Teil V

Relationale Entwurfstheorie

Relationale Entwurfstheorie

Relationenmodell

WEINE

WeinID	Name	Farbe	Jahrgang	Weingut
1042	La Rose Grand Cru	Rot	1998	Château La Rose
2168	Creek Shiraz	Rot	2003	Creek
3456	Zinfandel	Rot	2004	Helena
2171	Pinot Noir	Rot	2001	Creek
3478	Pinot Noir	Rot	1999	Helena
4711	Riesling Reserve	Weiß	1999	Müller
4961	Chardonnay	Weiß	2002	Bighorn

ERZEUGER

Weingut	Anbaugebiet	Region
Creek	Barossa Valley	South Australia
Helena	Napa Valley	Kalifornien
Château La Rose	Saint-Emilion	Bordeaux
Château La Pointe	Pomerol	Bordeaux
Müller	Rheingau	Hessen
Bighorn	Napa Valley	Kalifornien

Begriffe des Relationenmodells

Begriff	**Informale Bedeutung**
Attribut	Spalte einer Tabelle
Wertebereich	mögliche Werte eines Attributs (auch Domäne)
Attributwert	Element eines Wertebereichs
Relationenschema	Menge von Attributen
Relation	Menge von Zeilen einer Tabelle
Tupel	Zeile einer Tabelle
Datenbankschema	Menge von Relationenschemata
Datenbank	Menge von Relationen (Basisrelationen)

Begriffe des Relationenmodells /2

Begriff	Informale Bedeutung
Schlüssel	minimale Menge von Attributen, deren Werte ein Tupel einer Tabelle eindeutig identifizieren
Primärschlüssel	ein beim Datenbankentwurf ausgezeichneter Schlüssel
Fremdschlüssel	Attributmenge, die in einer anderen Relation Schlüssel ist
Fremdschlüsselbedingung	alle Attributwerte des Fremdschlüssels tauchen in der anderen Relation als Werte des Schlüssels auf

Formalisierung Relationenmodell

- **Attribute und Domänen**
  - $\mathcal{U}$ nichtleere, endliche Menge: Universum
  - $A \in \mathcal{U}$: Attribut
  - $\mathcal{D} = \{D_1, \ldots, D_m\}$ Menge endlicher, nichtleerer Mengen: jedes D_i: Wertebereich oder Domäne
  - total definierte Funktion $\mathrm{dom} : \mathcal{U} \longrightarrow \mathcal{D}$
  - $\mathrm{dom}(A)$: Domäne von A
    $w \in \mathrm{dom}(A)$: Attributwert für A

Formalisierung Relationenmodell /2

- **Relationenschemata und Relationen**
  - $R \subseteq \mathcal{U}$: Relationenschema
  - Relation r über $R = \{A_1, \ldots, A_n\}$ (kurz: $r(R)$) ist endliche Menge von Abbildungen $t : R \longrightarrow \bigcup_{i=1}^{m} D_i$, Tupel genannt
  - Es gilt $t(A) \in \mathrm{dom}(A)$ ($t(A)$ Restriktion von t auf $A \in R$)
  - für $X \subseteq R$ analog $t(X)$ X-Wert von t
  - Menge aller Relationen über R: $\mathbf{REL}(R) := \{r \mid r(R)\}$

Formalisierung Relationenmodell /3

- **Datenbankschema und Datenbank**
  - Menge von Relationenschemata $S := \{R_1, \ldots, R_p\}$: Datenbankschema
  - Datenbank über S: Menge von Relationen $d := \{r_1, \ldots, r_p\}$, wobei $r_i(R_i)$
  - Datenbank d über S: $d(S)$
  - Relation $r \in d$: Basisrelation

Integritätsbedingungen

- Identifizierende Attributmenge $K := \{B_1, \ldots, B_k\} \subseteq R$:

$$\forall t_1, t_2 \in r \; [t_1 \neq t_2 \quad \Longrightarrow \quad \exists B \in K : t_1(B) \neq t_2(B)]$$

- Schlüssel: ist minimale identifizierende Attributmenge
  - `{Name, Jahrgang, Weingut}` und
  - `{WeinID}` für `WEINE`
- Primattribut: Element eines Schlüssels
- Primärschlüssel: ausgezeichneter Schlüssel
- Oberschlüssel oder **Superkey**: jede Obermenge eines Schlüssels (= identifizierende Attributmenge)
- Fremdschlüssel: $X(R_1) \rightarrow Y(R_2)$

$$\{t(X) | t \in r_1\} \subseteq \{t(Y) | t \in r_2\}$$

Relationaler DB-Entwurf: Überblick

- 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 (s.u.)

Relation mit Redundanzen

WEINE	WeinID	Name	...	Weingut	Anbaugebiet	Region
	1042	La Rose Gr. Cru	...	Ch. La Rose	Saint-Emilion	Bordeaux
	2168	Creek Shiraz	...	Creek	Barossa Valley	South Australia
	3456	Zinfandel	...	Helena	Napa Valley	Kalifornien
	2171	Pinot Noir	...	Creek	Barossa Valley	South Australia
	3478	Pinot Noir	...	Helena	Napa Valley	Kalifornien
	4711	Riesling Res.	...	Müller	Rheingau	Hessen
	4961	Chardonnay	...	Bighorn	Napa Valley	Kalifornien

Redundanzen

- Redundanzen in Basisrelationen aus mehreren Gründen unerwünscht:
  - Redundante Informationen belegen unnötigen Speicherplatz
  - Änderungsoperationen auf Basisrelationen mit Redundanzen nur schwer korrekt umsetzbar: wenn eine Information redundant vorkommt, muss eine Änderung diese Information in allen ihren Vorkommen verändern
    - mit normalen relationalen Änderungsoperationen und den in relationalen Systemen vorkommenden lokalen Integritätsbedingungen (Schlüsseln) nur schwer realisierbar

