



0101seda010100
software engineering dependability

Software Entwicklung 2

Strukturierte Analyse 2
(Kurz-Einführung in relationale
Informationsmodellierung)

- Grundbegriffe der relationalen Informationsmodellierung
- Operationen auf Relationen
- Datenabhängigkeiten & Integritätsregeln
- ER-Modell
- Normalformen

- Die Begriffe im Zusammenhang mit relationaler Informationsmodellierung erläutern können (Entität, Relation, Attribut, Schlüssel, ...)
- Die Operationen auf Relationen erläutern können
- Datenabhängigkeiten und Integritätsregeln erläutern können
- Die Normalformen erklären und anwenden können
- Eine verbale Beschreibung in ein relationales Informationsmodell transformieren können

- Oft existieren umfangreiche Daten, die beschrieben werden müssen
 - Syntax → z.B. Data Dictionary
 - Semantik → IM
- In SA gibt es das Data Dictionary zur Beschreibung der Struktur von Daten und „Files“ (Datenspeicher) zur Beschreibung der Daten an sich
- Wir benötigen noch eine Technik zur Beschreibung der inhaltlichen Zusammenhänge zwischen Daten → IM

- Eine Software zur Verwaltung von Seminaren muss sich Informationen über Seminarveranstaltungen, Veranstaltungshotels, Seminarteilnehmer, usw. merken
- Dazu müssen jeweils unterschiedliche „Detailinformationen“ abgespeichert werden, z.B. zur Seminarveranstaltung das Datum, der Ort, der Titel, ..., zum Seminarteilnehmer sein Name und die Anschrift, ...
- Darüber hinaus stehen Seminarveranstaltungen und Teilnehmer in Beziehung zueinander, weil Seminarveranstaltungen von Teilnehmern gebucht werden. Zu Buchungen können weitere detaillierte Informationen relevant sein, z.B. der Teilnahmepreis eines bestimmten Teilnehmers für ein bestimmtes Seminar

- **Entität**
 - Eine betrachtete Instanz zu der unterschiedliche Informationen relevant sind, z.B. ein Gegenstand, eine Person, ein Ereignis (z.B. der Mitarbeiter Müller)
- **Entitätsmenge**
 - Entitäten, die sich durch gleichartige Eigenschaften auszeichnen (z.B. Mitarbeiter)
- **Attribut**
 - Eine relevante Eigenschaft einer Entitätsmenge (z.B. die Personalnummer)
 - Streng genommen muss der Attributbezeichner (z.B. die Personalnummer) vom Attributwert (z.B. 12345) unterschieden werden
 - Jedes Attribut A besitzt einen definierten Wertebereich $W(A)$ (sogenannte Domäne)

- **Tupel**
 - Menge von Attributen (z.B. das 3-Tupel (*Personalnummer = 12345, Name = Müller, Vorname = Walter*))
- **Relation**
 - Menge von Tupeln mit denselben Attributbezeichnern, die einem definierten **Relationenschema** genügen
 - **Relationenschema** wird durch Aufzählung der Attributbezeichner und ihrer Domäne angegeben (z.B. (*Personalnummer DECIMAL(5), Name CHAR(20), Vorname CHAR(20)*))
 - Übliche Kurzschreibweise (z.B. (*Personalnummer, Name, Vorname*))
 - Nullmarken sind ggf. möglich

- Es seien $W(A_1), W(A_2), \dots, W(A_n)$ endliche Mengen (die Wertebereiche (Domänen) der Attribute A_1, A_2, \dots, A_n)
- Die Menge aller Kombinationen ihrer Elemente heißt dann ihr kartesisches Produkt $[W(A_1) \times W(A_2) \times \dots \times W(A_n)]$
- Beispiel: Seien $W(\text{Anrede}) = \{\text{Hr.}; \text{Fr.}\}$ und $W(\text{Titel}) = \{\text{Dr.}; \text{Dipl.-Ing.}\}$ dann ist: $W(\text{Anrede}) \times W(\text{Titel}) = \{(\text{Hr.}, \text{Dr.}), (\text{Hr.}, \text{Dipl.-Ing.}), (\text{Fr.}, \text{Dr.}), (\text{Fr.}, \text{Dipl.-Ing.})\}$
- Jede Teilmenge R eines kartesischen Produktes $W(A_1) \times W(A_2) \times \dots \times W(A_n)$ heißt eine (n-stellige) Relation über $W(A_1) \times W(A_2) \times \dots \times W(A_n)$

- Beispiel

- Die Menge der Personen kann als Relation $\text{Person} \subseteq W(\text{Pers.-Nr.}) \times W(\text{Anrede}) \times W(\text{Titel}) \times \dots$ über den Wertebereichen der Attribute der Entitätsmenge Person aufgefasst werden
- Jede Relation $R \subseteq W(A_1) \times W(A_2) \times \dots \times W(A_n)$ kann als Tabelle (mit dem Namen R) dargestellt werden

Relations- bzw. Tabellennamen Schlüsselattribut (unterstrichen) Attribute

Person	<u>Pers.-Nr.</u>	Anrede	Titel	Vorname	Name	...	PLZ	Ort	Geb.-Datum
	11	Herr		Hans	Mayer		22307	Hamburg	200355
	22	Herr	Dipl. Ing.	Werner	Schulte		81675	München	040763
	129	Herr	Dr.	Kurt	Kurz		44879	Bochum	150945

Tupel (Inhalt)

- Relationenschema
 - Beschreibung der Relationen
 - Es können auch ein oder mehrere Schlüssel angegeben werden
- Schlüssel
 - Attributkombination, deren Wertekombinationen jeweils ein Tupel der Relation eindeutig bestimmen
 - Die den Schlüssel bildende Attributkombination muss minimal in dem Sinne sein, dass keine echte Teilmenge von ihr bereits die Schlüsseleigenschaft besitzt

- Beispiel

Veranstaltung	<u>Veranstaltungs-Nr.</u>	Dauer	...
	23	3	
	27	1	
	35	4	

Kunde	<u>Pers.-Nr.</u>	Funktion	Umsatz
	11	Berater	5000,-
	129	Projektleiter	8550,-
	22	Entwickler	2000,-

- Beispiel

- Die Beziehung »bucht« zwischen den Entitätsmengen »Kunde« und »Seminarveranstaltung« kann durch die Relation »bucht« mit den Attributen Pers.-Nr. und Veranstaltungs-Nr. dargestellt werden

Kunde	<u>Pers.-Nr.</u>	Funktion	Umsatz
	11	Berater	5000,-
	129	Projektleiter	8550,-
	22	Entwickler	2000,-

bucht	<u>Pers.-Nr.</u>	<u>Veranstaltungs-Nr.</u>
	11	27
	129	27
	11	35

Veranstaltung	<u>Veranstaltungs-Nr.</u>	Dauer	...
	23	3	
	27	1	
	35	4	

- Beispiel
 - Die Attributkombination (*Pers.-Nr.*, *Veranstaltungs-Nr.*) in der Tabelle »bucht« ist der Schlüssel dieser Tabelle
 - Jede Wertekombination bestimmt eindeutig ein Tupel der Relation »bucht«
 - Wird für die Tabelle »Person« die Attributkombination (*Pers.-Nr.*, *Name*) als Schlüssel verwendet, dann ist diese Attributkombination nicht minimal, da die Teilmenge *Pers.-Nr.* dieser Attributkombination bereits die Schlüsseleigenschaft besitzt

- Tabellendarstellung im Kurzformat
 - Hinter dem Tabellennamen werden in Klammern und durch Kommata getrennt die Spaltennamen aufgeführt
 - Schlüsselattribute werden z.B. durch das Nummernzeichen # hinter dem entsprechenden Spaltennamen gekennzeichnet
 - Beispiel
 - Person (Pers.-Nr.#, Anrede, Titel, Vorname,..)
 - Kunde (Pers.-Nr.#, Funktion, Umsatz)
 - Bucht (Pers.-Nr.#, Veranstaltungs-Nr.#,...)

