



Übung zur Vorlesung *Grundlagen: Datenbanken* im WS19/20

Christoph Anneser, Moritz Sichert, Lukas Vogel (gdb@in.tum.de)

<https://db.in.tum.de/teaching/ws1920/grundlagen/>

Blatt Nr. 08

Tool zum Üben von funktionalen Abhängigkeiten: <https://normalizer.db.in.tum.de/>.

Hausaufgabe 1

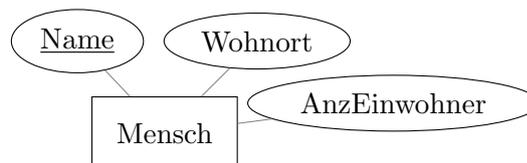
Bewerten Sie die folgende Aussage: Die Modellierung in einem ER-Diagramm führt immer zu Relationen, die mindestens in 4. Normalform sind.

- Unter welchen Voraussetzungen können aus einem ER-Diagramm FDs (oder MVDs) entstehen, die die 4. Normalform (oder eine der niedrigeren) verletzen? Geben Sie ein beispielhaftes ER-Diagramm an, in dem ein Zusammenhang nicht korrekt modelliert wurde.
- Wie hängt die Modellierung von Entitytypen und Relationships zusammen mit der Bestimmung von FDs und MVDs?
- Welche Auswirkung hat die Verfeinerung von Relationen aus dem ER-Modell auf die Normalformen?

Lösung:

- Falls es Zusammenhänge gibt, die im ER-Diagramm nicht modelliert wurden (z.B. Attribute, die eigentlich als eigener Entitytyp modelliert werden sollten), kann das zu Relationen führen, die nicht in 4. Normalform sind.

Beispiel:



Daraus resultiert die folgende Relation:

$$\text{Mensch} : \{[\underline{\text{Name}}, \text{Wohnort}, \text{AnzEinwohner}]\}$$

Für diese gilt offensichtlich die FD $\text{Name} \rightarrow \text{Wohnort}, \text{AnzEinwohner}$. Allerdings gilt hier auch die FD $\text{Wohnort} \rightarrow \text{AnzEinwohner}$. Damit verletzt diese Relation bereits die 3. Normalform.

Wären in dem ER-Diagramm Stadt und Land als eigene Entitytypen modelliert worden, gäbe es insgesamt die folgenden zwei (verfeinerten) Relationen, die jeweils in 4. Normalform sind:

$$\text{Mensch} : \{[\underline{\text{Name}}, \text{Wohnort}]\}$$
$$\text{Ort} : \{[\underline{\text{Ortsname}}, \text{AnzEinwohner}]\}$$

- b) Ein gegebener Zusammenhang, der im ER-Diagramm als Entitytyp oder Relationship modelliert wird, taucht im relationalen Modell immer wieder als FD oder MVD auf. Umgekehrt findet sich jede FD oder MVD im ER-Diagramm als Entitytyp oder Relationship wieder.

Die Modellierung eines ER-Diagramms (im Speziellen die Überlegungen welche Attribute als eigene Entitytypen oder Relationships modelliert werden) ähnelt also dem formalen Vorgehen im Dekompositionsalgorithmus sehr.

Wenn der Dekompositionsalgorithmus auf das Beispiel aus Teilaufgabe a) angewendet wird, erhält man dadurch exakt das selbe Ergebnis als hätte man „Wohnort“ als eigenen Entitytyp modelliert.

- c) Zwei Relationen werden in der Verfeinerung nur genau dann zusammengefasst, wenn sie den selben Schlüssel haben. Gibt es nach der Verfeinerung also eine FD oder MVD, die die 4. Normalform verletzt, muss es diese auch bereits vor der Zusammenfassung gegeben haben und dort bereits ebenfalls die 4. Normalform verletzt haben. Die Verfeinerung hat also keine Auswirkung auf die geltenden Normalformen.

Hausaufgabe 2

Betrachten Sie ein abstraktes Relationenschema $R : \{[A, B, C, D, E, F, G]\}$ mit den FDs:

$$\begin{aligned} A &\rightarrow BC \\ DE &\rightarrow B \\ F &\rightarrow A \\ E &\rightarrow BF \\ A &\rightarrow DE \\ C &\rightarrow A \end{aligned}$$

Überführen Sie die Relation verlustfrei und abhängigkeitsbewahrend in die dritte Normalform.

