

DATENBANKTECHNOLOGIEN SS 2022

Datenbankdesign

Prof. Dr. Eva-Maria Iwer
Mai/ Juni 2022

 Hochschule RheinMain

SCHLECHTES
DATENBANKDESIGN

08.05.2022 Dr. Carsten Lier - Hochschule RheinMain 38

 Hochschule RheinMain

ENTITY RELATIONSHIP
MODEL
Vom Entwurf zum Design

08.05.2022 Dr. Carsten Lier - Hochschule RheinMain 40

 Hochschule RheinMain

FUNKTIONALE
ABHÄNGIGKEITEN

08.05.2022 Dr. Carsten Lier - Hochschule RheinMain 46

 Hochschule RheinMain


SCHLÜSSEL

08.05.2022 Dr. Carsten Lier - Hochschule RheinMain 54

 Hochschule RheinMain

DEKOMPPOSITION VON
RELATIONEN

08.05.2022 Dr. Carsten Lier - Hochschule RheinMain 76

 Hochschule RheinMain

NORMALFORMEN
1NF, 2NF, 3NF

• 1NF: Alle Attribute müssen einen atomaren Wertebereich (Domain) haben

Bsp nicht in 1NF

Prof	Lehrveranstaltungen
Iwaner	(DB, SWT, DB2)
Meier	(DB, Prog1, Prog2)

Bsp in 1NF

Prof	Lehrveranstaltungen
Iwaner	DB
Iwaner	SWT
Iwaner	DB2
Meier	DB
Meier	Prog1
Meier	Prog2

08.05.2022 Dr. Carsten Lier - Hochschule RheinMain 77

 Hochschule RheinMain

BCNF – EINE ERWEITERUNG
DER NORMALFORM

08.05.2022 Dr. Carsten Lier - Hochschule RheinMain 81

SCHLECHTES DATENBANKDESIGN

SCHLECHTE RELATIONENSCHEMATA

Und ihre Probleme

- ProfVorl(PersNr, Name, Rang, Raum, VorlNr, Titel, CP)

PersNr	Name	Rang	Raum	VorlNr	Titel	CP
123	Iwer	W2	C011	7051	Datenbankt echnology	5
123	Iwer	W2	C011	3111	Softwaretec hnik	10
123	Iwer	W2	C011	4111	Softwaretec hnik	5
...
234	Pratz	C3	C011	1111	PM	5
256	Laftum	W4	C008	1234	Einführung in C	5

SCHLECHTE RELATIONENSCHEMATA

Updateanomalien

- Wenn ein Prof umzieht, muss der neue Raum in der DB gespeichert werden
- Die Information ist redundant
- Damit muss es in jeder entsprechenden Zeile gespeichert werden
- Probleme:
 - Mehr Speicherbedarf
 - Leistungseinbußen bei Änderungen

PersNr	Name	Rang	Raum	VorlNr	Titel	CP
123	Iwer	W2	C011	7051	Datenba nktechnol ogy	5
123	Iwer	W2	C011	3111	Software technik	10
123	Iwer	W2	C011	4111	Software technik	5
...
234	Pratz	C3	C011	1111	PM	5
256	Laftum	W4	C008	1234	Einführung in C	5

SCHLECHTE RELATIONENSCHEMATA

Einfügenaomalien

- Will man die Daten für neu berufene Professoren eintragen, die noch keine Vorlesung halten, so geht dies nur, indem man viele Attribute auf NULL setzt.
- Will man eine Vorlesung eintragen, wo es noch keinen Prof gibt, ist das je nach Schlüssel nicht möglich.

PersNr	Name	Rang	Raum	VorINr	Titel	CP
123	Iwer	W2	C011	7051	Datenba nktechnol ogy	5
123	Iwer	W2	C011	3111	Software technik	10
123	Iwer	W2	C011	4111	Software technik	5
...
234	Pratz	C3	C011	1111	PM	5
256	Laftum	W4	C008	1234	Einführung in C	5

SCHLECHTE RELATIONENSHEMATA

Löschanomalien

- Löschen des Profs Iwer ... Verlust von Daten zu den entsprechenden Vorlesungen
- Setzen auf NULL wäre Lösung
- Wenn Vorlesung von 2 Profs gehalten wird, könnte man den Prof löschen

PersNr	Name	Rang	Raum	VorlNr	Titel	CP
123	Iwer	W2	C011	7051	Datenba nktechnol ogy	5
123	Iwer	W2	C011	3111	Software technik	10
123	Iwer	W2	C011	4111	Software technik	5
...
234	Pratz	C3	C011	1111	PM	5
256	Laftum	W4	C008	1234	Einführung in C	5

