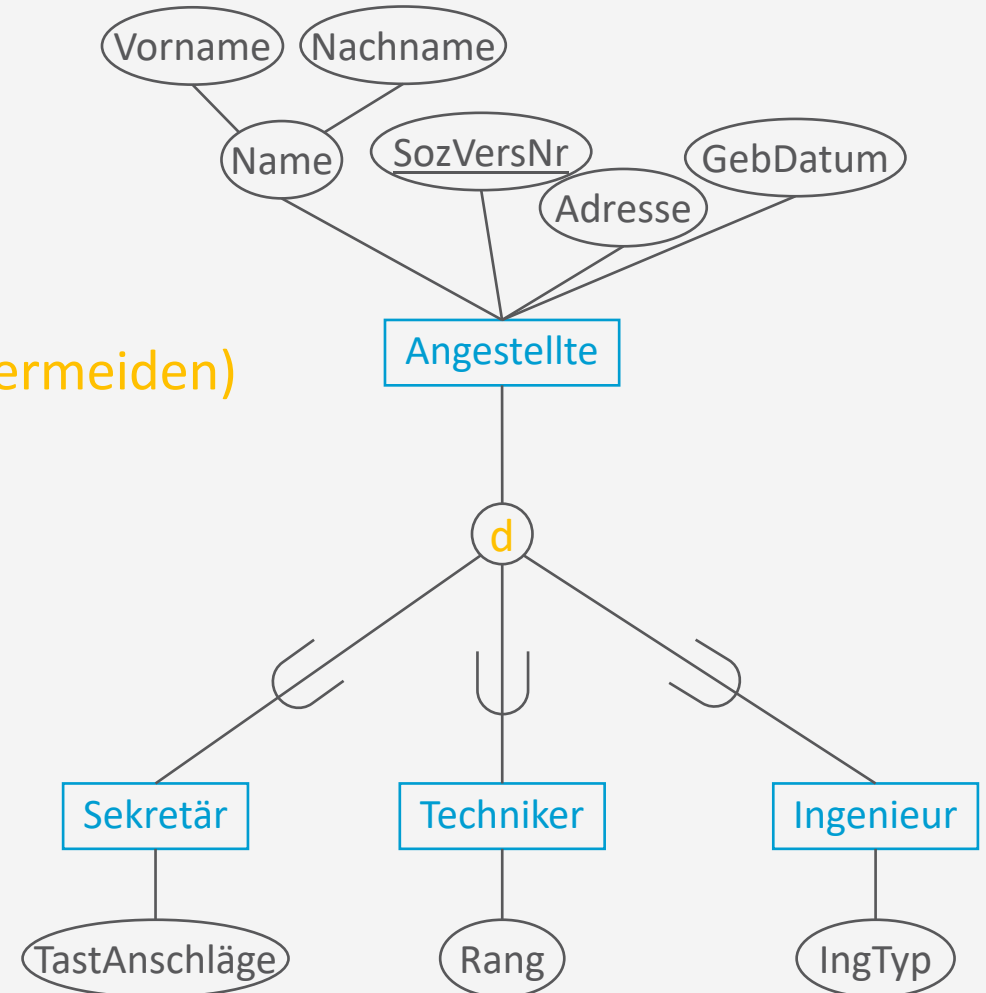
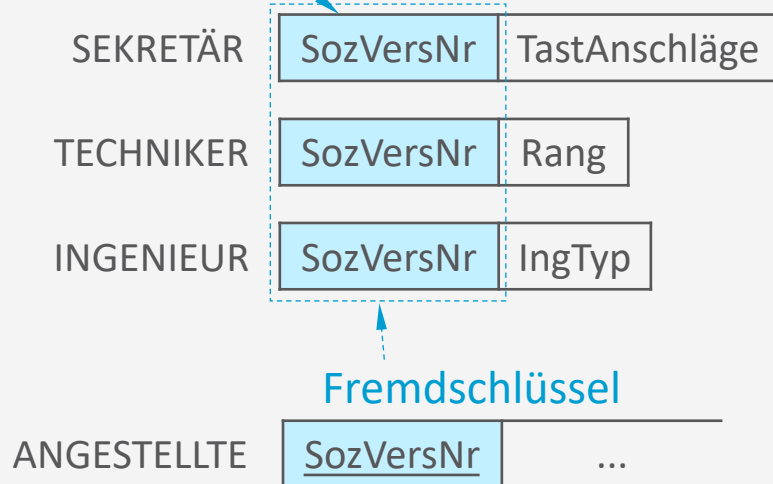


