(Bisherige) Gliederung der Vorlesung

0. Vorbemerkungen
1. Einführung in Datenbanken
2. Datenbankmodelle für den Entwurf
3. Datenbankmodelle für die Realisierung
4. Grundlagen von Anfragen und Änderungen
    - 4.1 Kritierien für Anfragesprachen
    - 4.2 Relationenalgebra
    - 4.3 Anfragekalküle
5. Datenbanksprache SQL
6. Datenbankentwurfsprozeß

7. Relationaler Datenbankentwurf

- zum Verfeinern des logischen Entwurfs
- Ziel: Vermeidung von Redundanzen durch Aufspalten von Relationenschemata, ohne gleichzeitig
  - semantische Informationen zu verlieren (Abhängigkeitstreue)
  - die Möglichkeit zur Rekonstruktion der Relationen zu verlieren (Verbundtreue)
- Redundanzvermeidung durch Normalformen
- Inhalt:
- 1. Funktionale Abhängigkeit
- 2. Schema-Eigenschaften (Normalformen)
- 3. Transformationseigenschaften
- 4. Entwurfsverfahren

7.1. Funktionale Abhängigkeiten

- **Funktionale Abhängigkeiten** einer Relation zwischen Attributmengen X und Y, wenn in jedem Tupel der Relation der Attributwert unter den X-Komponenten den Attributwert unter den Y-Komponenten festlegt
- **Funktionale Abhängigkeit**
  (kurz: FD, von functional dependency)
  Schreibweise: $X \rightarrow Y$

Gregor Engels
7. Relationaler Datenbankentwurf

Funktionale Abhängigkeiten

- Im Beispiel auf folgender Folie:
  ISBN → Titel, Verlagsname
- Nicht:
  ISBN → Autor, Stichwort
- Trivialerweise:
  ISBN → ISBN

WS 2000/01
Techniken des Softwareentwurfs II

Bücher-Relation mit Redundanzen

Bücher

ISBN	Titel	Autor	Version	Stichwort	Verlagsname
0-8053-1753-8	Princ. of DBS	Elmasri	1,1989	RDB	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Navathe	1,1989	RDB	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Elmasri	2,1994	RDB	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Navathe	2,1994	RDB	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Elmasri	1,1989	Lehrbuch	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Navathe	1,1989	Lehrbuch	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Elmasri	2,1994	Lehrbuch	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Navathe	2,1994	Lehrbuch	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Elmasri	1,1989	ER	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Navathe	1,1989	ER	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Elmasri	2,1994	ER	Benj./Cumm.
0-8053-1753-8	Princ. of DBS	Navathe	2,1994	ER	Benj./Cumm.

Problem: update-Anomalien

Beispiel: Einfügen einer neuen Version 3, 1996

Schlüssel als Spezialfall

- für Beispiel auf folgender Folie
  PANr → Vorname, Nachname, PLZ, Ort, Straße, Hausnummer
  Geburtsdatum
- Immer: PANr → PANr
- Dann gesamtes Schema auf rechter Seite
- Wenn linke Seite minimal: Schlüssel
- Formal: Schlüssel X liegt vor, wenn für Relationenschema R

  FD X → R gilt und X minimal

Ziel des Datenbankentwurfs: alle gegebenen funktionalen Abhängigkeiten in „Schlüsselabhängigkeiten“ umformen, ohne dabei semantische Information zu verlieren

Schlüssel im Beispiel

Personen

PANr	Vorname	Nachname	PLZ	Ort	Straße	HNr	Geb.datum
4711	Andreas	Heuer	18209	DBR	BHS	15	31.10.58
5588	Gunter	Saake	39106	MD	STS	55	05.10.60
6834	Michael	Korn	39104	MD	BS	41	24.09.74
7754	Andreas	Möller	18209	DBR	RS	31	25.02.76
8832	Tamara	Jagellovsk	38106	BS	GS	12	11.11.73
9912	Antje	Hellhof	18059	HRO	AES	21	04.04.70
9999	Christa	Loeser	69121	HD	TS	38	10.05.69

Pers_Telefon

PANr	Telefon
4711	038203-12230
4711	0381-498-3401
4711	0381-498-3427
5588	0391-345677
5588	0391-5592-3800
9999	06221-400177

Ableitung von FDs

r

A	B	C
a_1	b_1	c_1
a_2	b_1	c_1
a_3	b_2	c_1
a_4	b_1	c_1

- genügt $A \rightarrow B$ und $B \rightarrow C$
- dann gilt auch $A \rightarrow C$
- nicht ableitbar $C \rightarrow A$ oder $C \rightarrow B$