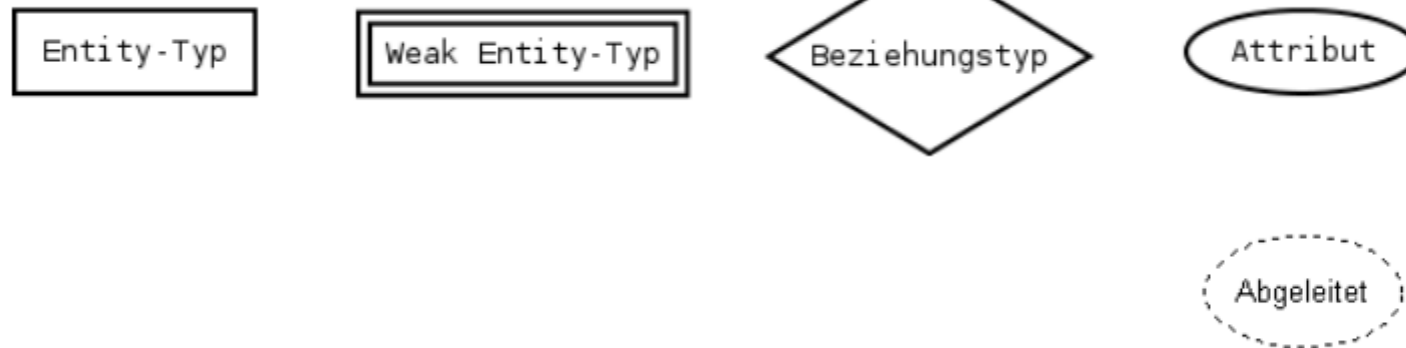
WIR BRAUCHEN EIN GUTES DESIGN.