Lösung:

Kanonische Überdeckung

Linksreduktion:

$$\begin{aligned} A &\rightarrow BC \\ \cancel{DE} &\rightarrow B \\ F &\rightarrow A \\ E &\rightarrow BF \\ A &\rightarrow DE \\ C &\rightarrow A \end{aligned}$$

Rechtsreduktion:

$$\begin{aligned} A &\rightarrow \cancel{B}C \\ E &\rightarrow \emptyset \\ F &\rightarrow A \\ E &\rightarrow BF \\ A &\rightarrow DE \\ C &\rightarrow A \end{aligned}$$

Zusammenfassen der FDs mit gleichen linken Seiten:

$$\begin{aligned} A &\rightarrow CDE \\ E &\rightarrow \emptyset \\ F &\rightarrow A \\ E &\rightarrow BF \\ C &\rightarrow A \end{aligned}$$

Entfernen von FDs mit leerer Menge auf der rechten Seite:

$$\begin{aligned} A &\rightarrow CDE \\ F &\rightarrow A \\ E &\rightarrow BF \\ C &\rightarrow A \end{aligned}$$

Synthesealgorithmus

Zerlegung anhand der kanonischen Überdeckung:

$$\begin{aligned} R_1 &: \{\underline{A}, C, D, E\} \\ R_2 &: \{A, \underline{E}\} \\ R_3 &: \{B, \underline{E}, F\} \\ R_4 &: \{A, \underline{C}\} \end{aligned}$$

Kandidatenschlüssel bestimmen:

$$\kappa_1 = \{A, G\}, \kappa_2 = \{F, G\}, \kappa_3 = \{C, G\}, \kappa_4 = \{E, G\}$$

Kandidatenschlüssel der Relationenzerlegung hinzufügen:

Da keiner der Kandidatenschlüssel in der Zerlegung enthalten ist,
wähle ein beliebiges κ_i und erstelle R_κ , z. B. $R_\kappa = \{\underline{A}, G\}$.

Redundanzen entfernen:

$$R_4 \text{ verwerfen, da } R_4 \subseteq R_1$$

Ergebnis:

$$\{\underline{A}, C, D, E\}, \{A, \underline{F}\}, \{B, \underline{E}, F\}, \{\underline{A}, \underline{G}\}$$

Hausaufgabe 3

Bestimmen Sie alle Kandidatenschlüssel der Relation R . Wenden Sie den Dekompositionsalgorithmus an, um die Relation R in die BCNF zu zerlegen und unterstreichen Sie die Schlüssel der Teilrelationen des Endergebnisses.

$$R : \{[A, B, C, D, E, F]\}$$

FDs:

1. $B \rightarrow DA$
2. $DEF \rightarrow B$
3. $C \rightarrow EA$

Prüfen Sie als erstes, ob FD 1) für die Zerlegung geeignet ist und - falls dies der Fall ist - verwenden Sie diese im ersten Zerlegungsschritt. Für diese Aufgabe ist zu bedenken, dass die oben angegebenen FDs eine Charakterisierung der insgesamt geltenden FDs sind. Die Menge der geltenden FDs ist größer. Wieso? Wie muss dies beim Dekompositionsalgorithmus genutzt werden?

Lösung:

- Dekompositionsalgorithmus:
 - Starte mit $Z := \{R\}$.
 - R in BCNF? - Nein, $B \rightarrow DA$ verletzt die BCNF.
 - * Zerlegung anhand FD $B \rightarrow DA$, da $\{B\}$ kein Superschlüssel:
 $R_1 : \{[A, B, D]\}$ mit den FDs $F_1 = \{B \rightarrow DA\}$,
 $R_2 : \{[B, C, E, F]\}$ mit den FDs $F_2 = \{C \rightarrow E\}$, FD (2) geht verloren und FD (3) geht "teilweise" verloren: wenn $C \rightarrow AE$ gilt, dann gilt auch $C \rightarrow A$ und $C \rightarrow E$ (Dekompositionsregel), aber lediglich $C \rightarrow E$ bleibt erhalten.
 $Z := \{R_1, R_2\}$
 - R_1 in BCNF? - Ja.
 - R_2 in BCNF? - Nein, $C \rightarrow E$ verletzt die BCNF.
 - * Zerlegung anhand FD $C \rightarrow E$, da $\{C\}$ kein Superschlüssel:
 $R_{2.1} : \{[C, E]\}$ mit den FDs $F_{2.1} = \{C \rightarrow E\}$,
 $R_{2.2} : \{[B, C, F]\}$ mit ausschließlich trivialen FDs.
 $Z := \{R_1, R_{2.1}, R_{2.2}\}$
 - $R_{2.1}$ in BCNF? - Ja.
 - $R_{2.2}$ in BCNF? - Ja.
- Ergebnis:

$$\begin{aligned} R_1 & : \{[A, \underline{B}, D]\} \\ R_{2.1} & : \{[\underline{C}, E]\} \\ R_{2.2} & : \{[\underline{B}, \underline{C}, \underline{F}]\} \end{aligned}$$

Im Allgemeinen ist eine gegebene FD-Menge weder minimal noch vollständig. Die angegebenen FDs enthalten also möglicherweise Redundanzen einerseits und andererseits werden triviale Abhängigkeiten i.d.R. nicht explizit mit angegeben. Bei der Ausführung des Dekompositionsalgorithmus müssen jedoch alle *geltenden* FDs betrachtet werden, die sich mit Hilfe der Axiome von Armstrong herleiten lassen (F^+). So gilt im obigen Beispiel in R_2 die FD $C \rightarrow E$, obwohl diese nicht explizit angegeben war.

