2. Übungsblatt

3.0 VU Datenmodellierung

2. Dezember 2015

Allgemeines

In diesem Übungsteil sollten Sie Aufgabenstellungen aus den Bereich SQL und Normalformentheorie bearbeiten.

Lösen Sie die Beispiele **eigenständig**, denn bei der Prüfung und in der Praxis sind Sie auch auf sich alleine gestellt. Wir weisen Sie darauf hin, dass abgeschriebene Lösungen mit 0 Punkten beurteilt werden.

Geben Sie ein einziges PDF Dokument ab (max. 5MB). Erstellen Sie Ihr Abgabedokument computerunterstützt. Wir akzeptieren keine gescannten handschriftlichen PDF-Dateien.

Deadlines

bis 18.12.	**06:55 Uhr**	Upload über TUWEL
ab 12.01.	13:00 Uhr	Feedback in TUWEL verfügbar
bis 12.01.	23:59 Uhr	Reservierung eines Termins für das Tutorengespräch
bis 12.01.	23:59 Uhr	Reservierung eines Termins für das Abgabegespräch

(Terminreservierung für Tutoren- und Abgabegespräch über TUWEL)

Tutorengespräch

1. Sie müssen sich über TUWEL zu einem Tutorengespräch anmelden. Bitte machen Sie das rechtzeitig, je später Sie sich anmelden, umso eingeschränkter ist das Terminangebot.

2. Wenn Sie kein Blatt abgegeben haben, sind Sie nicht zum Tutorengespräch zugelassen.

3. Sie kommen mit Ihrem Studierendenausweis zu der von Ihnen reservierten Zeit vorbei, und absolvieren das Tutorengespräch. Sie erhalten dabei wertvolles Feedback zu Ihrem Übungsblatt. Das Gespräch ist verpflichtend und dauert ca. 30 Minuten.

4. Nützen Sie das Tutorengespräch um wertvolles Feedback zu Ihrem Übungsblatt zu erhalten: Gehen Sie vor dem Gespräch Ihre korrigierte Lösungen durch, identifizieren Sie Unklarheiten und stellen Sie den TutorInnen entsprechende **Fragen**. Trauen Sie sich zuzugeben, wenn Sie etwas nicht verstanden haben. Ihre Fragen haben keinerlei (negativen) Einfluss auf Ihre Note!

Abgabegespräch

1. Sie müssen sich über TUWEL zu einem Abgabegespräch anmelden. Bitte machen Sie das rechtzeitig, je später Sie sich anmelden, umso eingeschränkter ist das Terminangebot.

2. Sie kommen mit Ihrem Studierendenausweis zu der von Ihnen reservierten Zeit vorbei, und absolvieren das Abgabegespräch. Stoffgebiet des Abgabegesprächs sind die mit dem Übungsblatt abgedeckten Themengebiete. Wir setzen voraus, dass Sie sich mit Ihrer korrigierten Abgabe auseinandergesetzt haben. Das Abgabegespräch hat Prüfungscharakter.

3. Sie absolvieren Ihr Abgabegespräch gemeinsam mit anderen KollegInnen. Das Gespräch dauert ca. 60 Minuten.

4. Sie können auf die Abgabe maximal 15 Punkte erreichen. Diese setzen sich wie folgt zusammen:

   5 Punkte auf das Übungsblatt

   10 Punkte auf das Abgabegespräch

5. Die Assistenten tragen die Punkte des Abgabegesprächs in TUWEL ein und Sie sehen dort, wieviele Punkte Sie bekommen haben.

6. Falls Sie nicht zu Ihrem Gesprächstermin erscheinen, bekommen Sie automatisch 0 Punkte auf das Gespräch.

SQL

Aufgabe 1 (eSQL) *[1.0 Punkte]*
Lösen Sie alle 10 (inkl. Unterpunkte) in unserer Online-Übungsumgebung zur Verfügung gestellten SQL-Aufgaben. Sie erreichen die Umgebung über TUWEL: Wählen Sie im Abschnitt "2.Übungsblatt" die Aktivität **eSQL Tool**. Sie benötigen kein weiteres Passwort, die Authentifizierung erfolgt über TUWEL.