ENTITY RELATIONSHIP MODEL

Vom Entwurf zum Design

HAUPTELEMENTE

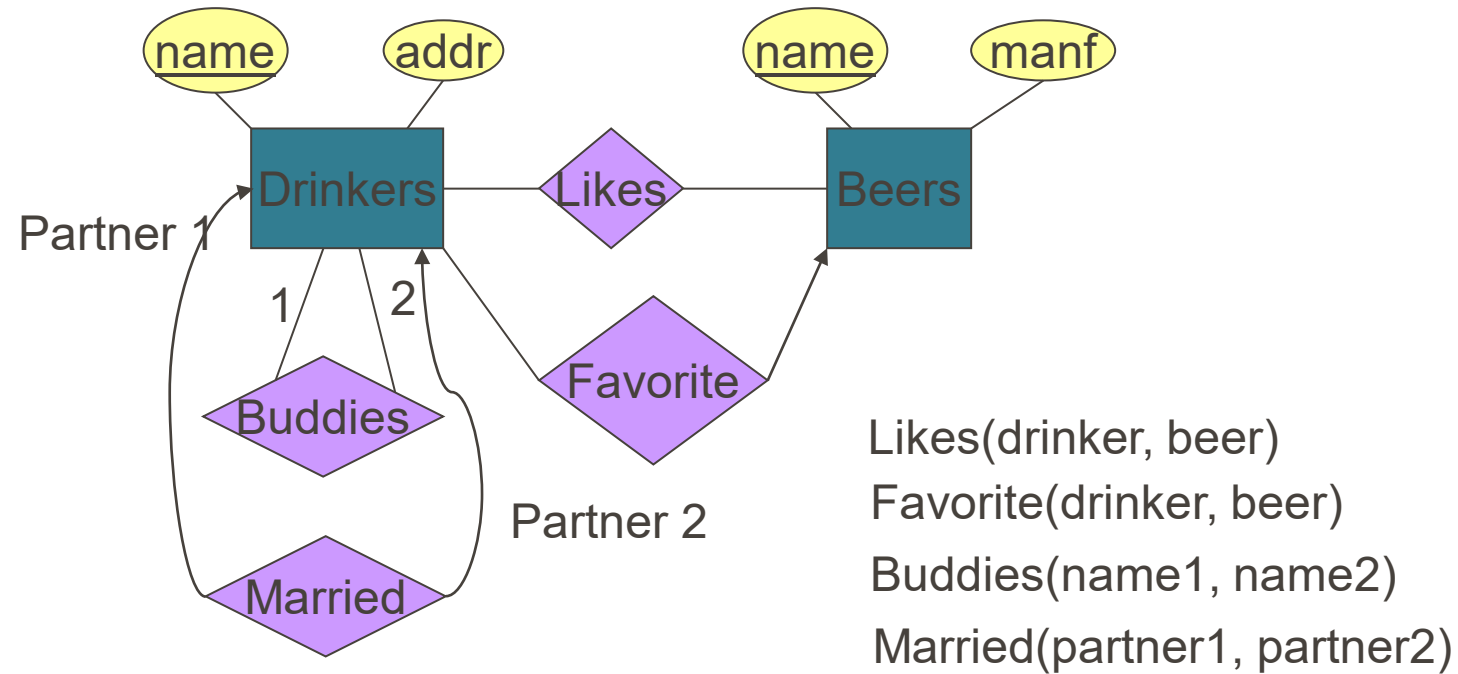
Kurze Wiederholung



VOM ER-DIAGRAM ZU RELATIONEN

- Entity Set -> Relation
 - Attribute -> Attribute.
- Beziehungen -> Relation mit folgenden Attributen:
 - Die Schlüsselattribute der verbundenen Entity Sets
 - Die Beziehungsattribute

Beziehungen werden zu Relation



FUNKTIONALE ABHÄNGIGKEITEN

DER WEG ZUM GUTEN DESIGN

Wenn ERM nicht mehr reichen

Funktionale Abhängigkeiten

Schlüssel

Tabellendesign

FUNKTIONALE ABHÄNGIGKEITEN

Definition

- Eine funktionale Abhängigkeit (engl. functional dependency FD) stellt eine Bedingungen an die möglichen gültigen Ausprägungen des Datenbankschemas dar. Eine funktionale Abhängigkeit wird wie folgt dargestellt:

$$A \rightarrow B$$

- A und B repräsentieren jeweils Mengen von Attributen
- Wenn $A \rightarrow B$ auf dem Relationenschema R definiert ist, dann heißt es, dass A und B Teilmengen von R sind.
- Beispiel 1: $R = \{x,y,z\}$, $A = \{x\}$, $B = \{y\}$
- Beispiel 2: $R = \{\text{DrachenID}, \text{DracheName}, \text{Geschlecht}\}$, $A = \{\text{DrachenID}\}$, $B = \{\text{DracheName}, \text{Geschlecht}\}$

FUNKTIONALE ABHÄNGIGKEITEN

Bedeutung

- $A \dashrightarrow B$
- Für alle Paare von Tupeln
 $r, t \in R$ mit $r.A = t.A$ muss auch gelten $r.B = t.B$
- $r.A = t.A$ ist eine Kurzform von $\forall X \in A: r.A = t.A$
- In andere Worten: Wenn $A \rightarrow B$ gilt, dann gilt für zwei Tupel mit den gleichen Werten in A, dass sie auch die gleichen Werte in B haben.
- Wir sagen, dass die A-Werte die B-Werte funktional bestimmen. Bzw B ist funktional abhängig von A.
- A ist die Determinante von B.

AUFSPALTUNG REGEL

Der rechten Seite



$X \rightarrow A_1 A_2 \dots A_n$ ist wahr für R genau dann wenn jede FD $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$ für R gilt.



Beispiel: $A \rightarrow BC$ ist das Gleiche wie $A \rightarrow B$ und $A \rightarrow C$.



Es gibt keine Aufspaltung der linken Seite



Wir probieren FD immer mit Singeltons (ein Attribut) darzustellen.

Drinkers(name, addr, beersLiked, manf, favBeer)



Abgeleitet FD's:

name \rightarrow addr
name \rightarrow favBeer

beersLiked \rightarrow manf

BEISPIEL 1

name	addr	beersLiked	manf	favBeer
Janeway	Voyager	Bud	A.B.	WickedAle
Janeway	Voyager	WickedAle	Pete's	WickedAle
Spock	Enterprise	Bud	A.B.	Bud

Weil name -> addr

Weil name -> favBeer

Weil beersLiked -> manf

- Funktionale Abhängigkeiten sind eine semantische Konsistenzbedingung, die sich aus der jeweiligen Anwendungssemantik und nicht aus der derzeitigen zufälligen Relationenausprägung ergeben
- Funktionale Abhängigkeiten stellen Konsistenzbedingungen dar, die zu allen Zeiten in jedem Datenbankzustand eingehalten werden müssen.