- Selektion

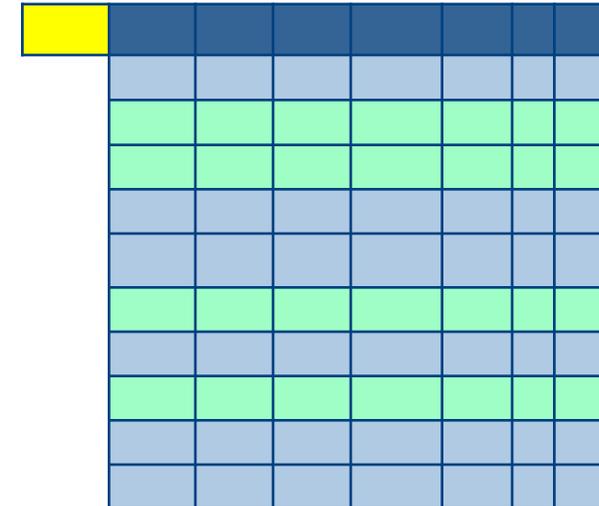
- Beispiel

- Anfrage »Gib alle Personen aus, die in Bochum wohnen«

- Aus der Tabelle »Person« alle Tupel selektieren, bei denen der Wert für das Attribut »Ort = 'Bochum'« ist

- Ergebnis

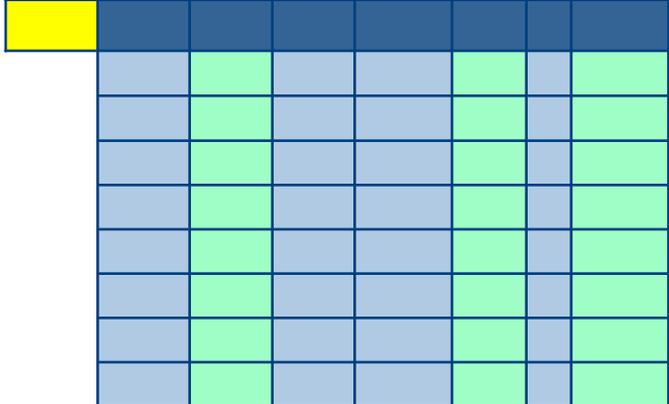
- 129 Herr Dr. Kurt Kurz... 44819 Bochum*



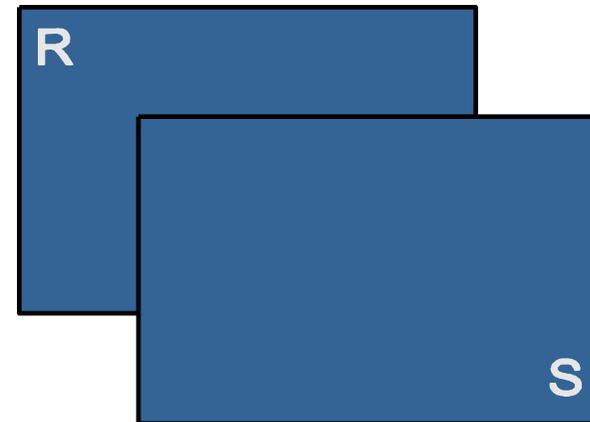
- Projektion

- Beispiel

- Anfrage »Gib den Vornamen, Namen und das Geburtsdatum aller Personen aus«
 - Aus der Tabelle »Person« alle Spalten außer Vorname, Name und Geburtsdatum ausblenden



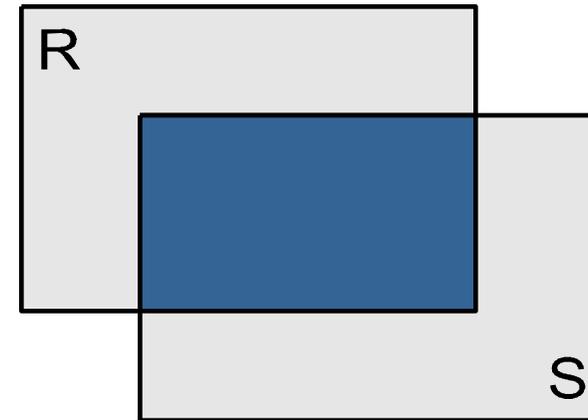
- Vereinigung (union)
 - Beispiel
 - Alle Dozenten und Mitarbeiter der Fa. Teachware erhalten eine Geburtstagskarte
 - Es soll eine Liste mit Anrede, Titel, Vorname, Name und Geburtsdatum ausgedruckt werden
 - Wenn ein Mitarbeiter auch als Dozent arbeitet, dann soll er nur einmal aufgeführt sein



- Durchschnitt (intersect)

- Beispiel

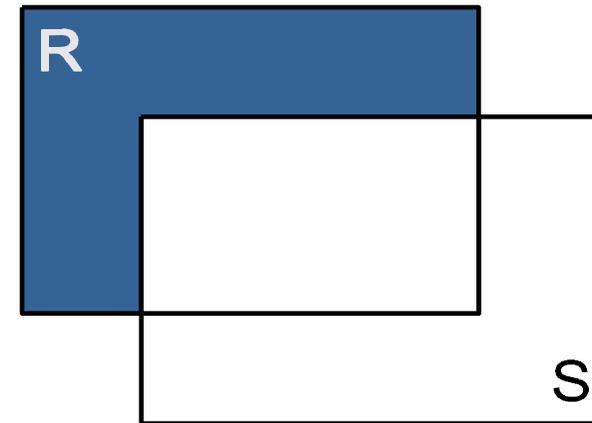
- Die Geschäftsführung der Firma Teachware möchte eine Liste aller *Pers.-Nr.* der Personen, die sowohl Kunden als auch Dozenten sind



- Differenz (difference)

- Beispiel

- Die Geschäftsführung der Firma Teachware möchte wissen, welche Kunden Privatkunden sind, d.h. zu keiner Firma gehören

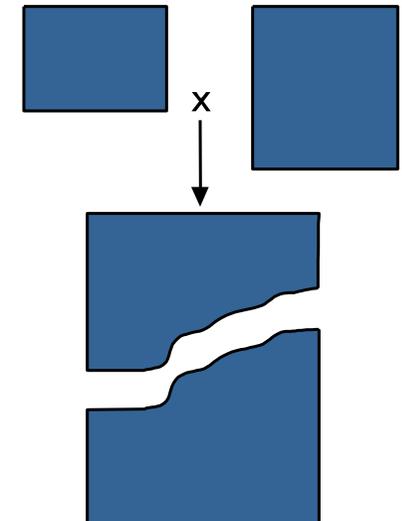


- Das kartesische Produkt $R \times S$ von zwei Relationen R und S hat als Attribute alle Attribute von R und S
- Beispiel

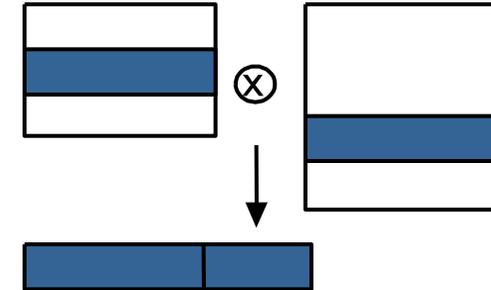
R	A	B
	a	b
	b	b

S	C	D	E
	1	2	3
	4	5	6
	7	8	9

RxS	A	B	C	D	E
	a	b	1	2	3
	a	b	4	5	6
	a	b	7	8	9
	b	b	1	2	3
	b	b	4	5	6
	b	b	7	8	9



- Verschmilzt zwei Relationen bzgl. gemeinsamer Attribute (gleicher Name und gleiche Wertebereiche) zu einer Relation höherer Stelligkeit



