

Datenbanksysteme

Kap 7: Entwurfstheorie - Normalisierung

- Relationale Entwurfstheorie
 - Gute Schemata, schlechte Schemata
 - Funktionale Abhängigkeiten
 - Normalisierung durch Zerlegung
 - Normalformen
- Lernziel
 - Problematik der Redundanzen verstehen
 - Normalformen (er)kennen
 - Zerlegungsalgorithmen kennen und anwenden können

Was ist Normalisierung?

- Ergebnis des Datenbankentwurf ist ein Datenbankschema
 - Besteht aus einer Menge von Relationenschemata
- Bewertung der Qualität eines Relationenschemas
 - Vermeidung von Redundanz
 - Einhaltung von Konsistenzbedingungen
- Ein initiales Relationenschema wird schrittweise umgeformt, um ein "gutes" Schema zu erhalten
 - Es gibt unterschiedliche **Normalformen**, mit denen bestimmte Arten von Redundanzen verhindert werden können
 - Ob sich ein relationales Schema in einer bestimmten Normalform befindet, kann mithilfe **formaler Tests** überprüft werden
 - Die Transformation eines relationalen Schemas in unterschiedliche Normalformen nennt man **Normalisierung**

1. Normalform

- Ein Schema ist in 1. Normalform (1NF), wenn
 - Alle Wertebereiche nur atomare Werte enthalten
 - Ein Attributwert nur einen Einzelwert aus seinem Wertebereich annehmen kann
- Unzulässig in 1NF:

Hochschule				
<u>HNr</u>	Name	Adresse		Studienfach
H1	HS Niederrhein	Reinarzstraße 49		{Informatik, BWL, MB}
		47805	Krefeld	

- Adresse ist komplex strukturiert
 - Besteht aus drei Komponenten
- Studienfach ist Menge von Werten
 - Kann selbst wieder als Relation betrachtet werden

1NF im relationalen Modell

- Die Bedingungen für 1NF sind definitionsgemäß Teil des relationalen Datenmodells
 - Daher ist ein relationales Schema grundsätzlich immer in 1NF
- Es gibt auch Datenmodelle, die diese Eigenschaft nicht unbedingt voraussetzen
 - sog. NF²-Datenmodelle – Non-First-Normal-Form-Datenmodelle
 - Erlauben sog. *Nested Relations* (Relationen als Attributwerte)
 - Objektrelationale DBMS (auch PostgreSQL)
 - Erlauben benutzerdefinierte, zusammengesetzte Datentypen und Arrays
 - Erfordert Möglichkeit, eigene Datentypen und Operationen darauf definieren zu können

Normalisierung in 1NF

Hochschule				
<u>HNr</u>	Name	Adresse		Studienfach
H1	HS Niederrhein	Reinarzstraße 49		{Informatik, BWL, MB}
		47805	Krefeld	

- Wie kann dieses Schema in 1NF gebracht werden?
- Komplexes Attribut **Adresse**:
 - Ersetze Adresse durch die drei Felder Straße, PLZ, Ort

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	Studienfach
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	{Informatik, BWL, MB}

- Mehrfachwerte für Studienfach:
 - Verschiedene Lösungen denkbar

Lösung 1

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	<u>Studienfach</u>
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	Informatik
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	BWL
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	MB

- Füge pro Studienfach ein Tupel ein
 - für eine Hochschule mit n Studienfächern erhalten wir dann n Tupel
 - Der Primärschlüssel HNr muss um das Attribut Studienfach erweitert werden (Warum?)

Lösung 2

Hochschule									
<u>HNr</u>	Name	Straße	PLZ	Ort	SF1	SF2	SF3	...	SFn
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	Inf.	BWL	MB	...	NULL

- Führe für jedes Studienfach eine eigene Spalte ein
– ersetze das Attribut Studienfach durch SF_1 , SF_2 , ... SF_n
- Nur möglich, wenn maximale Anzahl n von Studienfächern bekannt
- Bei weniger Studienfächern mit NULL auffüllen
- Schwierige Anfrageformulierung

Lösung 3

Hochschule				
<u>HNr</u>	Name	Straße	PLZ	Ort
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld

HS_SF	
<u>HNr</u>	<u>Studienfach</u>
H1	Informatik
H1	BWL
H1	MB

- Entferne Studienfach aus Relation Hochschule
- Bilde neue Relation HS_SF, die Studienfach und den Primärschlüssel von Hochschule (als FK) enthält
- Für jedes Studienfach ein Tupel in HS_SF

Vergleich der Lösungen

- Nachteile der Lösung 1
 - führt zu Redundanzen durch überflüssige Mehrfachspeicherung der Attribute Straße, PLZ, Ort:
 - Speicherplatzverschwendung
 - Gefahr von Inkonsistenzen bei Änderungen
- Nachteile der Lösung 2
 - begrenzt maximale Anzahl von Studienfächern
 - NULL-Werte bei Hochschulen mit weniger als n Studienfächern
 - Schwierige Anfrageformulierung
- ➔ Lösung 3 ist unbedingt vorzuziehen!

Redundanzen und Anomalien

- Die in Lösung 1 eingeführten Redundanzen sind besonders unangenehm, weil sie leicht zu Inkonsistenzen und Anomalien führen können
- Bei einer Änderung müssen in der Regel mehrere Tupel verändert werden, um Inkonsistenzen zu verhindern
- Alle Arten von Änderungsoperationen sind betroffen
 - Einfüge-Anomalien
 - Löschen-Anomalien
 - Update-Anomalien

Einfüge-Anomalien

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	Studienfach
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	Informatik
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	BWL
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	MB
H1	?	?	?	?	Gesundheitswesen

- Wenn für Hochschule H1 ein neues Studienfach eingefügt werden soll, müssen Name, Straße, PLZ, Ort passend belegt werden → sonst Inkonsistenz
- Lässt sich in SQL nicht über Integrity Constraints erzwingen (außer über Trigger)

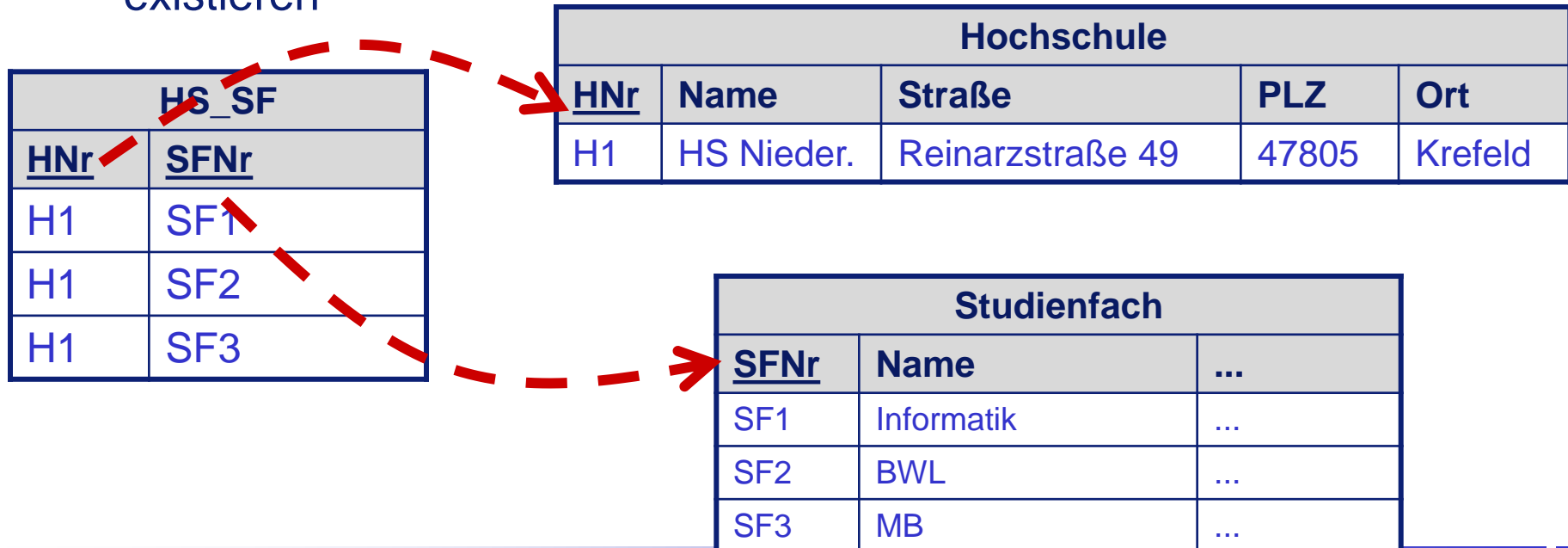
Einfüge-Anomalien

- Ein weiteres Beispiel:
 - was passiert, wenn ein neuer Professor ohne Vorlesungen eingefügt werden soll?