Ableitung von FDs

- Bezeichnungen: - F eine Menge von funktionalen Abhängigkeiten
  - f eine funktionale Abhängigkeit
  - $\mathbf{SAT}_R(F)$ die Menge von Relationen über dem Relationenschema R, die die funktionalen Abhängigkeiten aus F erfüllen.
- Gilt für f über R $\mathbf{SAT}_R(F) \subseteq \mathbf{SAT}_R(f)$, dann impliziert F die funktionale Abhängigkeit f (kurz: $F \models f$)
- obiges Beispiel:
  $F = \{A \rightarrow B, B \rightarrow C\} \models A \rightarrow C$
- Bemerkung:
  Falls F = {}, gilt: $\mathbf{SAT}_R(\{\}) = \mathbf{REL}\ (R)$

transitive Hülle

- die **(transitive) Hülle** einer Menge von funktionalen Abhängigkeiten F ist definiert durch
  $\mathbf{F^+ := \{ f \mid f\ funktionale\ Abhängigkeit \wedge F \models f \}}$
- Beispiel:
  $\{A \rightarrow C, AB \rightarrow C, A \rightarrow BC, AB \rightarrow AB\} \subseteq F^+$

Ableitungsregeln

• ***gültig*** (sound) (nur solche FDs, die logisch impliziert sind)

• ***vollständig*** (complete) (alle implizierten FDs)

• ***unabhängig*** (independent) oder auch bzgl. $\subseteq$ minimal (alle Regeln sind nötig)

Name	Regel		
$\boxed{R}$ Reflexivität	$\{\ \}$	$\Rightarrow$	$X \to X$
$\boxed{A}$ Akkumulation	$\{X \to Y\,Z, Z \to V\,W\}$	$\Rightarrow$	$X \to YZV$
$\boxed{P}$ Projektivität	$\{X \to Y\,Z\}$	$\Rightarrow$	$X \to Y$

(X, Y, Z, V, W sind Attributmengen bzw. einzelne Attribute)

Membership-Problem

Kann eine bestimmte FD $X \to Y$ aus der vorgegebenen Menge F abgeleitet werden, d. h. wird sie von F impliziert?

Membership-Problem: “$X \to Y \in F^+$?“

Problem: Berechnung von F^+ aufwendig, deshalb Beschränkung auf die von einem Attribut aus „erreichbaren“, d.h. abhängigen Attribute

Hülle einer Attributmenge X bzgl. F ist $X^*_F := \{A \mid X \to A \in F^+\}$

Das Membership-Problem kann nun durch das modifizierte

Membership-Problem (2): “$Y \subseteq X^*_F$?“

in linearer Zeit mit dem **RAP-Algorithmus** gelöst werden.

RAP-Algorithmus

1. Bestimme X, setze $X^* := X$ ($\boxed{\text{R}}$-Regel für X)
2. Gibt es FD $f_1 := X_1 \to Y_1 \in F$ mit $X_1 \subseteq X^*$?
3. Wenn ja, dann wird X^* gemäß $X^* := X^* \cup Y_1$ vergrößert ($\boxed{\text{A}}$-Regel).
4. Führe Schritt 2 und 3 so lange aus, bis X^* stabil (Hülle)
5. Ist $Y \subseteq X^*$, dann ist $X \to Y \in F^+$ ($\boxed{\text{P}}$ -Regel)

Überdeckungen

F heißt *äquivalent* zu G

(oder: F *Überdeckung* von G; kurz: $F \equiv G$)

falls $F^+ = G^+$

7.2. Schema-Eigenschaften

Relationenschemata, Schlüssel und Fremdschlüssel so wählen, daß

1. alle Anwendungsdaten aus den Basisrelationen hergeleitet werden können,
2. nur semantisch sinnvolle und konsistente Anwendungsdaten dargestellt werden können und
3. die Anwendungsdaten möglichst nicht-redundant dargestellt werden.