Änderungsanomalien

- Einfügen in die mit Redundanzen behaftete `WEINE`-Relation:

```
insert into WEINE (WeinID, Name, Farbe, Jahrgang,
      Weingut, Anbaugebiet, Region)
   values (4711, 'Chardonnay', 'Weiß', 2004,
      'Helena', 'Rheingau', 'Kalifornien')
```

  - WeinID 4711 bereits anderem Wein zugeordnet: verletzt FD `WeinID` → `Name`
  - Weingut Helena war bisher im Napa Valley angesiedelt: verletzt FD `Weingut` → `Anbaugebiet`
  - Rheingau liegt nicht in Kalifornien: verletzt FD `Anbaugebiet` → `Region`

- auch **update**- und **delete**-Anomalien

Funktionale Abhängigkeiten

- funktionale Abhängigkeit zwischen Attributmengen X und Y einer Relation

> Wenn in jedem Tupel der Relation der Attributwert unter den X-Komponenten den Attributwert unter den Y-Komponenten festlegt.

- Unterscheiden sich zwei Tupel in den X-Attributen nicht, so haben sie auch gleiche Werte für alle Y-Attribute
- Notation für funktionale Abhängigkeit (FD, von functional dependency): $X \rightarrow Y$
- Beispiel:

```
WeinID      → Name, Weingut
Anbaugebiet → Region
```

- aber nicht: `Weingut → Name`

Schlüssel als Spezialfall

- für Beispiel auf Folie 5-10

  `WeinID` $\rightarrow$ `Name`, `Farbe`, `Jahrgang`, `Weingut`, `Anbaugebiet`, `Region`

- Immer: `WeinID` $\rightarrow$ `WeinID`,
  dann gesamtes Schema auf rechter Seite
- Wenn linke Seite minimal: Schlüssel
- Formal: Schlüssel X liegt vor, wenn für Relationenschema R FD $X \rightarrow R$ gilt und X minimal

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

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 /2

- Gilt für f über R $\textbf{SAT}_R(F) \subseteq \textbf{SAT}_R(f)$, dann impliziert F die FD f (kurz: $F \models f$)
- obiges Beispiel:

$$F = \{A \rightarrow B, B \rightarrow C\} \models A \rightarrow C$$

- Hüllenbildung: Ermittlung aller funktionalen Abhängigkeiten, die aus einer gegebenen FD-Menge abgeleitet werden können
- Hülle $F_R^+ := \{f \mid (f \text{ FD über } R) \wedge F \models f\}$
- Beispiel:

$$\begin{aligned}\{A \rightarrow B, B \rightarrow C\}^+ \quad = \quad & \{A \rightarrow B, B \rightarrow C, A \rightarrow C, AB \rightarrow C, A \rightarrow BC, \ldots, \\ & AB \rightarrow AB, \ldots\}\end{aligned}$$

Ableitungsregeln

F1	Reflexivität	$X \supseteq Y$	$\Longrightarrow$	$X \rightarrow Y$
F2	Augmentation	$\{X \rightarrow Y\}$	$\Longrightarrow$	$XZ \rightarrow YZ$ sowie $XZ \rightarrow Y$
F3	Transitivität	$\{X \rightarrow Y, Y \rightarrow Z\}$	$\Longrightarrow$	$X \rightarrow Z$
F4	Dekomposition	$\{X \rightarrow YZ\}$	$\Longrightarrow$	$X \rightarrow Y$
F5	Vereinigung	$\{X \rightarrow Y, X \rightarrow Z\}$	$\Longrightarrow$	$X \rightarrow YZ$
F6	Pseudotransitivität	$\{X \rightarrow Y, WY \rightarrow Z\}$	$\Longrightarrow$	$WX \rightarrow Z$

F1-F3 bekannt als Armstrong-Axiome (sound, complete)

- *gültig* (sound): Regeln leiten keine FDs ab, die logisch nicht impliziert
- *vollständig* (complete): alle implizierten FDs werden abgeleitet
- *unabhängig* (independent) oder auch bzgl. $\subseteq$ minimal: keine Regel kann weggelassen werden

Beweis: F1

- Annahme: $X \supseteq Y$, $X, Y \subset R$, $t_1, t_2 \in r(R)$ mit $t_1(X) = t_2(X)$
- dann folgt: $t_1(Y) = t_2(Y)$ wegen $X \supseteq Y$
- daraus folgt: $X \rightarrow Y$

Beweis: **F2**

- Annahme: $X \rightarrow Y$ gilt in $r(R)$, jedoch nicht: $XZ \rightarrow YZ$
- dann müssen zwei Tupel $t_1, t_2 \in r(R)$ existieren, so dass gilt
  (1) $t_1(X) = t_2(X)$
  (2) $t_1(Y) = t_2(Y)$
  (3) $t_1(XZ) = t_2(XZ)$
  (4) $t_1(YZ) \neq t_2(YZ)$
- Widerspruch wegen $t_1(Z) = t_2(Z)$ aus (1) und (3), woraus folgt: $t_1(YZ) = t_2(YZ)$ (in Verbindung mit (4))

Beweis: **F3**

- Annahme: in $r(R)$ gelten:
  (1) $X \rightarrow Y$
  (2) $Y \rightarrow Z$
- demzufolge für zwei beliebige Tupel $t_1, t_2 \in r(R)$ mit $t_1(X) = t_2(X)$ muss gelten:
  (3) $t_1(Y) = t_2(Y)$ (wegen (1))
  (4) $t_1(Z) = t_2(Z)$ (wegen (3) und (2))
- daher gilt: $X \rightarrow Z$

Alternative Regelmenge

- B-Axiome oder RAP-Regeln

R Reflexivität $\{\} \implies X \rightarrow X$

A Akkumulation $\{X \rightarrow YZ, Z \rightarrow AW\} \implies X \rightarrow YZA$

P Projektivität $\{X \rightarrow YZ\} \implies X \rightarrow Y$

- Regelmenge ist vollständig, da Armstrong-Axiome daraus abgeleitet werden können

Membership-Problem

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

Membership-Problem: „$X \rightarrow Y \in F^+$?"