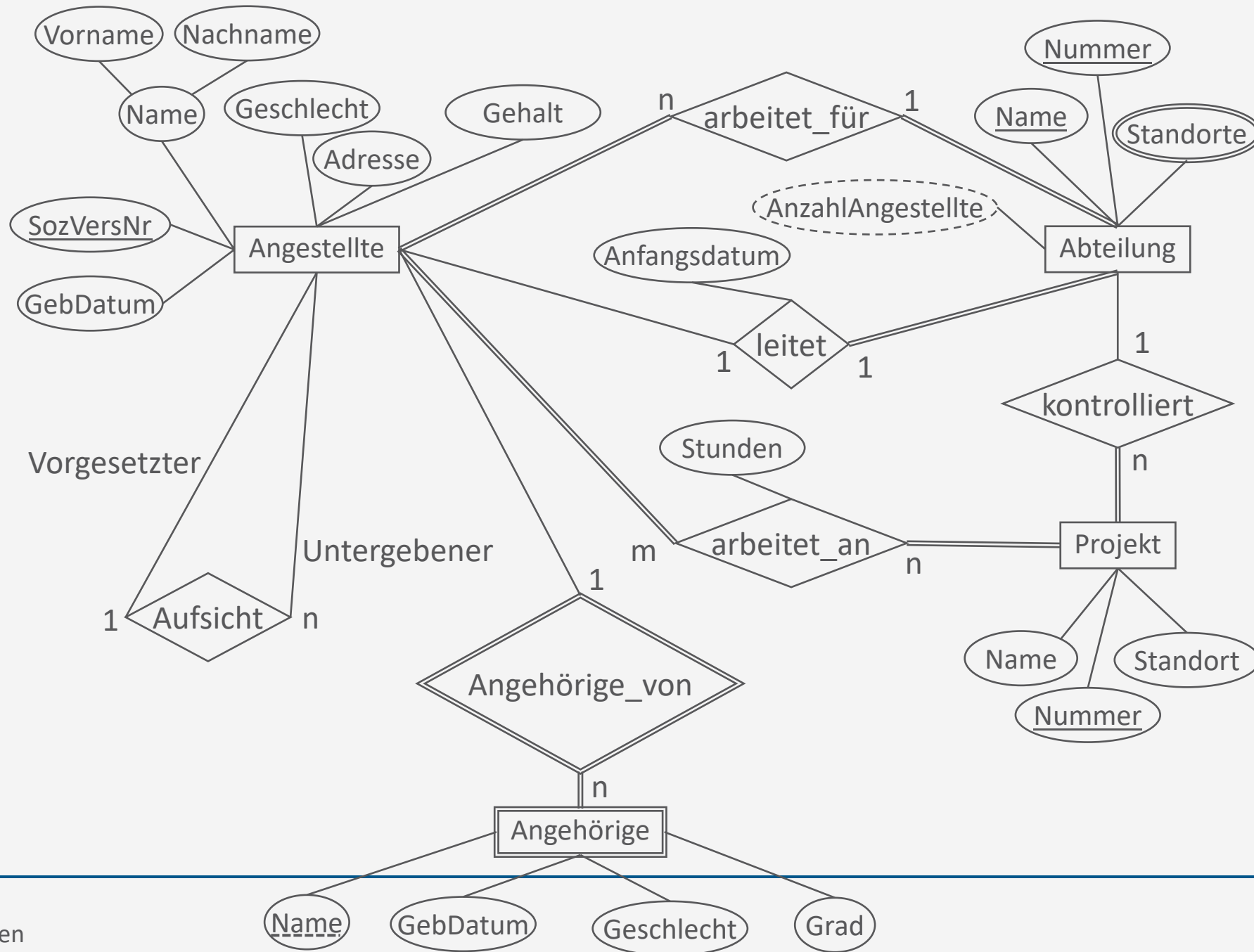
Name	Geb.	Geschl.	Grad	Soz.
Alice	A1
Eve	A1
Bob	A1
Charlie	B2
Judy	C4