Hier: Forderung 3

- Redundanzen innerhalb einer Relation: Normalformen
- globale Redundanzen: Minimalität

Update-Anomalien

Redundanzen in Basisrelationen unerwünscht:

- Belegen unnötigen Speicherplatz (eher unwichtig)
- Information redundant → Änderung muß diese Information in allen ihren Vorkommen verändern (in relationalen Systemen nur schwer zu realisieren)

Beispiel **insert**-Anomalie:

Bücher

ISBN	Titel	Autor	Version	Stichwort	Verlagsname
0-8053-1753-8	Princ. of DBS	Elmasri	3,1996	RDB	null

Erste Normalform

führt zunächst Redundanzen ein

Erste Normalform (1NF): nur atomare Attribute in Relationenschemata

Invnr	Titel	ISBN	Autoren
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer, Scholl
4711	Datenbanken	3-765	Vossen, Witt
4712	Datenbanken	3-891	Ullman
4717	Pascal	3-999	Wirth, Dijkstra

wäre in erster Normalform

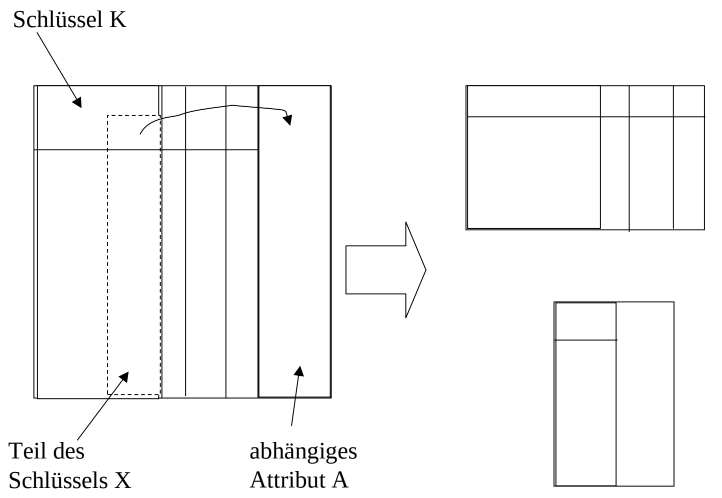
Invnr	Titel	ISBN	Autor
0007	Dr. No	3-125	James Bond
1201	Objektbanken	3-111	Heuer
1201	Objektbanken	3-111	Scholl
4711	Datenbanken	3-765	Vossen
4711	Datenbanken	3-765	Witt
4712	Datenbanken	3-891	Ullman
4717	Pascal	3-999	Wirth
4717	Pascal	3-999	Dijkstra

Zweite Normalform

Zweite und weitere Normalformen: aufgrund der Struktur von Abhängigkeiten Redundanzen entdecken

- **Zweite Normalform:** Keine partiellen Abhängigkeiten zwischen einem Schlüssel und weiteren Nicht-Primattributen (Attribute, die nicht in einem Schlüssel auftauchen)
- Partielle Abhängigkeit liegt vor, wenn ein Attribut funktional schon von einem Teil des Schlüssel abhängt
- Beispiel: Invnr $\rightarrow$ Titel
  und
  Invnr, Autor $\rightarrow$ Invnr, Titel, ISBN, Autor
  Invnr und Autor zusammen Schlüssel
  Titel hängt aber allein von Invnr ab, also Bücher nicht in 2. Normalform!
- Zweite Normalform erreichen durch Elimination der rechten Seite der partiellen Abhängigkeit und Kopie der linken Seite (siehe nächste Folie)