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	C4	226	5041	Ethik	4
2125	Sokrates	C4	226	5049	Mäeutik	2
2125	Sokrates	C4	226	4052	Logik	4
...
2132	Popper	C3	52	5259	Der Wiener Kreis	2
2137	Kant	C4	7	4630	Die 3 Kritiken	4
2138	Bacon	C3	17	?	?	?

Lösch-Anomalien

- Wenn eine Hochschule gelöscht wird, die zufälligerweise als einzige ein bestimmtes Studienfach anbietet, werden alle Informationen über dieses Studienfach mitgelöscht
- Lösung 3 leidet ebenfalls unter diesem Problem
 - Kann aber durch eine Zwischentabelle gelöst werden
 - Letztendlich entspricht die Zuordnung von Hochschulen und Studienfächern einer N:M-Beziehung
 - Studienfächer können dadurch auch unabhängig von Hochschulen existieren



Lösch-Anomalien


- Ein weiteres Beispiel
 - Angenommen, Kant liest als einziger die Vorlesung "Die 3 Kritiken"
 - Wenn Kant gelöscht wird, verschwindet auch die Information zu "Die 3 Kritiken" (z.B. SWS)

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	C4	226	5041	Ethik	4
2125	Sokrates	C4	226	5049	Mäeutik	2
2125	Sokrates	C4	226	4052	Logik	4
...
2132	Popper	C3	52	5259	Der Wiener Kreis	2
2137	Kant	C4	7	4630	Die 3 Kritiken	4

Update-Anomalien

- Wenn sich die Adresse von Hochschule H1 ändert, muss sie in allen drei Tupeln geändert werden
- Eine ungleiche bzw. nur teilweise Änderung lässt sich auf SQL-Ebene nicht abfangen

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	Studienfach
H1	HS Niederrhein	Reinarzstraße 49	47829	Krefeld	Informatik
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	BWL
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	MB



Informelle Qualitätsmaße

- Inhaltliche Bedeutung der Relationen und Attribute sollte leichtverständlich sein
- Schema sollte Ergebnis eines "vernünftigen" Entwurfs sein
 - Jede Relation sollte nur Informationen über einen Sachverhalt ("Entity") enthalten plus evt. Fremdschlüssel für N:1-Beziehung
 - Siehe auch ER-Modellierung (Kap. 8)
- Reduzierung redundanter Werte in Tupeln
- Reduzierung von Nullwerten
- Verhinderung der Erzeugung "unechter" Tupel

Güte von Relationenschemata

- Bei Normalisierung in 1NF entstanden Modelle mit selbem Informationsgehalt, aber unterschiedlicher "Güte"
 - Variante 1 führt zu unerwünschten Redundanzen

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	<u>Studienfach</u>

- Variante 2 hat diesen Nachteil nicht

Hochschule				
<u>HNr</u>	Name	Straße	PLZ	Ort

HS_SF	
<u>HNr</u>	<u>Studienfach</u>

- Ziele
 - Formale Definition von Redundanzen
 - Kriterien für Redundanzfreiheit

Redundanz eines Attributs

- Ein Attribut ist redundant, wenn einzelne Attributwerte ohne Informationsverlust weggelassen werden können.

Redundante Attribute

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	Studienfach
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	Informatik
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	BWL
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	MB

überflüssige Werte

- Die "roten" Attributwerte sind durch das erste H1-Tupel schon eindeutig festgelegt, also eigentlich überflüssig
- Wie kann man Redundanz formal definieren?

Funktionale Abhängigkeiten

- Gegeben
 - ein relationales Schema $R = (A_1, \dots, A_n)$ und r eine gültige, aber beliebige Ausprägung von R
 - X, Y seien Attributmengen von R
- Eine funktionale Abhängigkeit (FA) zwischen X und Y besteht genau dann, wenn gilt für beliebige Tupel t_1, t_2 gilt:

$$t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$$

- Mit anderen Worten:
 - In allen möglichen Ausprägungen von R bestimmen die X -Werte eindeutig die Y -Werte
 - Die Y -Werte sind eine Funktion der X -Werte: $t[Y] = f(t[X])$

Beispiel zu funktionalen Abhängigkeiten

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	<u>Studienfach</u>
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	Informatik
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	BWL
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	MB

- Folgende FAs gelten (u.a.)
 - {HNr, Studienfach} → {Name, Straße, PLZ, Ort}
 - {HNr} → {Name, Straße, PLZ, Ort}
 - {HNr} → {Straße, Ort}
 - {HNr, Name} → {PLZ}
- Folgende FAs gelten nicht:
 - {HNr} → {Studienfach}
 - {Studienfach} → {Name, PLZ}

Anmerkungen zu funktionalen Abhängigkeiten (1)

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	<u>Studienfach</u>
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	Informatik
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	BWL
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	MB

- FA ist **semantische** Eigenschaft eines Relationenschemas
 - Lässt sich nicht an einer einzelnen Ausprägung ablesen
 - In obiger Ausprägung könnte geschlossen werden: {Straße} → {Ort}
 - Ist trotzdem keine sinnvolle FA (warum?)
 - Ergibt sich aus Bedeutung der Attribute in der realen Welt
 - Nicht beweisbar, sondern wird auf Grund von Intuition und Anwendungswissen des Schema-Designers/Fachexperten festgelegt

Anmerkungen zu funktionalen Abhängigkeiten (2)

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	<u>Studienfach</u>
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	Informatik
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	BWL
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	MB

- Wenn $P = \{\text{HNr}, \text{Studienfach}\}$ die Menge der Primärschlüsselattribute ist, dann gilt für jede beliebige Attributmenge X : $P \rightarrow X$
- Aus gegebenen funktionalen Abhängigkeiten lassen sich weitere Abhängigkeiten ohne Kenntnis der Attributbedeutung ableiten:
 - Z.B. folgt aus:
 $\{\text{hnr}\} \rightarrow \{\text{Straße}, \text{Ort}\}$ automatisch $\{\text{HNr}\} \rightarrow \{\text{Straße}\}$
- Für die Ableitung weiterer Abhängigkeiten gibt es Regeln, die sog. Inferenzregeln (Inference Rules) nach Armstrong

Armstrongs Regeln

- IR1 (Reflexivität):

$$Y \subseteq X \Rightarrow X \rightarrow Y$$

- IR2 (Augmentation):

$$X \rightarrow Y \Rightarrow X \cup Z \rightarrow Y \cup Z$$

- IR3 (Transitivität)

$$X \rightarrow Y \text{ und } Y \rightarrow Z \Rightarrow X \rightarrow Z$$

Anmerkungen zu Armstrongs Regeln

- Die durch IR1 gegebenen Abhängigkeiten heißen trivial
- Armstrong hat 1974 gezeigt, dass die Regeln IR1-3 vollständig sind:
 - Wenn man diese Regeln solange auf eine Menge F von funktionalen Abhängigkeiten anwendet, bis keine neuen Abhängigkeiten mehr erzeugt werden, so erhält man alle Abhängigkeiten, die aus F herleitbar sind
- Die Menge aller aus F herleitbaren Abhängigkeiten heißt Hülle (Closure) von F (symbolisch F^+)

Beispiel

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	<u>Studienfach</u>
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	Informatik
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	BWL
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	MB

Aus {HNr} \rightarrow {Name, Straße, PLZ, Ort} lässt sich nur mit Armstrongs Regel folgendes ableiten (Wie?):

- {HNr} \rightarrow {Name}
- {HNr, Studienfach} \rightarrow {HNr, Name, Straße, PLZ, Ort, Studienfach}

Weitere aus IR1-IR3 ableitbare Inferenzregeln

- IR4 (Zerlegung)

$$X \rightarrow Y \cup Z \Rightarrow X \rightarrow Y \text{ und } X \rightarrow Z$$

- IR5 (Vereinigung)

$$X \rightarrow Y \text{ und } X \rightarrow Z \Rightarrow X \rightarrow Y \cup Z$$

- IR6 (Pseudotransitivität)

$$X \rightarrow Y \text{ und } W \cup Y \rightarrow Z \Rightarrow W \cup X \rightarrow Z$$

- IR7 (Komposition)

$$X \rightarrow Y \text{ und } V \rightarrow W \Rightarrow X \cup V \rightarrow Y \cup W$$