Relationale Darstellung: EER-Modell

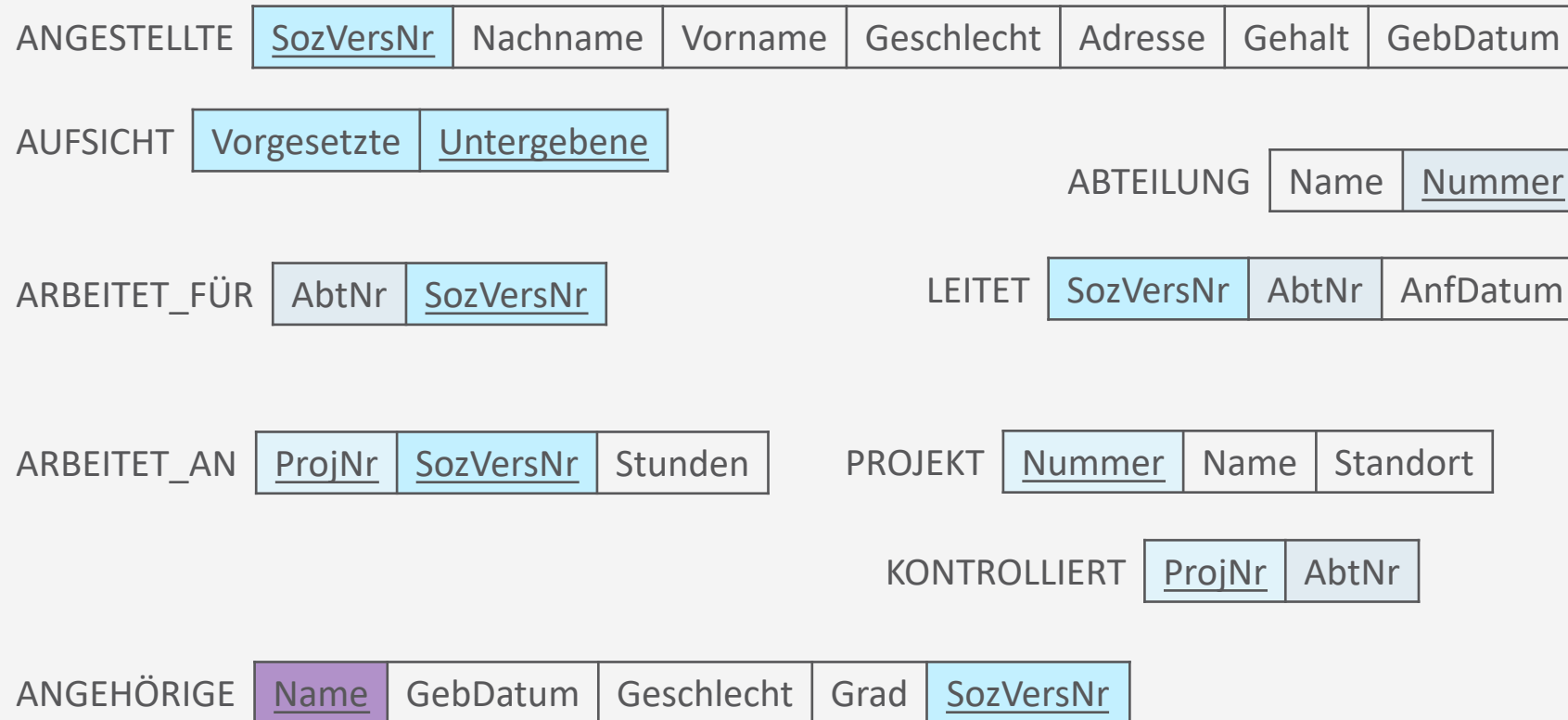
- EER-Konstrukte: Spezialisierung, Generalisierung
 - Subklasse wird zur Relation
 - Fremdschlüssel von Superklasse
 - Keine Vererbung der Attribute (doppelte Speicherung vermeiden)

Primärschlüssel





Entsprechendes relationales Schema



Neun Relationen

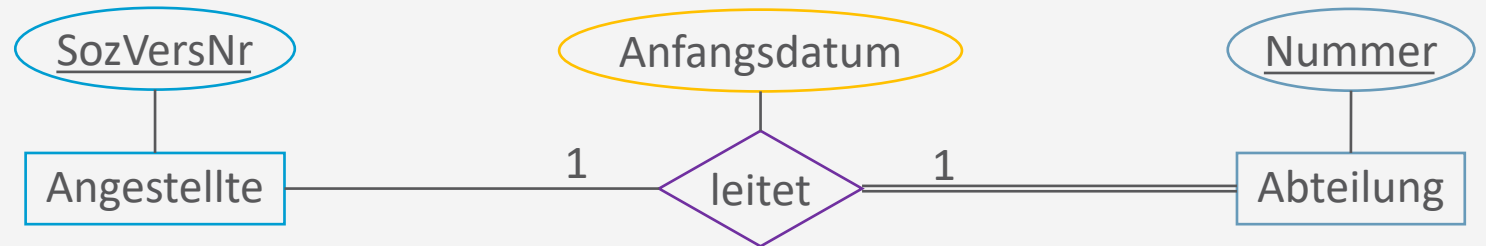
Verfeinerung des relationalen Schemas

- Relationen mit gleichem Schlüssel kann man zusammenfassen
- **Aber nur diese und keine anderen!**