FUNKTIONALE ABHÄNGIGKEITEN

Konventionen zur Notation

- Formal präzise Notation: $\{C,D\} \rightarrow \{A\}$
- Oft genutzt: $C, D \rightarrow A$ oder auch $CD \rightarrow A$

FUNKTIONALE ABHÄNGIGKEIT

Algorithmus zur Überprüfung ob $A \rightarrow B$ in R gültig ist

- Eingabe: Relation R und eine FD $A \rightarrow B$
- Ausgabe: ja, falls $A \rightarrow B$ in R erfüllt ist, sonst nein
- Einhaltung ($R, A \rightarrow B$)
 - Sortiere R nach A -Werten
 - Falls alle Gruppen bestehend aus Tupeln mit Gleichen A -Werten auch gleiche B -Werte aufweisen:
Ausgabe ja, sonst Ausgabe nein

ÜBUNGEN

Funktionelle Abhängigkeiten finden

ZEIT FÜR GRUPPENARBEIT

- Sie haben 30 Minuten Zeit
- Ein Team stellt danach eine Lösung vor.

A	B	C	D
A4	B2	C4	D3
A1	B1	C1	D1
A1	B1	C1	D2
A2	B2	C3	D2
A3	B2	C4	D3

1. Finden Sie alle gültigen FD aus der obigen Abbildung der Relation R.
2. Zeigen Sie, dass FDs der Art $A \rightarrow B$ mit $B \subseteq A$ trivial sind.
3. Zeigen Sie, dass die folgenden FDs nicht valide Regeln sind indem Sie Beispiele für eine Relation zeigen, in der die Wenn-Regel stimmt, aber nicht die Dann
 1. Wenn $A \rightarrow B$ dann $B \rightarrow A$
 2. Wenn $AB \rightarrow C$ und $A \rightarrow C$, dann $B \rightarrow A$
 3. Wenn $AB \rightarrow C$ dann $A \rightarrow C$ oder $B \rightarrow C$

SHOW YOUR WORK

SCHLÜSSEL

SUPERSCHLÜSSEL

In der Relation R ist K ein Superschlüssel, falls gilt:

$$K \rightarrow R, \text{ wenn } K \subseteq R$$

D.h. K bestimmt all anderen Attributwerte innerhalb der Relation R .

Wir nennen K in diesem Fall Superschlüssel, weil noch nichts darüber ausgesagt ist, ob K eine minimale Menge von Attributen enthält.

Eine Menge aus allen Attributen aus R ist damit auch ein Superschlüssel.

SUPERSCHLÜSSEL VS SCHLÜSSEL

- K ist ein Superschlüssel der Relation R , wenn
 K functional determiniert alle Attribute von R .
- K ist ein Schlüssel von R wenn
 K ein Superschlüssel ist, jedoch keine gültige Teilmenge von K ein Superschlüssel ist.

BEISPIEL 1

Drinkers(name, addr, beersLiked, manf, favBeer)

- {name, beersLiked} is ein Superschlüssel
 - name -> addr favBeer
 - beersLiked -> manf

- $\{\text{name, beersLiked}\}$ ist ein Schlüssel, da weder $\{\text{name}\}$ als auch $\{\text{beersLiked}\}$ ein Superschlüssel ist.
- Es gibt keine anderen Schlüssel, jedoch eine Menge Superschlüssel:
 - Jeder Supermenge von $\{\text{name, beersLiked}\}$.

VOLLE FUNKTIONALE ABHÄNGIGKEIT

- B ist voll funktional abhängig von A ($A \xrightarrow{o} B$) falls folgende Kriterien gelten:
 1. $A \rightarrow B$, d.h. B ist funktional abhängig von A und
 2. A kann nicht mehr „verkleinert“ werden, d.h. $\forall C \in A : A - \{A\} \rightarrow B$

Es kann also kein Attribut mehr aus A entfernt werden, ohne die FD zu „zerstören“

KANDIDATENSCHLÜSSEL

- Fall $A \xrightarrow{o} R$ gilt, bezeichnet man A als Kandidatenschlüssel von R.
- Im Allgemeinen wird einer der Kandidatenschlüssel als sogenannter Primärschlüssel ausgewählt.