Erinnerung: Schlüssel

- Relation $r(R) = r(A_1, \dots, A_n)$ sei gegeben
- Superschlüssel $S \subseteq R = \{A_1, \dots, A_n\}$
 - Superschlüssel ist eine Attribut-Teilmenge, die für jede gültige Ausprägung r ein Tupel eindeutig identifiziert:
 - $t_1 \neq t_2 \Rightarrow t_1[S] \neq t_2[S]$ bzw. $t_1[S] = t_2[S] \Rightarrow t_1 = t_2$
 - Funktionale Abhängigkeit: $S \rightarrow R$
- Schlüsselkandidat: minimaler Superschlüssel
 - Kein Attribut kann aus S entfernt werden, ohne Eindeutigkeit zu verletzen
 - Mehrere Schlüsselkandidaten pro Relation möglich
- Primärschlüssel(Schlüssel)
 - Ein vom Schemadesigner ausgewählter Schlüsselkandidat

Zurück zur Redundanzvermeidung

Hochschule					
<u>HNr</u>	Name	Straße	PLZ	Ort	<u>Studienfach</u>
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	Informatik
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	BWL
H1	HS Niederrhein	Reinarzstraße 49	47805	Krefeld	MB

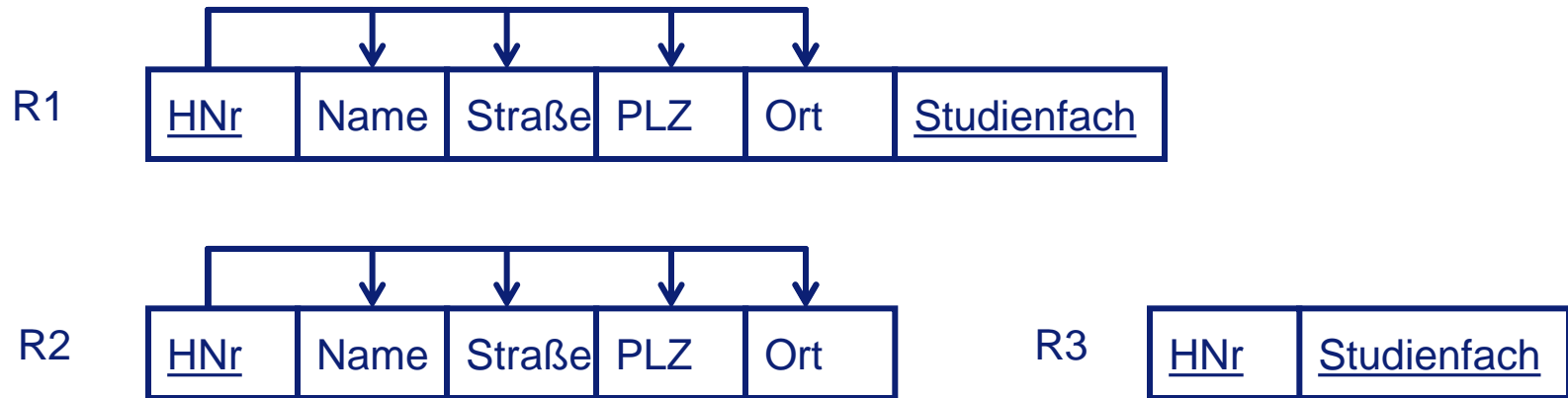
überflüssige Werte

- Frage:
 - wodurch entsteht die Redundanz der Attributwerte?
- Antwort:
 - Redundante Attribute hängen nur von einem Teil des Primärschlüssels {HNr, Studienfach} funktional ab, nämlich HNr!
 - Wie kann diesen Umstand formal definieren?

2. Normalform

- Definition (*Annahme: nur ein Schlüsselkandidat*)
 - Ein Relationschema ist in **2. Normalform** (2NF), wenn jedes **nicht-primäre** Attribut **voll funktional** vom Primärschlüssel abhängt
- Erläuterungen
 - Ein Attribut heißt **primär**, wenn es zum Primärschlüssel gehört
 - Alle anderen Attribute heißen **nicht-primär**
 - Eine Attributmenge Y heißt **voll funktional abhängig** von einer Attributmenge X, wenn sie von keiner echten Teilmenge von X funktional abhängig ist
 - Tut sie es doch, heißt Y **partiell abhängig** von X

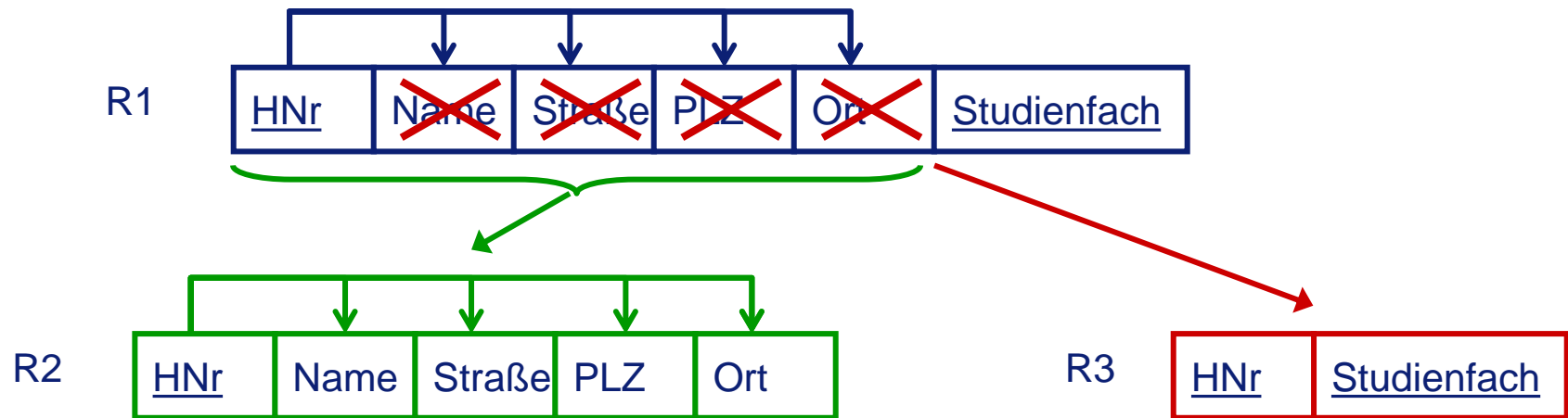
Beispiel zu 2. Normalform



- Relation R1 ist nicht in 2NF – Warum?
 - Name hängt bereits nur von HNr ab, aber nicht vom ganzen Primärschlüssel {HNr, Studienfach}, d.h. partielle Abhängigkeit
- Relation R2 ist in 2NF – Warum?
 - Primärschlüssel enthält nur ein Attribut, d.h. es kann gar keine partiellen Abhängigkeiten geben.
 - Oder anders ausgedrückt: bzgl. R2 ist HNr eindeutig
- Relation R3 ist in 2NF – Warum?
 - Es gibt keine nicht-primären Attribute, die die 2NF verletzen könnten

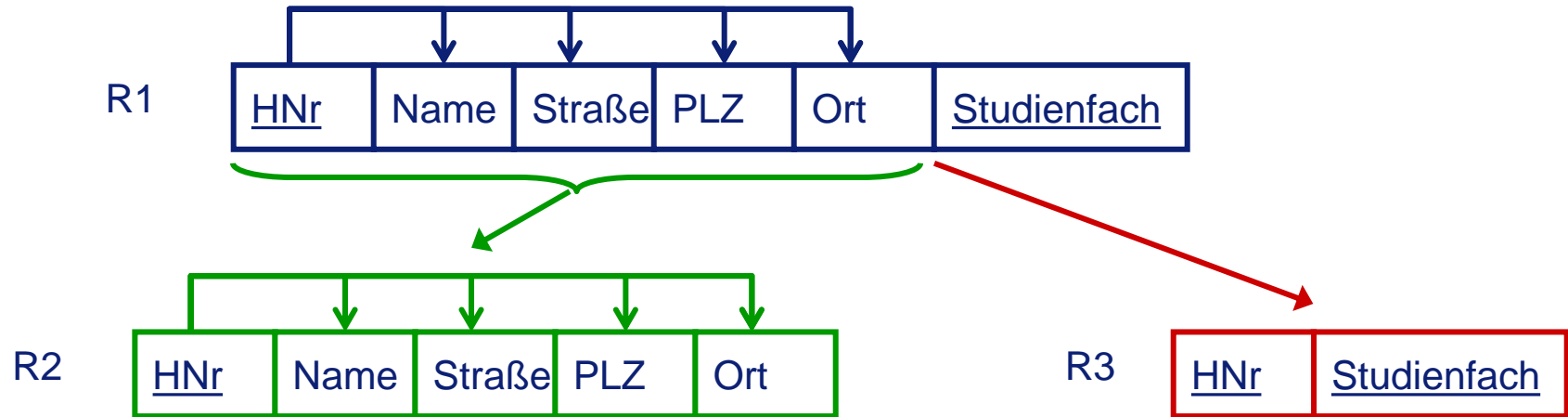
2NF-Dekomposition

- Ein Relationenschema, das nicht in 2NF ist, kann in mehrere Schemata in 2NF zerlegt werden
 - a) Fasse alle nicht-primären Attribute, die nur von einem Teilschlüssel abhängen, mit diesem Teilschlüssel als Primärschlüssel in einer eigenen Relation zusammen
 - b) Alle Attribute, die vom selben Teilschlüssel abhängen, müssen in derselben Relation zusammengefasst werden
 - c) Entferne die ausgelagerten nicht-primären Attribute aus der Ursprungsrelation und fasse die übriggebliebenen Attribute zu einer neuen Relation zusammen