- Hülle einer Attributmenge X bzgl. F ist $X_F^+ := \{A \mid X \rightarrow A \in F^+\}$
- Membership-Problem kann durch das modifizierte Problem

  Membership-Problem (2): „$Y \subseteq X_F^+$?"

  in linearer Zeit gelöst werden

Algorithmus CLOSURE

- Ermittlung von X_F^+: die Hülle von X bzgl. F

```
CLOSURE(F, X):
    X⁺ := X
    repeat
        X̄⁺ := X⁺   /* R-Regel */
        forall FDs Y → Z ∈ F
            if Y ⊆ X⁺ then X⁺ := X⁺ ∪ Z   /* A-Regel */
    until X⁺ = X̄⁺
    return X⁺

MEMBER(F, X → Y):   /* Test auf X → Y ∈ F⁺ */
    return Y ⊆ CLOSURE(F, X)   /* P-Regel */
```

- Beispiel: $A \rightarrow C \in \{\underbrace{A \rightarrow B}_{f_1}, \underbrace{B \rightarrow C}_{f_2}\}^+$?

Überdeckungen

- F heißt äquivalent zu G
- oder: F Überdeckung von G; kurz: $F \equiv G$ falls $F^+ = G^+$
- d.h.:

$$\forall g \in G : g \in F^+ \ \wedge \ \forall f \in F : f \in G^+$$

- wichtige Entwurfsaufgabe: Finden einer Überdeckung, die
  - einerseits so wenig Attribute wie möglich in ihren funktionalen Abhängigkeiten und
  - andererseits möglichst wenig funktionale Abhängigkeiten insgesamt enthält
- verschiedene Formen von Überdeckung: nicht-redundant, reduziert, minimal, ringförmig

Reduktionsoperationen

- Ziel: Entfernen überflüssiger Attribute auf linker bzw. rechter Seite von FDs
- Linksreduktion: entfernt unwesentliche Attribute auf der linken Seite einer FD
- Rechtsreduktion: entsprechend auf der rechten Seite
- erw. Relationenschema $\mathcal{R} = (R, \mathcal{K})$, FD-Menge F über R, A ist ein Attribut aus R und $X \rightarrow Y$ eine FD aus F

A heißt unwesentlich in $X \rightarrow Y$ bzgl. F, wenn

$$[X = AZ, Z \neq X \implies (F - \{X \rightarrow Y\}) \cup \{Z \rightarrow Y\} \equiv F] \quad \vee$$

$$[Y = AW, W \neq Y \implies (F - \{X \rightarrow Y\}) \cup \{X \rightarrow W\} \equiv F]$$

- A kann also aus der FD $X \rightarrow Y$ entfernt werden, ohne dass sich die Hülle von F ändert

Reduktionsoperationen /2

- FD $X \rightarrow Y$ heißt linksreduziert, wenn kein Attribut in X unwesentlich ist.
- FD $X \rightarrow Y$ heißt rechtsreduziert, wenn kein Attribut in Y unwesentlich ist.

Schemaeigenschaften

- 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

Normalformen

- legen Eigenschaften von Relationenschemata fest
- verbieten bestimmte Kombinationen von funktionalen Abhängigkeiten in Relationen
- sollen Redundanzen und Anomalien vermeiden

Erste Normalform

- erlaubt nur atomare Attribute in den Relationenschemata, d.h. als Attributwerte sind Elemente von Standard-Datentypen wie `integer` oder `string` erlaubt, aber keine Konstruktoren wie `array` oder `set`
- Nicht in 1NF:

Weingut	**Anbaugebiet**	**Region**	**WName**
Ch. La Rose	Saint-Emilion	Bordeaux	La Rose Grand Cru
Creek	Barossa Valley	South Australia	Creek Shiraz, Pinot Noir
Helena	Napa Valley	Kalifornien	Zinfandel, Pinot Noir
Müller	Rheingau	Hessen	Riesling Reserve
Bighorn	Napa Valley	Kalifornien	Chardonnay

Erste Normalform /2

- in erster Normalform:

Weingut	Anbaugebiet	Region	WName
Ch. La Rose	Saint-Emilion	Bordeaux	La Rose Grand Cru
Creek	Barossa Valley	South Australia	Creek Shiraz
Creek	Barossa Valley	South Australia	Pinot Noir
Helena	Napa Valley	Kalifornien	Zinfandel
Helena	Napa Valley	Kalifornien	Pinot Noir
Müller	Rheingau	Hessen	Riesling Reserve
Bighorn	Napa Valley	Kalifornien	Chardonnay

Zweite Normalform

- partielle Abhängigkeit liegt vor, wenn ein Attribut funktional schon von einem Teil des Schlüssels abhängt

