

Datenbanken und Informationssysteme

6. Relationale Entwurfstheorie

Prof. Dr. Stefan Decker

📍 RWTH Aachen

Warum gutes Design über die Datenqualität entscheidet

Von funktionalen Abhängigkeiten zu Zerlegung und Normalformen

Kapitelüberblick

1. Motivation - Warum „gutes“ Datenbankdesign?
2. Funktionale Abhängigkeiten
3. Armstrong-Kalkül
4. Kanonische Überdeckung
5. Zerlegung von Relationen: Zerlegungskriterien
6. Normalformen und Normalisierungen

Leitidee: Gute relationale Entwürfe reduzieren Redundanz, vermeiden Änderungsanomalien und begründen Zerlegungen formal.

1. Motivation

Warum „gutes“ Datenbankdesign?

Leitfrage: Warum können scheinbar einfache relationale Schemata bereits Redundanz, Anomalien und Inkonsistenzen erzeugen?



Fokus dieses Abschnitts

Wir untersuchen ein absichtlich problematisches Schema und leiten daraus ab, waran man schlechtes relationales Design erkennt.

☰ Kapitelpfad

1. **Motivation - Warum „gutes“ Datenbankdesign?**
2. Funktionale Abhängigkeiten
3. Armstrong-Kalkül
4. Kanonische Überdeckung
5. Zerlegung von Relationen: Zerlegungskriterien
6. Normalformen und Normalisierungen

Wo stehen wir?

- ▶ Anforderungsanalyse: fachliche Anforderungen verstehen
- ▶ Konzeptioneller Entwurf: z. B. ER-Modell
- ▶ Logischer Entwurf: relationales Schema (Tabellen)

Nächster Schritt: Relationale Entwurfstheorie

Warum?

- ▶ Nicht jedes relationale Schema ist automatisch gut.
- ▶ Ein schlechtes Schema kann zu Problemen führen.

Ziel der Entwurfstheorie

- ▶ Kriterien für „gute“ relationale Schemata entwickeln.
- ▶ Methoden zur Verbesserung „schlechter“ Schemata lernen.
- ▶ Sicherstellung von Datenintegrität und Vermeidung von Redundanz und Anomalien.

Motivation - Ein problematisches Schema

Betrachten wir ein *scheinbar einfaches* Schema, das Informationen über Professoren und ihre Vorlesungen kombiniert:

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
...	...	...	...	...	...	...
2132	Popper	W2	52	5259	Der Wiener Kreis	2
2137	Kant	W3	7	4630	Die 3 Kritiken	4

Erster Eindruck: Sieht auf den ersten Blick vielleicht okay aus, **ABER...**

Motivation - Änderungsanomalie

Problem 1: Redundanz führt zu Änderungsanomalien.

Frage: Was passiert, wenn Prof. Sokrates (2125) von Raum 226 nach Raum 233 umzieht?

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
...	...	...	...	...	...	...
2132	Popper	W2	52	5259	Der Wiener Kreis	2
2137	Kant	W3	7	4630	Die 3 Kritiken	4

Warum ist das problematisch?

- ▶ **Redundanz:** Die Information „Sokrates ist in Raum 226“ ist mehrfach gespeichert.
- ▶ **Update-Aufwand:** Alle Einträge von Sokrates müssen gefunden und geändert werden.
- ▶ **Inkonsistenz-Gefahr:** Wenn nicht alle Einträge geändert werden, steht Sokrates plötzlich in mehreren Räumen gleichzeitig.

Motivation - Einfügeanomalie

Problem 2: Die Zusammenfassung unterschiedlicher Sachverhalte führt zu Einfüge-Anomalien.

Zwei Einfügefälle

Frage 1: Wie fügen wir einen neuen Professor Platon (2140, W3, Raum 101) ein, bevor er eine Vorlesung hält?

- ▶ Das geht nicht direkt.
- ▶ Wir bräuchten Platzhalter oder NULL für VorlNr, Titel und SWS.
- ▶ Falls VorlNr Teil des Schlüssels ist, scheitert selbst das.

Frage 2: Wie fügen wir die neue Vorlesung „Metaphysik“ (5060, 4 SWS) ein, bevor ein Prof. zugeteilt ist?

- ▶ Ebenfalls nicht direkt.
- ▶ Dann bräuchten PersNr, Name, Rang und Raum NULL-Werte.

Ausgangsschema

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
...	...	...	...	...	...	...
2132	Popper	W2	52	5259	Der Wiener Kreis	2
2137	Kant	W3	7	4630	Die 3 Kritiken	4

Kernproblem: Das Schema zwingt uns, Professoren- und Vorlesungsinformationen gleichzeitig einzufügen, obwohl oft nur eine Seite vorliegt.

Motivation - Löschanomalie

Problem 3: Die Zusammenfassung führt auch zu Löschanomalien.

Frage: Was passiert, wenn Prof. Poppers einzige Vorlesung “Der Wiener Kreis” (5259) aus dem Programm genommen wird?

ProfVorl						
PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
...	...	...	...	...	...	...
2132	Popper	W2	52	5259	Der Wiener Kreis	2
...	...	...	...	...	...	...



Diesen einen Eintrag löschen?

Unbeabsichtigte Folge

Mit dem Löschen der Vorlesungsinformation verschwinden zugleich alle Informationen über Prof. Popper:

- ▶ Name
- ▶ Rang
- ▶ Raum

Der Professor geht verloren, obwohl nur die Vorlesung entfernt werden sollte. **Merksatz:** Löschen darf keinen unbeabsichtigten Informationsverlust auslösen.

Fazit aus dem Beispiel 'ProfVorl':

- ▶ **Redundanz:** Dieselbe Information erscheint mehrfach in verschiedenen Tupeln.
- ▶ **Anomalien:** Update-, Einfüge- und Löschooperationen haben unerwünschte Nebenwirkungen.

⚠ Redundanz und Anomalien sind Indikatoren für einen schlechten relationalen Entwurf.

1. Formale Kriterien

Wir brauchen **formale Kriterien**, um gutes und schlechtes Schemadesign unabhängig vom konkreten Beispiel zu unterscheiden.

2. Verbesserungsmethode

Wir brauchen mit der **Normalisierung** eine Methode, um Schemata anhand dieser Kriterien systematisch zu verbessern.

Nächster Schritt: Funktionale Abhängigkeiten liefern dafür die formale Grundlage.

Motivation - Übergang zum Werkzeug

Leitfrage: Warum führt das ProfVor1-Schema zu Problemen? Weil es *Abhängigkeiten* zwischen den Attributen gibt, die durch die Struktur ungünstig abgebildet werden.

Im Beispiel ProfVor1

- ▶ Die PersNr bestimmt eindeutig Name, Rang, Raum.
- ▶ Die Vor1Nr bestimmt eindeutig Titel, SWS.
- ▶ PersNr und Vor1Nr zusammen bestimmen das ganze Tupel.

Formalisierung

Wie können wir solche „A bestimmt B“-Beziehungen formal fassen?

Gesucht ist ein Werkzeug, das solche Abhängigkeiten präzise als Eigenschaft eines Schemas beschreibt.

Das zentrale Werkzeug dafür sind die Funktionalen Abhängigkeiten (FDs).

Nächster Teil: Funktionale Abhängigkeiten

2. Funktionale Abhängigkeiten

Das formale Werkzeug für relationales Design

Leitfrage: Wie beschreiben wir präzise, dass bestimmte Attribute andere Attribute eindeutig festlegen?

Fokus dieses Abschnitts

Wir definieren funktionale Abhängigkeiten, interpretieren sie als Schemaeigenschaften und nutzen sie anschliessend für Entwurfsregeln und Normalformen.

Kapitelpfad

1. Motivation - Warum „gutes“ Datenbankdesign?
2. **Funktionale Abhängigkeiten**
3. Armstrong-Kalkül
4. Kanonische Überdeckung
5. Zerlegung von Relationen: Zerlegungskriterien
6. Normalformen und Normalisierungen

Funktionale Abhängigkeiten - Definition

Zentrales Werkzeug der relationalen Entwurfstheorie: Funktionale Abhängigkeit
(Functional Dependency, FD)

Notation

$$\alpha \rightarrow \beta$$

Gelesen: „ α bestimmt β “ oder „ β ist funktional abhängig von α “.

- ▶ $\alpha, \beta \subseteq X$: Attributmengen des Schemas X
- ▶ α : Determinante
- ▶ β : abhängiger Teil

Formale Definition

Eine Relation R erfüllt $\alpha \rightarrow \beta$, wenn für alle Tupelpaare $t_1, t_2 \in R$ gilt:

$$t_1[\alpha] = t_2[\alpha] \implies t_1[\beta] = t_2[\beta]$$

Gleiche Werte auf der linken Seite erzwingen also gleiche Werte auf der rechten Seite.

Wichtig: Eine FD ist eine Schemaeigenschaft und muss für alle gültigen Instanzen der Relation gelten.

Triviale Abhängigkeit: Ist $\beta \subseteq \alpha$, dann gilt $\alpha \rightarrow \beta$ immer, z. B. $\{MatNr, Name\} \rightarrow \{Name\}$.

Konvention: Mengenklammern werden oft weggelassen: $\{MatNr, Name\} \rightarrow \{Name\}$ wird zu $MatNr, Name \rightarrow Name$.

Funktionale Abhängigkeiten - Beispiele

Frage: Welche FDs gelten in der Instanz, und welche sollen als Schema-FDs gelten?

Beispielrelation Student

Student			
MatNr	Name	#Semester	Status
1234	Michael	6	eingeschrieben
5678	Andrea	4	eingeschrieben
4711	Sabine	8	beurlaubt
0815	Franz	12	exmatrikuliert
9999	Michael	2	eingeschrieben

Schema vs. Instanz

- ▶ Eine FD gehört nur dann zum Schema, wenn sie in jeder gültigen Instanz gelten muss.
- ▶ $\text{Name} \rightarrow \text{MatNr}$ scheitert an den zwei Tupeln mit Michael; $\text{Semester} \rightarrow \text{Status}$ gilt hier nur zufällig.
- ▶ Wie bei ProfVorl: $\text{PersNr} \rightarrow \text{Name, Rang, Raum}$ ist eine typische Schema-FD.

Beurteilung möglicher FDs

Potentielle FD	Gilt in Instanz R?	Soll im Schema gelten?
$\text{MatNr} \rightarrow \text{Name, Semester, Status}$	JA (keine gleiche MatNr)	JA
$\text{Name} \rightarrow \text{Semester, Status, MatNr}$	NEIN (Michael)	NEIN
$\text{Semester} \rightarrow \text{Status}$	JA (zufällig)	NEIN
$\text{Status} \rightarrow \text{Semester}$	NEIN (eingeschrieben)	NEIN
$\text{MatNr, Name} \rightarrow \text{Semester, Status}$	JA ($\{\text{MatNr}\}$ genügt)	JA

Funktionale Abhängigkeiten - Überprüfung

Frage: Wie prüft man algorithmisch, ob $\alpha \rightarrow \beta$ in einer **gegebenen Instanz R** gilt?

Algorithmus zur Instanzprüfung

1. **Sortieren:** Sortiere alle Tupel primär nach den Attributwerten aus α , sodass gleiche α -Werte direkt hintereinander stehen.
2. **Gruppen scannen:** Gehe die sortierte Relation durch, identifiziere zusammenhängende Gruppen mit gleichem α -Wert und prüfe für jede Gruppe, ob alle β -Werte übereinstimmen.
3. **Entscheidung:** Falls alle Gruppen konsistent sind, gilt $\alpha \rightarrow \beta$ in dieser Instanz; sobald eine Gruppe verschiedene β -Werte hat, gilt die FD nicht in R .

