



**Übung zur Vorlesung *Grundlagen: Datenbanken* im WS22/23**  
Michael Jungmair, Stefan Lehner, Moritz Sichert, Lukas Vogel (gdb@in.tum.de)  
<https://db.in.tum.de/teaching/ws2223/grundlagen/>

**Blatt Nr. 08**

**Hausaufgabe 1**

Betrachten Sie das Relationenschema

Fahrplan: {[Linie, Verbund, von, nach, von GPS, nach GPS, Preis, #Fahrzeuge, Modus]}

mit der folgenden beispielhaften Ausprägung:

Linie	Verbund	von	nach	von GPS	nach GPS	Preis	#Fahrzeuge	Modus
U6	MVV	GF	G	0N 0W	1S 0W	1€	20	U-Bahn
U6	MVV	G	GH	1S 0W	2S 0W	1€	20	U-Bahn
U6	MVV	GH	FR	2S 0W	5S 0W	3€	20	U-Bahn
U3	MVV	MF	GI	8S 0W	9S 0W	1€	16	U-Bahn
690	MVV	GF	DI	0N 0W	1N 0W	1€	5	Bus
690	MVV	DI	NF	1N 0W	3N 1W	2€	5	Bus
690	MVV	NF	EH	3N 1W	5N 2W	2€	5	Bus
S1	MVV	NF	EH	3N 1W	5N 2W	3€	8	S-Bahn

- Bestimmen Sie die geltenden FDs.
- Bestimmen Sie die Kandidatenschlüssel.

**Lösung:**

- Im Relationenschema gelten die folgenden funktionalen Abhängigkeiten:

- {Linie} → {#Fahrzeuge, Modus}
- {von} → {von GPS}
- {nach} → {nach GPS}
- {von GPS} → {von}
- {nach GPS} → {nach}
- {Linie, von, nach} → {Preis}
- ∅ → {Verbund}

Anhand der Ausprägung werden manche FDs nicht widerlegt, wie z.B. {#Fahrzeuge} → {Linie}. Bei dieser Musterlösung haben wir aber angenommen, dass es noch weitere Tupel in der Ausprägung geben könnte, die diese FD widerlegt.

Natürlich gelten auch alle anderen funktionalen Abhängigkeiten, die mit Hilfe der Armstrong-Axiome daraus hergeleitet werden können.

b) Die möglichen Kandidatenschlüssel sind:

- {Linie, von, nach}
- {Linie, von GPS, nach}
- {Linie, von, nach GPS}
- {Linie, von GPS, nach GPS}

Aus „Linie“ können die Attribute „#Fahrzeuge“ und „Modus“, aus „Linie“, „von“ und „nach“ das Attribut „Preis“ abgeleitet werden. Da „von“ und „von GPS“ sowie „nach“ und „nach GPS“ jeweils in beide Richtungen abgeleitet werden können, enthält der Kandidatenschlüssel alle möglichen Kombinationen.

## Hausaufgabe 2

Betrachten Sie ein abstraktes Relationenschema  $\mathcal{R} = \{A, B, C, D, E, F\}$  mit den FDs

1.  $A \rightarrow BC$
2.  $C \rightarrow DA$
3.  $E \rightarrow ABC$
4.  $F \rightarrow CD$
5.  $CD \rightarrow BEF$

- a) Berechnen Sie die Attributhülle von  $A$ .
- b) Bestimmen Sie alle Kandidatenschlüssel.
- c) Bestimmen Sie zu den gegebenen FDs die kanonische Überdeckung.

**Lösung:**

### Attributhülle von A

Berechnung der Attributhülle von  $A$  mit Hilfe des bekannten *AttrHülle*-Algorithmus.

Aufruf:  $AttrHülle(FD, \{A\})$ .

Schritt	betrachtete FD	Ergebnis
init		{A}
1.	$A \rightarrow BC$	{A, B, C}
2.	$C \rightarrow DA$	{A, B, C, D}
3.	$CD \rightarrow BEF$	{A, B, C, D, E, F}

Damit enthält die Attributhülle von  $A$  alle Attribute.

### Kandidatenschlüssel

$\{A\}$  ist nach der vorherigen Berechnung (Attributhülle von  $A$ ) ein Superschlüssel. Da  $\{A\}$  außerdem minimal ist, ist  $\{A\}$  ein Kandidatenschlüssel. Da man aus  $\{C\}$  und  $\{E\}$  direkt  $A$  folgern kann, handelt es sich hier ebenfalls um Superschlüssel und da sie einelementig sind (also minimal sind) auch um Kandidatenschlüssel. Aus  $\{F\}$  wiederum kann  $C$  und somit  $A$  gefolgert werden. Damit ist  $\{F\}$  analog zu oben auch ein Kandidatenschlüssel.

$\{B\}$  und  $\{D\}$  sind dagegen keine Kandidatenschlüssel.  $\{B\}$  ist nicht einmal Superschlüssel.  $\{CD\}$  wäre zwar ein Superschlüssel, allerdings kein Kandidatenschlüssel, da nicht minimal.