Der verpflichtende SQL-Test am Ende des Semesters wird über die selbe Plattform abgewickelt. Es empfiehlt sich daher zusätzlich auch mit Beispielen aus den vorigen Semestern zu üben.

Aufgabe 2 (Allquantifizierung) *[0.5 Punkte]*

In einem all-quantifizierten Ausdruck wird z.B. nach jenen Studenten gesucht, die bereits *alle* angebotenen Vorlesungen besucht haben. Diesen Sachverhalt kann man in klassischer Prädikatenlogik erster Stufe wie folgt ausdrücken:

$$\varphi(k) \equiv \mathit{Student}(k) \wedge \forall p\ (\mathit{Vorlesung}(p) \rightarrow \mathit{besucht}(k, p))$$

Erklären Sie zwei Methoden, durch die ein Allquantor in SQL ausgedrückt werden kann, zunächst allgemein und dann auch an dem oben angeführten einfachen Beispiel. Geben Sie hierbei die entsprechenden Abfragen in SQL-Syntax an.

```
Student(skey)
Vorlesung(vkey)
besucht(Student.skey, Vorlesung.vkey)
```

Lösung:
Es gibt verschiedene Möglichkeiten einen All-Quantor in SQL auszudrücken. In der Vorlesung wurden vier verschiedene Herangehensweisen vorgestellt:

- Durch Ausnutzung der Äquivalenz $\forall x\ \varphi(x) \equiv \neg\exists x\ \neg\varphi(x)$, welche mittels einem `NOT EXISTS`, `NOT EXISTS` bzw. einem `NOT EXISTS`, `NOT IN` Konstrukt in SQL ausgedrückt werden kann. D.h. anstatt, dass wir jene Einträge suchen für die alle Werte eine bestimmte Eigenschaft erfüllen suchen wir jene Einträge, für die es keinen Wert gibt, der diese Eigenschaft nicht erfüllt.

  Im konkreten Beispiel heißt das, dass wir anstatt jene Studenten zu suchen die *alle Vorlesungen besucht* haben jene Studenten finden wollen, für die es *keine Vorlesung* gibt, die sie *nicht besucht* haben.

  ```
  SELECT s.skey
  FROM Student s
  WHERE not exists (SELECT * FROM Vorlesung v
                    WHERE NOT EXISTS (SELECT *
                                      FROM besucht b
                                      WHERE b.skey=s.skey AND
                                            b.vkey=v.vkey))
  ```

  bzw.

  ```
  SELECT s.skey
  FROM Student s
  WHERE not exists (SELECT * FROM Vorlesung v
                    WHERE v.vkey NOT IN (SELECT b.vkey
                                         FROM besucht b
                                         WHERE b.skey=s.skey))
  ```

- Mittels einem Konstrukt aus `NOT EXISTS` und `EXCEPT`, welches testet ob die Menge aller Tupel der betrachteten Domäne eine Teilmenge jener Tupel sind, welche die gewünschte Eigenschaft erfüllen. Daraus folgt, dass in der Tat alle Tupel der betrachteten Domäne die gewünschte Eigenschaft erfüllen. Dies kann in SQL durch Ausnutzung der Äquivalenz $A \subseteq B \Leftrightarrow A - B = \emptyset$ ausgedrückt werden. D.h. man überprüft (mittels `NOT EXISTS`) ob die Differenz (gebildet mittels `EXCEPT`) zwischen der Menge aller Tupeln der betrachteten Domäne und der Menge jener Tupeln welche die gewünschte Eigenschaft erfüllen kein Tupel enthält.