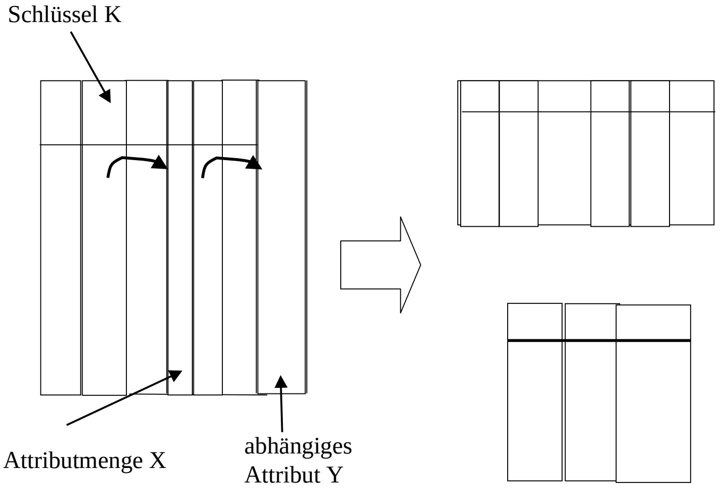
Veranschaulichung zweite Normalform



Dritte Normalform

- **Dritte Normalform:** Keine transitiven Abhängigkeiten zwischen einem Schlüssel und weiteren Nicht-Primattributen
- *transitive Abhängigkeit:* Schlüssel *K* bestimmt Attributmenge *X* funktional, diese aber auch eine Attributmenge *Y*
  transitive Abhängigkeit: $K \rightarrow X, X \not\rightarrow K, X \rightarrow Y, Y \not\subseteq KX$
- Beispiel: PANr → PLZ und PLZ → Ort
  Information, daß zur PLZ ‘18209‘ der Ort ‘DBR‘ gehört, ist redundant in der Relation Personen gespeichert
- Dritte Normalform erreichen durch Elimination von *Y* und Kopie von *X* (siehe nächste Folie)

Veranschaulichung dritte Normalform



Boyce-Codd-Normalform

Nicht nur Nicht-Primattribute betrachten

Im aktuellen Postleitzahlsystem der Deutschen Post innerhalb der Attribute
PLZ, Ort, Straße, Hausnummer
folgende funktionale Abhängigkeiten:

Ort, Straße, Hausnummer $\rightarrow$ PLZ,
PLZ $\rightarrow$ Ort

Schlüssel: Ort, Straße, Hausnummer und PLZ, Straße, Hausnummer

alle Attribute nun Primattribute, also dritte Normalform

Trotzdem Redundanz: PLZ, Straße, Hausnummer $\rightarrow$ PLZ $\rightarrow$ Ort

partielle (oder transitive) Abhängigkeit

Boyce-Codd-Normalform (BCNF) definiert transitive Abhängigkeiten nicht nur über Nicht-Primattribute

Minimalität

Global Redundanzen vermeiden

andere Kriterien (wie Normalformen) mit möglichst wenig Schemata erreichen

Beispiel: Attributmenge ABC, FD-Menge $\{A \to B, B \to C\}$

Datenbankschemata in dritter Normalform:

$$S=\{(AB, \{A\}), (BC, \{B\})\}$$

und

$$S'=\{(AB, \{A\}), (BC,\{B\}), (AC,\{A\})\}$$

Redundanzen in S'

Schema-Eigenschaften: Zusammenfassung

Kennung	Schemaeigenschaft	Kurzcharakteristik
	1 NF	Nur atomare Attribute
	2 NF	keine partielle Abhängigkeit eines Nicht-Primattributes von einem Schlüssel
S 1	3 NF	keine transitive Abhängigkeit eines Nicht-Primattributes von einem Schlüssel
	BCNF	keine transitive Abhängigkeit eines Attributes von einem Schlüssel
S 2	Minimalität	minimale Anzahl von Relationenschemata, die die anderen Eigenschaften erfüllt

7.3. Transformationseigenschaften

Erreichen von Normalformen durch Zerlegung von Relationenschemata

Dabei beachten:

1. Nur semantisch sinnvolle und konsistente Anwendungsdaten darstellen (***Abhängigkeitstreue***)
2. Alle Anwendungsdaten sollen aus Basisrelationen hergeleitet werden können (***Verbundtreue***)

Abhängigkeitstreue

Abhängigkeitstreue ist folgende Forderung:

• Allgemein: Menge der erfaßten Abhängigkeiten äquivalent zur Menge der im System darstellbaren Abhängigkeiten (etwa Schlüssel und Fremdschlüssel)

• Hier spezieller: Menge der FD‘s äquivalent zur Menge der Schlüsselabhängigkeiten

Beispiel: Attribute:
PLZ (P), Ort (O), Straße (S), Hausnummer (H)

funktionale Abhängigkeiten *F*:
OSH → P, P → O

Datenbankschema R:
(OSHP, {OSH})

Menge der zugehörigen Schlüsselabhängigkeiten
{OSH → OSHP}

nicht äquivalent zu *F*; also ist das Schema *R* nicht *abhängigkeitstreu*

Abhängigkeitstreue formal

- $S = \{(R_1, \mathcal{K}_1), ..., (R_p, \mathcal{K}_p)\}$ lokal erweitertes Datenbankschema,
- F Menge lokaler Abhängigkeiten
- S charakterisiert vollständig F (oder: ist abhängigkeitstreu bezüglich F) genau dann, wenn
  $$F \equiv \{K \to R \mid (R, \mathcal{K}) \in S, K \in \mathcal{K}\}$$
  d.h. F wird durch Schlüsselabhängigkeiten äquivalent dargestellt