Rekonstruktion

- R1 wurde zerlegt in R2 und R3

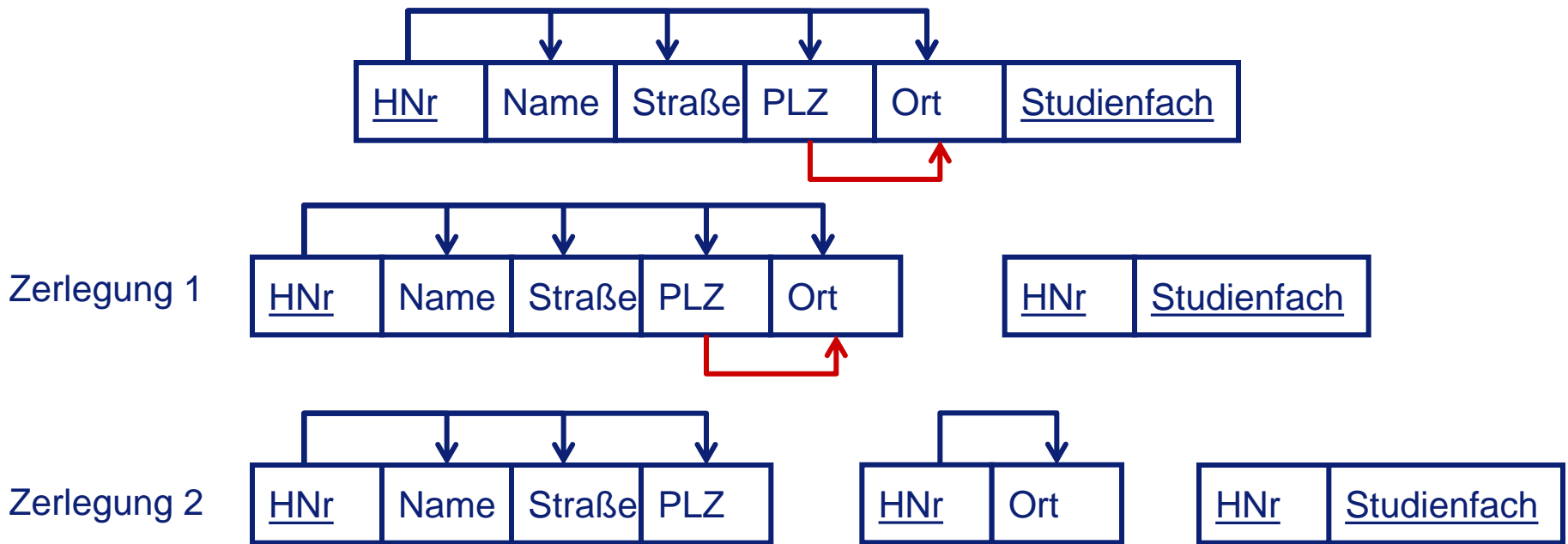


- Wie lässt sich dieser Vorgang rückgängig machen?

```
SELECT  R2.HNr, R2.Name, R2.Strasse,  
        R2.PLZ, R2.Ort, R3.Studienfach  
FROM    R2 [LEFT OUTER] JOIN R3 ON R2.hnr = R3.hnr;
```

- Wünschenswerte Eigenschaften
 - Kein Informationsverlust (später)
 - Abhängigkeitserhaltung

Abhängigkeitserhaltung



- Wofür brauchen wir Regel b?
- Zerlegung 1: Korrekt gemäß Regeln a-c
- Zerlegung 2: Verletzt Regel b
 - Ort und PLZ hängen vom gleichen Teilschlüssel HNr ab, sind aber nicht in der derselben Relation zusammengefasst
 - FA {PLZ} → {Ort} geht verloren
 - FAs können nicht über Relationengrenzen hinweg definiert werden

Redundanzen trotz 2NF

- 2NF verhindert bestimmte Redundanzen, andere aber nicht:

<u>HNr</u>	Name	Straße	PLZ	Ort
H2	FH Aachen	Eupener Straße 70	52066	Aachen
H3	KFH Aachen	Bayernallee	52066	Aachen

↑
redundanter Wert

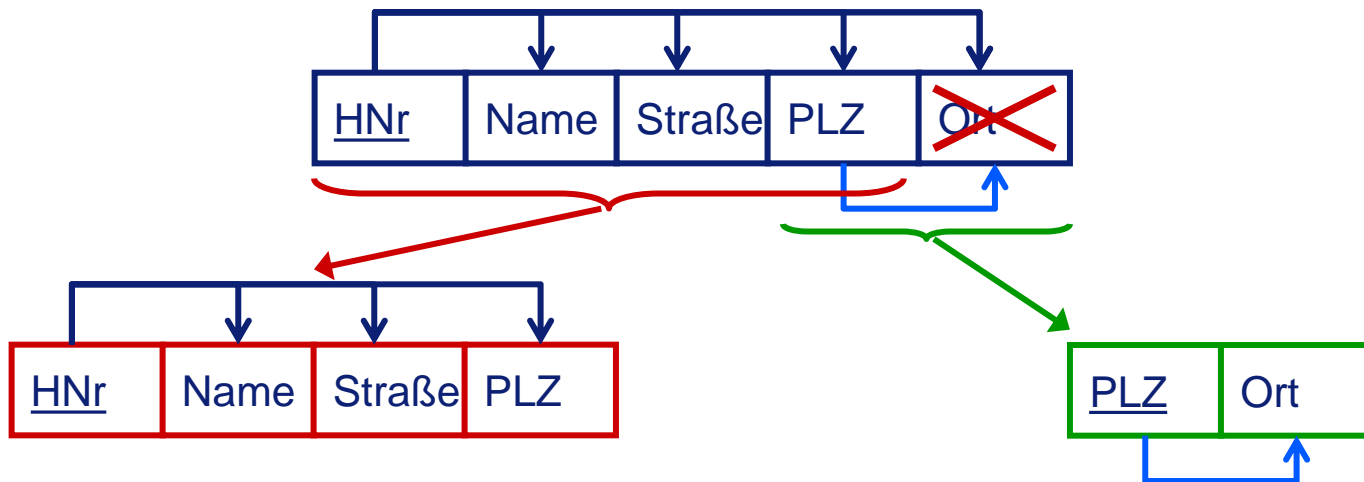
- Relation ist in 2NF – Warum?
- Attribut Ort ist trotzdem redundant
- Ursache:
 - funktionale Abhängigkeit von nicht-primären Attribut PLZ bzw. **transitive** Abhängigkeit HNr → PLZ → Ort

3. Normalform

- Definition (*Annahme: nur ein Schlüsselkandidat*)
 - Ein Relationenschema ist in dritter Normalform (3NF), wenn es in 2NF ist und kein nicht-primäres Attribut transitiv vom Primärschlüssel abhängt
- Erläuterungen:
 - C heißt **transitiv** von A abhängig, wenn es eine Attributmenge B gibt mit: $A \rightarrow B$ und $B \rightarrow C$
 - Die beiden Abhängigkeiten bei der Transitivität dürfen nicht **trivial** sein: $B \subseteq A$ oder $B \subseteq C$
 - Beispiel für eine triviale transitive Abhängigkeit
 - $\{HNr\} \rightarrow \{Name, Straße\} \rightarrow \{Straße\}$
 - $\{HNr, Studienfach\} \rightarrow \{HNr\} \rightarrow \{Name, Straße\}$