Name	Weingut	Farbe	Anbaugebiet	Region	Preis
La Rose Grand Cru	Ch. La Rose	Rot	Saint-Emilion	Bordeaux	39.00
Creek Shiraz	Creek	Rot	Barossa Valley	South Australia	7.99
Pinot Noir	Creek	Rot	Barossa Valley	South Australia	10.99
Zinfandel	Helena	Rot	Napa Valley	Kalifornien	5.99
Pinot Noir	Helena	Rot	Napa Valley	Kalifornien	19.99
Riesling Reserve	Müller	Weiß	Rheingau	Hessen	14.99
Chardonnay	Bighorn	Weiß	Napa Valley	Kalifornien	9.90

f_1: Name, Weingut $\rightarrow$ Preis

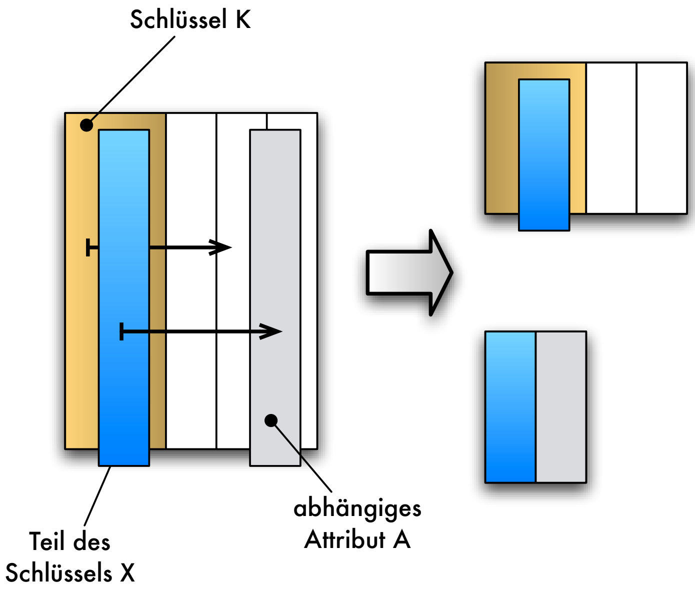
f_2: Name $\rightarrow$ Farbe

f_3: Weingut $\rightarrow$ Anbaugebiet, Region

f_4: Anbaugebiet $\rightarrow$ Region

- Zweite Normalform eliminiert derartige partielle Abhängigkeiten bei Nichtschlüsselattributen

Eliminierung partieller Abhängigkeiten



Zweite Normalform /2

- Beispielrelation in 2NF

  R1($\underline{\texttt{Name, Weingut}}$, Preis)

  R2($\underline{\texttt{Name}}$, Farbe)

  R3($\underline{\texttt{Weingut}}$, Anbaugebiet, Region)

Zweite Normalform /3

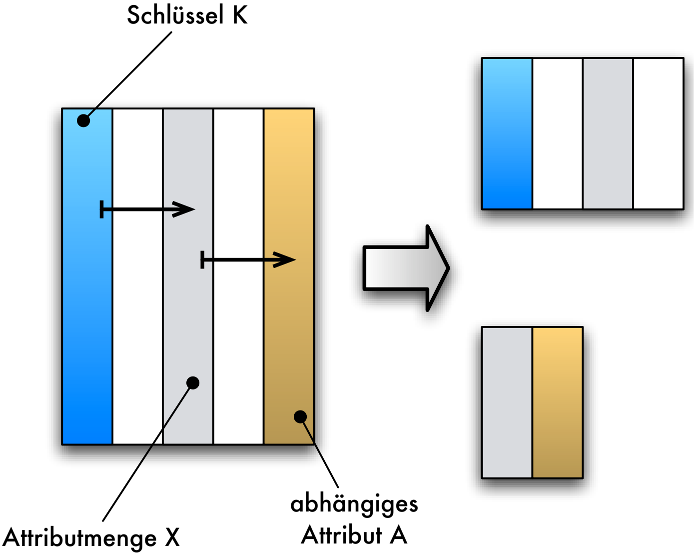
- Hinweis: partiell abhängiges Attribut stören nur, wenn es kein Primattribut ist
- 2NF formal: erweitertes Relationenschema $\mathcal{R} = (R, \mathcal{K})$, FD-Menge F über R

- Y **hängt partiell** von X bzgl. F ab, wenn die FD $X \rightarrow Y$ nicht linksreduziert ist
- Y **hängt voll** von X ab, wenn die FD $X \rightarrow Y$ linksreduziert ist
- $\mathcal{R}$ ist in 2NF, wenn $\mathcal{R}$ in 1NF ist und jedes Nicht-Primattribut von R voll von jedem Schlüssel von $\mathcal{R}$ abhängt

Dritte Normalform

- eliminiert (zusätzlich) transitive Abhängigkeiten
- etwa `Weingut` $\rightarrow$ `Anbaugebiet` und `Anbaugebiet` $\rightarrow$ `Region` in Relation auf Folie 5-30
- man beachte: 3NF betrachtet nur Nicht-Schlüsselattribute als Endpunkt transitiver Abhängigkeiten

Eliminierung transitiver Abhängigkeiten



Dritte Normalform /2

- transitive Abhängigkeit in `R3`, d.h. `R3` verletzt 3NF
- Beispielrelation in 3NF

  R3_1(<u>Weingut</u>, Anbaugebiet)

  R3_2(<u>Anbaugebiet</u>, Region)

Dritte Normalform: formal

- Relationenschema R, $X \subseteq R$ und F ist eine FD-Menge über R

- $A \in R$ heißt transitiv abhängig von X bezüglich F genau dann, wenn es ein $Y \subseteq R$ gibt mit $X \rightarrow Y, Y \not\rightarrow X, Y \rightarrow A, A \notin XY$
- erweitertes Relationenschema $\mathcal{R} = (R, \mathcal{K})$ ist in 3NF bezüglich F genau dann, wenn

$$\begin{aligned} \nexists A \in R: \quad & A \text{ ist Nicht-Primattribut in } R \\ & \wedge A \text{ transitiv abhängig von einem } K \in \mathcal{K} \text{ bezüglich } F. \end{aligned}$$