  Im konkreten Beispiel bedeutet das, dass wir jene Studenten suchen, für welche die Menge aller Vorlesungen eine Teilmenge der von dem Studenten besuchten Vorlesungen bildet. Dies kann in SQL überprüft werden indem wir für jeden Studenten die von dem Studenten besuchten Vorlesungen von der Menge aller Vorlesungen abziehen, und genau jene Studenten ausgeben für welche das Ergebnis dieser Mengendifferenz kein Tupel (also keine Vorlesung) enthält.

```
SELECT s.skey
FROM Student s
WHERE NOT EXISTS ((SELECT v.vkey
                   FROM Vorlesung v)
                  EXCEPT
                   (SELECT b.vkey
                    FROM besucht b
                    WHERE b.skey=s.skey))
```

- Mittels einem Konstrukt aus doppelter Mengendifferenz `EXCEPT`, indem man die Übersetzung des Divisionsoperators der relationalen Algebra in die primitiven Operatoren anwendet, d.h. durch Ausnutzung der Eigenschaft:

$$R \div S = \pi_{\mathcal{R}-\mathcal{S}}(R) - \pi_{\mathcal{R}-\mathcal{S}}((\pi_{\mathcal{R}-\mathcal{S}}(R) \times S) - R)$$

  Am konkreten Beispiel bedeutet das, dass man zuerst alle Paare von Studenten und Vorlesungen bildet (mittels Kreuzprodukt). Von dieser Menge entfernt man (mittels `EXCEPT`) alle Paare von Studenten und Vorlesungen, wo der Studierende die Vorlesung tatsächlich besucht hat. Übrig bleiben nun nur Paare aus Studenten und Vorlesungen, welche angeben welche Studenten welche Vorlesungen *nicht* besucht haben. Insbesonders gibt es für Stundeten die alle Vorlesungen besucht haben kein Tupel mehr in der verbleibenden Menge. Im nächsten Schritt projizieren wir die verbleibenden Paare auf die Studenten. Dadurch erhalten wir eine Liste mit genau jenen Studenten, die mindestens eine Vorlesung nicht besucht haben. Dabei handelt es sich nun genau um jene Studenten, die wir *nicht* ausgeben wollen. Daher berechnen wir nun noch die Differenz zwischen der Menge aller Studenten und der erhaltenen Menge an Studenten. D.h., wir entfernen aus der Menge aller Studenten genau jene, welche mindestens eine Vorlesung nicht besucht haben. Übrig bleiben demnach genau jene Studenten, die alle Vorlesungen besucht haben, also genau die gesuchten Studenten.

```
(SELECT sall.skey FROM Student sall)
EXCEPT
(SELECT sNotAll.skey FROM
       ((SELECT s.skey, s.vkey
         FROM Student s, Vorlesung v)
        EXCEPT
        (SELECT * FROM besucht)) as sNotAll)
```

- Mittels `COUNT`. Bei diesem Ansatz werden zunächst alle Tupel in der betrachteten Domäne gezählt. Dann werden die Tupel in der betrachteten Domäne gezählt, die überdies die gewünschte Bedingung erfüllen. Stimmen die beiden Ergebnisse überein, erfüllen alle betrachteten Tupel die Bedingung.

  Am konkreten Beispiel bedeutet das, dass man zunächst alle Vorlesungen zählt, und genau jene Studenten ausgibt für die die Anzahl der besuchten Vorlesungen gleich ist der Anzahl aller Vorlesungen.

```
SELECT skey
FROM Student NATURAL JOIN besucht b
GROUP BY skey
HAVING COUNT(*) = (SELECT COUNT(*) FROM Vorlesung);
```

Normalformentheorie

Aufgabe 3 (Funktionale Abhängigkeiten) *[0.5 Punkte]*
Gegeben sei das Relationsschema $\mathcal{R} = \{FN, LN, TO, DOB, OC\}$ der folgenden relation R:

R				
FN	LN	TO	DOB	OC
Mickey	Mouse	Duckburg	1928	Detektiv
Donald	Duck	Duckburg	1934	Matrose
Tom	Bombadil	Old Forest	1954	Gärtner
Bruce	Wayne	Gotham City	1915	Batman
Tom	Turbo	Wien	1993	Detektiv

Welche der folgenden FDs sind auf der Relation R gültig?

