

AG Datenbanken und Informationssysteme

Wintersemester 2006 / 2007

Prof. Dr.-Ing. Dr. h. c. Theo Härder
 Fachbereich Informatik
 Technische Universität Kaiserslautern



<http://wwwdvs.informatik.uni-kl.de>

4. Übungsblatt

Für die Übung am Donnerstag, 23 November 2006,
 von 15:30 bis 17:00 Uhr in 13/222.

Aufgabe 1: 4NF am Beispiel Kredite

Gegeben sei folgende Relation:

KREDITE (KONTONR, LFD_KREDITNR, DATUM, NAME, ADRESSE)

In dieser Relation sollen folgende Abhängigkeiten gelten:

- Ein Kunde kann mehrere Konten haben.
- Ein Kunde kann mehrere Adressen haben.
- Ein Konto kann mehrere Besitzer haben.
- Auf einem Konto können mehrere Kredite aufgenommen werden.
- Ein Kredit auf einem Konto läuft seit einem festgelegten Datum

Folgendes Beispiel erfüllt diese Bedingungen:

KONTONR	LFD_KREDITNR	DATUM	NAME	ADRESSE
100	1	01.01.2002	Müller	KL
100	1	01.01.2002	Maier	MA
100	1	01.01.2002	Maier	SB
100	2	01.03.2002	Müller	KL
100	2	01.03.2002	Maier	MA
100	2	01.03.2002	Maier	SB
110	1	15.11.2001	Müller	KL
110	2	01.02.2002	Müller	KL
120	1	01.06.2002	Maier	MA
120	1	01.06.2002	Maier	SB

- a) Welche Abhängigkeiten bestehen in dieser Relation?
 b) Wie lässt sich die Relation in eine 4NF zerlegen?

Seite 1

! Es kann durchaus vorkommen, dass die Lösungsvorschläge fehlerhaft oder unvollständig sind !

Lösung:

- a) Mehrwertige Abhängigkeiten sind direkt aus Aufgabenstellung abzulesen:
 NAME -->> KONTONR, ADRESSE (Kontonr, Adresse mwtig abh. von Name)
 KONTONR -->> NAME, LFD_KREDITNR (Name, Lfd_Kreditnr mwtig abh. von Kontonr)
 funktionale Abhängigkeit:
 KONTONR, LFD_KREDITNR -> DATUM

- b) Zerlegung in die 4NF-Relationen:
 R1 (KONTONR, LFD_KREDITNR, DATUM)
 R2 (KONTONR, NAME)
 R3 (NAME, ADRESSE)

mit den Inhalten:

KONTONR	LFD_KREDITNR	DATUM	KONTONR	NAME	NAME	ADRESSE
100	1	01.01.2002	100	Müller	Müller	KL
100	2	01.03.2002	100	Maier	Maier	MA
110	1	15.11.2001	110	Müller	Maier	SB
110	2	01.02.2002	120	Mayer		
120	1	01.06.2002				

Die durch die Zerlegung entstandenen Relationen sind laut Definition in 4NF, denn R1, R2 und R3 sind in BCNF und bei R1, R2 und R3 liegen keine mehrwertigen Abhängigkeiten vor.

Aufgabe 2: Synthese von Relationen in 3NF

Die Beziehungen der Miniwelt, einem anwendungsbezogenen Ausschnitt der realen Welt, sollen im Hinblick auf das Informationsbedürfnis der verschiedenen Anwendungen einer Datenbank unabhängig von ihrer physischen Realisierung im konzeptionellen Schema (logisches Datenmodell) beschrieben und dargestellt werden. Im Relationenmodell sind Relationen in 3NF zur Vermeidung bestimmter Anomalien beim Änderungsdienst als logische Datenstrukturen besonders gut geeignet.