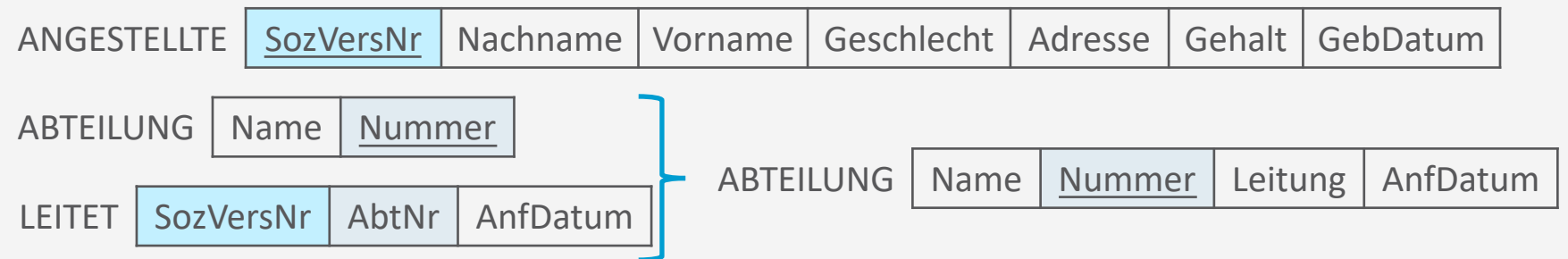
- Beispiel: Relation LEITET

- SozVersNr und AbtNr als Schlüssel möglich, jeweils Schlüssel von Angestellte bzw. Abteilung