HÜLLE (COVER)

- Im Allgemeinen sind wir bei einer gegebenen Menge F von FDs daran interessiert, die Menge F^+ aller daraus herleitbaren funktionalen Abhängigkeiten zu bestimmen.
- Die Menge F^+ bezeichnet man als Hülle (engl. Cover) der Menge F .
- Die Hülle einer Menge von FDs kann durch Anwendung von Herleitungsregeln – auch Inferenzregeln genannt – bestimmt werden.
- Für die Herleitung der vollständigen Hülle reichen die drei nachfolgend aufgeführten Armstrong-Axiome als Inferenzregeln aus

$\alpha, \beta, \gamma, \delta$ sind Teilmengen von R

1. Reflexivität: Falls β eine Teilmenge von α ist ($\beta \subseteq \alpha$) dann gilt immer $\alpha \rightarrow \beta$
2. Verstärkung: Falls $\alpha \rightarrow \beta$ gilt, dann gilt auch $\alpha\gamma \rightarrow \beta\gamma$. Hierbei stehe z.B. $\alpha\gamma$ für $\alpha \cup \gamma$.
3. Transitivität: Falls $\alpha \rightarrow \beta$ und $\beta \rightarrow \gamma$ gilt, dann gilt auch $\alpha \rightarrow \gamma$

Die Armstrong-Axiome sind korrekt und vollständig.

Die Korrektheit der Axiome besagt, dass sich mit Hilfe der Armstrong-Axiome aus einer Menge F von FDs nur solche weitere FDs ableiten lassen, die von jeder Relationsausprägung erfüllt sind, für die F erfüllt ist.

Die Vollständig der Axiome besagt, dass sich alle FDs ableiten lassen, die durch F logisch impliziert werden.

- Vereinigungsregel:
Falls $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$ gilt, dann gilt auch $\alpha \rightarrow \beta\gamma$.
- Dekompositionsregel:
Falls $\alpha \rightarrow \beta\gamma$ gilt, dann gelten auch $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$.

- Pseudotransitivitätsregel:
Falls $\alpha \rightarrow \beta$ und $\gamma\beta \rightarrow \delta$ gilt, dann gilt auch $\alpha\gamma \rightarrow \delta$

ÜBUNGEN

Bestimmung von Schlüsseln

ZEIT FÜR GRUPPENARBEIT

- Sie haben 30 Minuten Zeit
- Ein Team stellt danach eine Lösung vor.

Gegeben ist das Relationsschema

ProfAdr{PersNr, Name, Rang, Raum, Ort, Strasse, PLZ, Vorwahl, Bland, Landesregierung}

1. Bestimmen Sie alle funktionalen Abhängigkeiten.
2. Bestimmen Sie für alle Singeltons der Attribute die Hülle und für {Pnr, Ort}, {Pnr, Vorwahl}
3. Ausgehend von Aufgabe 2, was sind mögliche Kandidatenschlüssel?

SHOW YOUR WORK

Was sind Ihre Lösungen?

KANONISCHE ÜBERDECKUNG

- Es gibt viele unterschiedliche äquivalente Mengen von funktionalen Abhängigkeiten
- Zwei Mengen F und G von funktionalen Abhängigkeiten heißen genau dann äquivalent ($F \equiv G$), wenn ihre Hüllen gleich sind, d.h. $F^+ = G^+$
- Nachteilig wirkt sich eine große, redundante Menge von funktionalen Abhängigkeiten im Rahmen einer Konsistenzüberprüfung bei Datenbankmodifikationen aus
- Gesucht ist die kleinstmögliche noch äquivalente Menge von FDs interessiert.

- Zu einer gegebenen Menge F von FDs nennt man F_c eine kanonische Überdeckung, falls folgende drei Eigenschaften erfüllt sind:

1. $F_c \equiv F$, d.h. $F_c^+ = F^+$
2. In F_c existieren keine FDs $\alpha \rightarrow \beta$, bei denen α oder β überflüssige Attribute enthalten. D.h. es muss folgendes gelten:

$$\forall A \in \alpha: \left(F_c - (\alpha \rightarrow \beta) \cup ((\alpha - A) \rightarrow \beta) \right)! \equiv F_c$$
$$\forall B \in \beta: \left(F_c - (\alpha \rightarrow \beta) \cup (\alpha \rightarrow (\beta - B)) \right)! \equiv F_c$$

3. Jede linke Seite einer funktionalen Abhängigkeit in F_c ist einzigartig. Dies kann durch sukzessive Anwendung der Vereinigungsregel auf FDs der Art $\alpha \rightarrow \beta$ und $\alpha \rightarrow \gamma$ erzielt werden, so dass die beiden FDs durch $\alpha \rightarrow \beta\gamma$ ersetzt werden.