Ein Weg zur Ableitung dieser logischen Datenstrukturen ist die Relationensynthese. Ausgangspunkt dafür ist die Beschreibung der Miniwelt durch Attribute und FAen, die entsprechend dem Informationsbedarf der geplanten Anwendungen zu erheben sind. Attribute und FAen dienen als Erhebungsprimitive für den IST-Zustand. Alle möglichen FAen, die gegebenenfalls aus der erhobenen Menge abgeleitet sind, bilden eine transitiv abgeschlossene Menge, die sich als gerichteter Graph darstellen lässt. Für den Entwurf von Relationen interessiert eine nichtredundante Überdeckung der FAen der transitiven Hülle, die eine redundanzfreie logische Datenstruktur garantiert. Aus dieser minimalen Menge von FAen lassen sich schrittweise Relationen in optimaler 3NF erzeugen.

Für folgendes Anwendungsbeispiel ist die Relationensynthese für Relationen in optimaler 3NF durchzuführen:

Es seien in einer Miniwelt folgende Attribute, die die Entities Student (STUD), Fachbereich (FB), Vorlesung (VL), Professor (PR), Forschungsgruppe (FG) und Fachrichtung (FR) beschreiben, bekannt:

Name (NAME), Matrikelnummer (MATRNR), Geburtsdatum (GEB), Adresse (ADR), Semesteranzahl (SEMANZ), Fachbereichsnummer (FBNR), Fachbereichsname (FBNAME), Dekan (DEKAN), Fachbereichsadresse (FBADR), Professor (PROF), Vorlesung (VORL), Fachrichtung (FACHR), Forschungsgruppe (FG) und Stundenzahl (STDANZ).

Seite 2

! Es kann durchaus vorkommen, dass die Lösungsvorschläge fehlerhaft oder unvollständig sind !

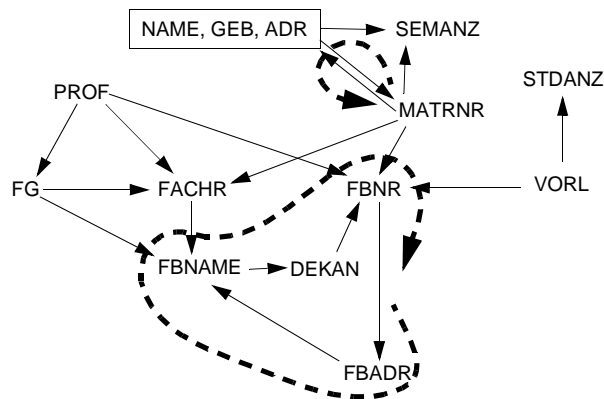
Folgende FAen seien erhoben worden:

- NAME, GEB, ADR → MATRNR, SEMANZ
- MATRNR → NAME, GEB, ADR, SEMANZ
- MATRNR → FACHR, FBNR
- FACHR → FBNAME
- FG → FBNAME, FACHR
- PROF → FACHR, FG, FBNR
- FBNAME → DEKAN
- DEKAN → FBNR
- FBNR → FBADR
- FBADR → FBNAME
- VORL → FBNR, STDANZ

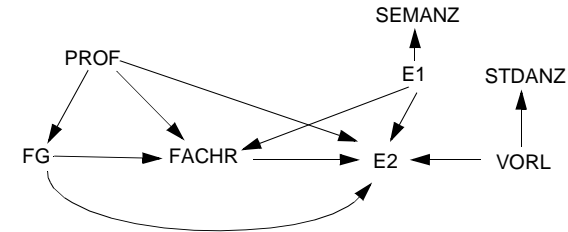
- a) Stellen Sie die Menge der FAen durch einen Graph dar.
- b) Der erhaltene FA-Graph hat Zyklen. Da jede FA-Beziehung eines der am Zyklus beteiligten Attribute für jedes Attribut des Zyklus gilt, kann der Zyklus separiert und durch ein Ersatzattribut (Ersatzknoten) im FA-Graph ersetzt werden. Zeichnen Sie den reduzierten FA-Graph nach Elimination der Zyklen.
- c) Für jeden Zyklus kann isoliert eine minimale Überdeckung gefunden werden. Finden Sie für die Zyklen geeignete minimale Überdeckungen.
- d) Finden Sie eine minimale Überdeckung für den reduzierten FA-Graph und ersetzen sie die Ersatzattribute wieder durch eine geeignete minimale Überdeckung des entsprechenden Zyklus. Damit haben sie eine minimale Überdeckung für das Anwendungsbeispiel gefunden.
- e) Bilden Sie aus den FAen der gefundenen minimalen Überdeckung Relationen in 3NF und diskutieren Sie das Ergebnis.