- $R \bowtie S$ der Relationen R und S hat als Attribute alle Attribute von R und S, wobei die gemeinsamen Attribute nur einmal aufgeführt werden
- Beispiel

R	A	B	C	S	C	D	E	F	$R \bowtie S$	A	B	C	D	E	F
	a	a	b		b	1	2	3		a	a	b	1	2	3
	a	b	a		c	4	5	6		a	a	b	7	8	9
	a	a	c		b	7	8	9		a	a	c	4	5	6



- Beispiel

Kunde	<u>Pers.-Nr.</u>	Funktion	Umsatz
	11	Berater	2000,-
	17	Entwickler	5000,-
	22	Analytiker	8000,-

Zahlungsverzug	<u>Pers.-Nr.</u>	<u>Stichtag</u>	Betrag
	11	12-12-04	1000,-
	18	10-08-04	2000,-
	22	22-10-04	500,-
	11	11-11-04	700,-

Kunde ⊗ Zahlungsverzug	<u>Pers.-Nr.</u>	Funktion	Umsatz	<u>Stichtag</u>	Betrag
	11	Berater	2000,-	12-12-04	1000,-
	11	Berater	2000,-	11-11-04	700,-
	22	Analytiker	8000,-	22-10-04	500,-

- Beim Θ -Verbund $R[A \Theta B] S$, wobei Θ einer der Vergleichsoperationen $=, \neq, \leq, \geq, <, >$ ist, wird jedes Tupel aus R mit den Tupeln aus S verknüpft, so dass der A -Wert des R -Tupels dem Vergleich Θ mit dem B -Wert jedes dieser S -Tupel genügt

R	A	B
	a	b
	c	d

S	C	D
	b	d
	a	b
	d	e

$R[B \neq C]S$	A	B	C	D
	a	b	a	b
	a	b	d	e
	c	d	b	d
	c	d	a	b

$R[B \leq C]S$	A	B	C	D
	a	b	b	d
	a	b	d	e
	c	d	d	e

- Ist die Θ -Vergleichsoperation die Gleichheit $\gg=«$, dann nennt man den Θ -Verbund auch equi-Join
- Unterschied zum natürlichen Verbund, die Attribute sind nicht notwendigerweise gleichbenannt und die verglichenen Spalten sind doppelt aufgeführt
- Der natürliche Verbund muss in SQL über den Θ -Verbund ($\Theta = \gg=«$) beschrieben werden

Relationale Informationsmodellierung

Operationen auf Relationen

Kunde	<u>Pers.-Nr.</u>	Funktion	Umsatz
	11	Berater	2000,-
	17	Entwickler	5000,-
	22	Analytiker	8000,-

Zahlungsverzug	<u>Pers.-Nr.</u>	<u>Stichtag</u>	Betrag
	11	12-12-04	1000,-
	18	10-08-04	2000,-
	22	22-10-04	500,-
	11	11-11-04	700,-

Kunde ⊗ Zahlungsverzug	<u>Pers.-Nr.</u>	Funktion	Umsatz	<u>Stichtag</u>	Betrag
	11	Berater	2000,-	12-12-04	1000,-
	11	Berater	2000,-	11-11-04	700,-
	22	Analytiker	8000,-	22-10-04	500,-

- Integritätsbedingungen
 - Legen fest, welche Teilmengen eines kartesischen Produktes von Wertebereichen »gültige« Relationen sind
 - Als (spezielle) Integritätsbedingungen werden hier sogenannte Datenabhängigkeiten betrachtet
- Abhängigkeiten innerhalb einer Tabelle
 - Funktionale Abhängigkeiten FDs
 - Mehrwertige Abhängigkeiten MVDs
- Abhängigkeiten zwischen Tabellen
 - Inklusions-Abhängigkeiten IDs
- Ein relationales Datenmodell muss zwei Integritätsbedingungen erfüllen
 - Entitäts-Integrität
 - Referentielle Integrität

- Funktionale Abhängigkeit FD

- Ist eine Aussage der syntaktischen Form $X \rightarrow Y$, wobei X und Y Teilmengen der Attributmenge eines Relationschemas sind
(zu lesen: X bestimmt funktional Y)
- Eine Relation über diese Attributmenge erfüllt die FD $X \rightarrow Y$, falls
 - je zwei Tupel aus ihr, die auf X übereinstimmen, auch auf Y übereinstimmen
- Zu jedem X-Wert oder X-Wertetupel in R gibt es also höchstens einen Y-Wert bzw. ein Y-Wertetupel
- Während des Entwurfs werden FDs festgelegt: Sie schränken mögliche Relationen auf gültige ein

- Beispiel
 - Für die Tabelle »Person« ist folgende FD denkbar
 $\{\text{Vorname, Name, Straße, PLZ, Ort}\} \rightarrow \{\text{Geburtsdatum}\}$
 - Eine Kombination von Werten der Attribute Vorname, Name, Straße, Ort bestimmt genau einen Wert des Attributes Geburtsdatum
 - D.h. in der Relation »Person« kann kein weiteres Tupel mit dem Wert (Hans, Meyer, Hauptstr. 12, 56666, Hauptdorf, 2/5/33) für Vorname, Name, Straße, Ort enthalten sein mit einem anderen Wert als 2/5/33 für Geburtsdatum
- Die FDs, die für eine Relation gelten, werden mit in ihr Relationenschema aufgenommen

- Schlüssel

- Sei X die Attributmengende des Relationenschemas R und K eine Teilmenge von X
- K definiert einen Schlüssel für R , falls jede Relation die FD $K \rightarrow X$ erfüllt und K keine echte Teilmenge enthält, die ebenfalls diese Eigenschaft besitzt

- Mehrwertige Abhängigkeiten MVD

- Bei einer FD $X \rightarrow Y$ bestimmt eine Wertekombination der Attribute aus X genau eine Wertekombination der Attribute aus Y
- Es gibt jedoch auch Fälle, in denen eine Wertekombination der Attribute aus X mehrere Wertekombinationen der Attribute aus Y bestimmt und alle diese Kombinationen unabhängig voneinander sind

Person	Pers.-Nr.	Vorname	Name	Beruf	Kind
	198	Fritz	Schultz	Informatiker	Franz
	198	Fritz	Schultz	Informatiker	Carola
	237	Hans	Lange	Ingenieur	Kerstin
	237	Hans	Lange	Ingenieur	Holger
	237	Hans	Lange	Ingenieur	Achim

- Beispiel

- Jede Person bestimmt seine Kinder (nicht funktional) und unabhängig davon auch seinen Beruf:
 $\{\text{Pers.-Nr.}\} \rightarrow \{\text{Beruf}\}$
- Es besteht hier die mehrwertige Abhängigkeit (multivalued dependency MVD)
 $\{\text{Pers.-Nr.}\} \twoheadrightarrow \{\text{Kind}\}$
(zu lesen: Pers.-Nr. bestimmt mehrfach Kind)

Person	Pers.-Nr.	Vorname	Name	Beruf	Kind
	198	Fritz	Schultz	Informatiker	Franz
	198	Fritz	Schultz	Informatiker	Carola
	237	Hans	Lange	Ingenieur	Kerstin
	237	Hans	Lange	Ingenieur	Holger
	237	Hans	Lange	Ingenieur	Achim

- MVDs bringen Redundanzen in einer Relation mit sich
 - Im obigen Beispiel müssen die jeweiligen Werte zu den anderen Attributen {Beruf...} mehrfach (bei 198 zweimal, bei 237 dreimal) aufgeführt werden