1. Führe jede FD $\alpha \rightarrow \beta \in F$ die Linksreduktion durch, also:
Überprüfe für alle $A \in \alpha$, ob A überflüssig ist, d.h. ob $\beta \subseteq \text{AttrHülle}(F, \alpha - A)$ gilt. Falls dies der Fall ist, ersetze $\alpha \rightarrow \beta$ durch $(\alpha - A) \rightarrow \beta$.
2. Führe für jede (verbliebene) FD $\alpha \rightarrow \beta$ die Rechtsreduktion durch, also:
Überprüfe für alle $B \in \beta$, ob $B \subseteq \text{AttrHülle}(F - (\alpha \rightarrow \beta) \cup (\alpha \rightarrow (\beta - B)), \alpha)$ gilt. In diesem Fall ist B auf der rechten Seite überflüssig und kann eliminiert werden, d.h. $\alpha \rightarrow \beta$ wird durch $\alpha \rightarrow (\beta - B)$ ersetzt.
3. Entferne die FDs der Form $\alpha \rightarrow \emptyset$, die im 2. Schritt möglicherweise entstanden sind.
4. Fasse mittels Vereinigungsregel FDs der Form $\alpha \rightarrow \beta_1, \beta_2, \dots, \beta_n$ zusammen, so dass $\alpha \rightarrow (\beta_1 \cup \beta_2 \cup \dots \cup \beta_n)$ verbleibt.

ÜBUNGEN

Auslesen von Informationen

ZEIT FÜR GRUPPENARBEIT

- Sie haben 30 Minuten Zeit
- Ein Team stellt danach eine Lösung vor.

Gegen ist das abstrakte Relationsschema $R=\{A, B, C, D, E, F\}$ mit den funktionalen Abhängigkeiten

$A \rightarrow BC$

$C \rightarrow DA$

$E \rightarrow ABC$

$F \rightarrow CD$

$CD \rightarrow BEF$

1. Bestimmen Sie die kanonische Überdeckung.
2. Berechnen Sie die Attributhülle von A.
3. Bestimmen Sie mind. 2 Kandidatenschlüssel

SHOW YOUR WORK

Was sind Ihre Lösungen?

DEKOMPOSITION VON RELATIONEN

NORMALFORMEN

1NF, 2NF, 3NF

- 1NF: Alle Attribute müssen einen atomaren Wertebereich (Domain) haben

Bsp nicht in 1NF

Prof	Lehrveranstaltungen
Iwer	{DB, SWT, DB2}
Meier	{DB, Prog1, Progr2}

Bsp in 1NF

Prof	Lehrveranstaltungen
Iwer	DB
Iwer	SWT
Iwer	DB2
Meier	DB
Meier	Prog1
Meier	Prog2

NORMALFORMEN

NFNF bzw. NF²

- Relationen, die nicht in 1. NF sind
- NF² ist Basis für objektrelationale Datenbanken

NORMALFORMEN

1NF – ja oder nein

VerkNr	VerkName	PLZ	VerkAdresse	Produktname	Umsatz
V1	Meier	80075	München	Waschmaschine	11000
V1	Meier	80075	München	Herd	5000
V1	Meier	80075	München	Kühlschrank	1000
V2	Schneider	70038	Stuttgart	Herd	4000
V2	Schneider	70038	Stuttgart	Kühlschrank	3000
V3	Müller	50083	Köln	Staubsauger	1000

NORMALFORMEN

1NF, 2NF, 3NF

- 2NF: 1NF ist erfüllt und eine Relation R mit zugehörigen FDs F ist in zweiter Normalform, falls jedes Nichtschlüssel-Attribut voll funktional abhängig ist von jedem Kandidatenschlüssel der Relation.

Bsp nicht in 2NF

MatrNr	VorlNr	Name	Semester
26098	1001	Albert	4
32093	1001	Berndum	8
24002	1002	Christal	6
24002	1001	Berndum	8

MatrNr → *Name, Semester*

NORMALFORMEN

1NF, 2NF, 3NF

- Einfügeanomalie, Updateanomalie, Löschanomalie

Bsp in 2NF

MatrNr	Name	Semester
26098	Albert	4
32093	Berndum	8
24002	Christal	6
24002	Berndum	8

MatrNr	VorlNr
26098	1001
32093	1001
24002	1002
24002	1001

NORMALFORMEN

1NF, 2NF, 3NF

- 3NF: 1NF und 2NF erfüllt UND für jede für R geltende funktionale Abhängigkeit der Form $\alpha \rightarrow B$ mit $\alpha \subseteq R$ und $B \in R$ mindestens eine von drei Bedingungen gilt:
 1. $B \in \alpha$, d.h. die FD ist trivial
 2. Das Attribut B ist in einem Kandidatenschlüssel von R enthalten – also B ist prim.
 3. α ist Superschlüssel von R