Lösung:

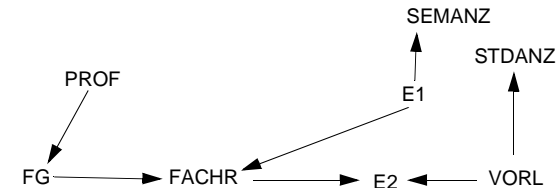
- a) Stellen Sie die Menge der erhobenen FAen durch einen Graph dar.



- b) Der erhaltene FA-Graph hat Zyklen. Da jede FA-Beziehung eines der am Zyklus beteiligten Attribute für jedes Attribut des Zyklus gilt, kann der Zyklus separiert und durch ein Ersatzattribut (Ersatzknoten) im FA-Graph ersetzt werden. Zeichnen Sie den reduzierten FA-Graph nach Elimination der Zyklen.



- c) Für jeden Zyklus kann isoliert eine minimale Überdeckung gefunden werden. Finden Sie für die Zyklen geeignete minimale Überdeckungen.
 E1: (MATRNR → (NAME GEB ADR)), ((NAME GEB ADR) → MATRNR)
 E2: (FBNR → FBADR), (FBADR → FBNAME), (FBNAME → DEKAN), (DEKAN → FBNR)
- d) Finden Sie eine minimale Überdeckung für den reduzierten FA-Graph und ersetzen sie die Ersatzattribute wieder durch eine geeignete minimale Überdeckung des entsprechenden Zyklus. Damit haben sie eine minimale Überdeckung für das Anwendungsbeispiel gefunden.



- e) Bilden Sie aus den FAen der gefundenen minimalen Überdeckung Relationen in 3NF. Diskutieren Sie das Ergebnis.
 Wähle für jeden Zyklus ein repräsentatives Element: E1:MATRNR, E2: FBNR

- R1: (E1 SEMANZ FACHR) = (MATRNR NAME GEB ADR SEMANZ FACHR)
- R2: (FACHR "Element aus (E2)") = (FACHR FBNR)
- R3: (PROF FG)
- R4: (FG FACHR)
- R5: (VORL "Element aus (E2)" STDANZ) = (VORL FBNR STDANZ)
- R6: (FBNR, FBADR)
- R7: (FBADR, FBNAME)
- R8: (FBNAME, DEKAN)

Zu diskutieren wäre evtl.:

- R6': (E2) = (FBNR FBNAME FBADR DEKAN) R6-R8 entfallen
- R3': (PROF FG FACHR FBNR) ==> unter der Voraussetzung, dass Relation sehr stabil

(Schreibweise: Primärschlüssel, Eindeutigkeit)

Aufgabe 3: Entwurf eines Data Warehouse

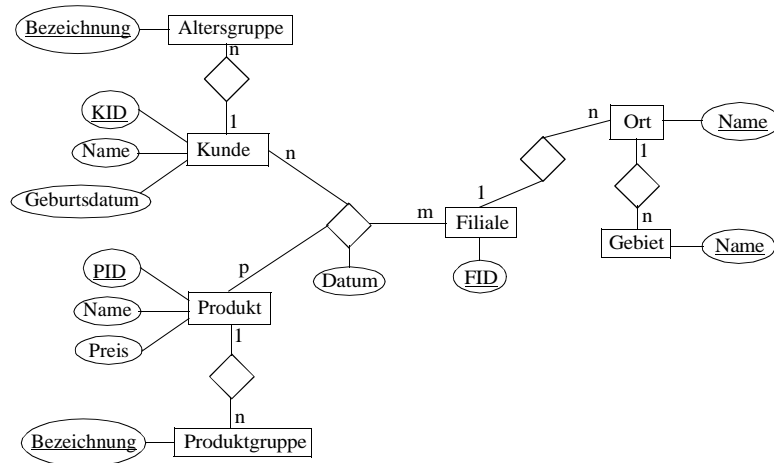
Für den aus dem folgenden Anwendungsszenario hervorgehenden Datenbestand eines Unternehmens soll ein Data Warehouse zur Analyse der Verkaufszahlen erstellt werden.