1. $\{DOB\} \rightarrow \{OC, LN\}$
2. $\{OC, LN\} \rightarrow \{FN\}$
3. $\{TO, OC\} \rightarrow \{FN, LN, TO, DOB, OC\}$
4. $\{FN\} \rightarrow \{DOB, OC\}$
5. $\{TO\} \rightarrow \{LN\}$
6. $\{LN\} \rightarrow \{TO\}$
7. $\{FN, LN, TO\} \rightarrow \{DOB, OC, LN\}$
8. $\{OC\} \rightarrow \{FN\}$
9. $\{OC\} \rightarrow \{OC\}$

Lösung:
Die Funktionalen Abhängigkeiten 1,2,3,6,7 und 9 sind gültig.

Aufgabe 4 (Armstrong Axiome) *[0.5 Punkte]*

(a) Zeigen Sie, dass für alle $n, m \in \mathbb{N}$ folgendes Axiom aus den Armstrong Axiomen hergeleitet werden kann:
Seien $A_1, \ldots, A_m, B_1, \ldots, B_n$ Attribute. Dann gilt $A_1 \ldots A_m \to B_1 \ldots B_n$ dann und genau dann, wenn für alle $i \in \mathbb{N}$ mit $1 \leq i \leq n$ $A_1 \ldots A_m \to B_i$ gilt.

(b) Gegeben ist ein Relationenschema $\mathcal{R} = ABCDEF$ und zwei Mengen F_1 und F_2 von funktionalen Abhängigkeiten.

$$F_1 = \{AC \to B, CE \to D, A \to E\}$$
$$F_2 = \{AC \to BCDE, CE \to D, A \to E\}$$

Sind F_1 und F_2 äquivalent? Begründen Sie Ihre Antwort formal und dokumentieren Sie den Lösungsweg.

Lösung:

(a) Zuerst nehmen wir an, dass $A_1 \ldots A_m \to B_1 \ldots B_n$ gilt. Durch n-malige Anwendung der Dekomposition erhalten wir direkt $A_1 \ldots A_m \to B_i$ für alle $1 \leq i \leq n$.
Um die andere Richtung der Äquivalenz zu zeigen, nehmen wir nun $A_1 \ldots A_m \to B_i$ für alle $1 \leq i \leq n$ an. Durch n-maliges anwenden der Vereinigung erhalten wir $A_1 \ldots A_m \to B_1 \ldots B_n$, und somit ist die Äquivalenz gezeigt.

(b) Ja, da $F_1^+ = F_2^+$. Es reicht aus, folgende zwei Eigenschaften zu zeigen:

$F_1 \subseteq F_2^+$:

Es reicht zu zeigen, dass $AC \to B$ aus den FDs in F_2 hergeleitet werden kann. Dazu wendet man das Axiom der Dekomposition einmalig auf die funktionela Abhängigkeit $AC \to BCDE$ an.

$F_2 \subseteq F_1^+$:

Es reicht zu zeigen, dass $AC \to BCDE$ aus den FDs in F_1 hergeleitet werden kann. Aus $A \to E$ können wir mithilfe der Verstärkung $AC \to CE$ ableiten. Somit folg aus $AC \to CE$ und $AC \to B$ durch Vereinigung $AC \to BCE$. Aus $AC \to CE$ und $CE \to D$ folgt mit Transitivität $AC \to D$, und mithilfe der Vereinigung erhalten wir somit aus $AC \to D$ und $AC \to BCE$ die Abhängigkeit $AC \to BCDE$.

Aufgabe 5 (Kanonische Überdeckung) *[0.5 Punkte]*

Bestimmen Sie eine kanonische Überdeckung folgender Mengen funktionaler Abhängigkeiten über dem Relationenschema $\mathcal{R}$ und dokumentieren Sie den Lösungsweg.