Merksätze

- ▶ Nur Instanztest: kein Beweis dafür, dass die FD fachlich zum Schema gehört.
- ▶ Nach dem Sortieren genügt es, Gruppen mit gleichem α -Wert auf gleiche β -Werte zu prüfen.
- ▶ Eine widersprüchliche Gruppe widerlegt die FD in R ; Komplexität durch Sortieren bestimmt: $O(n \log n)$.

Funktionale Abhängigkeiten - Schlüssel

FDs erlauben eine präzise Definition zentraler Schlüsselbegriffe für $R(X)$ mit FD-Menge F .

1. Superschlüssel (Superkey)

Definition: $\alpha \subseteq X$ heißt Superschlüssel, falls $\alpha \rightarrow X$ gilt.

Bedeutung: α bestimmt alle Attributwerte eindeutig.

Beispiel: X selbst ist immer ein Superschlüssel.

3. Schlüsselkandidat

Definition: $\alpha \subseteq X$ ist ein Schlüsselkandidat, wenn X voll funktional von α abhängig ist.

Äquivalent: α ist ein minimaler Superschlüssel.

2. Voll funktionale Abhängigkeit

Definition: β ist voll funktional abhängig von α , wenn $\alpha \rightarrow \beta$ gilt und keine echte Teilmenge $\alpha' \subset \alpha$ schon $\alpha' \rightarrow \beta$ erfüllt.

Bedeutung: Kein Attribut in α ist überflüssig.

4. Primärschlüssel

Definition: Einer der Schlüsselkandidaten, der vom Designer ausgewählt wird.

Funktionale Abhängigkeiten - Schlüsselbeispiel

Gegebene Situation

Beispiel: Relation ORTE **Schema:** Orte(Name, BLand, Vorwahl, EW)

Attributmenge: $X = \{\text{Name, BLand, Vorwahl, EW}\}$

Angenommene FDs

- ▶ F1: Name, BLand $\rightarrow$ Vorwahl, EW (Annahme: Name in BLand ist eindeutig)
- ▶ F2: Name, Vorwahl $\rightarrow$ BLand, EW (Annahme: Vorwahl allein reicht nicht)

Beispielrelation ORTE

Orte			
Name	BLand	Vorwahl	EW
Frankfurt	Hessen	069	690.000
Frankfurt	Brandenburg	0335	60.000
München	Bayern	089	1.378.000
Passau	Bayern	0851	50.000

Anwendung der Definitionen

1. **Superschlüssel:** Aus F1 folgt $\{\text{Name, BLand}\} \rightarrow X$ und aus F2 folgt $\{\text{Name, Vorwahl}\} \rightarrow X$; Obermengen wie $\{\text{Name, BLand, EW}\}$ sind ebenfalls Superschlüssel.
2. **Schlüsselkandidaten:** $\{\text{Name, BLand}\}$ und $\{\text{Name, Vorwahl}\}$ sind minimal; beide sind also Schlüsselkandidaten.
3. **Primärschlüssel:** $\{\text{Name, BLand}\}$ wird als Primärschlüssel ausgewählt.

Funktionale Abhängigkeiten - Implikationen

Bisher: Definition von FDs ($\alpha \rightarrow \beta$) und Schlüsseln basierend auf FDs.

Beobachtung / Nächstes Problem

Gegebene FDs F implizieren oft weitere FDs.

Beispiel

$PLZ \rightarrow Ort, BLand \quad Ort, BLand \rightarrow Vorwahl$

$\implies PLZ \rightarrow Vorwahl$

Die implizierte FD ist nicht explizit notiert, folgt aber logisch aus den gegebenen Abhängigkeiten.

Offene Frage

Wie erkennen wir systematisch, welche weiteren FDs aus einer gegebenen Menge F logisch folgen?

Einzelne Beispiele genügen nicht.
Wir brauchen dafür eine allgemeine und präzise Methode.

3. Armstrong-Kalkül

Implizierte FDs formal aus einer FD-Menge ableiten

Leitfrage: Wie leiten wir implizierte funktionale Abhängigkeiten formal aus einer gegebenen FD-Menge F ab?

Fokus dieses Abschnitts

Wir führen ein kleines Kalkül mit wenigen Axiomen ein. Damit können wir FDs syntaktisch herleiten, die Hülle F^+ systematisch beschreiben und später Korrektheit und Vollständigkeit beweisen.

Kapitelpfad

1. Motivation - Warum „gutes“ Datenbankdesign?
2. Funktionale Abhängigkeiten
3. **Armstrong-Kalkül**
4. Kanonische Überdeckung
5. Zerlegung von Relationen: Zerlegungskriterien
6. Normalformen und Normalisierungen

Armstrong-Kalkül - Warum die Hülle F^+ ?

Problem

Gegebene FDs F implizieren oft weitere FDs.

Beispiel

PLZ $\rightarrow$ Ort, BLand Ort, BLand $\rightarrow$ Vorwahl

$\implies$ PLZ $\rightarrow$ Vorwahl

Idee: Die Hülle F^+

F^+ ist die Menge *aller* FDs, die logisch aus F folgen.

Damit beschreibt F^+ die gesamte logische Konsequenz von F .

Warum ist es wichtig, F^+ zu kennen?

1. **Vollständiges Verständnis:** Welche Constraints gelten im Schema?
2. **Korrekte Schlüssel:** Ist ein Schlüsselkandidat minimal?
3. **Redundanz in F erkennen:** Ist eine FD notwendig oder folgt sie aus anderen?
4. **Basis für Normalisierung:** Zur Normalisierung muss man alle FDs im Schema kennen.

Ziel: Methoden, um F^+ zu bestimmen.

Nächster Schritt: Definition der **Semantischen Implikation**

Armstrong-Kalkül - Semantische Implikation

Ziel: Verstehen, was „FD f folgt logisch aus Menge F “ bedeutet.

Aus der Logik: $M \models \Phi$ heißt: M ist Model von Φ .

$R \models F$ und $Sat(F)$

$R \models F$: Eine Relation R erfüllt eine Menge FDs F , wenn R jede einzelne FD $f \in F$ erfüllt.

$Sat(F)$: Menge aller Relationen (über Schema X), die F erfüllen. Formal: $Sat(F) = \{R \mid R \models F\}$.

Semantische Implikation $F \models f$

Definition: F impliziert die FD f (semantisch), genau dann wenn gilt: Jede Relation R , die F erfüllt, erfüllt auch f .

Formal:

$$F \models f \text{ iff } \forall R : R \models F \Rightarrow R \models f$$

Äquivalent: $Sat(F) \subseteq Sat(\{f\})$

Bedeutung: f ist eine logische Konsequenz von F , die sich allein aus der Definition von FDs ergibt.

Problem: Wie prüft man $F \models f$? Das Testen endlich vieler Relationen ist nicht praktikabel. **Lösung:** Syntaktischer Ansatz mit Ableitungsregeln (Axiomen) $\Rightarrow$ **Der Armstrong-Kalkül** (Armstrong, 1974)

William Ward Armstrong: *Dependency Structures of Data Base Relationships*, page 580–583. IFIP Congress, 1974.

Armstrong-Kalkül - Die Axiome A_1, A_2, A_3

Armstrong-Axiome (Armstrong 1974):

Drei Grundregeln zur syntaktischen Ableitung von funktionalen Abhängigkeiten.

Seien $\alpha, \beta, \gamma \subseteq X$.

A_1 : Reflexivität

Regel: Wenn $\beta \subseteq \alpha$, dann gilt auch $\alpha \rightarrow \beta$.

Bedeutung: Eine Attributmenge bestimmt immer jede ihrer Teilmengen.

Beispiel:

$\{MatrNr, Name\} \rightarrow \{Name\}$
 $\{MatrNr, Name\} \rightarrow \{MatrNr\}$

A_2 : Verstärkung

Regel: Aus $\alpha \rightarrow \beta$ folgt $\alpha\gamma \rightarrow \beta\gamma$.
Mit $\alpha\gamma := \alpha \cup \gamma$

Bedeutung: Fügt man auf beiden Seiten die gleichen Attribute hinzu, bleibt die FD gültig.

Beispiel: Aus

$\{MatrNr\} \rightarrow \{Name\}$ folgt
 $\{MatrNr, Semester\} \rightarrow \{Name, Semester\}$

A_3 : Transitivität

Regel: Aus $\alpha \rightarrow \beta$ und $\beta \rightarrow \gamma$ folgt $\alpha \rightarrow \gamma$.

Bedeutung: Abhängigkeiten können verkettet werden.

Beispiel: Aus $\{PLZ\} \rightarrow \{Ort\}$ und
 $\{Ort\} \rightarrow \{BLand\}$
folgt $\{PLZ\} \rightarrow \{BLand\}$

Kernaussage: Diese drei Axiome bilden die Basis des Armstrong-Kalküls.

Übergang: Mit ihnen definieren wir als nächstes Ableitbarkeit und die Hülle F^+ .

W. W. Armstrong: *Dependency Structures of Data Base Relationships*.
IFIP Congress, 1974, pp. 580–583.

Armstrong-Kalkül - Ableitung & Hülle F^+

Neue Perspektive: Mit den Axiomen A_1 , A_2 und A_3 können wir funktionale Abhängigkeiten **syntaktisch ableiten**.

Ableitbarkeit ($F \vdash f$)

Definition: f ist aus F ableitbar, wenn f durch endlich viele Anwendungen von A_1 , A_2 oder A_3 aus F folgt.

Formal: Es gibt eine endliche Folge $f_1, \dots, f_n = f$, wobei jedes f_i entweder in F liegt oder aus $F \cup \{f_1, \dots, f_{i-1}\}$ mit einem Axiom gewonnen wird.

Beispiel: Aus $A \rightarrow B$ und $B \rightarrow C$ folgt $A \rightarrow C$ durch A_3 .

Hülle / Abschluss F^+

Definition: Die Hülle F^+ einer Menge FDs F ist die Menge aller aus F ableitbaren FDs:

$$F^+ = \{ f \mid F \vdash f \}$$

Bedeutung: F^+ repräsentiert alle syntaktischen Konsequenzen von F .

Verbindung zur Semantik: Ist der syntaktische Kalkül ($\vdash$) genau so mächtig wie die semantische Implikation ($\models$)?

Gilt also $F \vdash f \iff F \models f$?

$\Rightarrow$ Als nächstes: **Korrektheit & Vollständigkeit**

Satz: Korrektheit & Vollständigkeit des Armstrong-Kalküls

Ja: Die syntaktische Ableitung ($\vdash$) ist äquivalent zur semantischen Implikation ($\models$).

Der Armstrong-Kalkül (Axiome A_1 – A_3) ist **korrekt** und **vollständig** für funktionale Abhängigkeiten. Für jede FD f gilt: $F \vdash f \iff F \models f$.

Korrektheit

Aussage: $F \vdash f \Rightarrow F \models f$

Bedeutung: Alles, was wir mit den Axiomen ableiten können, ist auch eine logische Konsequenz von F .

Kein "Unsinn" wird abgeleitet.

Beweisidee: Zeige, dass jedes Axiom semantisch korrekt ist.

Vollständigkeit

Aussage: $F \models f \Rightarrow F \vdash f$

Bedeutung: Alles, was logisch aus F folgt, kann auch mit den Axiomen abgeleitet werden.

Keine gültige FD geht verloren.

Beweisidee: Falls f nicht ableitbar ist, konstruiert man ein Gegenbeispiel.