Das Unternehmen erfasst alle Kunden mit einer eindeutigen ID, deren Namen und Geburtsdatum. Jeder Kunde wird einer Altersgruppe (z. B. Teenager) zugeteilt. Die angebotenen Produkte werden mit einer Produkt-ID, Namen und einem Preis gespeichert. Jedes Produkt wird zusätzlich mit einer Produktgruppe klassifiziert. Das Unternehmen besitzt zahlreiche Filialen. Für jede Filiale ist der Ort gespeichert, der wiederum einem der Gebiete Süd, Ost, West oder Nord zugeordnet ist. Der Verkauf eines Produkts an einen Kunden in einer Filiale wird zusammen mit dem Datum in der Unternehmensdatenbank abgelegt.

- Erstellen Sie für das beschriebene Anwendungsszenario ein E/R-Diagramm.
- Bilden Sie das E/R-Diagramm mit SQL auf ein relationales Datenbankschema ab.
- Erstellen Sie mit SQL das Stern-Schema für ein Data Warehouse, in das die Daten aus der Datenbank von b) geladen und mit dem die in d) aufgeführten Fragen beantwortet werden können.
- Formulieren Sie die folgenden Anfragen in SQL auf dem Datenbankschema des Data Warehouse:
 - Wieviele Produkte wurden im ersten Quartal verkauft?
 - Wieviele Produkte wurden davon in der fünften Kalenderwoche verkauft?
 - Wieviele Produkte haben Kunden der Altersgruppe 'Teenager' im Gebiet 'Süd' gekauft?
 - Wie hoch ist das Durchschnittsalter aller Kunden der Altersgruppe 'Rentner', die im Gebiet 'Nord' Produkte der Gruppe 'Mobiltelefon' außerhalb der Weihnachtszeit gekauft haben?

Lösung:

- E/R-Diagramm für das Anwendungsszenario



- Abbildung des E/R-Diagramms auf ein SQL-Schema

```

CREATE TABLE altersgruppe
(
    bezeichnung VARCHAR(25) PRIMARY KEY
)

CREATE TABLE kunde
(
    kid INTEGER PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    geburtsdatum DATE NOT NULL,
    gruppe VARCHAR(25) NOT NULL REFERENCES altersgruppe (bezeichnung)
)

CREATE TABLE produktgruppe
(
    bezeichnung VARCHAR(25) PRIMARY KEY
)

CREATE TABLE produkt
(
    pid INTEGER PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    preis DECIMAL(6,2) NOT NULL,
    gruppe VARCHAR(25) NOT NULL REFERENCES produktgruppe (bezeichnung)
)

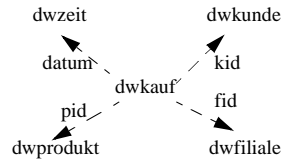
CREATE TABLE gebiet
(
    name VARCHAR(20) PRIMARY KEY
)

CREATE TABLE ort
(
    name VARCHAR(50) PRIMARY KEY,
    gebiet VARCHAR(20) NOT NULL REFERENCES gebiet (name)
)

CREATE TABLE filiale
(
    fid INTEGER PRIMARY KEY,
    ort VARCHAR(50) NOT NULL REFERENCES ort (name)
)

CREATE TABLE kauf
(
    datum DATE NOT NULL,
    kid INTEGER NOT NULL REFERENCES kunde (kid),
    pid INTEGER NOT NULL REFERENCES produkt (pid),
    fid INTEGER NOT NULL REFERENCES filiale (fid)
)
  