Kandidatenschlüssel sind:  $\{A\}, \{C\}, \{E\}, \{F\}$ .

## Kanonische Überdeckung

Gegeben ist die Ausgangsmenge  $F = \{A \rightarrow BC, C \rightarrow DA, E \rightarrow ABC, F \rightarrow CD, CD \rightarrow BEF\}$ .

1. Führe für jede FD  $\alpha \rightarrow \beta \in F$  die Linksreduktion durch.

Einzig in Betracht kommende FD ist  $CD \rightarrow BEF$ .

- Ist  $C$  überflüssig?  
 $AttrHülle(F, \{D\}) = \{D\} \not\supseteq \{B, E, F\}$
- Ist  $D$  überflüssig?  
 $AttrHülle(F, \{C\}) =$

Schritt	betrachtete FD	Ergebnis
init		$\{C\}$
1.	$C \rightarrow DA$	$\{A, C, D\}$
2.	$CD \rightarrow BEF$	$\{A, B, C, D, E, F\}$

$$\{C\}^+ = \{A, B, C, D, E, F\} \supseteq \{B, E, F\}$$

Damit kann  $CD \rightarrow BEF$  zu  $C \rightarrow BEF$  reduziert werden.

2. Führe für jede (verbliebene) FD  $\alpha \rightarrow \beta$  die Rechtsreduktion durch.

Bisheriges Zwischenergebnis:

$$A \rightarrow BC \quad (1)$$

$$C \rightarrow DA \quad (2)$$

$$E \rightarrow ABC \quad (3)$$

$$F \rightarrow CD \quad (4)$$

$$C \rightarrow BEF \quad (5)$$

**Betrachte FD (1):**

- Ist  $B$  überflüssig?  
 $B \in AttrHülle(F - FD (1) \cup (A \rightarrow C), A)$ , da  $A \rightarrow C \rightarrow BEF$ .
- Ist  $C$  überflüssig?  
 $C \notin AttrHülle(F - FD (1) \cup (A \rightarrow \emptyset), A)$ .

Damit erhält man für FD (1):  $A \rightarrow C$ .

**Betrachte FD (2):**

- Ist  $D$  überflüssig?  
 $D \in AttrHülle(F - FD (2) \cup (C \rightarrow A), C)$ , da  $C \rightarrow BEF, F \rightarrow CD$ .
- Ist  $A$  überflüssig?  
 $A \in AttrHülle(F - FD (2) \cup (C \rightarrow \emptyset), C)$ , da  $C \rightarrow BEF, E \rightarrow ABC$ .

Damit erhält man für FD (2):  $C \rightarrow \emptyset$ .

**Betrachte FD (3):**

- Ist  $A$  überflüssig?  
 $A \notin AttrHülle(F - FD (3) \cup (E \rightarrow BC), E)$ .
- Ist  $B$  überflüssig?  
 $B \in AttrHülle(F - FD (3) \cup (E \rightarrow AC), E)$ , da  $E \rightarrow AC, C \rightarrow BEF$ .

- Ist  $C$  überflüssig?  
 $C \in \text{AttrHülle}(F - \text{FD (3)} \cup (E \rightarrow A), E)$ , da  $E \rightarrow A, A \rightarrow C$ .

Damit erhält man für FD (3):  $E \rightarrow A$ .

**Betrachte FD (4):**

- Ist  $C$  überflüssig?  
 $C \notin \text{AttrHülle}(F - \text{FD (4)} \cup (F \rightarrow D), F)$ .
- Ist  $D$  überflüssig?  
 $D \notin \text{AttrHülle}(F - \text{FD (4)} \cup (F \rightarrow C), F)$ .

Damit bleibt FD (4) unverändert.

**Betrachte FD (5):**

- Ist  $B$  überflüssig?  
 $B \notin \text{AttrHülle}(F - \text{FD (5)} \cup (C \rightarrow EF), C)$ .
- Ist  $E$  überflüssig?  
 $E \notin \text{AttrHülle}(F - \text{FD (5)} \cup (C \rightarrow BF), C)$ .
- Ist  $F$  überflüssig?  
 $F \notin \text{AttrHülle}(F - \text{FD (5)} \cup (C \rightarrow BE), C)$ .

Damit bleibt FD (5) unverändert.

3. Entferne die FDs der Form  $\alpha \rightarrow \emptyset$ .

Bisheriges Zwischenergebnis:

$$\begin{array}{l}
 A \rightarrow C \\
 C \rightarrow \emptyset \\
 E \rightarrow A \\
 F \rightarrow CD \\
 C \rightarrow BEF
 \end{array} \tag{6}$$

FD (6) wird eliminiert.

4. Fasse mittels der Vereinigungsregel FDs der Form  $\alpha \rightarrow \beta_1, \dots, \alpha \rightarrow \beta_n$  zusammen.

Bisheriges Zwischenergebnis:

$$F_c = \left\{ \begin{array}{l} A \rightarrow C \\ E \rightarrow A \\ F \rightarrow CD \\ C \rightarrow BEF \end{array} \right.$$

Es werden keine FDs vereinigt, da es keine zwei FDs mit gleicher linker Seite gibt.