Verbundtreue

Orginalrelation soll aus zerlegten Relationen mit natürlichem Verbund zurückgewonnen werden können: *Verbundtreue*

Beispiel: Relationenschema

$$R = ABC$$

in

$$R_1 = AB \text{ und } R_2 = BC$$

zerlegt; ist bei

$$F = \{A \to B, C \to B\}$$

nicht verbundtreu, bei

$$F' = \{A \to B, B \to C\}$$

verbundtreu

Kriterium: Attributmenge im Schnitt der entstandenen Relationenschemata (hier: B) bestimmt eines der beiden Relationenschemata (hier: BC) funktional (ist also Schlüssel)

Beispielrelationen zur Verbundtreue

Originalrelation:

A	B	C
1	2	3
4	2	5

Dekomposition:

A	B
1	2
4	2

B	C
2	3
2	5

Verbund (nicht verbundtreu):

A	B	C
1	2	3
4	2	5
1	2	5
4	2	3

Originalrelation:

A	B	C
1	2	3
4	2	3

Dekomposition:

A	B
1	2
4	2

B	C
2	3

Verbund (verbundtreu):

A	B	C
1	2	3
4	2	3

Verbundtreue formal

- Definition: Dekomposition einer Attributmenge X in $X_1, ..., X_p$ mit $X = \bigcup^p_{i=1} X_i$ heißt *verbundtreu* ($\pi \bowtie$ -treu, lossless) bezüglich einer Menge von Abhängigkeiten G über X genau dann, wenn
  $\forall$ Relationen r, die G erfüllen, gilt:

$$\pi_{x1} (r) \bowtie ... \bowtie \pi_{x_p} (r) = r$$

- Einfaches Kriterium für zwei Relationenschemata: Dekomposition von X in X_1 und X_2 ist verbundtreu bzgl. F, wenn $X_1 \cap X_2 \rightarrow X_1 \in F^+$ oder $X_1 \cap X_2 \rightarrow X_2 \in F^+$

Verbundtreue formal

- Allgemeineres Kriterium: G Menge funktionaler Abhängigkeiten
  $\exists i \in \{1, ..., p\} : X_i \rightarrow X \in G^+ \quad \Rightarrow$
  Die Dekomposition von X in $X_1, ..., X_p$ ist verbundtreu bezüglich G
- minimale Teilmenge von X_i: *Universalschlüssel*
- Beispiel erster Fall: AC der einzige Universalschlüssel, in keinem Relationenschema enthalten
- Beispiel zweiter Fall: Universalschlüssel A

Transformationseigenschaften: Zusammenfassung

Kennung	Transformationseigenschaft	Kurzcharakteristik
T 1	Abhängigkeitstreue	Alle gegebenen Abhängigkeiten sind durch Schlüssel repräsentiert
T 2	Verbundtreue	Die Originalrelationen können durch den Verbund der Basisrelationen wiedergewonnen werden

7.4. Entwurfsverfahren

- Universum U und FD-Menge F gegeben
- lokal erweitertes Datenbankschema
  $S = \{(R_1, K_1), ..., (R_p, K_p)\}$
  berechnen mit

$\boxed{\mathbf{T}}\boxed{\mathbf{1}}$ S charakterisiert vollständig F

$\boxed{\mathbf{S}}\boxed{\mathbf{1}}$ S ist in 3NF bezüglich F

$\boxed{\mathbf{T}}\boxed{\mathbf{2}}$ Die Dekomposition von U in $R_1,..., R_p$ ist verbundtreu bezüglich F

$\boxed{\mathbf{S}}\boxed{\mathbf{2}}$ Minimalität, d. h. $\nexists S' : S'$ erfüllt $\boxed{\mathbf{T}}\boxed{\mathbf{1}}$, $\boxed{\mathbf{S}}\boxed{\mathbf{1}}$, $\boxed{\mathbf{T}}\boxed{\mathbf{2}}$ und $|S'| < |S|$

- Datenbankschemata schlecht entworfen, wenn nur eins dieser vier Kriterien nicht erfüllt
- Beispiel: S = $\{(AB, \{A\}), (BC, \{B\}), (AC, \{A\})\}$ erfüllt $\boxed{\mathbf{T}}\boxed{\mathbf{1}}$, $\boxed{\mathbf{S}}\boxed{\mathbf{1}}$ und $\boxed{\mathbf{T}}\boxed{\mathbf{2}}$ bezüglich $F = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$
  in dritter Relation AC-Tupel redundant oder inkonsistent
- korrekt: $S' = \{(AB, \{A\}), (BC, \{B\})\}$