Hausaufgabe 4

Gegeben sei die durch folgende SQL-Statements definierte Ausprägung einer Relation.

```
create table kinder_fahrraeder (
  person varchar(100) not null,
  kind_name varchar(100) not null,
  kind_alter integer not null,
  fahrrad_typ varchar(100) not null,
  fahrrad_farbe varchar(100) not null
);
insert into kinder_fahrraeder values
('Thomas', 'Markus', 10, 'Trekking-Fahrrad', 'schwarz'),
('Thomas', 'Markus', 10, 'Mountainbike', 'rot'),
('Thomas', 'Johanna', 5, 'Trekking-Fahrrad', 'schwarz'),
('Thomas', 'Johanna', 5, 'Mountainbike', 'rot');
```

Es gelten die beiden komplementären MVD

1. $person \twoheadrightarrow \{kind_name, kind_alter\}$ und
2. $person \twoheadrightarrow \{fahrrad_typ, fahrrad_farbe\}$

sowie die FD

3. $kind_name \rightarrow kind_alter$.

- a) Laura, das dritte Kind von Thomas, wird geboren. Fügen Sie Laura per SQL-Insert-Statement hinzu und beachten Sie dabei die MVDs. Formulieren Sie Ihr Statement so, dass es auch ohne Kenntnis der Fahrräder von Thomas funktioniert (d.h. nicht `insert ... 'Mountainbike', 'rot'`);).
- b) Allgemein gesprochen: In eine Relation $R : \{[A, B, C]\}$ mit den MVDs $A \twoheadrightarrow B$ und $A \twoheadrightarrow C$ soll für ein bestimmtes a in Spalte A ein neuer Wert b in Spalte B eingefügt werden. Wie viele Tupel müssen hinzugefügt werden, damit die MVDs weiterhin gelten?
- c) Was passiert, wenn Thomas seine beiden Fahrräder verkauft?
- d) Überführen Sie die Relation `kinder_fahrraeder` mit dem Dekompositionsalgorithmus in die 4. NF.
- e) Schreiben Sie ein SQL-Statement um zu prüfen ob die MVDs der Relation `kinder_fahrraeder` erfüllt sind.

Lösung:

- a) Zunächst werden alle Fahrräder von Thomas ermittelt, anschließend wird Laura kombiniert mit jedem Fahrrad in die Tabelle eingefügt.

```

with fahrraeder as (
  select distinct person, fahrrad_typ, fahrrad_farbe
  from kinder_fahrraeder
  where person='Thomas'
)
insert into kinder_fahrraeder (
  select person, 'Laura', '0', fahrrad_typ, fahrrad_farbe
  from fahrraeder
);

```

- b) b muss kombiniert mit jedem Wert in C eingefügt werden, die Anzahl neuer Tupel ist daher:

```
select count(distinct C) from R where A=a;
```

- c) Da die fahrradbezogenen Spalten nicht nullable sind, müssen sämtliche Tupel gelöscht werden, sodass auch die Informationen zu den Kindern verloren gehen. Durch Normalisierung kann diese Anomalie verhindert werden.
- d) Zunächst wird anhand der MVDs zerlegt, wir erhalten die Relationen:

$$R_1 : \{ \underline{\text{person}}, \text{kind_name}, \text{kind_alter} \} \text{ und}$$

$$R_2 : \{ \underline{\text{person}}, \text{fahrrad_typ}, \text{fahrrad_farbe} \}.$$

Anschließend wird R_1 anhand der FD weiter zerlegt:

$$R_{1a} : \{ \underline{\text{person}}, \text{kind_name} \} \text{ und}$$

$$R_{1b} : \{ \underline{\text{kind_name}}, \text{kind_alter} \} \text{ und}$$

$$R_2 : \{ \underline{\text{person}}, \text{fahrrad_typ}, \text{fahrrad_farbe} \}.$$

Alternativ kann auch mit der Zerlegung anhand der FD begonnen werden, die finale Zerlegung ist dieselbe.

- e) Eine Möglichkeit ist es, das Kreuzprodukt aus allen Fahrrädern und Kindern pro Person zu bilden, und die tatsächliche Ausprägung abzuziehen. Ist das Ergebnis des SQL-Statements leer, so sind die MVDs erfüllt.

```

with kinder as (
  select distinct person, kind_name, kind_alter
  from kinder_fahrraeder
), fahrraeder as (
  select distinct person, fahrrad_typ, fahrrad_farbe
  from kinder_fahrraeder
)
select f.person, k.kind_name, k.kind_alter, f.fahrrad_typ, f.fahrrad_farbe
from fahrraeder f, kinder k
where f.person=k.person
except
select * from kinder_fahrraeder;

```