Person	Pers.-Nr.	Vorname	Name	Beruf	Kind
	198	Fritz	Schultz	Informatiker	Franz
	198	Fritz	Schultz	Informatiker	Carola
	237	Hans	Lange	Ingenieur	Kerstin
	237	Hans	Lange	Ingenieur	Holger
	237	Hans	Lange	Ingenieur	Achim

- Abhängigkeiten zwischen Attributen verschiedener Relationenschemata
 - Interrelationale Abhängigkeiten
 - Eine häufig auftretende Abhängigkeit dieser Art ist die Inklusions-Abhängigkeit ID
 - Beispiel
 - Es kann z.B. ein Kunde nur Seminarveranstaltungen buchen, die auch wirklich angeboten werden
 - Das bedeutet, dass in der Relation »bucht« unter dem Attribut Veranstaltungs-Nr. nur Werte auftreten dürfen, die in der Relation Seminarveranstaltung als Wert unter dem Attribut Veranstaltungs-Nr. bereits vorhanden sind

- Inklusions-Abhängigkeiten

- Gewisse Tupel in einer Relation sind erst dann zugelassen, wenn in einer anderen dazu bereits Entsprechungen existieren
- Bei IDs handelt es sich um Aussagen der Form $RS_1[V] \supseteq RS_2[Z]$ wobei RS_1 und RS_2 zwei Relationenschemata mit den Attributen X bzw. Y sind und es gelte: $Y \supseteq Z, X \supseteq V$
- Sind R_1 und R_2 Relationen über RS_1 bzw. RS_2 , so ist diese ID erfüllt, falls zu jedem Zeitpunkt die Projektion von R_2 auf Z eine Teilmenge der Projektion von R_1 auf V ist
- Voraussetzung: Z und V sind vom gleichen Typ

- Fremdschlüssel

- Ist V Primärschlüssel von RS_1 , dann ist Z ein Fremdschlüssel in RS_2
- Beispiel

RS_2	Y_1	$Z=Y_2$	Y_3	Y_4
bucht	<u>Pers.-Nr.</u>	<u>Veranstaltungs-Nr.</u>	angemeldet am	...
$R_2 \rightarrow$	12	23	12-03-05	
	24	27	27-04-05	
	19	23	14-03-05	

RS_1	$V=X_1$	X_2	X_3
Veranstaltung	<u>Veranstaltungs-Nr.</u>	Dauer	...
$R_1 \rightarrow$	23	3	
	27	1	
	35	4	

Es gilt: $Y \supseteq Z, X \supseteq V$

- Zwei fundamentale Integritätsregeln des Relationsmodells müssen eingehalten werden
 - Entitäts-Integrität
 - Die Wertebereiche der Primärschlüssel-Attribute dürfen keine Nullwerte enthalten
 - Referentielle Integrität
 - Falls ein Fremdschlüssel Z einer Relation R_2 mit dem Primärschlüssel Y der Relation R_1 korrespondiert, muss entweder jeder Wert von Z in R_2 als Wert von Y in R_1 vorkommen oder er muss (in jeder Komponente) einen Nullwert enthalten
 - Referentielle Integrität kann auch innerhalb einer Relation bestehen

- Beispiel

- Ein Kunde kann zu einer Firma gehören, dann muss als Fremdschlüssel der Kurzname eingetragen sein
- Alle eingetragenen Kurznamen, d.h. alle Fremdschlüssel, müssen als Primärschlüssel in der Tabelle »Firma« vorhanden sein

Z

Kunde	<u>Pers.-Nr.</u>	...	Kurzname
R ₂ {	12		Hard&Soft
	24		Softech
	19		-

V

Firma	<u>Kurzname</u>	Name	...
R ₁ {	Hard&Soft		
	Softech		

- Umgangssprachlich ausgedrückt
 - Wenn in einer Tabelle ein Fremdschlüssel vorhanden ist, dann muss der Fremdschlüsselwert auch als Primärschlüsselwert in der korrespondierenden Tabelle auftauchen

Z

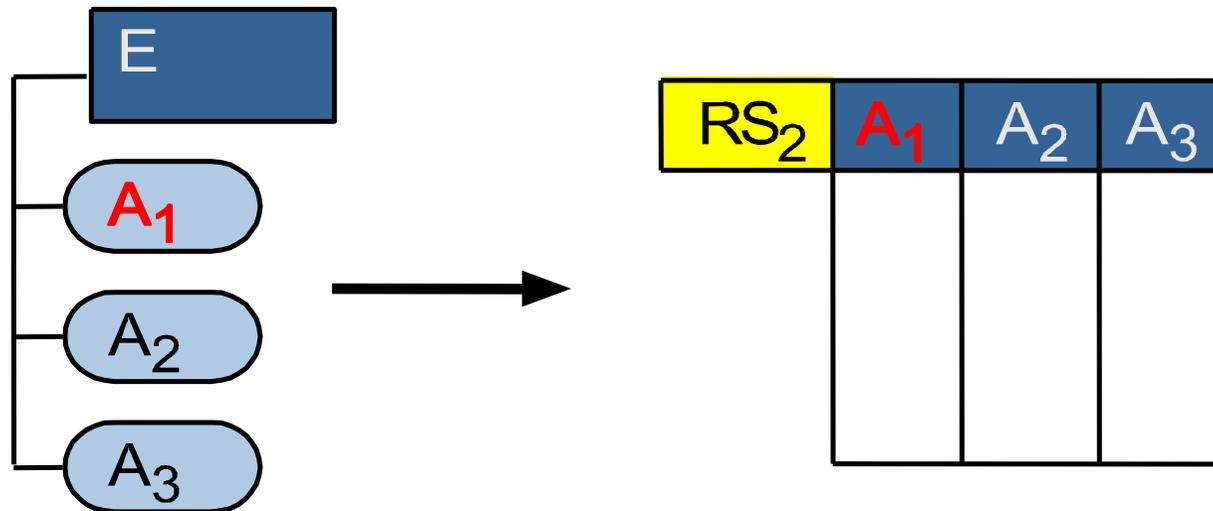
Kunde	<u>Pers.-Nr.</u>	...	Kurzname
R ₂ {	12		Hard&Soft
	24		Softech
	19		-