BCNF und Abhängigkeitstreue unvereinbar

Attribute

PLZ (P), Ort (O), Straße (S), Hausnummer (H)

funktionale Abhängigkeiten F

OSH $\rightarrow$ P, P $\rightarrow$ O

Datenbankschema S

(OSHP, {OSH, PSH})

PSH auch Schlüssel, da PSH $\rightarrow$ OSHP mit PSH minimal

Schema in 3NF, da alle Attribute Primattribute

Schema nicht in BCNF, da

{PSH $\rightarrow$ P $\rightarrow$ O}

transitive Abhängigkeit des Primattributs O

Jede Zerlegung von OSHP zerstört Abhängigkeit

OSH $\rightarrow$ P

Abhängigkeitstreue nicht gewährleistet

Dekomposition: Start

Start: initiales Relationenschema *R* mit allen Attributen und einer von erfaßten Abhängigkeiten implizierten Schlüsselmenge

- Attributmenge U und eine FD-Menge F
- suche alle $K \rightarrow U$ mit K minimal, für die $K \rightarrow U \in F^+$ gilt ($\mathcal{K}(F)$)
- (U, $\mathcal{K}(F)$) initiales Realationenschema

Dekomposition: Normalisierung

Normalisierungsschritt: falls $K \rightarrow Y \rightarrow A$, aus R Attributmenge A eliminieren und mit Y in ein neues Relationenschema stecken

- $\mathcal{R} = (R, \mathcal{K})$ und F über R gegeben
- Falls $\mathcal{R}$ in 3NF ist: fertig
- Sonst: existiert für Schlüssel K

$$K \rightarrow Y,\ Y \not\rightarrow K,\ Y \rightarrow A,\ A \notin KY$$

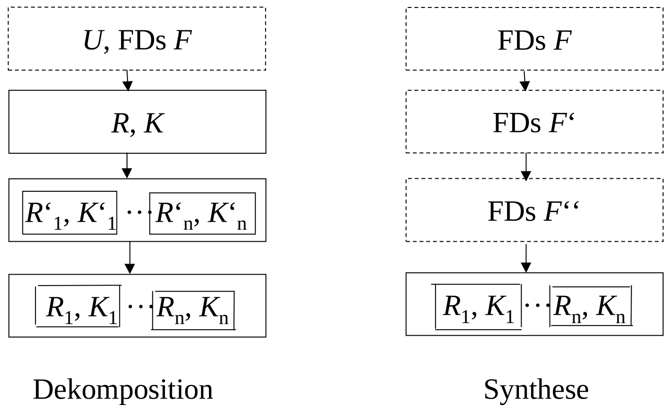
Wähle dann

$$R_1 := R - A \qquad R_2 := YA$$
$$\mathcal{R}_1 := (R_1, \mathcal{K}) \qquad \mathcal{R}_2 := (R_2, \mathcal{K}_2 = \{Y\})$$

(neue FD-Mengen durch Projektion auf noch existierende Attribute)

- Vorteile: 3NF, Verbundtreue
- Nachteile: restliche Kriterien nicht, reihenfolgeabhängig, NP-vollständig (Schlüsselsuche)

Vergleich Dekomposition - Synthese



Syntheseverfahren

- Prinzip: Synthese formt Original-FD-Menge F in resultierende Menge von Schlüsselabhängigkeiten G so um, daß $F \equiv G$ gilt
- “Abhängigkeitstreue“ im Verfahren verankert
- 3NF und Minimalität wird auch erreicht, reihenfolgeunabhängig
- Zeitkomplexität: quadratisch

Verfahren (grob)

- Eliminiere Redundanzen durch Entfernen überflüssiger FDs und Attribute.
  - FDs aus F überflüssig, wenn *redundant* d.h. F - $\{f\} \equiv F$, dann f redundant
  - Attribute aus F überflüssig, wenn unwesentlich (siehe Formalia 2NF)
- Fasse FDs zu “Äquivalenzklassen“
  FDs in eine Klasse, die gleiche oder äquivalente linke Seiten haben; pro Äquivalenzklasse ein Relationenschema
- Trick Verbundtreue: Original-FD-Menge F um $U \to \delta$ erweitern, δ Dummy-Attribut, das nach Synthese entfernt wird