1. Wenn „SozVersNr“ Schlüssel
→ LEITET zu ANGESTELLTE
2. Wenn „AbtNr“ Schlüssel
→ LEITET zu ABTEILUNG

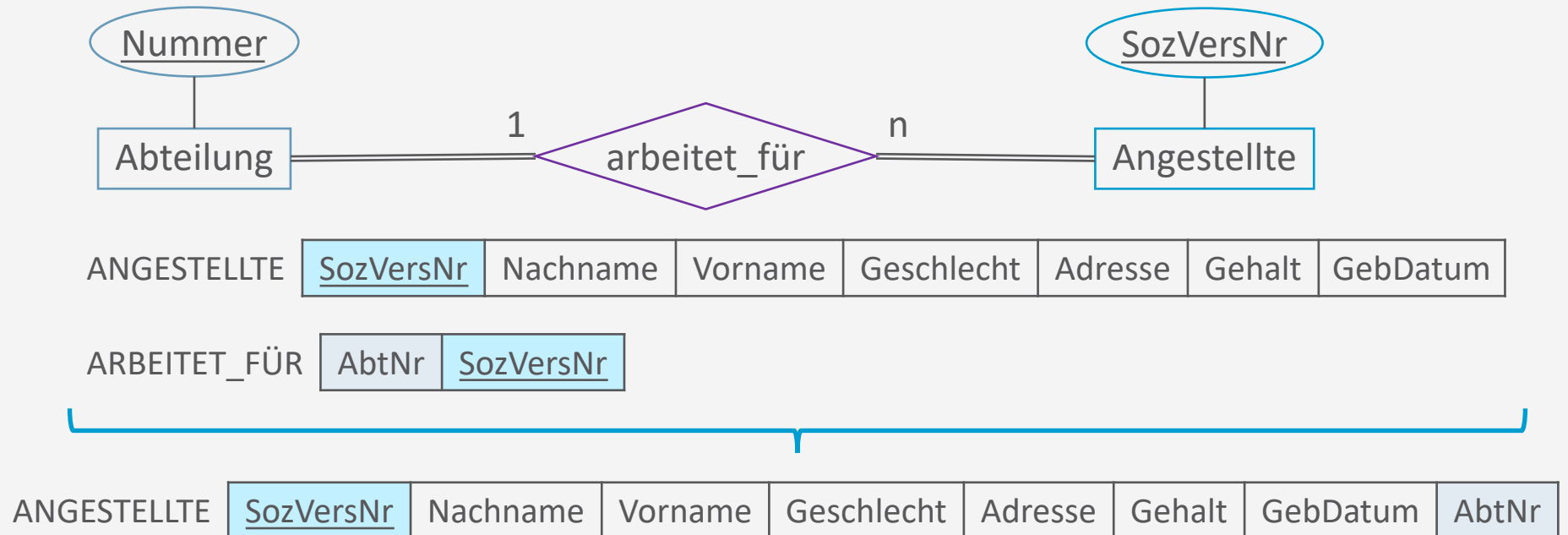


- Da jede Abteilung eine Leitung hat, aber nicht jeder Angestellte leitet, ist 2. hier besser



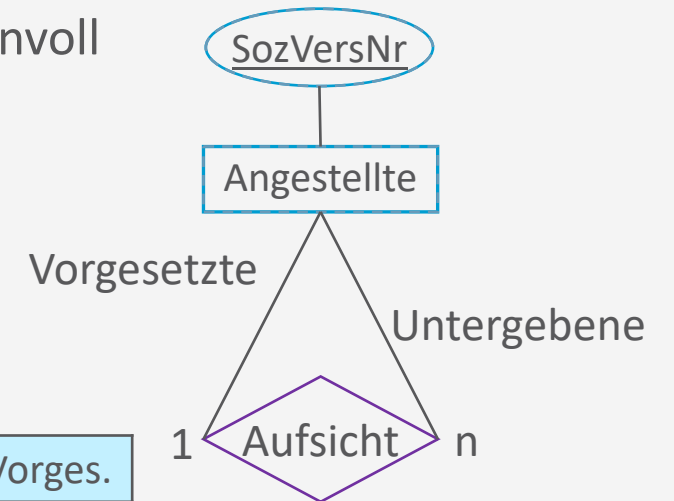
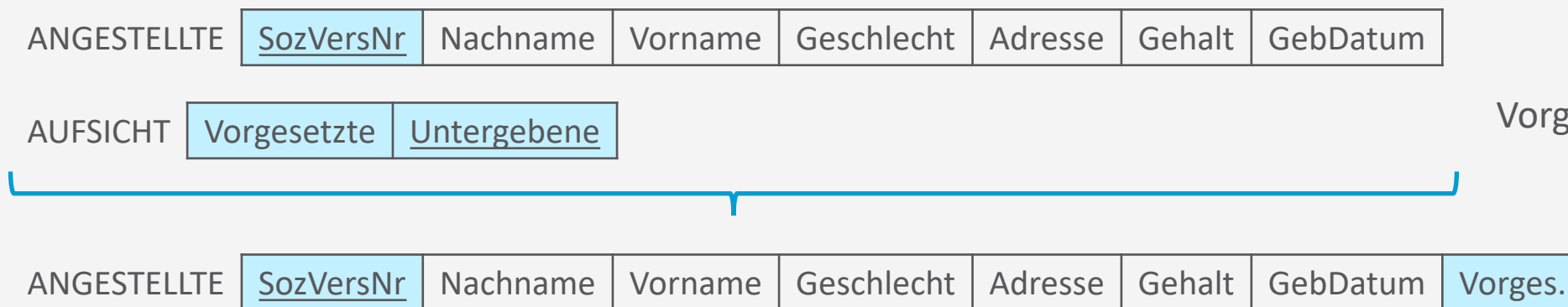
Verfeinerung des relationalen Schemas

- 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



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
 - Also Achtung: wenn viele Zellen mit NULL Werten nötig, Beziehung nicht so schnell ersichtlich
 - Beispiel:
 - Wenn fast jeder eine vorgesetzte Person hat, dann Zusammenschluss sinnvoll