3NF-Dekomposition

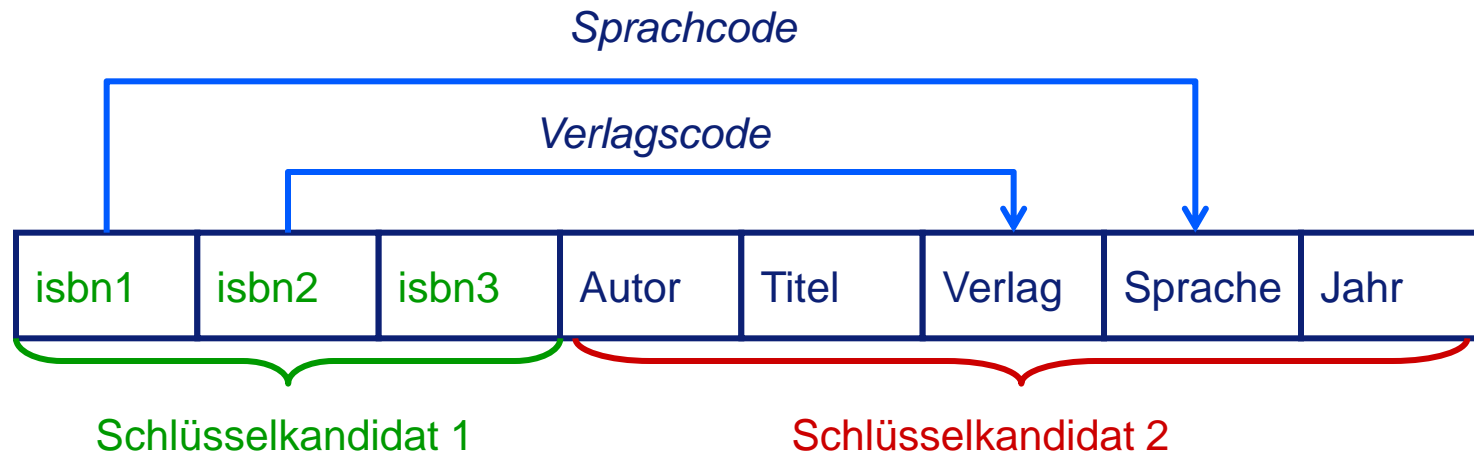
- Ein Relationenschema, das nicht in 3NF ist, kann in mehrere Schemata in 3NF zerlegt werden
 - Fasse die transitiv-abhängigen nicht-primären Attribute zusammen mit den Attributen, von denen sie nur direkt abhängen, in einer eigenen Tabelle zusammen
 - Entferne die ausgelagerten abhängigen Attribute aus der Ursprungstabelle und fasse die übriggebliebenen Attribute zu einer neuen Relation zusammen



Verallgemeinerung: mehr als 1 Schlüsselkandidat

- Bisher haben wir angenommen, dass es genau einen Schlüsselkandidaten gibt (der dann Primärschlüssel ist)
 - Jetzt: Verallgemeinerung auf mehr als einen SK
- Zwei Varianten
 - 2. Normalform nach Codd (2NF-Codd):
Jedes Attribut, das zu keinem Schlüsselkandidaten gehört, ist von *jedem* Schlüsselkandidaten voll-funktional abhängig
 - 2. Normalform nach Kent (2NF-Kent):
Jedes Attribut im Komplement eines Schlüsselkandidaten ist von *diesem* Schlüsselkandidaten voll-funktional abhängig

Beispiel



- Relation Buch hat mehrere Schlüsselkandidaten
- Relation ist in 2NF-Codd – Warum?
 - es gibt gar keine Attribute, die zu keinem SK gehören
- Relation ist nicht 2NF-Kent – Warum?
 - Sprache ist partiell abhängig vom Schlüsselteil ISBN1

Unterschied 2NF-Codd/2NF-Kent

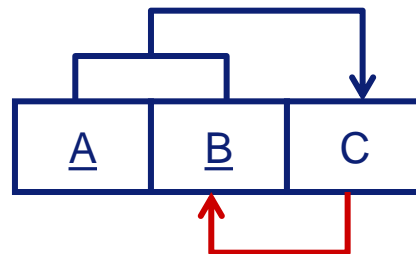
- 2NF-Codd betrachtet alle Schlüsselkandidaten auf einmal
- 2NF-Kent betrachtet jeden Schlüsselkandidaten für sich
- Im allgemeinen Fall ist 2NF-Codd schwächer, d.h. lässt mehr Redundanzen zu als 2NF-Kent
- Definitionen identisch, wenn es nur 1 Schlüsselkandidaten gibt

Verallgemeinerung 3NF auf mehrere SK

- 3NF-Codd
 - Die Relation ist in 2NF-Codd und kein Attribut, das zu keinem Schlüsselkandidaten gehört, ist von einem Schlüsselkandidaten transitiv abhängig
- 3NF-Kent
 - Die Relation ist in 2NF-Kent und kein Attribut im Komplement eines Schlüsselkandidaten ist von diesem Schlüsselkandidaten transitiv abhängig
- Bemerkung:
 - Definition nur für einen Schlüsselkandidaten identisch
 - 3NF-Kent betrachtet jeden Schlüsselkandidaten für sich
 - 3NF-Codd ist wieder schwächer als 3NF-Kent
- Wenn wir keine Angabe machen, gehen wir immer von der Codd'schen Definition aus (bei 2NF und 3NF)

Boyce/Codd-Normalform

- Definition Boyce/Codd-Normalform
 - Für jede nicht-triviale Abhängigkeit $X \rightarrow A$ gilt: X ist ein Superschlüssel der Relation
- Bemerkung
 - Erfordert keine Überprüfung von 2NF
 - Ursprünglich als einfachere Definition von 3NF vorgeschlagen; später hat sich herausgestellt, dass BCNF strenger als 3NF ist
 - Beispiel: Relation ist in 3NF, aber nicht in BCNF – Warum?



Noch ein Beispiel für 3NF vs. BCNF

- Relation Stadt(Ort, BLand, Ministerpräsident, EW)
- Geltende FAs:
 - $\{\text{Ort, BLand}\} \rightarrow \{\text{EW}\}$
 - $\{\text{BLand}\} \rightarrow \{\text{Ministerpräsident}\}$
 - $\{\text{Ministerpräsident}\} \rightarrow \{\text{BLand}\}$
- Schlüsselkandidaten
 - $\{\text{Ort, BLand}\}$
 - $\{\text{Ort, Ministerpräsident}\}$
- Stadt ist in 3NF, aber nicht in BCNF
 - EW ist das einzige Attribut, das nicht zu einem SK gehört, aber wir haben keine transitiven FAs zu EW, daher in 3NF
 - BLand und Ministerpräsident sind keine Superschlüssel, daher nicht in BCNF

Weitere Normalformen

- Es gibt noch weitere Normalformen 4NF und 5NF, die nicht auf funktionalen Abhängigkeiten basieren, sondern auf sogenannten mehrwertigen Abhängigkeiten bzw. Verbundabhängigkeiten
- In der Praxis beschränkt man sich auf 3NF und BCNF
 - Die 4NF wird nur in seltenen Fällen verletzt
 - Verletzungen der 5NF sind nur schwer zu erkennen
- Weitere exotische Normalformen
 - z.B. Domain-Key-NF
 - Umformung beruht jedoch nicht ausschließlich Projektion für die Zerlegung und Join für die Rekonstruktion

Übersicht: Normalformen

- Jede Stufe definiert echt strengere Kriterien

1NF: beliebige Relationen mit atomaren Werten

2NF: alle Attribute voll vom Schlüssel abhängig

3NF: keine transitiven Abhängigkeiten

BCNF: nur Abhängigkeiten vom Schlüssel

4NF: keine Mehrwert-Abhängigkeiten

5NF: keine trivialen Verbundabhängigkeiten

Zerlegung vs. Rekonstruktion

- Wir haben Algorithmen kennen gelernt, mit deren Hilfe man ein nicht-normalisiertes Ausgangsschema in eine Menge von normalisierten Schemata zerlegen kann
 - Zerlegungsregeln führen zu weniger redundanten Schemata
 - Projektive Dekomposition
- Stillschweigende Annahme
 - Die zerlegten Schemata sind von ihrem Informationsgehalt gleichwertig zum Ausgangsschema
 - Das Ausgangsschema lässt sich durch Joins rekonstruieren
 - Ist dies tatsächlich der Fall?