V

Firma	<u>Kurzname</u>	Name	...
R ₁ {	Hard&Soft		
	Softech		

- Entitätsmenge

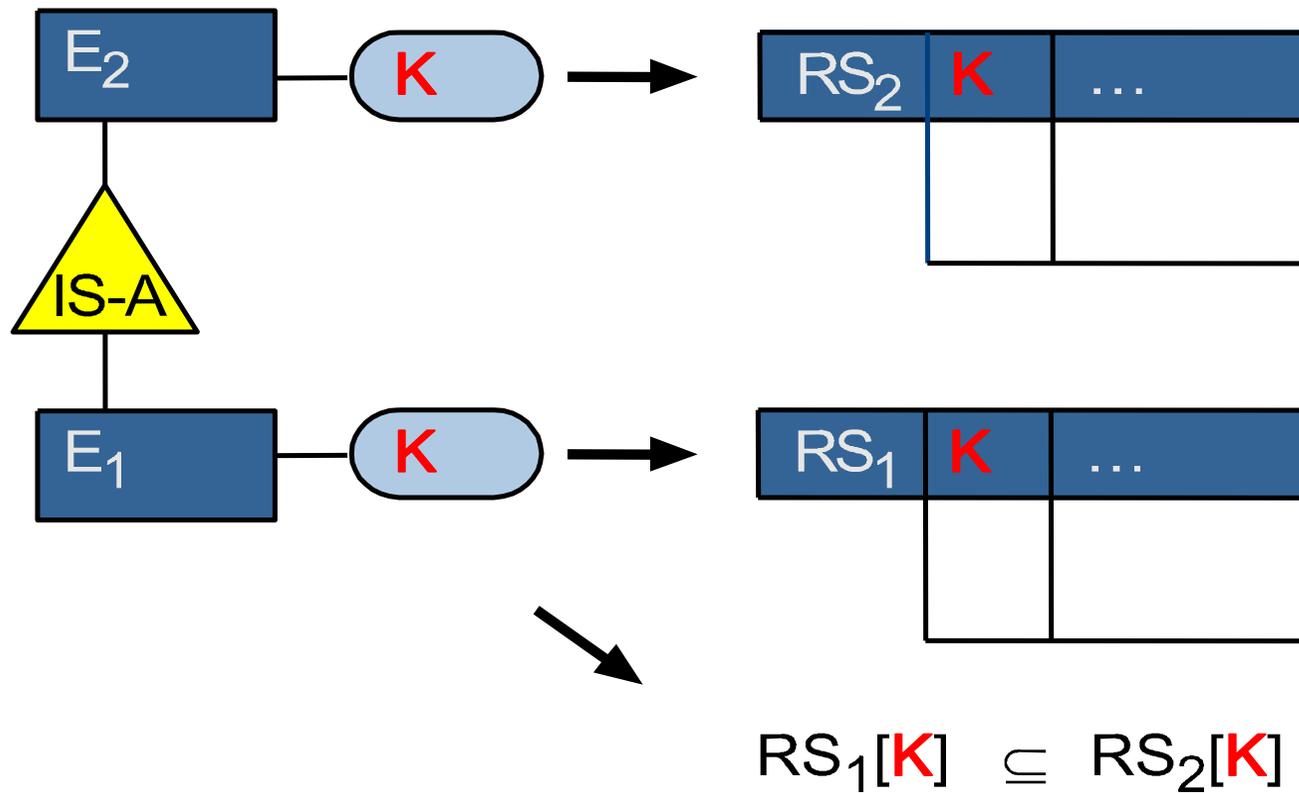
- Für jede Entitäts-Menge wird ein Relationenschema definiert, dessen Attributmenge alle Attribute der Menge umfasst
- Jedes Tupel in einer Relation über dieses Schema stellt eine Entität dar
- Beispiel



- IS-A-Vererbung

- Bei einer IS-A-Beziehung bzw. Vererbungsstruktur zwischen zwei Entitäts-Mengen E_1 und E_2 , die in die Relationschemata RS_1 und RS_2 transformiert werden, entsteht eine interrelationale Inklusions-Abhängigkeit
- Falls E_1 IS-A E_2 gilt und K der Schlüssel von E_2 bzw. RS_2 ist, der auch in der Attributmenge von RS_1 vorkommt, dann ist die Inklusions-Abhängigkeit $RS_2[K] \supseteq RS_1[K]$ in die Menge der Inklusions-Abhängigkeiten des Datenbankschemas aufzunehmen