Bsp nicht in 3NF

PersNr	Name	Rang	Raum	VorlNr	Titel	CP
123	Iwer	W2	C011	7051	Datenba nktechnol ogy	5
123	Iwer	W2	C011	3111	Software technik	10
123	Iwer	W2	C011	4111	Software technik	5
...
234	Pratz	C3	C011	1111	PM	5
256	Laftum	W4	C008	1234	Einführung in C	5

BESTIMMUNG OB 3NF ZUTRIFFT ODER NICHT

1. FDs bestimmen

$PersNr \rightarrow Name, Rang, Raum$

$VorlNr \rightarrow Titel, CP, PersNr, Name, Rang$

2. Kandidatenschlüssel bzw. Prim-Attribute bestimmen

{PersNr, VorlNr} ist Kandidatenschlüssel

PersNr, VorlNr sind Prim-Attribute

3. Überprüfen der FDs ob valide

$PersNr \rightarrow Name, Rang, Raum$

PersNr ist kein Superschlüssel

Name, Rang, Raum sind keine Prim-Attribute

FD ist nicht trivial

NICHT in 3NF

PersNr	Name	Rang	Raum	VorlNr	Titel	CP
123	Iwer	W2	C011	7051	Datenba nktechnol ogy	5
123	Iwer	W2	C011	3111	Softwaret echnik	10
123	Iwer	W2	C011	4111	Softwaret echnik	5
...
234	Pratz	C3	C011	1111	PM	5
256	Laftum	W4	C008	1234	Einführun g in C	5

RELATION IN 3NF ZERLEGEN

Synthesealgorithmus

1. Bestimme die kanonische Überdeckung F_c zu F
2. Für jede funktionale Abhängigkeit $\alpha \rightarrow \beta \in F_c$:
 1. Kreiere ein Relationenschema $R_\alpha := \alpha \cup \beta$
 2. Ordne R_α die FDs $F_c := \{\alpha' \rightarrow \beta' \in F_c \mid \alpha' \cup \beta' \subseteq R_\alpha\}$ zu
3. Falls eines der in Schritt 2 erzeugten Schemata R_α einen Kandidatenschlüssel von R bzgl. F_c enthält, sind wir fertig, sonst wähle einen Kandidatenschlüssel $k \subseteq R$ aus und definiere folgendes zusätzliches Schema:
 1. $R_k := k$
 2. $F_k := \emptyset$
4. Eliminiere diejenigen Schemata R_α , die in einem anderen Relationenschema $R_{\alpha'}$ enthalten sind, d.h. $R_\alpha \subseteq R_{\alpha'}$

RELATION IN 3NF ZERLEGEN

Beispiel

PersNr	Name	Rang	Raum	VorINr	Titel	CP
123	Iwer	W2	C011	7051	Datenba nktechnolog ogy	5
123	Iwer	W2	C011	3111	Softwaret echnik	10
123	Iwer	W2	C011	4111	Softwaret echnik	5
...
234	Pratz	C3	C011	1111	PM	5
256	Laftum	W4	C008	1234	Einführun g in C	5

RELATION IN 3NF ZERLEGEN

Beispiel 2

VerkNr	VerkName	PLZ	VerkAdresse	Produktname	Umsatz
V1	Meier	80075	München	Waschmaschine	11000
V1	Meier	80075	München	Herd	5000
V1	Meier	80075	München	Kühlschrank	1000
V2	Schneider	70038	Stuttgart	Herd	4000
V2	Schneider	70038	Stuttgart	Kühlschrank	3000
V3	Müller	50083	Köln	Staubsauger	1000

ÜBUNGEN

3NF

ZEIT FÜR GRUPPENARBEIT

- Sie haben 30 Minuten Zeit
- Ein Team stellt danach eine Lösung vor.

Bringen Sie folgendes Relationenschema mittels des Synthesealgorithmus in die dritte Normalform. Gehen Sie dabei schrittweise vor, d.h.