Korrektheitskriterien für die Dekomposition

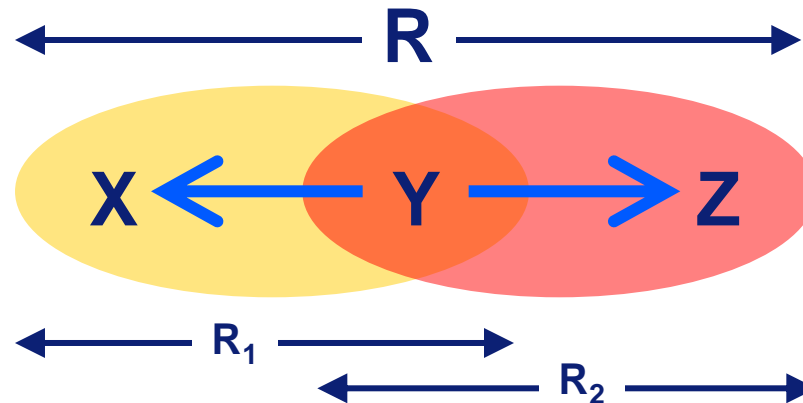
- Annahme:
 - ein Relationenschema R soll in die Relationschemata R_1, \dots, R_n zerlegt werden
 - r, r_1, \dots, r_n seien Ausprägungen von R, R_1, \dots, R_n
- Abhängigkeitserhaltung
 - Die für R geltenden FAs müssen auf R_1, \dots, R_n übertragbar sein und dort jeweils isoliert überprüft werden können
- Verlustlosigkeit
 - Der Verbund (Join) der zerlegten Relationen r_1, \dots, r_n muss wieder die ursprüngliche Relation r ergeben, d.h. die in r enthaltenen Informationen müssen wieder komplett aus r_1, \dots, r_n rekonstruierbar sein

Kriterium für Verlustlosigkeit

- $R = (X, Y, Z)$, $R_1 = (X, Y)$, $R_2 = (Y, Z)$, $R = R_1 \cup R_2$
 - r sei eine Ausprägung von R
 - $r_1 := \text{SELECT } X, Y \text{ FROM } R$ (Projektion auf X, Y -Attribute)
 - $r_2 := \text{SELECT } Y, Z \text{ FROM } R$ (Projektion auf Y, Z -Attribute)
- Die Zerlegung von R in R_1 und R_2 ist verlustlos, wenn für jede (gültige) Ausprägung r von R gilt
 - $r = r_1 \bowtie r_2$
- Bemerkung
 - \bowtie ist der Relationale Algebra-Operator für Join

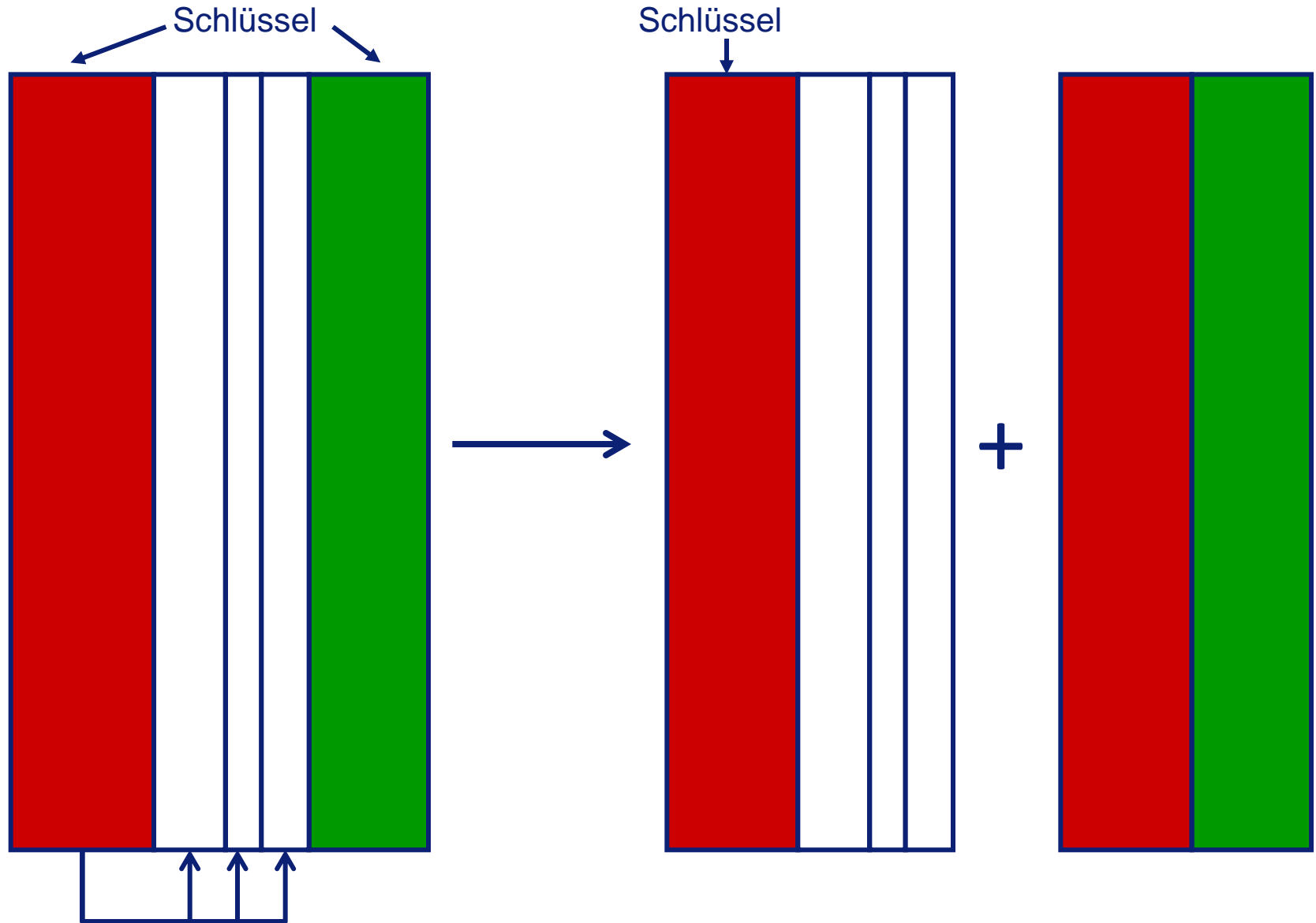
Theorem von Heath

- Hinreichende Bedingung für Verlustlosigkeit (Theorem von Heath)
 - $Y := (R_1 \cap R_2)$, $Y \rightarrow R_1$ oder $Y \rightarrow R_2$
 - Zerlege R so in R_1 und R_2 , dass eine Attributmenge Y in beiden Zerlegungen enthalten ist und (zumindest) für eine Zerlegung R_1 oder R_2 Schlüsselfunktion hat