Boyce-Codd-Normalform

- Verschärfung der 3NF: Eliminierung transitiver Abhängigkeiten auch zwischen Primattributen

Name	Weingut	Händler	Preis
La Rose Grand Cru	Château La Rose	Weinkontor	39.90
Creek Shiraz	Creek	Wein.de	7.99
Pinot Noir	Creek	Wein.de	10.99
Zinfandel	Helena	GreatWines.com	5.99
Pinot Noir	Helena	GreatWines.com	19.99
Riesling Reserve	Müller	Weinkeller	19.99
Chardonnay	Bighorn	Wein-Dealer	9.90

- FDs:

$$\begin{aligned} \texttt{Name}, \texttt{Weingut} &\rightarrow \texttt{Preis} \\ \texttt{Weingut} &\rightarrow \texttt{Händler} \\ \texttt{Händler} &\rightarrow \texttt{Weingut} \end{aligned}$$

- Schlüsselkandidaten: { `Name`, `Weingut` } und { `Name`, `Händler` }
- in 3NF, nicht jedoch in BCNF

Boyce-Codd-Normalform /2

- erweitertes Relationenschema $\mathcal{R} = (R, \mathcal{K})$, FD-Menge F
- BCNF formal:

$$\nexists A \in R : A \text{ transitiv abhängig von einem } K \in \mathcal{K} \text{ bezüglich } F.$$

- Schema in BCNF:

  WEINE($\underline{\texttt{Name, Weingut}}$, Preis)
  WEINHANDEL($\underline{\texttt{Weingut}}$, Händler)

- BCNF kann jedoch Abhängigkeitstreue verletzen, daher oft nur bis 3NF

Minimalität

- Global Redundanzen vermeiden
- andere Kriterien (wie Normalformen) mit möglichst wenig Schemata erreichen
- Beispiel: Attributmenge ABC, FD-Menge $\{A \rightarrow B, B \rightarrow C\}$
- Datenbankschemata in dritter Normalform:

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

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

Redundanzen in S'

Schemaeigenschaften

Kennung	Schemaeigenschaft	Kurzcharakteristik
	1NF	nur atomare Attribute
	2NF	keine partielle Abhängigkeit eines Nicht-Primattributes von einem Schlüssel
S 1	3NF	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

Transformationseigenschaften

- Bei einer Zerlegung einer Relation in mehrere Relationen ist darauf zu achten, dass
  1. nur semantisch sinnvolle und konsistente Anwendungsdaten dargestellt (Abhängigkeitstreue) und
  2. alle Anwendungsdaten aus den Basisrelationen hergeleitet werden können (Verbundtreue)

Abhängigkeitstreue

- Abhängigkeitstreue: eine Menge von Abhängigkeiten kann äquivalent in eine zweite Menge von Abhängigkeiten transformiert werden
- spezieller: in die Menge der Schlüsselabhängigkeiten, da diese vom Datenbanksystem effizient überprüft werden kann
  - die Menge der Abhängigkeiten soll äquivalent zu der Menge der Schlüsselbedingungen im resultierenden Datenbankschema sein
  - Äquivalenz sichert zu, dass mit den Schlüsselabhängigkeiten semantisch genau die gleichen Integritätsbedingungen ausgedrückt werden wie mit den funktionalen oder anderen Abhängigkeiten vorher

Abhängigkeitstreue: Beispiel

- Zerlegung des Relationenschemas `WEINE` (Folie 5-31) in 3NF:

  R1(<u>Name, Weingut</u>, Preis)
  R2(<u>Name</u>, Farbe)
  R3_1(<u>Weingut</u>, Anbaugebiet)
  R3_2(<u>Anbaugebiet</u>, Region)

  mit Schlüsselabhängigkeiten

  $$\begin{array}{ll} \texttt{Name, Weingut} & \rightarrow \texttt{Preis} \\ \texttt{Name} & \rightarrow \texttt{Farbe} \\ \texttt{Weingut} & \rightarrow \texttt{Anbaugebiet} \\ \texttt{Anbaugebiet} & \rightarrow \texttt{Region} \end{array}$$

- äquivalent zu FDs $f_1 \ldots f_4$ (Folie 5-31) $\leadsto$ abhängigkeitstreu

Abhängigkeitstreue: Beispiel /2

- Postleitzahl-Struktur der Deutschen Post

  `PLZ (P), Ort (O), Strasse(S), Hausnummer(H)`

  und funktionalen Abhängigkeiten F

  $$\mathtt{OSH} \rightarrow \mathtt{P},\ \mathtt{P} \rightarrow \mathtt{O}$$

- für ein Datenbankschema S bestehend aus dem einzigen Relationenschema

  $$(\mathtt{OSHP}, \{\mathtt{OSH}\}),$$

  ist Menge der Schlüsselabhängigkeiten

  $$\{\ \mathtt{OSH} \rightarrow \mathtt{OSHP}\ \}$$

  nicht äquivalent zu F und somit S nicht abhängigkeitstreu

Abhängigkeitstreue formal

- lokal erweitertes Datenbankschema $S = \{(R_1, \mathcal{K}_1), \ldots, (R_p, \mathcal{K}_p)\}$; ein Menge F lokaler Abhängigkeiten

S charakterisiert vollständig F (oder: ist abhängigkeitstreu bezüglich F) genau dann, wenn

$$F \equiv \{K \rightarrow R \mid (R, \mathcal{K}) \in S, K \in \mathcal{K}\}$$

Verbundtreue

- zur Erfüllung des Kriteriums der Normalformen müssen Relationenschemata teilweise in kleinere Relationenschemata zerlegt werden
- für Beschränkung auf „sinnvolle" Zerlegungen gilt Forderung, dass die Originalrelation wieder aus den zerlegten Relationen mit dem natürlichen Verbund zurückgewonnen werden kann
  $\rightsquigarrow$ Verbundtreue