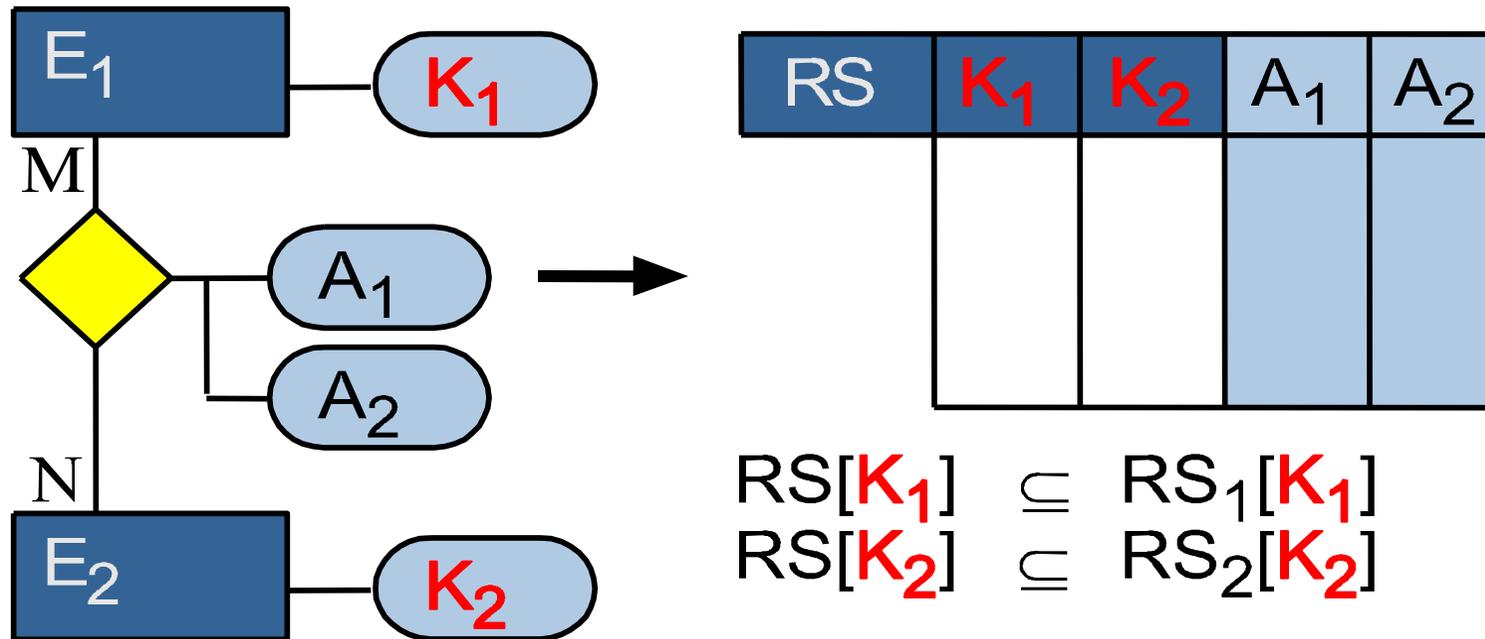
- IS-A-Vererbung
 - Beispiel



- M:N-Assoziation

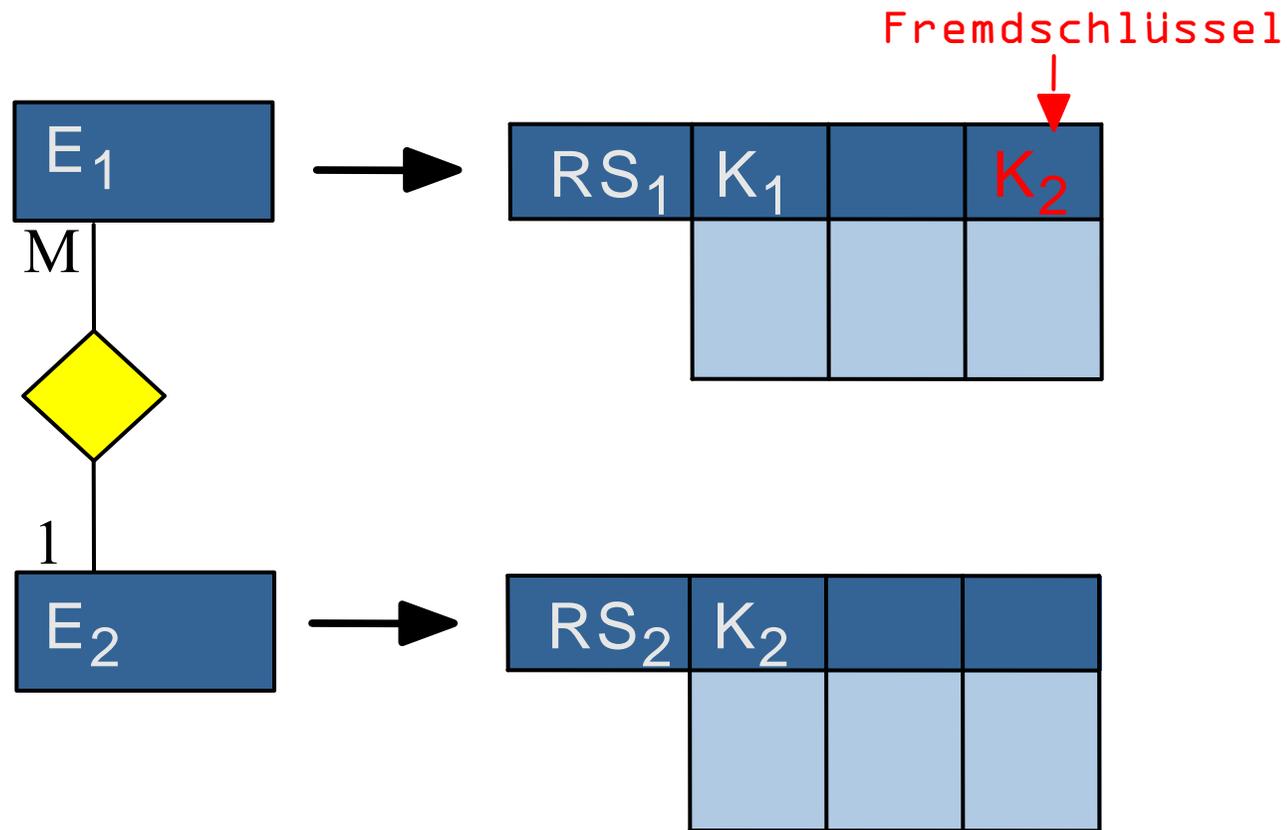
- Jede M:N-Assoziation zwischen k Entitäts-Mengen E_1, \dots, E_k wird in ein Relationenschema transformiert, dessen Attributmenge neben allen speziellen Attributen der Assoziation (falls vorhanden) je einen Schlüssel für E_1, \dots, E_k umfasst
- Wenn zwei Entitäts-Mengen ein Attribut mit dem gleichen Namen enthalten, diese Attribute jedoch verschiedene Dinge bezeichnen, dann sind diese Namenskonflikte aufzulösen

- M:N-Assoziation
 - Beispiel

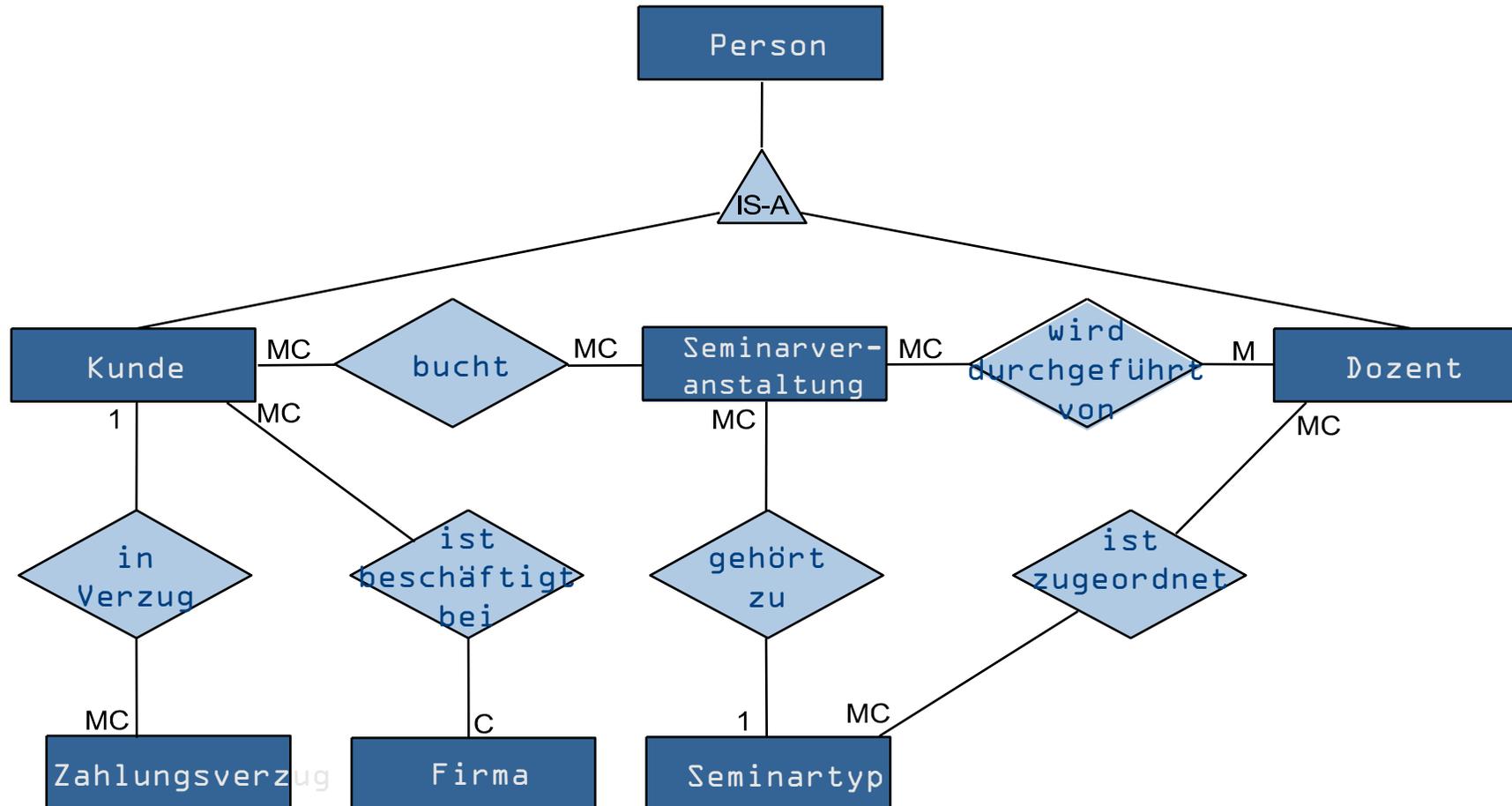


- M:1- und 1:1 Assoziation
 - Verbindet eine 1:1-Assoziation oder eine M:1-Assoziation zwei Entitäts-Mengen E_1 und E_2 , dann kann auf ein eigenes Relationsschema für diese Assoziation verzichtet werden
 - In diesem Fall wird der Primärschlüssel von E_2 als Fremdschlüssel in E_1 eingetragen