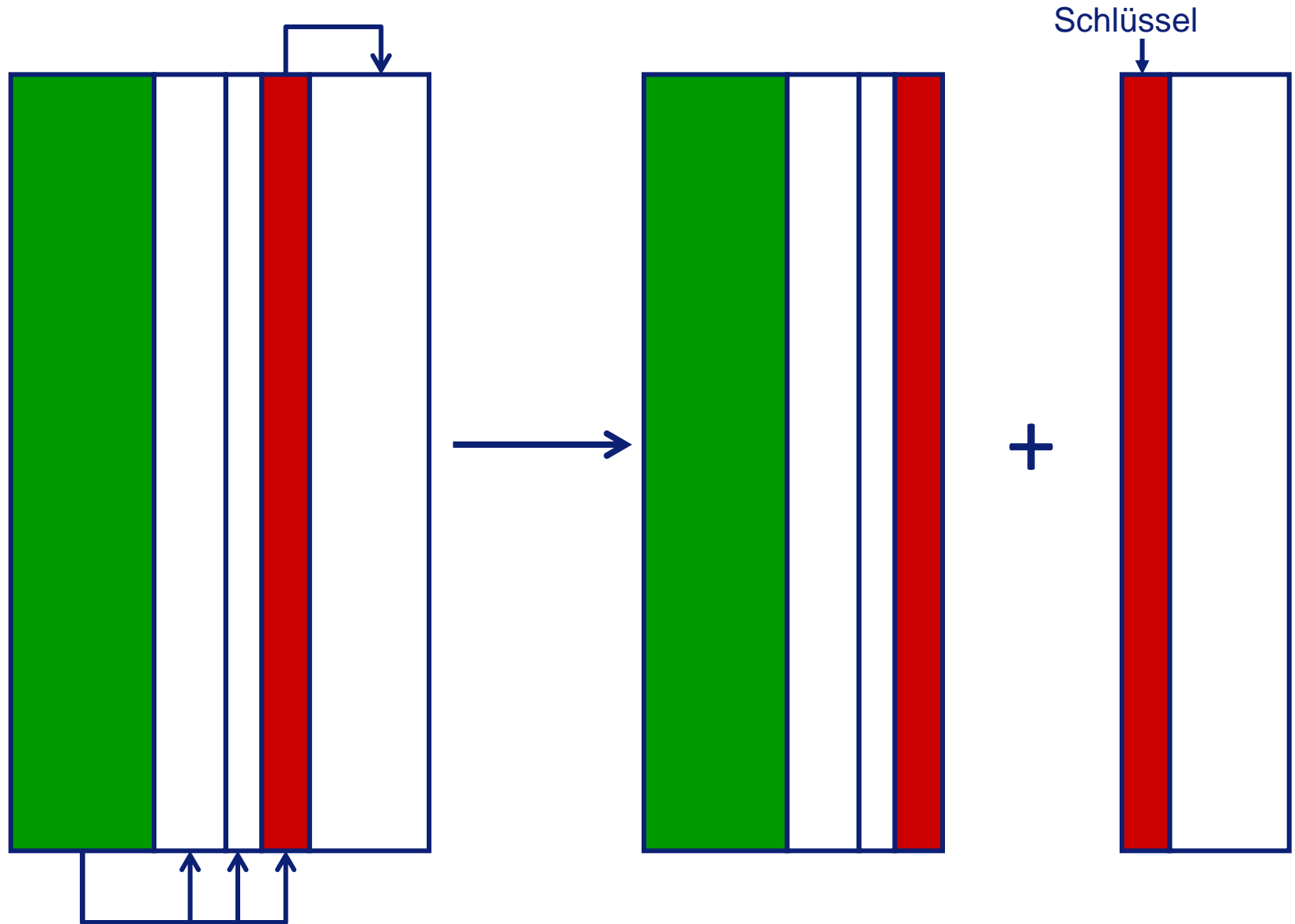


- Die Zerlegungsregeln von 2NF und 3NF (und BCNF) sind so gewählt, dass die Voraussetzung des Theorems von Heath erfüllt ist

Heath-Bedingung bei 2NF



Heath-Bedingung bei 3NF



Nicht-verlustfreie Zerlegung

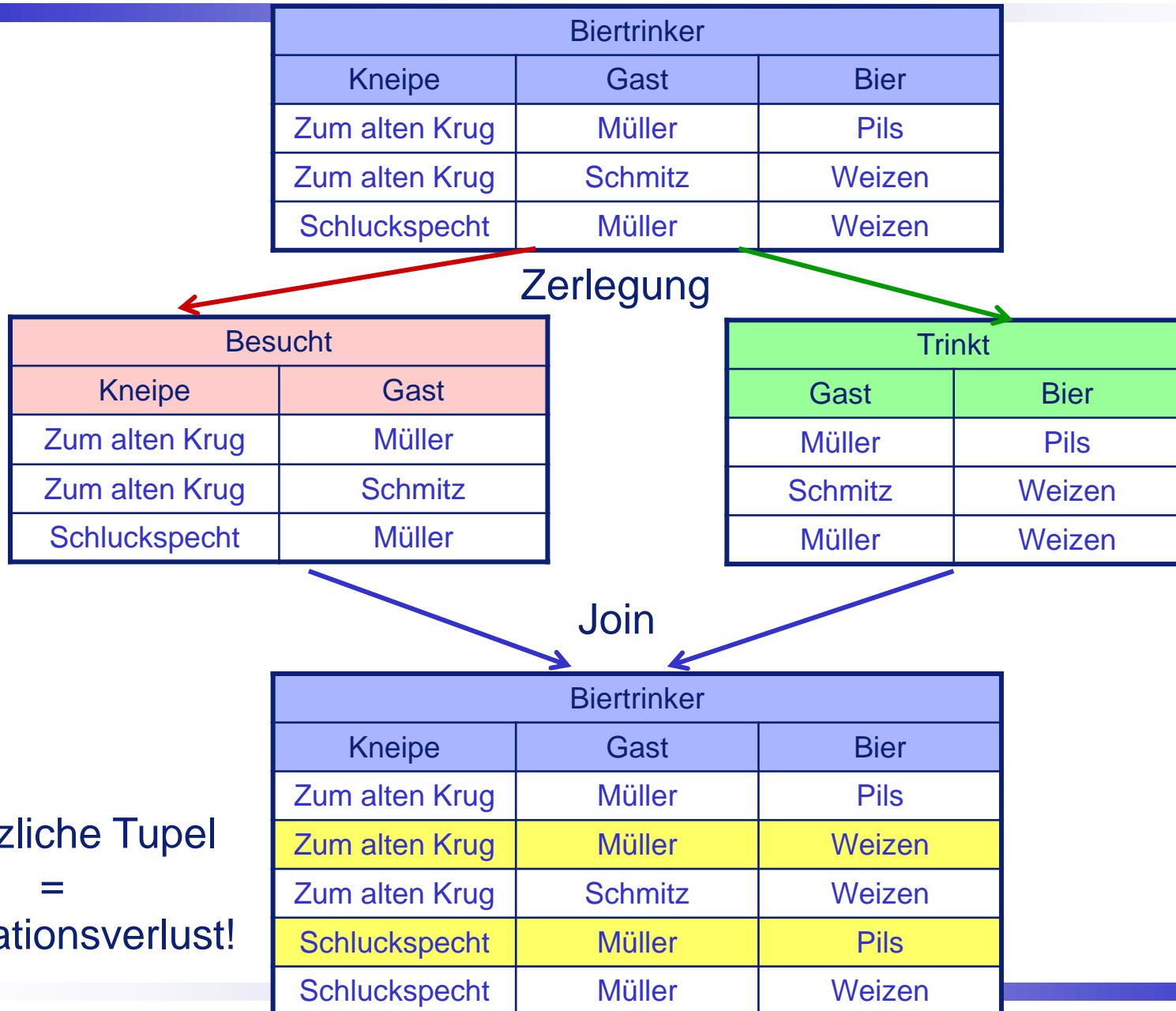
Biertrinker		
Kneipe	Gast	Bier
Zum alten Krug	Müller	Pils
Zum alten Krug	Schmitz	Weizen
Schluckspecht	Müller	Weizen

Zerlegung

Besucht	
Kneipe	Gast
Zum alten Krug	Müller
Zum alten Krug	Schmitz
Schluckspecht	Müller

Trinkt	
Gast	Bier
Müller	Pils
Schmitz	Weizen
Müller	Weizen

Informationsverlust



Warum ist die Zerlegung im Beispiel nicht verlustfrei?

- Die hinreichende Bedingung für eine verlustfreie Zerlegung war verletzt:
 - Einzige (nicht-triviale) FA:
 - $\{\text{Kneipe}, \text{Gast}\} \rightarrow \{\text{Bier}\}$
 - Die beiden möglichen, die Verlustlosigkeit garantierenden FAs gelten nicht:
 - $\{\text{Gast}\} \rightarrow \{\text{Bier}\}$
 - $\{\text{Gast}\} \rightarrow \{\text{Kneipe}\}$
- Das liegt daran, dass die Gäste (hier: Müller) in unterschiedlichen Kneipen verschiedene Biere trinken
 - In derselben Kneipe trinkt ein Gast allerdings immer das gleiche Bier
 - Diese Information geht in der Zerlegung verloren.
 - Informations-"Verlust" = zusätzliche unerwünschte Tupel

Abhängigkeitserhaltung

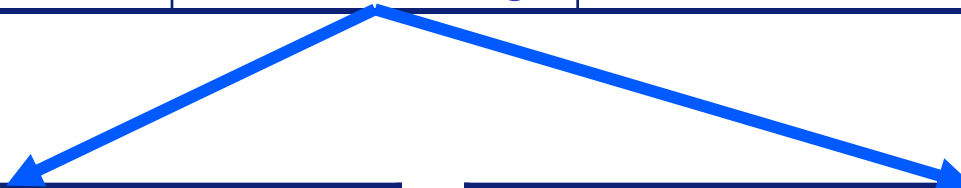
- Bei der Zerlegung von R in R_i , $i=1..n$ werden nur solche FAs übernommen, deren beteiligte Attribute jeweils komplett in R_i enthalten sind
- Grund:
 - FAs können nicht Relationenübergreifend definiert werden
- Formale Definition:
 - F_R : Menge der FAs, die auf Relationenschema R definiert sind
 - $F_{R_i} := \{X \rightarrow Y \in F_R / X \cup Y \in R_i \text{ für ein } i\}$
- Eine Zerlegung von R in R_1, \dots, R_n ist abhängigkeiterhaltend, wenn gilt:
 - $F_R = F_{R_1} \cup \dots \cup F_{R_n}$