$F_c$  ist eine kanonische Überdeckung zur Ausgangsmenge  $F$ .

### Hausaufgabe 3

Ist die kanonische Überdeckung  $F_c$  einer Menge  $F$  von funktionalen Abhängigkeiten eindeutig? Begründen Sie Ihre Antwort oder finden Sie ein Gegenbeispiel.

**Lösung:**

Die kanonische Überdeckung  $F_c$  zu einer Menge von funktionalen Abhängigkeiten  $F$  ist nicht eindeutig.

Begründung: Im Algorithmus zur Bestimmung der kanonischen Überdeckung ist nicht festgelegt, in welcher Reihenfolge die FDs bearbeitet werden.

Als Beispiel seien folgende funktionale Abhängigkeiten gegeben:

1.  $A \rightarrow BC$
2.  $B \rightarrow AC$

Wird die erste FD in der Rechtsreduktion zuerst abgearbeitet, ergibt sich:

$$F_c = \{A \rightarrow B, B \rightarrow AC\}$$

Wird die zweite FD in der Rechtsreduktion zuerst abgearbeitet, erhält man hingegen:

$$F_c = \{A \rightarrow BC, B \rightarrow A\}$$

#### Hausaufgabe 4

Gegeben sei ein erweitertes Universitätsschema mit den folgenden zusätzlichen Relationen *StudentenGF* und *ProfessorenF*:

*StudentenGF* : {[MatrNr : integer, Name : varchar(20), Semester : integer,  
Geschlecht : char, Fakultaet : varchar(20)]}  
*ProfessorenF* : {[PersNr : integer, Name : varchar(20), Rang : char(2),  
Raum : integer, Fakultaet : varchar(20)]}

Die erweiterten Tabellen sind auch in der Webschnittstelle angelegt.

- a) Ermitteln Sie den Männeranteil an den verschiedenen Fakultäten in SQL!
- b) Ermitteln Sie in SQL die Studenten, die alle Vorlesungen ihrer Fakultät hören. Geben Sie zwei Lösungen an, höchstens eine davon darf auf Abzählen basieren.

#### Lösung:

```
a) with
FakultaetTotal as (
  select Fakultaet, count(*) as Total
  from StudentenGF
  group by Fakultaet),
FakultaetMaenner as (
  select Fakultaet, count(*) as Maenner
  from StudentenGF
  where Geschlecht = 'M'
  group by Fakultaet)
select
  FakultaetTotal.Fakultaet,
  (case when Maenner is null then 0 else Maenner end)/(total*1.00)
from FakultaetTotal left outer join FakultaetMaenner
on FakultaetTotal.Fakultaet = FakultaetMaenner.Fakultaet
```

Wir müssen beachten, dass nicht jede Fakultät Männer beherbergt, weswegen diese Fakultäten (in der Standardausprägung im SQL Interface ist dies für Theologie der Fall) dann aus dem Ergebnis herausfallen würden. Aus diesem Grund verwenden wir einen `left outer join` um die Zahl der Männer und die Zahl der Studenten insgesamt zu verbinden, wodurch auch die Theologie-Fakultät im Ergebnis enthalten ist, auch wenn es keine Männer gibt.

Das `case`-Konstrukt dient in der oberen Anfrage dazu, den `NULL` Wert, die durch den `Left Join` für die Anzahl der Männer entstehen, wenn es keine Männer gibt, durch die Zahl 0 zu ersetzen. Alternativ ist dies möglich, indem man `coalesce(maenner,0) / (total*1.00)` verwendet.

Alternativ können wir das `case`-Konstrukt verwenden, um die Anzahl der Männer an den jeweiligen Fakultäten zu ermitteln. Den Männeranteil erhalten wir dann, indem wir die Anzahl der Männer durch die Gesamtanzahl der Studenten an der Fakultät teilen.

```
select Fakultaet,
       (sum(case when Geschlecht = 'M' then 1.00 else 0.00 end)) / count(*)
from StudentenGF
group by Fakultaet
```

- b) Wir fordern hier, dass es keine Vorlesung an der Fakultät des Studenten (d.h. von einem Professor der gleichen Fakultät gelesen) geben darf, die vom Studenten nicht gehört wird.

```
select s.*
from StudentenGF s
where not exists (select *
                  from Vorlesungen v, ProfessorenF p
                  where v.gelesenVon = p.PersNr
                        and p.Fakultaet = s.Fakultaet
                        and not exists
                            (select *
                             from hoeren h
                             where h.VorlNr = v.VorlNr
                                   and h.MatrNr = s.MatrNr));
```

Alternativ:

```
select * from StudentenGF s
where
  (select count(*)
   from Vorlesungen v, ProfessorenF p
   where v.gelesenVon = p.PersNr and p.Fakultaet = s.Fakultaet)
=
  (select count(*)
   from hoeren h, Vorlesungen v, ProfessorenF p
   where
     h.MatrNr = s.MatrNr and
     h.VorlNr = v.VorlNr and
```

```
p.PersNr = v.gelesenVon and  
p.Fakultaet = s.Fakultaet
```

```
)
```