(a) $\mathcal{R} = ABCDEFG$,
$F^1 = \{ABD \to EF,\ BC \to AB,\ BC \to E,\ G \to F,\ AC \to FE,\ B \to C\}$

(b) $\mathcal{R} = ABCDEF$,
$F^2 = \{A \to E,\ BC \to F,\ FE \to C,\ BD \to AF,\ DE \to EF,\ F \to ADC,\ D \to F\}$

Lösung:

(a) Die Kanonische Überdeckung lässt sich mithilfe der folgenden vier Schritten berechnen:

- Dekomposition:

  Mit der Regel der Dekomposition erhalten wir die Menge
  $\{ABD \to E,\ ABD \to F,\ BC \to A,\ BC \to B,\ BC \to E,\ G \to F,\ AC \to F,\ AC \to E,\ B \to C\}$.

- Linksreduktion:

  Wir müssen uns jetzt sukzessive die Frage stellen, ob für eine funktionale Abhängigkeit ein Attribut auf der linken Seite überflüssig ist. Beginnen wir mit $ABD \to E$. Wenn wir das Attribut D aus der linken Seite streichen, ist $AB \to E$ noch immer eine gültige funktionale Abhängigkeit: Aus AB kann man B ableiten, und da $B \to C$ gilt, gilt auch $B \to BC$ und somit $AB \to BC$. Da $BC \to E$ eine geltende funktionale Abhängigkeit ist, gilt somit $AB \to E$. Diese funktionale Abhängigkeit ersetzt $ABD \to E$, und muss ebenfalls auf reduzierbarkeit untersucht werden. Mit diesem Verfahren ergibt die Linksreduktion das Folgende:

$$
\begin{aligned}
ABD \to E &: \text{wird reduziert zu } AB \to E \\
AB \to E &: \text{wird reduziert zu } B \to E \\
ABD \to F &: \text{wird reduziert zu } AB \to F \\
AB \to F &: \text{wird reduziert zu } B \to F \\
BC \to A &: \text{wird reduziert zu } B \to A \\
BC \to B &: \text{wird reduziert zu } B \to B \\
BC \to E &: \text{wird reduziert zu } B \to E \\
G \to F &: \text{bereits reduziert} \\
AC \to F &: \text{bereits reduziert} \\
AC \to E &: \text{bereits reduziert} \\
B \to C &: \text{bereits reduziert}
\end{aligned}
$$

Somit ist die Menge der funktionalen Abhängigkeiten nach der Linksreduktion gegeben als
$\{B \to F,\ B \to A,\ B \to E,\ G \to F,\ AC \to F,\ AC \to E,\ B \to C\}$ (Die Abhängigkeit $B \to B$ wurde sofort gestrichen, da sie trivial ist; weiters wurde die doppelte Abhängigkeit $B \to E$ gestrichen).

- Rechtsreduktion:

  Bei der Rechtsreduktion muss man für jede funktionelle Abhängigkeit überprüfen, ob sie noch gültig ist, selbst wenn man sie streicht. Das heisst für die FD $B \to F$ gilt es festzustellen, ob sich aus B das Attribut F ableiten lässt, selbst wenn man $B \to F$ aus der Menge der FDs streicht. Für $B \to F$ ist das der Fall: Aus B folgt A und C, somit gilt auch $B \to AC$. Da $AC \to F$ gilt, kann $B \to F$ gestrichen werden. Somit erhalten wir mit der Rechtsreduktion das Folgende:

$$\begin{aligned}
B \to F &: \text{wird gestrichen} \\
B \to A &: \text{wird nicht gestrichen} \\
B \to E &: \text{wird gestrichen} \\
G \to F &: \text{wird nicht gestrichen} \\
AC \to F &: \text{wird nicht gestrichen} \\
AC \to E &: \text{wird nicht gestrichen} \\
B \to C &: \text{wird nicht gestrichen}
\end{aligned}$$

  Somit ist die Menge der funktionalen Abhängigkeiten nach der Rechtsreduktion gegeben als
  $\{B \to A,\ G \to F,\ AC \to F,\ AC \to E,\ B \to C\}$.

- Vereinigung: Die FDs $B \to A$ und $B \to C$ werden vereinigt, als auch die FDs $AC \to F$ und $AC \to E$.