Konsequenz: Die syntaktisch definierte Hülle F^+ ist *genau* die Menge aller semantisch implizierten FDs.

Übergang: Als nächstes beginnt der Beweis mit der **Korrektheit**.

Armstrong-Kalkül - Korrektheit Beweis

Ziel: Zeige die Korrektheit $F \vdash f \Rightarrow F \models f$.

Beweisidee: Zeige, dass jedes Axiom A_1 – A_3 semantisch korrekt ist. Dann ist auch jede endliche Ableitung korrekt.

Prüfung A_1 : Reflexivität

Zu zeigen: Aus $\beta \subseteq \alpha$ folgt $\alpha \rightarrow \beta$.

Seien $R \in \text{Sat}(F)$, $\alpha \subseteq X$, $\beta \subseteq \alpha$ und $t_1, t_2 \in R$ mit $t_1[\alpha] = t_2[\alpha]$.

Dann stimmen t_1 und t_2 insbesondere auch auf allen Attributen aus β überein. Also gilt $t_1[\beta] = t_2[\beta]$.

Folgerung: $\alpha \rightarrow \beta$ gilt in R .

Prüfung A_2 : Verstärkung

Zu zeigen: Aus $\alpha \rightarrow \beta$ folgt $\alpha\gamma \rightarrow \beta\gamma$.

Seien $R \in \text{Sat}(F)$, $\alpha, \beta, \gamma \subseteq X$ und $\alpha \rightarrow \beta \in F$. Außerdem seien $t_1, t_2 \in R$ mit $t_1[\alpha \cup \gamma] = t_2[\alpha \cup \gamma]$.

Dann gilt sowohl $t_1[\alpha] = t_2[\alpha]$ als auch $t_1[\gamma] = t_2[\gamma]$. Wegen $\alpha \rightarrow \beta$ folgt $t_1[\beta] = t_2[\beta]$.

Also gilt insgesamt $t_1[\beta \cup \gamma] = t_2[\beta \cup \gamma]$.

Folgerung: $\alpha\gamma \rightarrow \beta\gamma$ gilt in R .

Zwischenstand: A_1 und A_2 erhalten die FD-Semantik. zur Korrektheit.

Als nächstes: A_3 (Transitivität) und der Schluss

Armstrong-Kalkül - Korrektheit Beweis (2/2)

Letzter Schritt: Es fehlt noch A_3 (Transitivität).

Danach folgt die Korrektheit des Kalküls durch den strukturellen Schluss über endliche Axiomanwendungen.

Prüfung A_3 : Transitivität

Zu zeigen: Aus $\alpha \rightarrow \beta$ und $\beta \rightarrow \gamma$ folgt $\alpha \rightarrow \gamma$.

Seien $R \in \text{Sat}(F)$, $\alpha, \beta, \gamma \subseteq X$ und $\alpha \rightarrow \beta, \beta \rightarrow \gamma \in F$. Außerdem seien $t_1, t_2 \in R$ mit $t_1[\alpha] = t_2[\alpha]$.

Wegen $\alpha \rightarrow \beta$ gilt dann $t_1[\beta] = t_2[\beta]$.

Wegen $\beta \rightarrow \gamma$ folgt daraus weiter $t_1[\gamma] = t_2[\gamma]$.

Folgerung: Also gilt $\alpha \rightarrow \gamma$ in R .

Schluss: Korrektheit

1. Jedes Armstrong-Axiom erhält die $\models$ -Beziehung.
2. Jede Ableitung ist eine endliche Kette solcher Axiomanwendungen.
3. Daher gilt:

$$F \vdash f \Rightarrow F \models f$$

Ergebnis: Der Kalkül ist korrekt.

Abschluss: Der Korrektheitsbeweis ist komplett.
Werkzeug: die **Attributhülle**.

Übergang: Für die Vollständigkeit brauchen wir ein weiteres

Statt Hülle F^+ : Welche Attribute sind durch α bestimmt?

Definition Attributhülle (α_F^+)

Die Attributhülle α^+ von α unter F ist die Menge aller Attribute A , so dass $\alpha \rightarrow A$ aus F ableitbar ist:

$$\alpha_F^+ = \{ A \mid F \vdash \alpha \rightarrow A \}$$

Superschlüsseltest

$\kappa \subseteq X$ ist genau dann ein Superschlüssel, falls gilt:

$$\kappa^+ = X$$

Algorithmus Attributhülle(F, α)

1. **Initialisierung:** $H_0 := \alpha, i := 0$
2. **Wiederhole:**
 - ▶ Setze $H_{i+1} := H_i$ und erhöhe $i := i + 1$.
 - ▶ Betrachte jede FD $\beta \rightarrow \gamma$ in F .
 - ▶ Wenn $\beta \subseteq H_{i-1}$, dann erweitere: $H_i := H_i \cup \gamma$.
3. **Stoppe, wenn:** $H_i = H_{i-1}$
4. **Ausgabe:** H_i als H_{final}

Leitfrage: Warum berechnet dieser Algorithmus genau die richtige Attributhülle?

Als nächstes: Beispiel, Korrektheit und Vollständigkeit.

Armstrong-Kalkül - Beispiel: Attributhülle(F, α)

Gegeben

FD-Menge: $F = \{A \rightarrow C, B \rightarrow A, AB \rightarrow C\}$ **Gesucht:** Attributhülle($F, \{B\}$) **Startwert:** $H_0 = \{B\}$

Iterationsablauf

Schritt	Ausgangsmenge	Anwendbare FD	Neue Hülle
Initialisierung	$H_0 = \{B\}$	–	Start mit $\{B\}$
Iteration 1	$H_0 = \{B\}$	$B \rightarrow A$, da $\{B\} \subseteq H_0$	$H_1 = \{A, B\}$
Iteration 2	$H_1 = \{A, B\}$	$A \rightarrow C$, da $\{A\} \subseteq H_1$	$H_2 = \{A, B, C\}$
Iteration 3	$H_2 = \{A, B, C\}$	keine neue FD liefert ein weiteres Attribut	$H_3 = H_2$
Abbruch	$H_3 = H_2$, also ist der Fixpunkt erreicht und die Schleife endet.		

Ergebnis: $H_{\text{final}} = \{A, B, C\}$. Die Attributhülle von $\{B\}$ ist also $\{A, B, C\}$. Neue Attribute werden genau so lange ergänzt, bis ein Fixpunkt erreicht ist.

Armstrong-Kalkül - Korrektheit und Vollständigkeit des Algorithmus: Attributhülle(F, α)

Ziel: $A \in \text{Attributhülle}(F, \alpha) \Leftrightarrow A \in \alpha_F^+$

Leitidee: Korrektheit zeigt $H_{final} \subseteq \alpha_F^+$, Vollständigkeit zeigt $\alpha_F^+ \subseteq H_{final}$.

Korrektheit

1. Invariante: Nach jeder Iteration ist alles in H_i bereits aus α ableitbar.

2. Basis: $H_0 = \alpha$, also $F \vdash \alpha \rightarrow H_0$ durch Reflexivität.

3. Schritt: Wird H_{i+1} wegen $\beta \rightarrow \gamma$ erweitert und gilt $\beta \subseteq H_i$, dann folgt aus der Induktionshypothese $F \vdash \alpha \rightarrow \beta$. Mit Transitivität erhält man $F \vdash \alpha \rightarrow \gamma$.

Folge: Der Algorithmus erzeugt keine nicht ableitbaren Attribute.

Vollständigkeit

1. Idee: Betrachte einen minimalen Ableitungsbaum für $\alpha \rightarrow A$ und induziere über die maximale Pfadlänge l von den Blättern zur Wurzel.

2. Basis: $l = 0 \Rightarrow A \in \alpha \Rightarrow A \in H_0$.

3. Schritt: Für $l > 0$ gibt es eine FD $\beta \rightarrow \gamma \in F$ mit $A \in \gamma$. Für alle $D \in \beta$ hat die Ableitung von $\alpha \rightarrow D$ kleinere Höhe; also gilt nach Induktionshypothese $D \in H_{k_D}$.

Wegen der Monotonität $H_i \subseteq H_{i+1}$ ist damit β spätestens in $H_{k_{max}}$ enthalten; folglich wird A in Schritt $H_{k_{max}+1}$ hinzugefügt.

Folge: Jedes ableitbare Attribut erscheint irgendwann in der berechneten Hülle.

Ergebnis: $H_{final} = \alpha_F^+$.

Armstrong-Kalkül - Vollständigkeit: Konstruktion der Relation R

Zu zeigen: $F \models f \Rightarrow F \vdash f$. **Technik:** Kontraposition.

Kontraposition: $F \not\models f \Rightarrow F \not\vdash f$.

Annahme: $f = \alpha \rightarrow \beta$, $F \not\models f$; wir konstruieren R mit $R \models F$, aber $R \not\models f$. Kurz: $\alpha^+ := \alpha_F^+$.

Ausgangspunkt

Aus $F \not\models \alpha \rightarrow \beta$ folgt $\beta \notin \alpha^+$.

Daher ist $X \setminus \alpha^+ \neq \emptyset$.

Wir schreiben: $\alpha^+ = \{a_1, \dots, a_n\}$,

$X \setminus \alpha^+ = \{b_1, \dots, b_m\}$.

Außerdem gilt: $\gamma \subseteq \alpha^+ \Rightarrow \gamma_F^+ \subseteq \alpha^+$.

Konstruierte Relation R

R	$a_1, \dots, a_n$	$b_1, \dots, b_m$
r	$1, \dots, 1$	$1, \dots, 1$
t	$1, \dots, 1$	$0, \dots, 0$

Nächste zwei Behauptungen: 1. $R \models F$ 2. $R \not\models f$

Schlüsselidee der Konstruktion

Die beiden Tupel werden so gebaut, dass sie auf allen Attributen in α^+ gleich und auf allen Attributen in $X \setminus \alpha^+$ verschieden sind.

Armstrong-Kalkül - Vollständigkeit: Behauptung 1

Behauptung 1: Die konstruierte Relation erfüllt alle FDs aus F , also $R \models F$.

Beweisidee: Unterschiede zwischen r und t können nur außerhalb von α^+ auftreten.

Widerspruchskette

Annahme: Es gibt $V \rightarrow W \in F$ mit $R \not\models V \rightarrow W$.

Dann: $r[V] = t[V]$, aber $r[W] \neq t[W]$.

Aus der Konstruktion folgt: $V \subseteq \alpha^+$, aber $W \not\subseteq \alpha^+$.

Andererseits: Weil $V \rightarrow W \in F$ und $V \subseteq \alpha^+$ gilt, folgt mit der Definition von α^+ und Transitivität auch $W \subseteq \alpha^+$.

Widerspruch.

Relation R

R	$a_1, \dots, a_n$	$b_1, \dots, b_m$
r	$1, \dots, 1$	$1, \dots, 1$
t	$1, \dots, 1$	$0, \dots, 0$

Lesart: Gleichheit bedeutet: innerhalb von α^+ .

Ungleichheit bedeutet: außerhalb von α^+ .

Folgerung: Keine FD aus F wird in R verletzt. Also gilt $R \models F$.

Armstrong-Kalkül - Vollständigkeit: Behauptung 2

Behauptung 2: Die konstruierte Relation verletzt $f = \alpha \rightarrow \beta$, also $R \not\models f$.

Beweisidee: Wegen $\beta \not\subseteq \alpha^+$ sind r und t auf α gleich, auf β aber verschieden.