- Bestimmen Sie die geltenden FDs
- Bestimmen Sie die Kandidatenschlüssel
- Bestimmen Sie die kanonische Überdeckung der FDs
- Wenden Sie den Synthesealgorithmus an

Mitarbeiter{PersNr, Name, Fachgebiet, BossPersNr, BossName, MatrNr, SName, Semester, SWohnOrt}

MatrNr, Sname, Semester, SWohnOrt sind die von den Mitarbeitern betreuten Studenten
BossNr, BossName sind die Daten der Professoren, bei denen die Mitarbeiter angestellt sind

SHOW YOUR WORK

Was sind Ihre Lösungen?
Gäbe es Optimierungsmöglichkeiten?

BCNF – EINE ERWEITERUNG DER NORMALFORM

NORMALFORMEN

BCNF

- Boyce-Codd Normalform (BCNF) stellt nochmals eine Verschärfung dar. Das Ziel der BCNF besteht darin, dass Informationseinheiten nicht mehrmals, sondern genau nur einmal gespeichert werden dürfen.
- Ein Relationenschema R mit FDs F ist in BCNF, falls für jede funktionale Abhängigkeit $\alpha \rightarrow \beta$ mindestens eine der folgenden zwei Bedingungen gilt:
 1. $\beta \in \alpha$, d.h. die FD ist trivial
 2. α ist Superschlüssel von R

Beispiel in 3NF aber nicht in BCNF: Stadt(Ort, BLand, Ministerpräsident/in, Einwohner)

FD: Ort, BLand \rightarrow Einwohner
Ministerpräsident/in \rightarrow BLand
BLand \rightarrow Ministerpräsident/in

Kandidatenschlüssel: $k1=\{\text{Ort, BLand}\}$, $k2=\{\text{Ort, Ministerpräsident/in}\}$

RELATIONEN IN BCNF ZERLEGEN

Dekompositionsalgorithmus

• Die Menge $Z = \{R_1, R_2, \dots, R_n\}$ wird von Zerlegungen sukzessive generiert:

1. Starte mit $Z = \{R\}$
2. Solange es noch ein Relationenschema $R_i \in Z$ gibt, das nicht in BCNF ist, mache folgendes:
 1. Finde eine für R_i geltende nicht-triviale FD $(\alpha \rightarrow \beta)$ mit
 1. $\alpha \cap \beta = \emptyset$
 2. $\alpha \not\rightarrow R_i$Man sollte die funktionalen Abhängigkeiten so wählen, dass β alle von α funktional abhängigen Attribute $B \in (R_i - \alpha)$ enthält, damit der Dekompositionsalgorithmus möglichst schnell terminiert.
 2. Zerlege R_i in $R_{i1} := \alpha \cup \beta$ und $R_{i2} := R_i - \beta$
 3. Entferne R_i aus Z und füge R_{i1} und R_{i2} ein, also $Z := (Z - R_i) \cup \{R_{i1}\} \cup \{R_{i2}\}$

Sobald dieser Algorithmus beendet ist, enthält Z eine Menge von BCNF-Relationen, die eine verlustlose Zerlegung von R darstellen.

RELATION IN BCNF ZERLEGEN

Beispiel

- $R(A,B,C)$
 $A \rightarrow B, B \rightarrow C$

Schritte: Überprüfung ob die linke Seite ein Superschlüssel ist. Wenn ja für alle FD, dann ist Relation in BCNF, wenn nein, muss Relation gespalten werden.

Hülle von linker Seite berechnen und daraus neue Relationen erstellen.

ÜBUNGEN

BCNF

ZEIT FÜR GRUPPENARBEIT

- Sie haben 30 Minuten Zeit
- Ein Team stellt danach eine Lösung vor.

Gegen ist das abstrakte Relationenschema $R=\{A, B, C, D, E, F\}$ mit den funktionalen Abhängigkeiten

$A \rightarrow BC$

$C \rightarrow DA$

$E \rightarrow ABC$

$F \rightarrow CD$

$CD \rightarrow BEF$

1. Ist die Relation in BCNF? Wenn nein, erstellen Sie Sie in BCNF!
2. Ist die Relation in 3NF? Wenn nein, erstellen Sie Sie in 3NF!

SHOW YOUR WORK

Was sind Ihre Lösungen?
Gäbe es Optimierungsmöglichkeiten?

ZUSAMMENFASSUNG

- Ziel ist die dritte Normalform, bzw. BCNF.
- Wir benötigen Wissen zu funktionaler Abhängigkeit!
- Wir erzeugen Relationen mit möglichst einfachen Schlüsselkandidaten

QUELLENVERZEICHNIS

Muster für Verzeichnis

1. Datenbanksysteme – Eine Einführung, Kemper, Eickler