- M:1- und 1:1 Assoziation
 - Beispiel



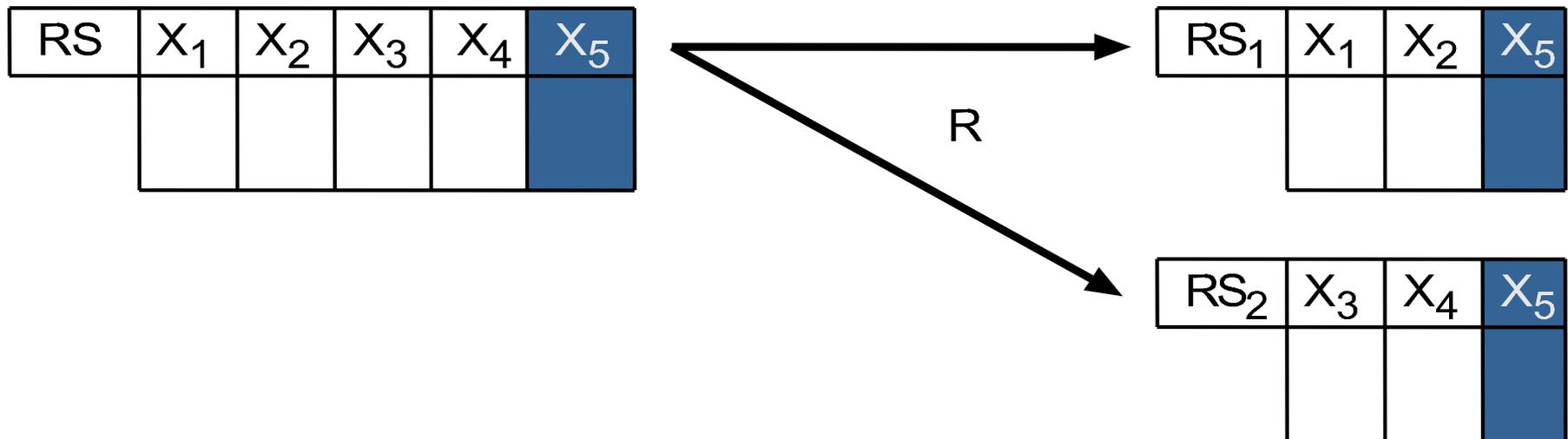
- ER-Diagramm »Seminarorganisation«



- Intrarelationale Abhängigkeiten im logischen Schema können zu Redundanzen führen
 - Im Betrieb der Datenbank kann dies zu Integritätsverletzungen der Schema-Inhalte führen
 - Die einzelnen Relationenschemata werden daher einem Normalisierungsprozess unterzogen
 - Jedes Relationenschema wird unter Umständen in neue Schemata, sogenannte Basisrelationenschemata, zerlegt, um Redundanzen zu vermeiden

- Normalisierungsprozess
 - Nicht normalisierte Relationenschemata werden in solche überführt, die einer bestimmten Normalform genügen
- Die Zerlegung muss dabei den verlustfreien Verbund (lossless join) garantieren
 - Wird ein Relationenschema RS mit der Attributmengemenge X in die 2 Schemata RS_1 mit der Attributmengemenge X_1 und RS_2 mit X_2 zerlegt, und werden die Relationen R_1 bzw. R_2 durch Projektion einer Relation R über RS auf X_1 bzw. X_2 erzeugt, d.h. $R_1 = R[X_1]$ und $R_2 = R[X_2]$, dann soll der natürliche Verbund bzgl. der gemeinsamen Attribute von R_1 und R_2 , genau die Relation R ergeben, d.h. $R = R_1 \otimes R_2$

- Beispiel



- Die erste Normalform garantiert eine feste Länge jeder „Tabellenzeile“
 - In jeder Tabellenposition steht immer nur ein Wert, niemals eine Liste von Werten
 - Der Wertebereich jedes Attributs darf nur atomare Werte enthalten, aber keine Wiederholungsgruppen
 - Alle bisherigen Beispiele waren in erster Normalform
 - Gegenbeispiel: Kunde bei mehreren Firmen beschäftigt

Kunde	<u>Pers.-Nr.</u>	Funktion	Umsatz	Kurznamen
	11	Berater	5000,-	Hard&Soft, Softtech
	129	Projektleiter	8550,-	Innosoft
	22	Entwickler	2000,-	Hard&Soft, Orgware, Casetech

- Eine Relation ist in der ersten Normalform, wenn die Werte der Attribute elementar sind

- Voraussetzung: 1.Normalform
- Jedes Nicht-Schlüsselattribut muss vom Primärschlüssel voll funktional abhängig sein
 - Ein Nicht-Schlüsselattribut ist ein Attribut, das in keinem Schlüssel dieser Tabelle vorkommt
 - Voll funktional abhängig bedeutet
 - Ein Attribut Y ist funktional abhängig von Attribut X, aber nicht von einer echten Teilmenge von X
 - Umgekehrt ausgedrückt bedeutet dies
 - Eine Tabelle ist nicht in 2.Normalform, wenn ein Nicht-Schlüsselattribut eine Eigenschaft einer Teilmenge eines Schlüssels beschreibt
 - Dies ist dann der Fall, wenn das Nicht-Schlüsselattribut bereits von einer Teilmenge des Schlüssels funktional abhängt

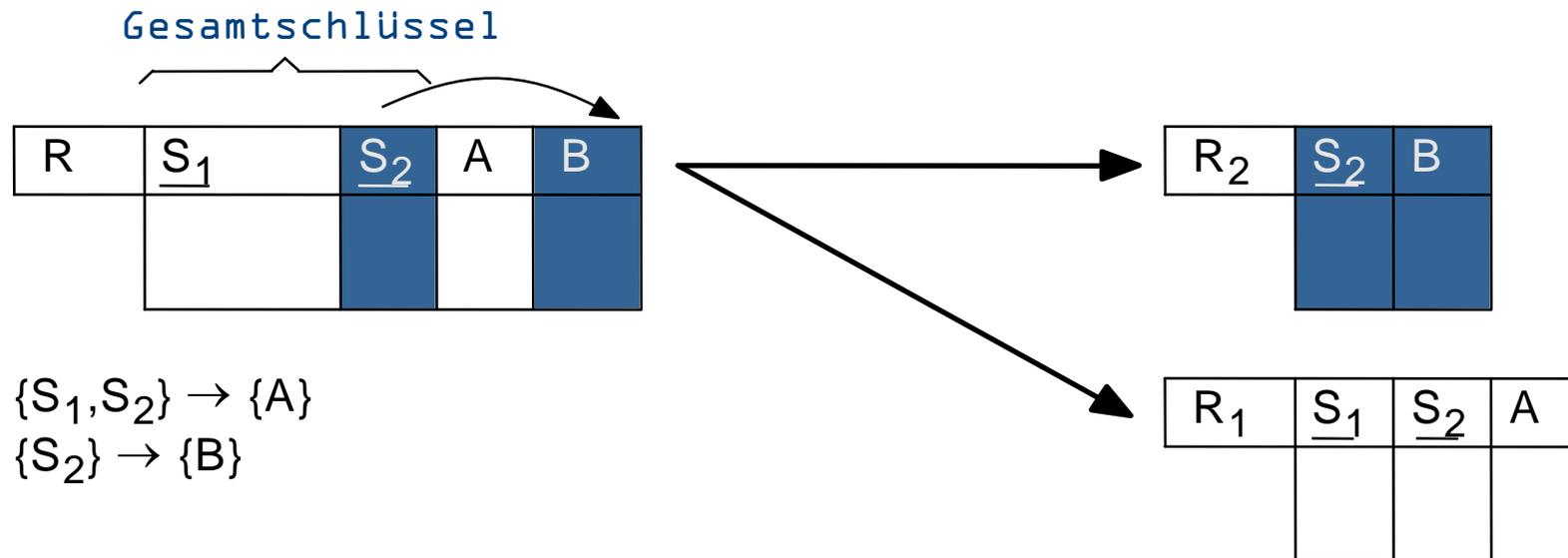
- Gegenbeispiel: Jeder Dozent verteilt ein Skript zu seinem Vortrag

Skript	<u>Veranst.-Nr.</u>	<u>Personal-Nr.</u>	Name	Skript-Nr.	Preis
	112	198	Schulz	2	25,-
	112	237	Lange	9	44,-
	112	11	Meyer	2	25,-
	202	198	Schulz	4	22,-

- Der Name ist funktional abhängig von der Personal-Nr.
 $\{\text{Personal-Nr.}\} \rightarrow \{\text{Name}\}$, d.h. von einer Teilmenge des Schlüssels
- Die Skript-Nr. ist voll funktional abhängig von der Kombination aus Veranst.-Nr. und Personal-Nr.
 $\{\text{Veranst.-Nr.}, \text{Personal-Nr.}\} \rightarrow \{\text{Skript-Nr.}\}$
- Der Preis ist voll funktional abhängig von der Skript-Nr.
 $\{\text{Skript-Nr.}\} \rightarrow \{\text{Preis}\}$
- Wegen der funktionalen Abhängigkeit $\{\text{Personal-Nr.}\} \rightarrow \{\text{Name}\}$ ist diese Tabelle nicht in der 2. Normalform!