Beweiskette

1. Aus $F \not\models \alpha \rightarrow \beta$ folgt $\beta \not\subseteq \alpha^+$.
2. Wähle ein Attribut $b_k \in \beta \setminus \alpha^+$.
3. Weil $\alpha \subseteq \alpha^+$ gilt, folgt aus der Konstruktion $r[\alpha] = t[\alpha]$.
4. Gleichzeitig gilt $r[b_k] = 1 \neq 0 = t[b_k]$.
5. Also gilt $R \not\models \alpha \rightarrow \beta$.

Gewähltes Attribut

Das Attribut b_k liegt in β , aber nicht in α^+ .
Genau daran unterscheiden sich die Tupel r und t .

Relation R

R	$a_1, \dots, a_n$	$b_1, \dots, b_m$
r	$1, \dots, 1$	$1, \dots, 1$
t	$1, \dots, 1$	$0, \dots, 0$

Zusammen: $R \models F$ und $R \not\models f$. Also $F \not\models f$. Durch Kontraposition: $F \models f \Rightarrow F \vdash f$.

Armstrong-Kalkül: Erweiterung des Kalküls - die Axiome A_4 , A_5 , A_6

Nützliche Kurzregeln: A_4 bis A_6 vereinfachen Ableitungen, fügen dem Kalkül aber keine neue Ausdrucksmacht hinzu. Sie sind aus A_1 bis A_3 ableitbar.

Voraussetzung: Seien α, β, γ Teilmengen der Attributmenge X .

A_4 : Vereinigung

Regel: Gelten $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$, so gilt auch $\alpha \rightarrow \beta\gamma$.

Bedeutung: Wenn α sowohl β als auch γ bestimmt, dann auch ihre Vereinigung.

Beispiel:

MatrNr $\rightarrow$ Name

MatrNr $\rightarrow$ Semester

$\Rightarrow$ MatrNr $\rightarrow$ Name, Semester

A_5 : Dekomposition

Regel: Gilt $\alpha \rightarrow \beta\gamma$, so gelten auch $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$.

Bedeutung: Bestimmt α die Menge $\beta\gamma$, dann bestimmt α auch jeden Teil davon einzeln.

Beispiel:

MatrNr $\rightarrow$ Name, Semester

$\Rightarrow$ MatrNr $\rightarrow$ Name

MatrNr $\rightarrow$ Semester

A_6 : Pseudotransitivität

Regel: Gelten $\alpha \rightarrow \beta$ und $\beta\gamma \rightarrow \delta$, so gilt auch $\alpha\gamma \rightarrow \delta$.

Bedeutung: Wenn α schon β bestimmt, kann man β links in einer anderen FD durch α ersetzen.

Beispiel:

VorlNr $\rightarrow$ PersNr

PersNr, RaumNr $\rightarrow$ Zeit

$\Rightarrow$ VorlNr, RaumNr $\rightarrow$ Zeit

Armstrong-Kalkül: Ableitung der Axiome A_4 , A_5 , A_6

Idee: A_4 bis A_6 sind keine neuen Grundaxiome. Sie lassen sich direkt aus A_1 bis A_3 herleiten.

Notation: $\alpha\gamma := \alpha \cup \gamma$, $\beta\gamma := \beta \cup \gamma$ und $\alpha, \beta, \gamma, \delta \subseteq X$.

A_4 : Vereinigung

Annahme: $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$.

$$\begin{aligned}(A_2) \quad & \alpha \rightarrow \gamma \Rightarrow \alpha \rightarrow \alpha\gamma \\(A_2) \quad & \alpha \rightarrow \beta \Rightarrow \alpha\gamma \rightarrow \beta\gamma \\(A_3) \quad & \alpha \rightarrow \alpha\gamma, \alpha\gamma \rightarrow \beta\gamma \\& \Rightarrow \alpha \rightarrow \beta\gamma\end{aligned}$$

A_6 : Pseudotransitivität

Annahme: $\alpha \rightarrow \beta$ und $\beta\gamma \rightarrow \delta$.

$$\begin{aligned}(A_2) \quad & \alpha \rightarrow \beta \Rightarrow \alpha\gamma \rightarrow \beta\gamma \\(A_3) \quad & \alpha\gamma \rightarrow \beta\gamma, \beta\gamma \rightarrow \delta \\& \Rightarrow \alpha\gamma \rightarrow \delta\end{aligned}$$

A_5 : Dekomposition

Annahme: $\alpha \rightarrow \beta\gamma$.

$$\begin{aligned}(A_1) \quad & \beta\gamma \rightarrow \beta, \beta\gamma \rightarrow \gamma \\(A_3) \quad & \alpha \rightarrow \beta\gamma, \beta\gamma \rightarrow \beta \Rightarrow \alpha \rightarrow \beta \\(A_3) \quad & \alpha \rightarrow \beta\gamma, \beta\gamma \rightarrow \gamma \Rightarrow \alpha \rightarrow \gamma\end{aligned}$$

Folgerung: A_4 , A_5 und A_6 sind nur bequeme Kurzformen für Ableitungen mit den drei Armstrong-Axiomen.

Armstrong-Kalkül: Beispiel

Schema:

ProfessorenAdressen(PersNr, Name, Rang, Raum, Ort, Straße, Hausnummer, PLZ, Vorwahl, Bundesland)

Relevante FDs aus F

Gegeben:

$f_1 = \text{Ort, Bundesland} \rightarrow \text{Vorwahl}$

$f_3 = \text{Ort, Bundesland, Straße, Hausnummer} \rightarrow \text{PLZ}$

...

Ziel

Zu zeigen:

$f = \text{Ort, Bundesland, Straße, Hausnummer} \rightarrow \text{PLZ, Vorwahl}$

ist aus F ableitbar.

Ableitung

(A_1) $f_4 = \text{Ort, Bundesland, Straße, Hausnummer} \rightarrow \text{Ort, Bundesland}$

(A_3) $f_5 = \text{Ort, Bundesland, Straße, Hausnummer} \rightarrow \text{Vorwahl}$ aus f_4 und f_1

(A_4) $\mathbf{f} = \mathbf{f_6} = \text{Ort, Bundesland, Straße, Hausnummer} \rightarrow \text{PLZ, Vorwahl}$

Muster: Zuerst wird mit A_1 und A_3 eine zusätzliche FD hergeleitet; danach fasst A_4 die gegebene und die hergeleitete Abhängigkeit zu einer stärkeren FD zusammen.

Armstrong-Kalkül: Zusammenfassung und Ausblick

Abschluss des Abschnitts: Der Armstrong-Kalkül liefert ein formales Werkzeug, um implizierte FDs herzuleiten und mit Attributhüllen effizient zu prüfen.

Was haben wir gelernt?

- ▶ FDs implizieren weitere FDs ($F \models f$).
- ▶ Der Armstrong-Kalkül ($\vdash$) ist korrekt und vollständig, um diese Implikationen zu finden.
- ▶ Die Attributhülle (α^+) ist ein effizientes Werkzeug zur Berechnung aller von α determinierten Attribute.
- ▶ Mit α^+ können wir FD-Implikationen und Superschlüssel-Eigenschaften prüfen.

Ausblick

- ▶ Eine Menge F von FDs enthält oft Redundanzen. Wie finden wir eine minimale, äquivalente Menge?
- ▶ Diese Frage führt zur „Kanonischen Überdeckung“ als kompakter Darstellung einer FD-Menge.
- ▶ Wenn wir Relationen zerlegen müssen, was macht eine „gute“ Zerlegung aus? Verlustlosigkeit und Abhängigkeitserhaltung.

Übergang: Als Nächstes betrachten wir nicht mehr, welche FDs aus F folgen, sondern wie wir F selbst möglichst kompakt und redundanzfrei machen.

4. Kanonische Überdeckung

FD-Mengen kompakt, äquivalent und redundanzfrei darstellen

Leitfrage: Wie finden wir zu einer FD-Menge F eine möglichst kleine, aber äquivalente Darstellung?



Fokus dieses Abschnitts

Wir identifizieren Redundanzen in FD-Mengen, entfernen überflüssige Abhängigkeiten und Attribute und definieren daraus die kanonische Überdeckung als kompakte Standardform.

☰ Kapitelpfad

1. Motivation - Warum „gutes“ Datenbankdesign?
2. Funktionale Abhängigkeiten
3. Armstrong-Kalkül
4. **Kanonische Überdeckung**
5. Zerlegung von Relationen: Zerlegungskriterien
6. Normalformen und Normalisierungen

Rückblick & Problemstellung

Methodenwechsel: Wir wechseln von der Ableitung einzelner FDs zu Fragen über ganze FD-Mengen und später über Zerlegungen von Schemata.

Rückblick

- ▶ Wir können mit FDs rechnen (Armstrong-Axiome $F \vdash f$).
- ▶ Wir können prüfen, was aus α folgt (Attributhülle α^+).
- ▶ Wir können FD-Implikation und Superschlüssel prüfen.

Nächste Leitfragen

Ziel 1: Wie finden wir zu einer FD-Menge F eine möglichst kompakte, aber äquivalente Darstellung?

⇒ **Kanonische Überdeckung**

Ziel 2: Wenn wir Schemata später für die Normalisierung zerlegen: Woran erkennen wir eine „gute“ Zerlegung?

⇒ **Verlustlosigkeit & Abhängigkeitserhaltung**

Kanonische Überdeckung (F_c) - Motivation

Fokus auf Ziel 1: Gesucht ist eine FD-Menge mit gleicher Bedeutung wie F , aber in einer möglichst einfachen Form.

Warum reicht F nicht?

- ▶ F kann redundant sein.
- ▶ Es kann FDs enthalten, die bereits aus anderen FDs in F folgen.
- ▶ Es kann überflüssige Attribute auf linker oder rechter Seite geben.

Warum nicht F^+ ?

- ▶ F^+ enthält alle aus F folgenden FDs.
- ▶ Diese Hülle ist meist sehr groß.
- ▶ Damit ist sie für Praxis und weitere Algorithmen unhandlich.

Warum eine kompakte Form?

- ▶ **Effizienz:** Integritätsprüfungen sollen auf wenigen, einfachen FDs basieren.
- ▶ **Verständnis:** Eine minimale Menge ist leichter zu lesen.
- ▶ **Algorithmen:** Z.B. die 3NF-Synthese braucht eine kompakte Startmenge.

Ziel

Gesucht: eine Menge F_c , die

- ▶ äquivalent zu F ist: $F_c^+ = F^+$
- ▶ minimal ist: keine redundanten FDs, keine redundanten Attribute

Kanonische Überdeckung (F_c) - Definition

Sei F eine Menge von FDs. F_c heißt **Kanonische Überdeckung** von F , wenn die folgenden drei Bedingungen erfüllt sind.

1. Äquivalenz

F_c ist äquivalent zu F : $F_c \equiv F$, also $F_c^+ = F^+$.
Beide Mengen beschreiben also dieselben Constraints.

2. Minimale FDs

Jede FD $\alpha \rightarrow \beta$ in F_c ist minimal:

Links-minimal: Kein Attribut $A \in \alpha$ ist überflüssig, d.h.

$$F_c \not\equiv (F_c \setminus \{\alpha \rightarrow \beta\}) \cup \{\alpha \setminus \{A\} \rightarrow \beta\}$$

Rechts-minimal: Kein Attribut $B \in \beta$ ist überflüssig, d.h.

$$F_c \not\equiv (F_c \setminus \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow \beta \setminus \{B\}\}$$

3. Eindeutige linke Seiten

Jede Attributmenge erscheint höchstens einmal als linke Seite in F_c .

Falls $\alpha \rightarrow \beta_1 \in F_c$ und $\alpha \rightarrow \beta_2 \in F_c$, dann müssen sie zu $\alpha \rightarrow \beta_1 \cup \beta_2$ zusammengefasst sein.