```

c) SQL-Schema für das Data Warehouse



```

CREATE TABLE dwzeit
(
  datum DATE PRIMARY KEY,
  kw INTEGER NOT NULL,
  quartal INTEGER NOT NULL,
  saison VARCHAR(30) NOT NULL
)

CREATE TABLE dwkunde
(
  kid INTEGER PRIMARY KEY,
  alter INTEGER NOT NULL,
  gruppe VARCHAR(25) NOT NULL
)

CREATE TABLE dwprodukt
(
  pid INTEGER PRIMARY KEY,
  gruppe VARCHAR(25) NOT NULL
)

CREATE TABLE dwfiliale
(
  fid INTEGER PRIMARY KEY,
  ort VARCHAR(50) NOT NULL,
  gebiet VARCHAR(20) NOT NULL
)

CREATE TABLE dwkauf
(
  datum DATE NOT NULL REFERENCES dwzeit(datum),
  kid INTEGER NOT NULL REFERENCES dwkunde(kid),
  pid INTEGER NOT NULL REFERENCES dwprodukt(pid),
  fid INTEGER NOT NULL REFERENCES dwfiliale(fid)
)

```

d) Anfragen an das Data Warehouse

```

(1) SELECT COUNT(*)
FROM dwkauf k, dwzeit z
WHERE k.datum=z.datum AND z.quartal=1

```

Seite 7

! Es kann durchaus vorkommen, dass die Lösungsvorschläge fehlerhaft oder unvollständig sind !

```

(2) SELECT COUNT(*)
FROM dwkauf k, dwzeit z
WHERE k.datum=z.datum AND z.kw=5

(3) SELECT COUNT(*)
FROM dwkauf ka, dwkunde ku, dwfiliale f
WHERE ka.kid=ku.kid AND ka.fid=f.fid AND ku.gruppe='Teenager' AND f.gebiet='Süd'

(4) SELECT AVG(ku.alter)
FROM dwkauf ka, dwzeit z, dwkunde ku, dwfiliale f, dwprodukt p
WHERE ka.datum=z.datum AND ka.kid=ku.kid AND ka.fid=f.fid AND ka.pid=p.pid
AND ku.gruppe='Rentner' AND f.gebiet='Nord' AND p.gruppe='Mobiltelefon'
AND z.saison <> 'Weihnachten'

```

Aufgabe 4: SQL-Anfragen und Views am Beispiel „Personal-DB“

Gegeben sei die folgende Datenbank, die von der Finanzabteilung zur Berechnung der Löhne und Gehälter der Mitarbeiter (MA) der verschiedenen Abteilungen (ABT) genutzt wird.

```

MA (MANR, MANAME, MAVORNAME, ABTNR, FIRMENZUGEHORIGKEIT,
  KINDER, STEUERKLASSE, GEHALT, KRANKENKASSE, BEITRAGSSATZ)
ABT (ABTNR, ABTNAME, ABTLEITER, ABTORT)

```

ABTLEITER hat denselben Wertebereich wie MANR und ist Fremdschlüssel. Zur Erstellung verschiedener Statistiken sollen dynamische Sichten erzeugt werden, und zwar:

- Eine Sicht, die die Abteilungsnummer, den Abteilungsnamen, die Anzahl der Mitarbeiter der Abteilung, den Durchschnitt der Firmenzugehörigkeit und des Gehalts, das höchste Gehalt der Abteilung und die Differenz zwischen dem höchsten und niedrigsten Gehalt in der Abteilung umfasst.
- Eine Sicht, die, gestaffelt nach Krankenkasse und Kinderzahl, den durchschnittlichen Beitragssatz für Mitglieder von Abteilungen in Frankfurt, München oder Stuttgart beinhaltet.
- Eine Sicht, die Name, Vorname und Gehalt der Mitarbeiter enthält, die in Abteilungen arbeiten, deren Durchschnittsgehalt größer als 50.000 ist.
- Eine Sicht, die die Daten der Mitarbeiter in Steuerklasse 1 enthält, und eine weitere, die nur Mitarbeiter in Steuerklasse 1 mit mehr als 5 Jahren Firmenzugehörigkeit enthält.
- Formulieren Sie auf der ersten der beiden letzten Sichten die Anfrage nach den Daten aller Mitarbeiter, deren Abteilungsleiter 'Müller' heißt.
- Was passiert bei Änderungen auf Sichten, die Aggregatfunktionen beinhalten?