Verbundtreue: Beispiele

- Zerlegung des Relationenschemas $R = ABC$ in

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

- Dekomposition bei Vorliegen der Abhängigkeiten

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

ist nicht verbundtreu

- dagegen bei Vorliegen von

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

verbundtreu

Verbundtreue Dekomposition

- 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

Nicht verbundtreue Dekomposition

- 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

Verbundtreue formal

Die Dekomposition einer Attributmenge X in $X_1, \ldots, X_p$ mit $X = \bigcup_{i=1}^{p} X_i$ heißt verbundtreu ($\pi \bowtie$-treu, lossless) bezüglich einer Menge von Abhängigkeiten F über X genau dann, wenn

$$\forall r \in \mathbf{SAT}_X(F) : \pi_{X_1}(r) \bowtie \cdots \bowtie \pi_{X_p}(r) = r$$

gilt.

- einfaches Kriterium für Verbundtreue bei Dekomposition in 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 $X_1 \cap X_2 \rightarrow X_2 \in F^+$

Transformationseigenschaften

Kennung	**Transformationseigenschaft**	**Kurzcharakteristik**
T1	Abhängigkeitstreue	alle gegebenen Abhängigkeiten sind durch Schlüssel repräsentiert
T2	Verbundtreue	Originalrelationen können durch den Verbund der Basisrelationen wiedergewonnen werden

Entwurfsverfahren: Ziele

- Universum $\mathcal{U}$ und FD-Menge F gegeben
- lokal erweitertes Datenbankschema $S = \{(R_1, \mathcal{K}_1), \ldots, (R_p, \mathcal{K}_p)\}$ berechnen mit
  - $\boxed{\text{T}}\boxed{1}$: S charakterisiert vollständig F
  - $\boxed{\text{S}}\boxed{1}$: S ist in 3NF bezüglich F
  - $\boxed{\text{T}}\boxed{2}$: Dekomposition von $\mathcal{U}$ in $R_1, \ldots, R_p$ ist verbundtreu bezüglich F
  - $\boxed{\text{S}}\boxed{2}$: Minimalität, d.h.
    $\not\exists S' : S'$ erfüllt $\boxed{\text{T}}\boxed{1}$, $\boxed{\text{S}}\boxed{1}$, $\boxed{\text{T}}\boxed{2}$ und $|S'| < |S|$

Entwurfsverfahren: Beispiel

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

Dekomposition

- Geg.: initiales Universalrelationenschema $\mathcal{R} = (\mathcal{U}, \mathcal{K}(F))$ mit allen Attributen und einer von erfassten FDs F über R implizierten Schlüsselmenge
  - Attributmenge $\mathcal{U}$ und eine FD-Menge F
  - suche alle $K \rightarrow \mathcal{U}$ mit K minimal, für die $K \rightarrow \mathcal{U} \in F^+$ gilt ($\mathcal{K}(F)$)
- Ges.: Zerlegung in $D = \{\mathcal{R}_1, \mathcal{R}_2, \dots\}$ von 3NF-Relationenschemata

Dekomposition: Algorithmus

DECOMPOSE($\mathcal{R}$)
  Setze $D := \{\mathcal{R}\}$
  while $\mathcal{R}' \in D$, das 3NF nicht erfüllt
    / Finde Attribut A, das transitiv von K abhängig ist */*
    if Schlüssel K mit $K \rightarrow Y, Y \not\rightarrow K, Y \rightarrow A, A \notin KY$
    then
      / Zerlege Relationenschema R bzgl. A */*
      $R_1 := R - A$, $R_2 := YA$
      $\mathcal{R}_1 := (R_1, \mathcal{K})$, $\mathcal{R}_2 := (R_2, \mathcal{K}_2 = \{Y\})$
      $D := (D - \mathcal{R}') \cup \{\mathcal{R}_1\} \cup \{\mathcal{R}_2\}$
    end if
  end while
  return D

Dekomposition: Beispiel

- initiales Relationenschema $R = ABC$
- funktionale Abhängigkeiten $F = \{A \rightarrow B, B \rightarrow C\}$
- Schlüssel $K = A$

Dekomposition: Beispiel /2

- initiales Relationenschema R mit `Name`, `Weingut`, `Preis`, `Farbe`, `Anbaugebiet`, `Region`
- funktionale Abhängigkeiten

$$
\begin{array}{lll}
f_1: & \texttt{Name, Weingut} & \rightarrow \texttt{Preis} \\
f_2: & \texttt{Name, Weingut} & \rightarrow \texttt{Weingut} \\
f_3: & \texttt{Name, Weingut} & \rightarrow \texttt{Name} \\
f_4: & \texttt{Name} & \rightarrow \texttt{Farbe} \\
f_5: & \texttt{Weingut} & \rightarrow \texttt{Anbaugebiet, Region} \\
f_6: & \texttt{Anbaugebiet} & \rightarrow \texttt{Region}
\end{array}
$$

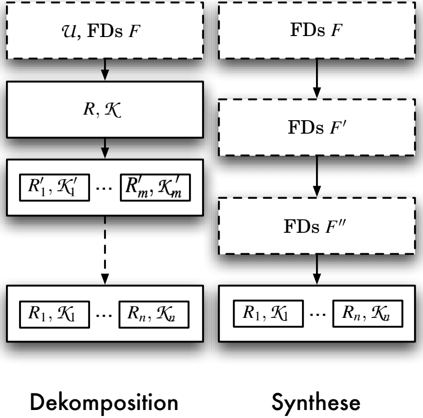
Dekomposition: Bewertung

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

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

Vergleich Dekomposition — Synthese



Syntheseverfahren: Ablauf

- Geg.: Relationschema R mit FDs F
- Ges.: verlustfreie und abhängigkeitstreue Zerlegung in $R_1, \ldots R_n$, wobei alle R_i in 3NF sind
- Algorithmus:

SYNTHESIZE(F):
  $\hat{F}$:= **MINIMALCOVER**(F) */* Bestimme minimale Überdeckung */*
  Bilde Äquivalenzklassen C_i von FDs aus $\hat{F}$ mit gleichen oder
    äquivalenten linken Seiten, d.h. $C_i = \{X_i \rightarrow A_{i1}, X_i \rightarrow A_{i2}, \ldots\}$
  Bilde zu jeder Äquivalenzklasse C_i ein Schema der Form
    $R_{C_i} = \{X_i \cup \{A_{i1}\} \cup \{A_{i2}\} \cup \ldots\}$
  if keines der Schemata R_{C_i} enthält einen Schlüssel von R
  then erzeuge weiteres Relationenschema R_K mit Attributen
    aus R, die Schlüssel bilden
  return $\{R_K, R_{C_1}, R_{C_2}, \ldots\}$

Minimale Überdeckung

- Auswahl der kleinsten aller nicht-redundanten Überdeckungen
- FD-Menge F heißt minimal gdw.

$$\forall F' \left[F' \equiv F \Rightarrow |F| \leq |F'| \right]$$

- Bestimmung etwa durch reduzierte (kanonische) Überdeckung

Reduzierte Überdeckung

```
REDUCEDCOVER(F):
   forall FD X → Y ∈ F  /* Linksreduktion */
      forall A ∈ X  /* A unwesentlich ? */
         if Y ⊆ CLOSURE(F, X − {A})
         then ersetze X → Y durch (X − A) → Y in F
      end for
   end for

   forall verbleibende FD X → Y ∈ F  /* Rechtsreduktion */
      forall B ∈ Y  /* B unwesentlich ? */
         if B ⊆ CLOSURE(F − {X → Y} ∪ {X → (Y − B)}, X)
         then ersetze X → Y durch X → (Y − B)
      end for
   end for

   Eliminiere FDs der Form X → ∅
   Vereinige FDs der Form X → Y1, X → Y2, ... zu X → Y1 Y2 ...
return resultierende FDs
```

Reduzierte Überdeckung: Beispiel

- Geg.: FD-Menge

$$F = \{f_1 : A \rightarrow B, f_2 : AB \rightarrow C, f_3 : A \rightarrow C, f_4 : B \rightarrow A, f_5 : C \rightarrow E\}$$

1. Linksreduktion: bei FD f_2 Attribut A streichen, da $C \subseteq \textsc{Closure}(F, \{A\})$ (wegen f_3)
2. Rechtsreduktion: FD f_3 durch $A \rightarrow \{\}$ ersetzt, da $C \subseteq \textsc{Closure}(\{A \rightarrow B, B \rightarrow C, A \rightarrow \{\}, B \rightarrow A, C \rightarrow E\}, \{A\})$
3. Streichen von $A \rightarrow \{\}$

- Ergebnis:

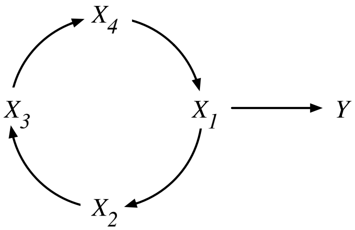
$$\textsc{ReducedCover}(F) = \{A \rightarrow B, B \rightarrow C, B \rightarrow A, C \rightarrow E\}$$

Äquivalenzklassen

- Klasse von FDs, die gleiche oder äquivalente linke Seiten besitzen
- linke Seiten sind äquivalent, wenn sie sich gegenseitig funktional bestimmen
- Relationenschema R mit $X_i, Y \subset R$, FD-Menge
  $X_i \rightarrow X_j$ und $X_i \rightarrow Y$ mit $1 \leq i,j \leq n$ kann durch

  $$(X_1, X_2, \ldots, X_n) \rightarrow Y$$

  dargestellt werden



Äquivalenzklassen: Beispiel

- FD-Menge

$$F = \{A \rightarrow B, AB \rightarrow C, A \rightarrow C, B \rightarrow A, C \rightarrow E\}$$

- minimale Überdeckung

$$\hat{F} = \{A \rightarrow B, B \rightarrow C, B \rightarrow A, C \rightarrow E\}$$

- Zusammenfassung zu Äquivalenzklassen

$$\begin{aligned} C_1 &= \{A \rightarrow B, B \rightarrow C, B \rightarrow A\} \\ C_2 &= \{C \rightarrow E\} \end{aligned}$$

- Syntheseergebnis

$$(ABC, \{\{A\}, \{B\}\}), (CE, \{C\})$$

Erreichung der Verbundtreue

- Erreichen der Verbundtreue durch einfachen „Trick“:
  - Erweitern der Original-FD-Menge F um $\mathcal{U} \rightarrow \delta$ um Dummy-Attribut δ
  - δ wird nach Synthese entfernt
- Beispiel: $\{A \rightarrow B, C \rightarrow E\}$
  - Syntheseergebnis $(AB, \{A\}), (CE, \{C\})$ ist nicht verbundtreu, da Universalschlüssel in keinem Schema enthalten ist
  - Dummy-FD $ABCE \rightarrow \delta$; reduziert auf $AC \rightarrow \delta$
  - liefert drittes Relationenschema

$$(AC, \{AC\})$$

Synthese: Beispiel

- Relationenschema und FD-Menge von Folie 5-59
- Ablauf
  1. minimale Überdeckung: Entfernen von f_2, f_3 sowie `Region` in f_5
  2. Äquivalenzklassen:

$$
\begin{aligned}
C_1 &= \{\texttt{Name}, \texttt{Weingut} \rightarrow \texttt{Preis}\} \\
C_2 &= \{\texttt{Name} \rightarrow \texttt{Farbe}\} \\
C_3 &= \{\texttt{Weingut} \rightarrow \texttt{Anbaugebiet}\} \\
C_4 &= \{\texttt{Anbaugebiet} \rightarrow \texttt{Region}\}
\end{aligned}
$$