Beispiel: Eine kanonische Überdeckung von $F = \{A \rightarrow C, B \rightarrow A, AB \rightarrow C\}$ ist $F_c = \{A \rightarrow C, B \rightarrow A\}$.

Nächste Frage:

Wie berechnet man F_c systematisch?

Kanonische Überdeckung (F_c) - Algorithmus

Algorithmus

Eingabe: FD-Menge F

Ausgabe: eine Kanonische Überdeckung F_c

$F_c \leftarrow F$

Schritt 1: Linksreduktion

for all $\alpha \rightarrow \beta \in F_c, A \in \alpha$ **do**

if $\beta \subseteq (\alpha \setminus \{A\})_{F_c}^+$:

ersetze $\alpha \rightarrow \beta$ durch $\alpha \setminus \{A\} \rightarrow \beta$

Schritt 2: Rechtsreduktion

for all $\alpha \rightarrow \beta \in F_c, B \in \beta$ **do**

$F'_c \leftarrow (F_c \setminus \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow \beta \setminus \{B\}\}$

if $B \in \alpha_{F'_c}^+$:

ersetze $\alpha \rightarrow \beta$ durch $\alpha \rightarrow \beta \setminus \{B\}$

Schritt 3: Zusammenfassen

mit Regel A_4 : $\alpha \rightarrow \beta_1, \alpha \rightarrow \beta_2 \Rightarrow \alpha \rightarrow \beta_1 \cup \beta_2$

Schritt 4: Triviale FDs entfernen

lösche alle FDs der Form $\alpha \rightarrow \emptyset$

return F_c

Merke ► Hüllen auf dem *aktuellen* F_c berechnen

► $\alpha_F^+ = \text{Attributhülle}(\alpha, F)$

► Bei Rechtsreduktion prüfen wir das Hilfs- F'_c

► F_c ist nicht eindeutig; die Reihenfolge beeinflusst das Ergebnis

Kanonische Überdeckung - Beispiel (Linksreduktion)

Gegeben (Schritt 0): $F_c = \{A \rightarrow C, B \rightarrow A, AB \rightarrow C\}$

Schritt 1: Linksreduktion (Ist ein Attribut auf der linken Seite entbehrlich?)

FD $\alpha \rightarrow \beta$	Prüfe	Hülle $\alpha \setminus \{A\}_{F_c}^+$	Ergebnis & Aktion
$A \rightarrow C$	–	$ \alpha = 1 \Rightarrow \text{minimal}$	Nichts zu tun
$B \rightarrow A$	–	$ \alpha = 1 \Rightarrow \text{minimal}$	Nichts zu tun
$AB \rightarrow C$	A	$\{B\}_{F_c}^+ = \{A, B, C\}$ (da $B \rightarrow A \rightarrow C$)	$C \in \text{Hülle} \Rightarrow A$ ist überflüssig! $\Rightarrow$ Ersetze durch $B \rightarrow C$
Aktualisiertes $F_c = \{A \rightarrow C, B \rightarrow A, B \rightarrow C\}$			
$B \rightarrow C$	–	$ \alpha = 1 \Rightarrow \text{minimal}$	Nichts zu tun

Kanonische Überdeckung - Beispiel (Rechtsreduktion)

Aktueller Stand: $F_c = \{A \rightarrow C, B \rightarrow A, B \rightarrow C\}$

Schritt 2: Rechtsreduktion (*Ist ein Attribut auf der rechten Seite entbehrlich?*)

FD	Prüfe	Hilfsmenge F'_c	Hülle $\alpha_{F'_c}^+$	Ergebnis & Aktion
$A \rightarrow C$	C	$\{B \rightarrow A, B \rightarrow C, A \rightarrow \emptyset\}$	$A_{F'_c}^+ = \{A\}$	$C \notin \text{Hülle} \Rightarrow \text{bleibt}$
$B \rightarrow A$	A	$\{A \rightarrow C, B \rightarrow C, B \rightarrow \emptyset\}$	$B_{F'_c}^+ = \{B, C\}$	$A \notin \text{Hülle} \Rightarrow \text{bleibt}$
$B \rightarrow C$	C	$\{A \rightarrow C, B \rightarrow A, B \rightarrow \emptyset\}$	$B_{F'_c}^+ = \{A, B, C\}$	$C \in \text{Hülle} \Rightarrow C$ überflüssig! $\Rightarrow$ Ersetze durch $B \rightarrow \emptyset$

Zwischenergebnis: $F_c = \{A \rightarrow C, B \rightarrow A, B \rightarrow \emptyset\}$

Kanonische Überdeckung - Beispiel (Abschluss)

Aktueller Stand: $F_c = \{A \rightarrow C, B \rightarrow A, B \rightarrow \emptyset\}$

Schritt 3: FDs zusammenfassen

Fasse alle FDs mit **gleicher linker Seite** zusammen:

$A \rightarrow C \quad \Rightarrow \quad \text{bleibt (keine weiteren FDs mit LHS } A)$
 $B \rightarrow A \text{ und } B \rightarrow \emptyset \quad \Rightarrow \quad B \rightarrow A \cup \emptyset \quad \Rightarrow \quad B \rightarrow A$

Schritt 4: Triviale FDs entfernen

Gibt es verbleibende FDs mit **leerer rechter Seite** ($\alpha \rightarrow \emptyset$)?

$\Rightarrow$ Nein. (Wurden im vorherigen Schritt implizit mit vereinigt.)

Endergebnis: Die Kanonische Überdeckung lautet $F_c = \{A \rightarrow C, B \rightarrow A\}$

5. Zerlegung von Relationen

Verlustlosigkeit und Abhängigkeitserhaltung
als Gütekriterien

Leitfrage: Wann ist eine Zerlegung eines Schemas fachlich sinnvoll und welche Eigenschaften muss sie dabei bewahren?



Fokus dieses Abschnitts

Wir zerlegen Relationen in kleinere Schemata und untersuchen, wann dabei keine Information verloren geht und funktionale Abhängigkeiten weiterhin geeignet erhalten bleiben.

☰ Kapitelpfad

1. Motivation - Warum „gutes“ Datenbankdesign?
2. Funktionale Abhängigkeiten
3. Armstrong-Kalkül
4. Kanonische Überdeckung
5. **Zerlegung von Relationen:
Zerlegungskriterien**
6. Normalformen und Normalisierungen

Schema-Zerlegung: Warum und Wie?

Leitidee: Wir wollen kleinere, bessere Schemata erhalten, ohne Informationen oder wichtige Constraints zu verlieren.

Warum Zerlegung?

- ▶ Wir haben gesehen: „schlechte“ Schemata führen zu Redundanz und Anomalien.
- ▶ Wir können Constraints formal beschreiben und analysieren: Funktionale Abhängigkeiten, Hüllen und kanonische Überdeckungen.
- ▶ Ziel: „schlechte“ Schemata systematisch verbessern.
- ▶ Methode: Zerlegung (Dekomposition) eines Schemas $\mathcal{R}$ in mehrere kleinere Schemata $\mathcal{R}_1, \dots, \mathcal{R}_n$.

Worauf kommt es an?

Eine Zerlegung ist nicht automatisch „gut“. Was müssen wir sicherstellen?

1. Verlustlosigkeit: Lassen sich die Originaldaten durch Join exakt rekonstruieren? Kein Informationsverlust, keine zusätzlichen falschen Tupel?

2. Abhängigkeitserhaltung: Können alle ursprünglichen FDs weiterhin effizient, also lokal in den zerlegten Tabellen, überprüft werden?

Gütekriterium 1: Verlustlosigkeit - Definition

Setup

Zerlegung von Schema $\mathcal{R}$ mit FDs F in $\mathcal{R}_1, \mathcal{R}_2$.
Seien X, X_1 und X_2 die entsprechenden Attributmengen. Die Zerlegung von $\mathcal{R}$ heißt *gültig*, falls gilt $X = X_1 \cup X_2$.

Projektion

Für eine Instanz R von $\mathcal{R}$ seien $R_1 := \pi_{X_1}(R)$ und $R_2 := \pi_{X_2}(R)$ die Projektionen auf die Teilattribute.

Definition Verlustlosigkeit

Die Zerlegung heißt *verlustlos* (lossless join decomposition), wenn für **jede** gültige Instanz R von $\mathcal{R}$ (d.h. $R \models F$) gilt:

$$R = R_1 \bowtie R_2 \quad \left(\text{bzw. } R = \pi_{X_1}(R) \bowtie \pi_{X_2}(R) \right)$$

Bedeutung: Der Natural Join der Projektionen muss exakt die ursprüngliche Relation ergeben. Es dürfen keine Tupel fehlen und es dürfen **keine falschen Tupel** hinzukommen.

Verlustlosigkeit: Beispiel (Nicht verlustlos)

Ausgangsschema: *Biertrinker*(Kneipe, Gast, Bier) mit FD $\{ Kneipe, Gast \rightarrow Bier \}$.

$R_1 = \text{Besucht}$

Kneipe	Gast
Domkeller	Kemper
Domkeller	Eickler
Die Kiste	Eickler

Biertrinker

Kneipe	Gast	Bier
Domkeller	Kemper	Pils
Domkeller	Eickler	Hefeweizen
Die Kiste	Eickler	Pils

$R_2 = \text{Trinkt}$

Gast	Bier
Kemper	Pils
Eickler	Hefeweizen
Eickler	Pils

Join-Ergebnis: $R_1 \bowtie R_2$

Kneipe	Gast	Bier
Domkeller	Kemper	Pils
Domkeller	Eickler	Hefeweizen
<i>Domkeller</i>	<i>Eickler</i>	<i>Pils</i>
<i>Die Kiste</i>	<i>Eickler</i>	<i>Hefeweizen</i>
Die Kiste	Eickler	Pils

Problem: Der Join erzeugt zusätzliche, **falsche Tupel**.

Beispiel: (Domkeller, Eickler, Pils).
Daher ist die Zerlegung **nicht** verlustlos.

Verlustlosigkeit: Kriterium für binäre Zerlegung

Theorem: Eine Zerlegung von $\mathcal{R}$ mit FDs F in $\mathcal{R}_1$ und $\mathcal{R}_2$ mit Attributmengen $X = X_1 \cup X_2$ ist **verlustlos**, wenn *mindestens eine* der beiden Bedingungen gilt.

Bedingung 1

Die gemeinsamen Attribute bestimmen alle Attribute von $\mathcal{R}_1$:

$$(X_1 \cap X_2) \rightarrow X_1 \in F^+$$

Bedingung 2

Die gemeinsamen Attribute bestimmen alle Attribute von $\mathcal{R}_2$:

$$(X_1 \cap X_2) \rightarrow X_2 \in F^+$$

In anderen Worten

$$\begin{aligned} X_1 &\subseteq X_1 \cap X_2^+ \\ &\text{oder} \\ X_2 &\subseteq X_1 \cap X_2^+ \end{aligned}$$

Intuition: Das Joinattribut $(X_1 \cap X_2)$ muss ein **Super-schlüssel (Superkey)** für mindestens eine der beiden Teilrelationen $\mathcal{R}_1$ oder $\mathcal{R}_2$ sein.

Beispielbezug: Im Biertrinker-Beispiel ist $X_1 \cap X_2 = \{\text{Gast}\}$ nicht stark genug, um als Schlüssel zu dienen.

Verlustlosigkeit: Beispiel (Verlustlos)

Ausgangsschema: *Eltern*(Vater, Mutter, Kind) mit FDs { $Kind \rightarrow Vater$, $Kind \rightarrow Mutter$ }.