Verfeinertes relationales Schema

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

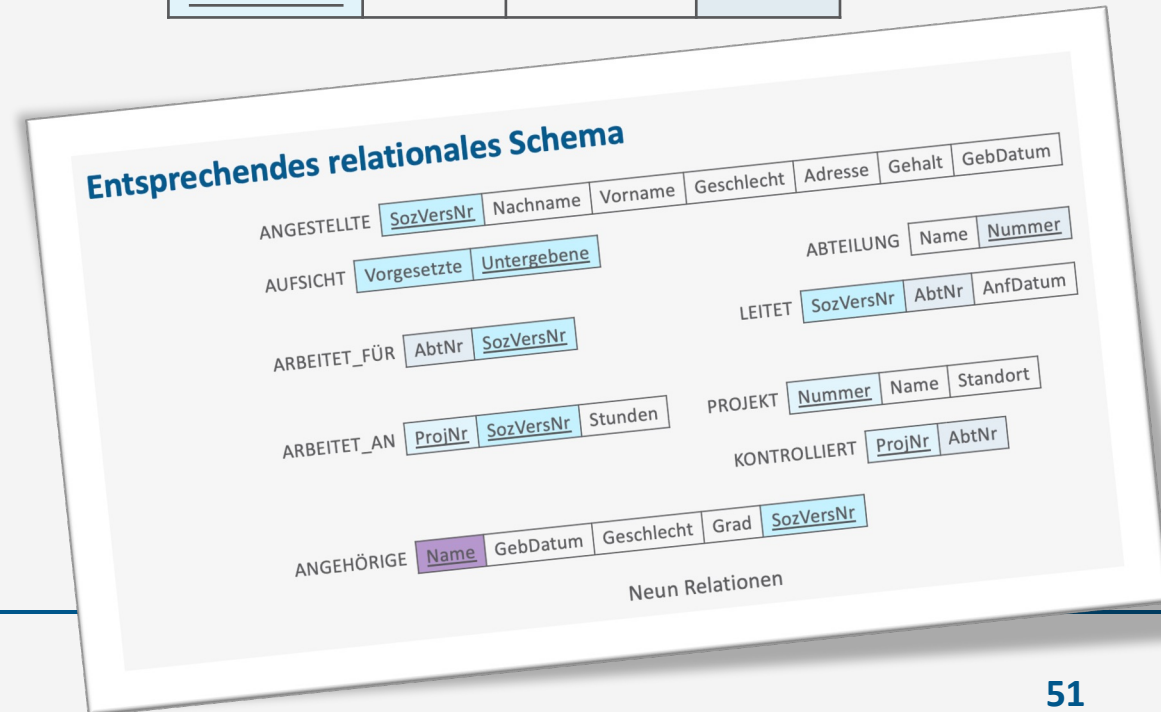
ABTEILUNG	Name	<u>Nummer</u>	Leitung	AnfDatum
-----------	------	---------------	---------	----------

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>
------------	-------------	----------	------------	------	------------------

Fünf Relationen

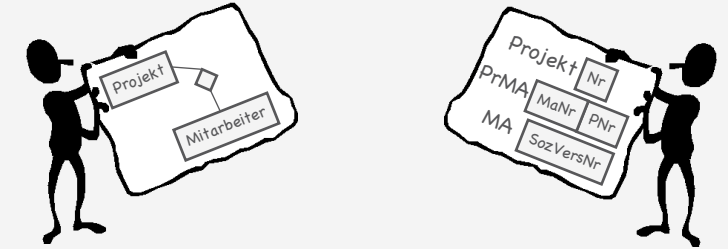


Entwurf relationaler Schemata

- 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)

Zwischenzusammenfassung

- Entität mit Attributen wird zu Relationenschema
 - Entität als Tabelle mit Attributen als Spaltennamen
 - Schlüssel ausgehend von identifizierenden Attributen, aber nun minimaler Superschlüssel
 - Mehrwertige Attribute als eigene Relation (so auch schon im ER-Diagramm möglich)
 - Zusammengesetzte Attribute nicht direkt übersetzbar
- Assoziation mit Attributen wird zu Relationenschema mit Schlüsselattributen der verknüpften Entitäten als zusätzliche Attribute
 - Schlüssel gemäß Kardinalitäten wählen
 - Abgeleitete Attribute: Anfrage
 - Schwache Entität: eigene Schlüssel plus Primärschlüssel der starken Entität
- Spezialisierung: eigenes Relationenschema mit Primärschlüsseln der spezialisierten Entität
- Verfeinerung durch Zusammenfassung von Relationen mit gleichem Schlüssel



Übersicht: 3. Das relationale Datenmodell

A. *Relationales Datenmodell*

- Relationen, Attribute, relationale Datenbanken und –schemata
- Schlüssel: Primärschlüssel, Fremdschlüssel, referentielle Integrität

B. *Entwurf relationaler Schemata*

- Vom ER-Diagramm zum relationalen Modell

C. *Relationale Algebra*

- $\pi, \rho, \sigma, \cup, \cap, -, \times, \bowtie$
- Minimalität
- Aggregieren, gruppieren
- Einfügen, löschen, aktualisieren