  3. Ableitung der Relationenschemata

Weitere Abhängigkeiten

- Mehrwertige Abhängigkeit (kurz: MVD)
  - innerhalb einer Relation r wird einem Attributwert von X eine Menge von Y-Werten zugeordnet, unabhängig von den Werten der restlichen Attribute $\leadsto$ Vierte Normalform
- Verbundabhängigkeit (kurz: JD)
  - kann R ohne Informationsverlust in $R_1, \ldots, R_p$ aufgetrennt werden: $\bowtie [R_1, \ldots, R_p]$
- Inklusionsabhängigkeit (kurz: IND)
  - auf der rechten Seite einer Fremdschlüsselabhängigkeit nicht unbedingt den Primärschlüssel einer Relation

Mehrwertige Abhängigkeiten

- Folge der 1NF
- Mehrwertige Abhängigkeiten erzeugen Redundanz:

WEIN_EMPFEHLUNG

WName	Jahrgang	Gericht
Chardonnay	2002	Geflügel
Chardonnay	2002	Fisch
Chardonnay	2003	Fisch
Chardonnay	2003	Geflügel
Shiraz	2003	Wild
Shiraz	2003	Lamm
Shiraz	2004	Wild
Shiraz	2004	Lamm

  - eine (oder mehrere) Gruppe von Attributwerten ist von einem Schlüssel bestimmt, unabhängig von anderen Attributen
  - hier: Menge von Jahrgängen plus Menge von Gerichten

$$\texttt{WName} \to\!\!\to \texttt{Jahrgang}, \texttt{WName} \to\!\!\to \texttt{Gericht}$$

  - Resultat: Redundanz durch Bildung aller Kombinationen

Mehrwertige Abhängigkeiten formal

- Relation $r(R)$ mit $X, Y \subseteq R$, $Z := R - (X \cup Y)$ genügt der MVD $X \rightarrow\!\!\!\rightarrow Y$ gdw.

$$
\begin{aligned}
\forall t_1, t_2 \in r: \quad & [(t_1 \neq t_2 \wedge t_1(X) = t_2(X)) \\
\Longrightarrow \quad & \exists t_3 \in r : t_3(X) = t_1(X) \wedge t_3(Y) = t_1(Y) \wedge \\
& t_3(Z) = t_2(Z)]
\end{aligned}
$$

- Relation $r(R)$ mit $R = XYZ$ und $X \rightarrow\!\!\!\rightarrow Y$:
  - wenn $(x_1, y_1, z_1) \in r$ und $(x_1, y_2, z_2) \in r$
  - dann auch: $(x_1, y_1, z_2) \in r$ und $(x_1, y_2, z_1) \in r$
- Bsp.: wegen ('Chardonnay', 2002, 'Geflügel') und ('Chardonnay', 2003, 'Fisch') müssen auch ('Chardonnay', 2002, 'Fisch') und ('Chardonnay', 2003, 'Geflügel') enthalten sein

Mehrwertige Abhängigkeiten und 4NF

- wünschenswerte Schemaeigenschaft bei Vorliegen von MVDs: vierte Normalform
- fordert die Beseitigung derartiger Redundanzen: keine zwei MVDs zwischen Attributen einer Relation
- Beispiel von Folie 5-72 verletzt diese Forderung
- Prinzip
  - Elimination der rechten Seite einer der beiden mehrwertigen Abhängigkeiten,
  - linke Seite mit dieser rechten Seite in neue Relation kopiert

Vierte Normalform

WEIN_JAHR

WName	Jahrgang
Chardonnay	2002
Chardonnay	2003
Shiraz	2003
Shiraz	2004

WEIN_GERICHT

WName	Gericht
Chardonnay	Geflügel
Chardonnay	Fisch
Shiraz	Wild
Shiraz	Lamm

Vierte Normalform formal

- Relationenschema R mit $X, Y \subseteq R$, MVD-Menge M über R
- MVD $X \twoheadrightarrow Y$ heißt trivial genau dann, wenn $Y \subseteq X$ oder $X \cup Y = R$

erweitertes Relationenschema $\mathcal{R} = (R, \mathcal{K})$ ist in vierter Normalform (4NF) bezüglich M genau dann, wenn für alle $X \twoheadrightarrow Y \in M^+$ gilt:

$$X \twoheadrightarrow Y \text{ ist trivial oder } X \supseteq K \text{ für ein } K \in \mathcal{K}.$$

Nichttriviale MVDs

- Erweiterung der Relation WEIN_JAHR von Folie 5-75 um Attribute Farbe und Restsüße
- MVD WName $\rightarrow\!\!\!\rightarrow$ Jahrgang ist nicht mehr trivial
- Zerlegung:

  WEIN_JAHR1(<u>WName, Jahrgang</u>)

  WEIN_JAHR2(<u>WName</u>, Farbe, Restsüße)

Zusammenfassung

- funktionale Abhängigkeiten
- Normalformen (1NF - 3NF, BCNF)
- Abhängigkeitstreue und Verbundtreue
- Entwurfsverfahren
- mehrwertige Abhängigkeiten