$R_1 = \text{Väter}$

Vater	Kind
Johann	Else
Johann	Theo
Heinz	Cleo

Eltern

Vater	Mutter	Kind
Johann	Martha	Else
Johann	Maria	Theo
Heinz	Martha	Cleo

$R_2 = \text{Mütter}$

Mutter	Kind
Martha	Else
Maria	Theo
Martha	Cleo

Join-Ergebnis: $R_1 \bowtie R_2$

Vater	Mutter	Kind
Johann	Martha	Else
Johann	Maria	Theo
Heinz	Martha	Cleo

Prüfung: $X_1 \cap X_2 = \{Kind\}$.

Kriterium: $Kind \rightarrow X_1$ und $Kind \rightarrow X_2$.

Fazit: Das Joinattribut ist ein Superschlüssel; der Join rekonstruiert die Ausgangsrelation exakt.

Gütekriterium 2: Abhängigkeitserhaltung

Motivation

Ziel: Effiziente Konsistenzprüfung nach Zerlegung.

Anforderung: Alle ursprünglichen FDs F sollen weiterhin prüfbar sein, idealerweise **lokal** auf jeder neuen Relation R_i .

Lokale FD-Mengen

Für jede Teilrelation mit Attributmenge X_i betrachten wir die dort formulierbaren FDs:

$$F_i = \{\alpha \rightarrow \beta \in F^+ \mid \alpha \cup \beta \subseteq X_i\}$$

Definition Abhängigkeitserhaltung

Eine Zerlegung ist **abhängigkeitserhaltend**, wenn alle FDs aus F durch die *lokalen* FDs zusammen repräsentiert werden.

Auch Hüllentreue: Die Hülle der lokalen FD-Mengen ist gleich der Hülle der ursprünglichen FD-Menge.

$$\left(\bigcup_{i=1}^n F_i \right)^+ = F^+$$

Verlustlos, aber nicht abhängigkeiterhaltend

Ausgangsschema: PLZVerzeichnis(BLand, Ort, Straße, PLZ)
FDs: $F = \{ \text{PLZ} \rightarrow \text{Ort}, \text{BLand}, \text{BLand}, \text{Ort}, \text{Straße} \rightarrow \text{PLZ} \}$.

Annahmen

- ▶ Ortsnamen sind innerhalb eines Bundeslandes eindeutig.
- ▶ PLZ ändern sich nicht innerhalb einer Straße.

Zerlegung

$$R_1 = \text{Straßen}(\text{PLZ}, \text{Straße})$$
$$R_2 = \text{Orte}(\text{PLZ}, \text{Ort}, \text{BLand})$$

Verlustlos: Das Joinattribut ist $\{\text{PLZ}\}$, und es gilt $\text{PLZ} \rightarrow \text{Ort}, \text{BLand}$.
Damit bestimmt $\{\text{PLZ}\}$ alle Attribute von R_2 .

Nicht abhängigkeiterhaltend: Die kritische FD ist $\text{BLand}, \text{Ort}, \text{Straße} \rightarrow \text{PLZ}$.
 Straße liegt in R_1 , aber $\{\text{BLand}, \text{Ort}\}$ liegen in R_2 . Die FD ist damit auf die Basisrelationen **zerrissen** und lokal nicht mehr prüfbar.

Die versteckte Gefahr (Fehlende Abhängigkeitserhaltung)

Zerlegung: $R_1 = \text{Straßen}(\text{PLZ}, \text{Straße})$ und $R_2 = \text{Orte}(\text{PLZ}, \text{Ort}, \text{BLand})$

Straßen (R_1)

PLZ	Straße
60313	Göthestraße
60437	Galgenstraße
15234	Göthestraße

Orte (R_2)

Ort	BLand	PLZ
Frankfurt	Hessen	60313
Frankfurt	Hessen	60437
Frankfurt	Brandenburg	15234

Join-Ergebnis: Straßen $\bowtie$ Orte

Ort	BLand	Straße	PLZ
Frankfurt	Hessen	Göthestraße	60313
Frankfurt	Hessen	Galgenstraße	60437
Frankfurt	Brandenburg	Göthestraße	15234

Beobachtung: Lokal sehen R_1 und R_2 unauffällig aus. Auch der Join rekonstruiert hier noch genau die ursprünglichen Tupel.

Die versteckte Gefahr (Fehlende Abhängigkeitserhaltung)

Zerlegung: $R_1 = \text{Straßen}(\text{PLZ}, \text{Straße})$ und $R_2 = \text{Orte}(\text{PLZ}, \text{Ort}, \text{BLand})$

Straßen (R_1)

PLZ	Straße
60313	Göthestraße
60437	Galgenstraße
15234	Göthestraße
<i>15235</i>	<i>Göthestraße</i>

Orte (R_2)

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

Join-Ergebnis: Straßen $\bowtie$ Orte

Ort	BLand	Straße	PLZ
Frankfurt	Hessen	Göthestraße	60313
Frankfurt	Hessen	Galgenstraße	60437
Frankfurt	Brandenburg	Göthestraße	15234
<i>Frankfurt</i>	<i>Brandenburg</i>	<i>Göthestraße</i>	<i>15235</i>

Problem beim Join: Beide lokalen Tabellen wirken weiterhin konsistent. Erst der Join zeigt die Verletzung von $\text{BLand}, \text{Ort}, \text{Straße} \rightarrow \text{PLZ}$.

Beide Kriterien erfüllt: Ein Positiv-Beispiel

Ausgangsschema: *Betreuung*(MatrNr, Projekt, Betreuer)

FDs: $F = \{ \text{MatrNr} \rightarrow \text{Projekt}, \text{Projekt} \rightarrow \text{Betreuer} \}$.

Zerlegung

$R_1(\text{MatrNr}, \text{Projekt})$

$R_2(\text{Projekt}, \text{Betreuer})$

Lokale FD-Mengen

$F_1 = \{ \text{MatrNr} \rightarrow \text{Projekt} \}$

$F_2 = \{ \text{Projekt} \rightarrow \text{Betreuer} \}$

1. Verlustlos?

Schnittmenge: $X_1 \cap X_2 = \{ \text{Projekt} \}$.

Kriterium: Da $\text{Projekt} \rightarrow \text{Betreuer} \in F$, ist das Joinattribut ein Superschlüssel für R_2 .

Ergebnis: Die Zerlegung ist verlustlos.

2. Abhängigkeitserhaltend?

Vereinigung: $F_1 \cup F_2 = \{ \text{MatrNr} \rightarrow \text{Projekt}, \text{Projekt} \rightarrow \text{Betreuer} \}$.

Beobachtung: Dies entspricht exakt der ursprünglichen Menge F .

Ergebnis: Alle FDs bleiben lokal prüfbar.

Zusammenfassung: Gütekriterien für Zerlegungen

Merke: Jede Zerlegung soll **verlustlos** sein; **Abhängigkeitserhaltung** ist für lokale Konsistenzprüfungen besonders wünschenswert.

Verlustlosigkeit

Sinn & Zweck: Sichert Datenintegrität. Verhindert Informationsverlust und das Entstehen von **falschen Tupeln**.

Stellenwert: **Zwingend erforderlich!** Jedes Design muss verlustlos sein.

Theorem-Test: Das Join-Attribut muss in mindestens einer Teilrelation Superschlüssel sein.

Abhängigkeitserhaltung

Sinn & Zweck: Sichert Performanz. Erlaubt die Überprüfung von Konsistenzbedingungen (FDs), ohne teure Joins.

Stellenwert: **Hochgradig wünschenswert!**
Sie ist aber nicht immer theoretisch erreichbar.

Ausblick: Normalformen Nun suchen wir Algorithmen zur Bereinigung von Schemata:

3NF-Synthese: garantiert beide Gütekriterien.

BCNF-Dekomposition: garantiert Verlustlosigkeit, opfert aber in seltenen Fällen die Abhängigkeitserhaltung.

6. Normalformen und Normalisierungen

Normalformen als Maß für Schemaqualität

Leitfrage: Wie bewerten wir relationale Schemata formal und verbessern sie durch Zerlegung systematisch?



Fokus dieses Abschnitts

Wir beschreiben Qualitätsstufen für relationale Schemata und nutzen sie als Leitfaden, um Redundanzen und Anomalien durch Zerlegung zu reduzieren.



Kapitelpfad

1. Motivation – Warum „gutes“ Datenbankdesign?
2. Funktionale Abhängigkeiten
3. Armstrong-Kalkül
4. Kanonische Überdeckung
5. Zerlegung von Relationen: Zerlegungskriterien
6. **Normalformen und Normalisierungen**

Rückblick: Anomalien und ihre Ursachen

Was wir gesehen haben

- ▶ **Einfügeanomalie:** Neuer Prof ohne Vorlesung → **nicht speicherbar**
- ▶ **Änderungsanomalie:** Raumwechsel → **mehrfaches Update** in Vorlesungen
- ▶ **Löschanomalie:** Letzte Vorlesung löschen → **Prof-Daten gehen verloren**

Die gemeinsame Ursache

Die problematische FD

PersNr → Name, Rang, Raum

- ▶ **PersNr ist kein Superschlüssel** (nur Teil von {PersNr, VorlNr}).
- ▶ **Folge:** Prof-Daten werden pro Vorlesung redundant wiederholt.
- ▶ **Fazit:** Diese Konstellation ist der Auslöser für alle drei Anomalien.

Leitidee der Normalformen: Zusammengehörende Daten gehören in genau eine Tabelle. Normalformen unterscheiden sich darin, wie dieses „Zusammengehören“ über funktionale Abhängigkeiten formalisiert wird.

2NF (partielle FDs) → **3NF** (transitive FDs) → **BCNF** (alle FDs)

Kernziel der Normalisierung

Systematische Zerlegung schlechter Schemata zur **dauerhaften Eliminierung** von Redundanz und Anomalien.

1. Formale Qualitätsstufen

- ▶ Normalformen dienen als **messbare Level** der Schemaqualität.
- ▶ Sie werden strikt durch **erlaubte FDs** definiert.

2. Sichere Dekomposition

- ▶ Höhere Normalformen erreichen, aber...
- ▶ **Verlustlosigkeit** zwingend erhalten!
- ▶ **Abhängigkeitserhaltung** möglichst wahren.

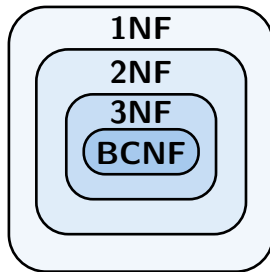
Hierarchie der Normalformen: $BCNF \subset 3NF \subset 2NF \subset 1NF$

Mengenbeziehung: Jede BCNF-Relation ist auch in 3NF, 2NF und 1NF.

Die vier Stufen

- ▶ **1NF:** Erste Normalform
- ▶ **2NF:** Zweite Normalform
- ▶ **3NF:** Dritte Normalform
- ▶ **BCNF:**
Boyce-Codd-Normalform

Verschachtelung der Klassen



Lesart: Von 1NF nach BCNF werden die Bedingungen an zulässige funktionale Abhängigkeiten strenger; dadurch werden Redundanzen und Anomalien systematischer ausgeschlossen.

Erste Normalform (1NF): Die Basis

Definition und Bedeutung

Definition: Eine Relation ist in 1NF, wenn die Domänen aller Attribute **atomare Werte** enthalten.

Bedeutung: Keine zusammengesetzten Attribute (z. B. Adresse als ein String), keine mengenwertigen Attribute (z. B. {Skill1, Skill2}), keine Tabellen als Attributwerte.