$F_C^1 = \{B \to AC,\ G \to F,\ AC \to EF\}$

(b) Da bei der Links- bzw. Rechtsreduktion die Reihenfolge der gewählten Funktionalen Abhängigkeiten nicht fix ist, können unterschiedliche (jedoch äquivalente) kanonische Überdeckungen berechnet werden.

$F_C^2 = \{A \to E,\ BC \to F,\ F \to ADC,\ D \to F\}$
(oder auch $\{A \to E,\ BC \to F,\ D \to AF,\ F \to CD\}$)

Aufgabe 6 (Schlüsselbestimmung) *[0.5 Punkte]*

Bestimmen Sie für die folgenden Relationenschemata samt funktionalen Abhängigkeiten alle Schlüssel und alle Superschlüssel.

(a)

$$\begin{aligned}
\mathcal{R} &= ABCDE \\
F_d &= \{A \to C,\ CD \to B,\ D \to C\}
\end{aligned}$$

Lösung:
Der einzige Schlüssel ist ADE. Die Menge der Superschlüssel:

$$\{ADE, ABDE, ACDE, ABCDE\}$$

(b)

$$\mathcal{R} = ABCD$$
$$F_d = \{AB \to C,\ DC \to A,\ C \to D\}$$

Lösung:
Die Schlüssel sind AB und BC. Die Menge der Superschlüssel:

$$\{AB, BC, ABC, ABD, BCD, ABCD\}$$

(c)

$$\mathcal{R} = ABCDE$$
$$F_d = \{AC \to BE,\ BD \to CE,\ B \to A,\ E \to B\}$$

Lösung:
Die Schlüssel sind BD, ACD und DE. Die Menge der Superschlüssel:

$$\{BD, DE, ABD, BCD, BDE, ADE, CDE, ACD, ACDE, ABDE, ABCD, BCDE, ABCDE\}$$

Aufgabe 7 (Normalformen) *[0.5 Punkte]*

Gegeben sei folgendes Relationenschema samt funktionalen Abhängigkeiten:

$$\mathcal{R} = ABCDE$$
$$F_d = \{AB \to CD,\ BC \to E,\ BD \to AE,\ CE \to AD\}$$

Geben Sie an, ob $\mathcal{R}$

(a) in dritter Normalform ist,

(b) in Boyce-Codd-Normalform ist,

und begründen Sie Ihre Antworten.

Lösung:
AB, BC und BD sind die Schlüssel von $\mathcal{R}$. Die drei Anforderungen für funktionale Abhängigkeiten der Form $\alpha \to B$, die man benötigt um ein Relationenschema $\mathcal{R}$ auf dritte Normalform bzw. Boyce-Codd Normalform zu untersuchen, sind wie folgt:

1. $B \in \alpha$, d.h. die FD ist trivial
2. α ist Superschlüssel von $\mathcal{R}$
3. das Attribut B ist in einem Schlüssel von $\mathcal{R}$ enthalten

Dabei ist ein Relationsschema $\mathcal{R}$ in Boyce-Codd Normalform, wenn für jede funktionale Abhängigkeit Bedingungen 1 oder 2 erfüllt ist, und $\mathcal{R}$ is in dritter Normalform wenn für jede funktionale Abhängigkeit Bedingungen 1,2 oder 3 erfüllt ist. Um dies zu untersuchen wenden wir die Dekomposition an, und kennzeichnen für jede FD, welche Bedingungen sie erfüllt:

$$\mathcal{F} = \{\underbrace{AB \to C}_{2,3}, \underbrace{AB \to D}_{2,3}, \underbrace{BC \to E}_{2}, \underbrace{BD \to A}_{2,3}, \underbrace{BD \to E}_{2}, \underbrace{CE \to A}_{3}, \underbrace{CE \to D}_{3}\}$$

Für jede FD trifft entweder Bedingung 2 oder 3 zu. Zudem gibt es mindestens eine FD (z.B. $CE \to A$), für die Bedingung 1 und 2 nicht zutreffen. Deswegen ist $\mathcal{R}$ in dritter Normalform, aber nicht in BCNF.