Seite 8

! Es kann durchaus vorkommen, dass die Lösungsvorschläge fehlerhaft oder unvollständig sind !

Lösung:

- a) Eine Sicht, die die Abteilungsnummer, den Abteilungsnamen, die Anzahl der Mitarbeiter der Abteilung, den Durchschnitt der Firmenzugehörigkeit und des Gehalts, das höchste Gehalt der Abteilung und die Differenz zwischen dem höchsten und niedrigsten Gehalt in der Abteilung umfasst.

```
CREATE VIEW ABTSTATISTIK
(ABTNR, ABTNAME, MITANZ, DFIRMENZUGEHORIGKEIT, DGEHALT, MAXGEHALT,
DIFFGEHALT )
AS
SELECT ABT.ABTNR, ABTNAME, COUNT(MANR),
AVG (FIRMENZUGEHORIGKEIT), AVG(GEHALT),
MAX (GEHALT), MAX(GEHALT) - MIN(GEHALT)
FROM MA, ABT
WHERE MA.ABTNR = ABT.ABTNR
GROUP BY ABT.ABTNR, ABTNAME
```

- b) Eine Sicht, die, gestaffelt nach Krankenkasse und Kinderzahl, den durchschnittlichen Beitragssatz für Mitglieder von Abteilungen in Frankfurt, München oder Stuttgart beinhaltet.

```
CREATE VIEW KK_KINDER_BEITRAGSSATZ_STAEDTE
( KRANKENKASSE, KINDER, DBEITRAGSSATZ )
AS
SELECT KRANKENKASSE, KINDER, AVG(BEITRAGSSATZ)
FROM MA, ABT
WHERE MA.ABTNR = ABT.ABTNR
AND ABT.ABTORT IN ('Frankfurt', 'München', 'Stuttgart' )
GROUP BY KRANKENKASSE, KINDER
```

- c) Eine Sicht, die Name, Vorname und Gehalt der Mitarbeiter enthält, die in Abteilungen arbeiten, deren Durchschnittsgehalt größer als 50.000 ist.

```
CREATE VIEW ANG_IN_TOP_ABT ( MANAME, MAVORNAME, GEHALT )
AS
SELECT MANAME, MAVORNAME, GEHALT
FROM MA
WHERE ABTNR IN
( SELECT ABT.ABTNR
FROM MA, ABT
WHERE MA.ABTNR = ABT.ABTNR
GROUP BY ABT.ABTNR
HAVING AVG(GEHALT) > 50000 )
```

- d) Eine Sicht, die die Daten der Mitarbeiter in Steuerklasse 1 enthält, und eine weitere, die nur Mitarbeiter in Steuerklasse 1 mit mehr als 5 Jahren Firmenzugehörigkeit enthält.

```
CREATE VIEW STKL1 AS
SELECT *
FROM MA
WHERE STEUERKLASSE = 1

CREATE VIEW STKL1_FZ5 AS
SELECT *
FROM MA
WHERE STEUERKLASSE = 1 AND
FIRMENZUGEHORIGKEIT > 5

CREATE VIEW STKL1_FZ5 AS
SELECT *
```

```
FROM STKL1
WHERE FIRMENZUGEHORIGKEIT > 5
```

- e) Formulieren Sie auf der ersten der beiden letzten Sichten die Anfrage nach den Daten aller Mitarbeiter, deren Abteilungsleiter 'Müller' heißt.

```
SELECT STKL1.*
FROM STKL1, ABT, MA
WHERE STKL1.ABTNR = ABT.ABTNR AND
ABT.ABTLEITER = MA.MANR AND
MA.MANAME = 'MÜLLER'
```

- f) Was passiert bei Änderungen auf Sichten, die Aggregatfunktionen beinhalten? Sichten mit Aggregatfunktionen lassen sich nicht ändern.