Nicht in 1NF: Eltern-1

Vater	Mutter	Kind
Johann	Martha	{Else, Lucia}
Johann	Maria	{Theo, Jose}
Heinz	Martha	{Cleo}

In 1NF: Eltern-2

Vater	Mutter	Kind
Johann	Martha	Else
Johann	Martha	Lucia
Johann	Maria	Theo
Johann	Maria	Jose
Heinz	Martha	Cleo

Hinweis: NF²-Modelle („non-first-normal-form Modelle“) erlauben solche nicht-atomaren Attributwerte explizit. **Status:** Das klassische relationale Modell setzt 1NF voraus; im Folgenden gehen wir immer von Relationen in 1NF aus.

Zweite Normalform (2NF): Motivation

Problem: Anomalien können auftreten, wenn Nicht-Schlüsselattribute (NSA) nur von einem *Teil* eines zusammengesetzten Schlüsselkandidaten abhängen (**partielle Abhängigkeit**).

Gegeben

Beispielschema:

ProfVorl(PersNr, Name, ...,
VorlNr, Titel, ...)

Schlüsselkandidat:

$k_1 = \{\text{PersNr}, \text{VorlNr}\}$

Nicht-Schlüsselattribute:

Name, Rang, Raum, Titel, SWS

Partielle FDs:

PersNr $\rightarrow$ Name, Rang, Raum

VorlNr $\rightarrow$ Titel, SWS

Beispielinstanz

PersNr	Name	Rang	Raum	VorlNr	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
...	...	...	...	...	...	...

Folge: Professoren- und Vorlesungsdaten werden redundant wiederholt, weil beide FDs NSA schon über echte Teile von k_1 festlegen. **2NF** will genau solche partiellen Abhängigkeiten vermeiden.

Zweite Normalform (2NF): Definition

Begriff: Nicht-Schlüsselattribut (NSA)

Ein Attribut, das zu **keinem** Kandidatenschlüssel gehört.

Definition der 2NF: Ein Relationsschema R ist in 2NF, wenn es in 1NF ist und **jedes Nicht-Schlüsselattribut** von **jedem Kandidatenschlüssel voll funktional abhängig** ist.

Bedeutung

Zusammengehörende Daten in 2NF: Alle Attribute, die durch den **ganzen** Kandidatenschlüssel bestimmt werden.

Es darf keine partielle Abhängigkeit eines NSA von einer echten Teilmenge eines Kandidatenschlüssels geben.

Verletzung im Beispiel

Name ist ein NSA und es gilt $\text{PersNr} \rightarrow \text{Name}$. *PersNr* ist aber nur eine echte Teilmenge von $\{\text{PersNr}, \text{VorlNr}\}$.

Also ist *ProfVorl* **nicht** in 2NF.

2NF erreichen durch Zerlegung

Ziel: Gegeben sei ein Schema R mit FD-Menge F . Gesucht ist eine Zerlegung, die alle **partiellen Abhängigkeiten** entfernt, also Abhängigkeiten eines NSA von einer echten Teilmenge eines Kandidatenschlüssels.

Verfahren (vereinfacht)

1. Partielle Abhängigkeiten identifizieren

Finde alle FDs $\alpha \rightarrow B \in F^+$ mit $\alpha \subset \kappa$, wobei κ ein Kandidatenschlüssel und B ein Nicht-Schlüsselattribut ist.

2. Auslagern

Erzeuge für jede solche linke Seite α eine neue Relation $R'(\alpha \cup \{B \mid \alpha \rightarrow B\})$; Primärschlüssel von R' ist α .

3. Bereinigen

Entferne die ausgelagerten Nicht-Schlüsselattribute aus der Ursprungstabelle. Die Restrelation behält den vollen Schlüssel und nur noch die davon **voll** abhängigen Attribute.

Hinweis: Der 2NF-Prozess ist eigentlich iterativ; hier vereinfacht dargestellt.

2NF – Zerlegung – Beispiel: Schritt 1

Ausgangsschema:

ProfVorl(PersNr, Name, Rang, Raum, VorINr, Titel, SWS)

Kandidatenschlüssel: {PersNr, VorINr}

Von *PersNr* bestimmt

Professorendaten:

- ▶ PersNr → Name
- ▶ PersNr → Rang
- ▶ PersNr → Raum

Von *VorINr* bestimmt

Vorlesungsdaten:

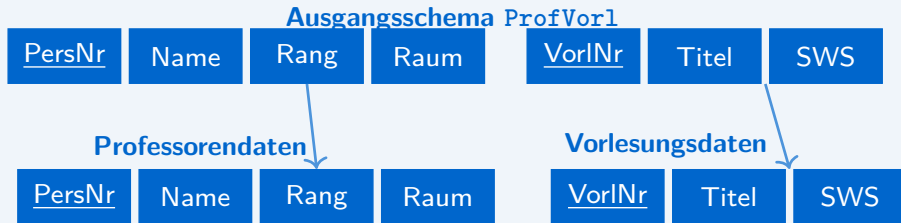
- ▶ VorINr → Titel
- ▶ VorINr → SWS

Beobachtung: Alle gezeigten FDs gehen von *echten Teilmengen* des zusammengesetzten Schlüssels aus. Damit sind genau diese beiden Attributgruppen die Kandidaten für die Auslagerung in Schritt 2.

2NF – Zerlegung – Beispiel: Schritt 2

Ausgangspunkt: PersNr $\rightarrow$ Name, Rang, Raum und VorlNr $\rightarrow$ Titel, SWS

Auslagern

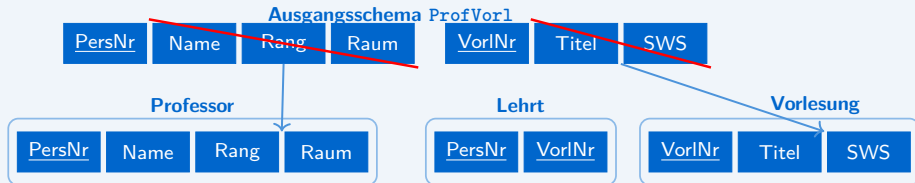


Beobachtung: In Schritt 2 entstehen zwei neue Relationen. Die Ursprungsrelation wird erst in Schritt 3 auf die Restrelation mit {PersNr, VorlNr} bereinigt.

2NF – Zerlegung – Beispiel: Schritt 3

Ausgangspunkt: Die in Schritt 2 ausgelagerten Attribute werden jetzt aus der Ursprungsrelation entfernt. Übrig bleibt nur noch die Zuordnung zwischen Professor und Vorlesung.

Bereinigen



Ergebnis: Nach dem Bereinigen bleiben die drei Relationen Professor, Lehrt und Vorlesung. In Lehrt stehen nur noch PersNr und VorlNr.

2NF – Zerlegung – Beispiel: Resultierende Tabellen

Wirkung der Zerlegung: Aus der Ausgangsrelation ProfVor1 entstehen drei Relationen, in denen Professorendaten und Vorlesungsdaten nicht mehr in jeder Zuordnungszeile wiederholt werden.

Vorher: ProfVor1

<u>PersNr</u>	Name	Rang	Raum	<u>VorlNr</u>	Titel	SWS
2125	Sokrates	W3	226	5041	Ethik	4
2125	Sokrates	W3	226	5049	Mäeutik	2
2125	Sokrates	W3	226	4052	Logik	4
...	...	...	...	...	...	...

Beobachtung: Name, Rang, Raum sowie Titel und SWS werden in der Ausgangsrelation für jede Lehrzuordnung erneut gespeichert.



Nachher: getrennte Relationen

Professor				Lehrt	
<u>PersNr</u>	Name	Rang	Raum	<u>PersNr</u>	<u>VorlNr</u>
2125	Sokrates	W3	226	2125	5041
...	...	...	...	2125	5049
				2125	4052
				...	...

Vorlesung		
<u>VorlNr</u>	Titel	SWS
5041	Ethik	4
5049	Mäeutik	2
4052	Logik	4
...	...	...

Interpretation: Die Redundanz aus ProfVor1 verschwindet, weil Professorendaten, Lehrzuordnungen und Vorlesungsdaten getrennt gespeichert werden.

Dritte Normalform (3NF) - Motivation

Warum reicht 2NF nicht aus? Auch in 2NF können noch Redundanzen und Anomalien entstehen, wenn Nichtschlüsselattribute von **anderen** Nichtschlüsselattributen abhängen. Man spricht dann von **transitiver Abhängigkeit**.

Beispiel: ProfessorenAdressen

Erweiterung des Universitäts-Beispiels um Adressdaten.

Schema:

PersNr, Name, Rang, Raum, Ort, Straße, Hausnr, PLZ, Vorwahl, BLand

Beobachtung: Das Schema ist bereits in 2NF, aber zum Beispiel gilt $PLZ \rightarrow Ort$. Damit hängt ein NSA (*Ort*) von einem anderen NSA (*PLZ*) ab.

Instanz-Skizze

PersNr	Name	...	Ort	...	PLZ	...
2125	Sokrates	...	Aachen	...	52052	...
2132	Popper	...	Aachen	...	52052	...
...	...	...	...	...	...	...

Transitiver Kern: $PLZ \rightarrow Ort$. Die Zuordnung *PLZ-Ort* wird deshalb für jede Person mit derselben PLZ erneut gespeichert. **Änderungsanomalie möglich.**

Dritte Normalform (3NF): Definition

Leitidee: 3NF verallgemeinert die 2NF: Ein Nicht-Schlüsselattribut darf nicht transitiv über andere Nicht-Schlüsselattribute bestimmt werden.

Zusammengehörende Daten in 3NF: Ein Schlüssel und alle Attribute, die direkt von ihm abhängen.

Formale Bedingung

Ein Relationsschema R mit FD-Menge F ist in **3NF**, wenn für jede **nicht-triviale** FD $\alpha \rightarrow B \in F^+$ gilt:

Ist B ein Nicht-Schlüsselattribut (NSA), dann muss α ein Superschlüssel von R sein.

Ein Superschlüssel muss dabei nicht notwendig minimal sein.

Verletzung im Beispiel

In *ProfessorenAdressen* sind etwa

Ort, BLand $\rightarrow$ Vorwahl

PLZ $\rightarrow$ Ort, BLand

Warum? Rechts stehen NSAs, links aber keine Superschlüssel.

Also ist das Schema nicht in 3NF.

Merksatz: Bei jeder nicht-trivialen FD zu einem NSA muss links ein Superschlüssel stehen.

3NF – Beispiel *ProfessorenAdressen*

Ausgangspunkt: *ProfessorenAdressen*(PersNr, Name, . . . , BLand)

Prüfidee: Bei jeder FD zu einem NSA muss links ein Superschlüssel stehen.

Unkritische FDs

Superschlüssel links:

PersNr $\rightarrow X$

Raum $\rightarrow$ PersNr

Raum $\rightarrow X$

Folge: keine Verletzung von 3NF.

Problematische FDs

NSA rechts, aber kein Superschlüssel links:

Ort, BLand $\rightarrow$ Vorwahl

Ort, Straße, Hausnr, BLand $\rightarrow$ PLZ

PLZ $\rightarrow$ Ort, BLand

Folge: Das Schema ist so nicht in 3NF.

Nächster Schritt: Statt jede problematische FD einzeln zu reparieren, verwenden wir den **3NF-Synthesealgorithmus**.

3NF – Der Synthesealgorithmus

Ziel: Zerlege ein Schema R mit Attributmenge X und FD-Menge F in Relationen $R_1, \dots, R_n$, so dass alle R_i in 3NF sind und die Zerlegung **verlustlos** und **abhängigkeitserhaltend** bleibt.