Beispiel für Abhängigkeitsverlust

- PLZVerzeichnis(Straße, Ort, BLand, PLZ)
- Annahmen
 - Orte werden durch ihren Namen (Ort) und das Bundesland (BLand) eindeutig identifiziert
 - Innerhalb einer Straße ändert sich die Postleitzahl nicht
 - Postleitzahlengebiete gehen nicht über Ortsgrenzen und Orte nicht über Bundeslandgrenzen hinweg
- Daraus resultieren die FAs
 - $\{PLZ\} \rightarrow \{Ort, BLand\}$
 - $\{Straße, Ort, BLand\} \rightarrow \{PLZ\}$
- Betrachte die Zerlegung
 - Straßen(PLZ, Straße)
 - Orte(PLZ, Ort, BLand)

Zerlegung der Relation PLZVerzeichnis

PLZverzeichnis			
<u>Ort</u>	<u>BLand</u>	<u>Straße</u>	PLZ
Frankfurt	Hessen	Goethestraße	60313
Frankfurt	Hessen	Galgenstraße	60437
Frankfurt	Brandenburg	Goethestraße	15234



Straßen	
<u>PLZ</u>	<u>Straße</u>
15234	Goethestraße
60313	Goethestraße
60437	Galgenstraße

Orte		
<u>Ort</u>	<u>BLand</u>	<u>PLZ</u>
Frankfurt	Hessen	60313
Frankfurt	Hessen	60437
Frankfurt	Brandenburg	15234

Die FA $\{\text{Straße, Ort, BLand}\} \rightarrow \{\text{PLZ}\}$ ist im zerlegten Schema nicht mehr enthalten → Einfügen inkonsistenter Tupel möglich

Einfügen zweier Tupel, die die FA verletzen

PLZverzeichnis			
<u>Ort</u>	<u>BLand</u>	<u>Straße</u>	PLZ
Frankfurt	Hessen	Goethestraße	60313
Frankfurt	Hessen	Galgenstraße	60437
Frankfurt	Brandenburg	Goethestraße	15234

Straßen

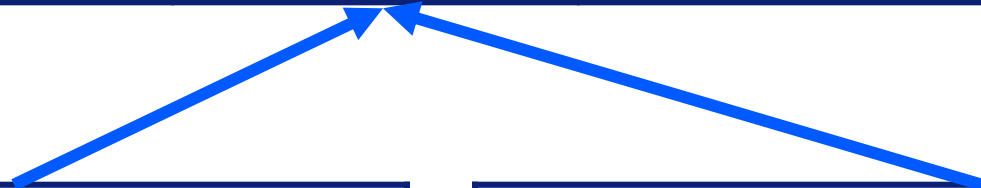
<u>PLZ</u>	<u>Straße</u>
15234	Goethestraße
60313	Goethestraße
60437	Galgenstraße
15235	Goethestraße

Orte

<u>Ort</u>	<u>BLand</u>	<u>PLZ</u>
Frankfurt	Hessen	60313
Frankfurt	Hessen	60437
Frankfurt	Brandenburg	15234
Frankfurt	Brandenburg	15235

Join verletzt FA {Straße, Ort, BLand} → {PLZ}

PLZverzeichnis			
<u>Ort</u>	<u>BLand</u>	<u>Straße</u>	PLZ
Frankfurt	Hessen	Goethestraße	60313
Frankfurt	Hessen	Galgenstraße	60437
Frankfurt	Brandenburg	Goethestraße	15234
Frankfurt	Brandenburg	Goethestraße	15235

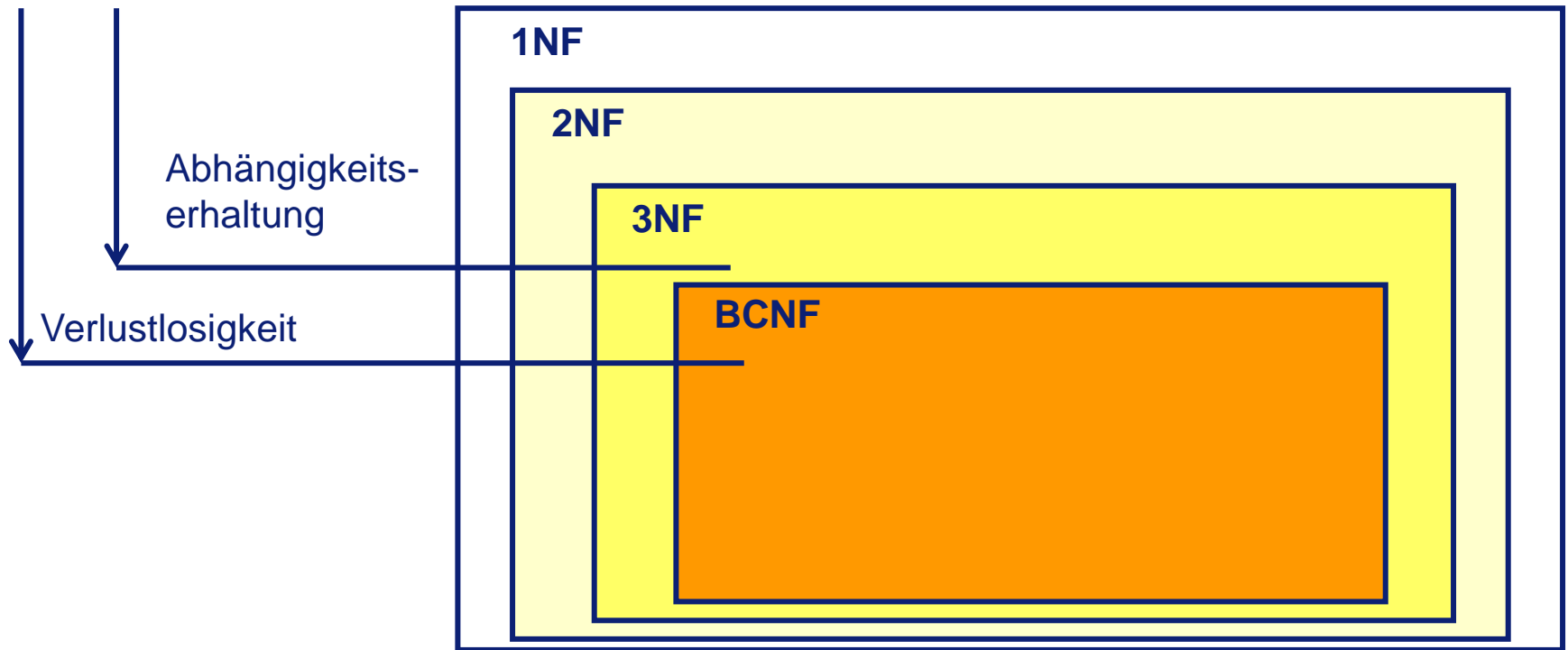


Straßen	
<u>PLZ</u>	<u>Straße</u>
15234	Goethestraße
60313	Goethestraße
60437	Galgenstraße
15235	Goethestraße

Orte		
<u>Ort</u>	<u>BLand</u>	<u>PLZ</u>
Frankfurt	Hessen	60313
Frankfurt	Hessen	60437
Frankfurt	Brandenburg	15234
Frankfurt	Brandenburg	15235

Verlustlosigkeit und Abhängigkeitserhaltung

- Die Verlustlosigkeit ist für alle Zerlegungsalgorithmen in alle Normalformen garantiert (sogar bis 4NF)
- Die Abhängigkeitserhaltung kann leider nur bis zur 3NF garantiert werden



Zusammenfassung

- Beim DB-Entwurf können schlechte Schemata entstehen (Redundanzen)
- **Normalformen** und **FAs** definieren ein formal überprüfbares Qualitätsmaß für die Güte eines Schemas, indem bestimmte Redundanzen auf Schemaebene ausgeschlossen werden
- **Normalisierung** bezeichnet die schrittweise Zerlegung eines Schemas in mehrere Teil-Schemata, die höheren Normalformen genügen
- Jedes Schema bis 3NF zerlegbar, so dass **Verlustlosigkeit** und **Abhängigkeitserhaltung** gewahrt bleiben
- Abschließende Bemerkung
 - Nicht-normalisierte Schemata entstehen meist bei einem unüberlegten **Ad-hoc**-Entwurf
 - Fast alle Redundanzprobleme werden von vorneherein verhindert bei sauberem Entwurf nach **ER-Methodik** → nächstes Kapitel