- Zerlegungsmethode

- Die Nicht-Schlüsselattribute einer Tabelle R, die von einer Teilmenge eines Schlüssels abhängen, bilden mit dieser Teilmenge als Schlüssel eine neue Tabelle R2
- Die restlichen Attribute bilden mit dem Schlüssel von R eine neue Tabelle R1



- Beispiel

Skript	<u>Veranst.-Nr.</u>	<u>Personal-Nr.</u>	Name	Skript-Nr.	Preis
	112	198	Schulz	2	25,-
	112	237	Lange	9	44,-
	112	11	Meyer	2	25,-
	202	198	Schulz	4	22,-

Dozent	<u>Personal-Nr.</u>	Name
	198	Schulz
	237	Lange
	11	Meyer

Skript2	<u>Veranst.-Nr.</u>	<u>Personal-Nr.</u>	Skript-Nr.	Preis
	112	198	2	25,-
	112	237	9	44,-
	112	11	2	25,-
	202	198	4	22,-

- Voraussetzung: 2.Normalform
- Es gibt keine transitiven funktionalen Abhängigkeiten der Art aus $A \rightarrow B$ und $B \rightarrow C$ folgt, dass auch $A \rightarrow C$ gilt
- Umgekehrt ausgedrückt bedeutet dies
Eine Tabelle ist nicht in 3.Normalform, wenn eine funktionale Abhängigkeit zwischen Nicht-Schlüsselattributen existiert

- Gegenbeispiel

Dozent	<u>Personal-Nr.</u>	Name
	198	Schulz
	237	Lange
	11	Meyer

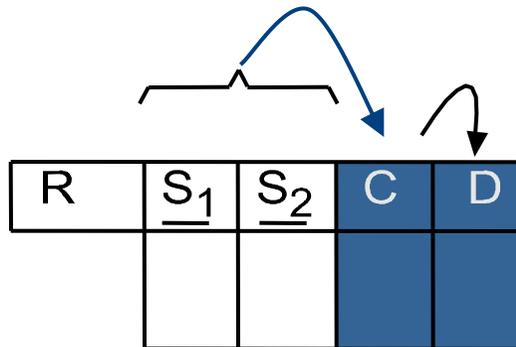
Skript2	<u>Veranst.-Nr.</u>	<u>Personal-Nr.</u>	Skript-Nr.	Preis
	112	198	2	25,-
	112	237	9	44,-
	112	11	2	25,-
	202	198	4	22,-

- Der Primärschlüssel von »Skript 2« besteht aus der Kombination der Primattribute Veranst.-Nr. und Personal-Nr.
- Es gelten folgende funktionale Abhängigkeiten
 - {Veranst.-Nr., Personal-Nr.} → {Skript-Nr}
 - {Skript-Nr} → {Preis}
- Daraus ergibt sich, dass der Preis transitiv vom Primärschlüssel abhängt und der Wert eines Nicht-Schlüsselattribut ein anderes vollständig bestimmt
- Die Tabelle »Skript 2« ist daher nicht in 3.Normalform

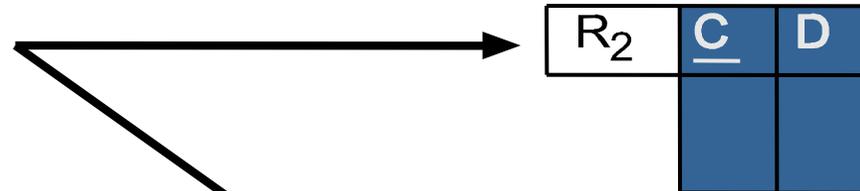
- Zerlegungsmethode

- Eine Tabelle R mit der Attributmenge Z und der transitiven Abhängigkeit $X \rightarrow Y \rightarrow A$ wird in 2 Tabellen R_1 und R_2 zerlegt
- Die Tabelle R_1 erhält alle Attribute aus Z ohne A, die funktionalen Abhängigkeiten aus R ohne $Y \rightarrow A$ und den Schlüssel von R
- Die Tabelle R_2 erhält alle Attribute aus Y und das Attribut A, die funktionale Abhängigkeit $Y \rightarrow A$ und den Schlüssel Y

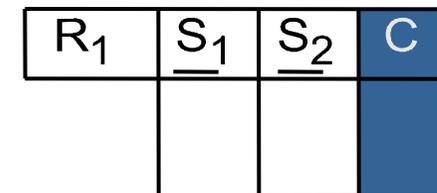
- Zerlegungsmethode



$\{S_1, S_2\} \rightarrow \{C\} \rightarrow \{D\}$
 $X \rightarrow Y \rightarrow A$
 $Z = \{S_1, S_2, C, D\}$



$\{C\} \rightarrow \{D\}$
 $Y \rightarrow A$



$\{S_1, S_2\} \rightarrow \{C\}$
 $X \rightarrow Y$

- Beispiel

Skript2	<u>Veranst.-Nr.</u>	<u>Personal-Nr.</u>	Skript-Nr.	Preis
	112	198	2	25,-
	112	237	9	44,-
	112	11	2	25,-
	202	198	4	22,-

Skript3	<u>Veranst.-Nr.</u>	<u>Personal-Nr.</u>	Skript-Nr.
	112	198	2
	112	237	9
	112	11	2
	202	198	4

Skriptpreis	<u>Skript-Nr.</u>	Preis
	2	25,-
	9	44,-
	4	22,-

FD {Veranst.-Nr., Personal-Nr.} → {Skript-Nr.} → {Preis}

- Das Zerlegen eines Relationenschemas führt zu entsprechend vielen Tabellen in der Datenbank
 - Diese Tabellen müssen zur Berechnung der Antwort auf eine Anfrage wieder miteinander verknüpft werden (Verbund-Operationen), was bei Tabellen mit vielen Tupeln zu längeren Antwortzeiten führt
- Falls überwiegend Anfragen ohne Änderungen stattfinden ist eine Normalisierung nicht unbedingt notwendig, da Inkonsistenzen durch Redundanzen nur selten auftreten werden. Im Fall einer Änderung müssen ggf. Inkonsistenzen beseitigt werden
- update-query-tradeoff: Abschätzung, welche Situation akzeptabler ist
 - Überwachen der Integrität durch den Benutzer, dafür gute Antwortzeiten bei Abfragen
 - oder höhere Antwortzeiten bei normalisierten Relationen, dafür kein Aufwand zur Beseitigung von Inkonsistenzen

...und wie Sie sehen können, haben alle Artikel in unserem Lager einen festen Platz mit Lagerplatznummer. Außerdem soll zu jedem Artikel auch die Bezeichnung, der Preis und eine kurze Beschreibung angezeigt werden.

Eine Rechnung enthält, neben den Rechnungsposten, eine eindeutige Rechnungsnummer, die Kundennummer, ein Rechnungsdatum und zwei Adressen. Eine ist die Rechnungsadresse, die andere die Lieferadresse. Wir trennen beide selbst dann, wenn die Adressen identisch sind. Wir unterscheiden zwischen Privatadressen und Firmenadressen. Bei Firmenadressen gibt es die Felder Vorname und Name nicht, dafür aber den Firmen- und Abteilungsnamen. Die Kundennummer ist einfach eine Zahl. Vielleicht können wir sie in Zukunft nutzen um alle Bestellungen eines Kunden zu suchen. Ist es möglich, die Software bereits in 4 Wochen...