Algorithmus

Eingabe: Schema R mit Attributmenge X und FD-Menge F

Ausgabe: 3NF-Zerlegung von R

1. **Kanonische Überdeckung:** Berechne F_c von F .
2. **Relationsschemata:** Für jede FD $\alpha \rightarrow \beta \in F_c$ erzeuge $R_{\alpha\beta}$ mit Attributmenge $\alpha \cup \beta$ und ordne $F_{\alpha\beta} = \{\alpha' \rightarrow \beta' \in F_c \mid \alpha', \beta' \subseteq \alpha \cup \beta\}$ zu.
3. **Schlüsselschema:** Enthält noch keines der erzeugten Schemata einen Kandidatenschlüssel von R , füge R_κ mit $F_\kappa = \{\kappa \rightarrow \kappa\}$ hinzu (κ Kandidatenschlüssel von R).
4. **Bereinigung:** Entferne jedes Schema R_i , dessen Attribute vollständig in einem anderen Schema R_j enthalten sind.

Garantie: Die resultierende Zerlegung ist in 3NF, verlustlos und abhängigkeitserhaltend.

3NF-Synthese – Beispiel *ProfessorenAdressen*

Beispiel zu Schritt 1: Aus der ursprünglichen FD-Menge F bestimmen wir die kanonische Überdeckung F_c .

Ausgangslage: Schema und FD-Menge F

Schema:

ProfessorenAdressen(PersNr, Name, Rang, Raum, Ort, Straße,
Hausnr, PLZ, Vorwahl, BLand)

FD-Menge F :

$f_1 = \text{PersNr} \rightarrow \text{Name, Rang, Raum, Ort, Straße, Hausnr, PLZ, Vorwahl, BLand}$
 $f_2 = \text{Raum} \rightarrow \text{PersNr}$
 $f_3 = \text{Ort, Straße, Hausnr, BLand} \rightarrow \text{PLZ}$
 $f_4 = \text{Ort, BLand} \rightarrow \text{Vorwahl}$
 $f_5 = \text{PLZ} \rightarrow \text{Ort, BLand}$

Ergebnis von Schritt 1: F_c

$f_1 = \text{PersNr} \rightarrow \text{Raum, Name, Rang, Ort, Straße, Hausnr, BLand}$
 $f_2 = \text{Raum} \rightarrow \text{PersNr}$
 $f_3 = \text{Ort, Straße, Hausnr, BLand} \rightarrow \text{PLZ}$
 $f_4 = \text{Ort, BLand} \rightarrow \text{Vorwahl}$
 $f_5 = \text{PLZ} \rightarrow \text{Ort, BLand}$

Änderung in f_1 : *PLZ* und *Vorwahl* entfallen, weil sie aus den anderen FDs transitiv folgen.

Nächster Schritt: Aus jeder FD in F_c wird nun ein Relationsschema erzeugt.

3NF-Synthese – Beispiel *ProfessorenAdressen* (Fortsetzung)

Schritt 2: Schemata aus F_c

$R_1 = \text{Professor}(\text{PersNr}, \text{Name}, \text{Rang}, \text{Raum}, \text{Ort}, \text{Straße}, \text{Hausnr}, \text{BLand})$

$R_2 = (\text{Raum}, \text{PersNr})$

$R_3 = \text{PLZVerzeichnis}(\text{Ort}, \text{Straße}, \text{Hausnr}, \text{BLand}, \text{PLZ})$

$R_4 = \text{Vorwahlverzeichnis}(\text{Ort}, \text{BLand}, \text{Vorwahl})$

$R_5 = (\text{PLZ}, \text{Ort}, \text{BLand})$

$F_1 = \{f_1, f_2\}$

$F_2 = \{f_2\}$

$F_3 = \{f_3, f_5\}$

$F_4 = \{f_4\}$

$F_5 = \{f_5\}$

Schritt 3 und 4: Schlüssel-Check und Bereinigung

R_1 enthält den Kandidatenschlüssel $\{\text{PersNr}\}$, R_2 den Kandidatenschlüssel $\{\text{Raum}\}$.

Schlüssel-Check: Kein zusätzliches Schlüsselschema ist nötig.

Bereinigung: Da $R_2 \subseteq R_1$ und $R_5 \subseteq R_3$ gelten, entfallen R_2 und R_5 .

Ergebnis der 3NF-Synthese:

Professor(PersNr, Name, Rang, Raum, Ort, Straße, Hausnr, BLand)

PLZVerzeichnis(Ort, Straße, Hausnr, BLand, PLZ)

Vorwahlverzeichnis(Ort, BLand, Vorwahl)

Ist nun das Optimum erreicht? Nein $\rightarrow$ BCNF

Boyce-Codd Normalform (BCNF) – Definition

Zusammengehörende Daten in BCNF: Ein Kandidatenschlüssel und genau die Attribute, die durch diesen Schlüssel bestimmt werden, ohne zusätzliche funktionale Abhängigkeiten von Nicht-Schlüsseln.

Formale Definition

Ein Relationsschema R mit Attributmenge X und FDs F ist in *Boyce-Codd Normalform (BCNF)*, wenn für alle nicht-trivialen FDs $\alpha \rightarrow \beta \in F^+$ gilt: α ist ein Superschlüssel von R .

$\forall$ nicht-triviale $\alpha \rightarrow \beta \in F^+ : \quad \alpha$ ist Superschlüssel von R

Kernaussage

Nicht-triviale FDs sind nur dann erlaubt, wenn ihre Determinante ein Superschlüssel ist.

Vergleich zu 3NF

BCNF ist strikter als 3NF. Daher gilt:

$BCNF \Rightarrow 3NF$ aber nicht umgekehrt.

BCNF – Dekompositionsalgorithmus und Trade-off

Ziel: Zerlege ein Schema $R(X)$ mit FDs F in Teilrelationen $R_1, \dots, R_n$ so, dass alle R_i in BCNF sind und die Zerlegung verlustlos bleibt.

Algorithmus (iterativ)

Wiederhole, solange ein Teilschema $R_i(X_i)$ nicht in BCNF ist:

1. Finde eine verletzende FD $\alpha \rightarrow \beta \in F_i^+$, d. h. α ist kein Superschlüssel von R_i , und $\alpha \cap \beta = \emptyset$.
2. Zerlege R_i in R_{i1} und R_{i2} mit $X_{i1} = \alpha \cup \beta$ und $X_{i2} = X_i \setminus \beta$.
3. Projiziere die FDs neu: $F_{i1} = \Pi_{X_{i1}}(F_i^+)$ und $F_{i2} = \Pi_{X_{i2}}(F_i^+)$.
4. Ersetze R_i durch R_{i1} und R_{i2} .

$$\Pi_X(F) := \{ \gamma \rightarrow \delta \in F \mid (\gamma \cup \delta) \subseteq X \}$$

Warum verlustlos?

Gemeinsame Attribute sind genau α . In $R_{i1}(\alpha \cup \beta)$ gilt $\alpha \rightarrow \beta$; damit ist α dort Schlüssel.

Garantie / Preis

Garantie: Jeder Schritt ist verlustlos; am Ende sind alle Teilrelationen in BCNF.

Preis: Nicht immer Abhängigkeitserhaltend.

BCNF-Dekomposition - Beispiel 1: Städte

Günstiger Fall: Die BCNF-Zerlegung bleibt hier abhängigkeiterhaltend.

Ausgangsschema

Schema:

Städte(Ort, BLand,
MinisterpräsidentIn, EW)

Kandidatenschlüssel:

- ▶ $\kappa_1 = \{\text{Ort, BLand}\}$
- ▶ $\kappa_2 = \{\text{Ort, MinisterpräsidentIn}\}$

FDs:

- ▶ $f_1 : \text{Ort, BLand} \rightarrow \text{EW}$
- ▶ $f_2 : \text{BLand} \rightarrow \text{MinisterpräsidentIn}$
- ▶ $f_3 : \text{MinisterpräsidentIn} \rightarrow \text{BLand}$

Zerlegung über f_3

Verletzende FD: $f_3 : \text{MinisterpräsidentIn} \rightarrow \text{BLand}$

Teilrelationen:

$R_1 = \text{Regierungen}(\text{BLand, MinisterpräsidentIn})$

$R_2 = \text{Städte}_1(\text{Ort, MinisterpräsidentIn, EW})$

Lokale FDs:

- ▶ In R_1 liegen die Attribute aus f_3 ; dort gelten f_2 und f_3 .
- ▶ In R_2 bleibt $f'_1 : \text{MinisterpräsidentIn, Ort} \rightarrow \text{EW}$.

Ergebnis: R_1 und R_2 sind in BCNF; die FDs bleiben erhalten.

BCNF-Dekomposition - Beispiel 2: PLZVerzeichnis

Problematischer Fall: Die BCNF-Zerlegung ist hier **nicht** abhängigkeiterhaltend.

Ausgangsschema

Schema:

PLZVerzeichnis(Straße, Ort, BLand, PLZ)

Kandidatenschlüssel:

$\kappa_1 = \{\text{Straße, Ort, BLand}\}$

$\kappa_2 = \{\text{Straße, PLZ}\}$

FDs:

$f_1 : \text{Straße, Ort, BLand} \rightarrow \text{PLZ}$

$f_2 : \text{PLZ} \rightarrow \text{Ort, BLand}$

Verletzung: BCNF wird durch f_2 verletzt, weil *PLZ* kein Superschlüssel ist.

Beobachtung: Dieselbe FD

$\text{PLZ} \rightarrow \text{Ort, BLand}$

für viele Straßen erneut auf.

Zerlegung über f_2

$R_1 = (\underline{\text{PLZ}}, \text{Ort, BLand})$

$R_2 = (\text{Straße}, \underline{\text{PLZ}})$

Lokale FDs: $F_1 = \{f_2\}$, $F_2 = \emptyset$.

Keine Rekonstruktion von f_1 :

$\{\text{Straße, Ort, BLand}\}_{F_1 \cup F_2}^+ = \{\text{Straße, Ort, BLand}\}$

PLZ ist also nicht ableitbar; die Zerlegung ist **nicht abhängigkeiterhaltend**.

Konsequenz: Oft ist 3NF der bevorzugte Kompromiss zwischen BCNF und Abhängigkeitserhaltung.

Zusammenfassung Normalformen

Ziele der Normalisierung: Redundanz ↓, Anomalien ↓, Integrität ↑.

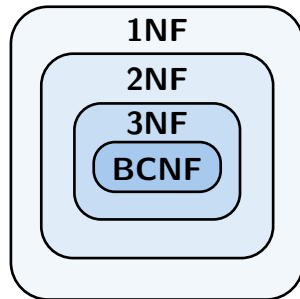
Fokus der Normalformen

- 1NF:** atomare Werte.
2NF: keine partiellen Abhängigkeiten von Schlüsselteilen.
3NF: keine transitiven Abhängigkeiten zwischen Nichtschlüsselattributen.
BCNF: jede linke Seite einer nicht-trivialen FD ist ein Superschlüssel.

Algorithmen & Garantien

Algorithmus	Zielform	Verlustlos	Abh.-erhaltend
3NF-Synthese	3NF	✓	✓
BCNF-Dekomposition	BCNF	✓	nicht immer

Ausblick: Höhere NFs (4NF, 5NF) für MVDs, JDs. Für FDs sind 3NF und BCNF die zentralen Zielgrößen.



Merke: $BCNF \subsetneq 3NF \subsetneq 2NF \subsetneq 1NF$