Aufgabe 8 (Synthesealgorithmus) *[0.5 Punkte]*

Gegeben sei folgendes Relationenschema samt funktionalen Abhängigkeiten:

$$\mathcal{R} = ABCDEF$$
$$F_d = \{AB \to C,\ CD \to E,\ C \to B,\ B \to E,\ BF \to D,\ E \to F\}$$

Gesucht ist eine verlustlose und abhängigkeitserhaltende Zerlegung in dritter Normalform. Wenden Sie hierzu den Synthesealgorithmus an und dokumentieren Sie das Ergebnis der einzelnen Schritte. Bestimmen Sie alle Schlüssel von $\mathcal{R}$ und allen Relationen der Zerlegung.

Lösung:

1. Bestimmung der kanonischen Überdeckung:

   $$F_c = \{AB \to C,\ C \to B,\ B \to DE,\ E \to F\}$$

   (Anmerkung zur Linksreduktion: Die FD $BF \to D$ kann zu $B \to D$ reduziert werden. Ausgehend von dem Attribut B und den FDs $B \to E$ und $E \to F$ gilt $B \to F$ und somit auch $B \to BF$. Da man beim Aufbau der Attributenhülle im Zuge der Linksreduktion <u>nicht</u> die FD streicht, die man links reduziert, erhält man somit mit $BF \to D$ die Kette $B \to BF \to D$ und somit $B \to D$.)

2. Erstelle Relationenschemata für jedes Element von F_c:

Relationenschema	Geltende FDs
$\mathcal{R}_1 = ABC$	$F_1 = \{AB \to C,\ C \to B\}$
$\mathcal{R}_2 = BC$	$F_2 = \{C \to B\}$
$\mathcal{R}_3 = BDE$	$F_3 = \{B \to DE\}$
$\mathcal{R}_4 = EF$	$F_4 = \{E \to F\}$

3. Bestimmung aller Kandidatenschlüssel von $\mathcal{R}$ bzgl. F_c: AB, AC.

   Der Schlüssel AB ist in einem der erzeugten Teilschemata enthalten ($\mathcal{R}_1$) Es muss kein weiteres Schema erzeugt werden.

4. Das Schema $\mathcal{R}_2$ kann eliminiert werden, da $\mathcal{R}_2 \subseteq \mathcal{R}_1$ gilt.

Ergebnis (Schlüssel sind unterstrichen):

Relationenschema	Geltende FDs
$\mathcal{R}_1 = \underline{AB}C$	$F_1 = \{AB \to C, C \to B\}$
$\mathcal{R}_3 = \underline{B}DE$	$F_3 = \{B \to DE\}$
$\mathcal{R}_4 = \underline{E}F$	$F_4 = \{E \to F\}$

Aufgabe 9 (Dekompositionsalgorithmus) *[0.5 Punkte]*

Gegeben sei folgendes Relationenschema samt funktionalen Abhängigkeiten:

$$R = ABCDEFG$$
$$F_d = \{ABD \to EC,\ DEF \to G,\ EC \to BF\}$$

Gesucht ist eine verlustlose Zerlegung in Boyce-Codd-Normalform. Wenden Sie hierzu den Dekompositionsalgorithmus an und dokumentieren Sie das Ergebnis der einzelnen Schritte. Bestimmen Sie alle Schlüssel von $\mathcal{R}$ und allen Relationen der Zerlegung. Ist die Zerlegung abhängigkeitserhaltend? Wenn die Zerlegung nicht abhängigkeitserhaltend ist, geben Sie an, welche Abhängigkeiten verloren gegangen sind.

Lösung:
Die Schlüssel von $\mathcal{R}$ sind ABD und $ACDE$. Die FDs $DEF \to G$ und $EC \to BF$ verletzen die BCNF, da sie nicht trivial sind, und DEF bzw. EC keine Superschlüssel sind. Daher haben wir 2 Möglichkeiten für die Zerlegung:

- Zuerst entscheiden wir uns für $DEF \to G$. Damit erhalten wir:

  $\mathcal{R}_1 = DEFG$ $\quad F_1 = \{DEF \to G\}$ $\quad$ Schlüssel: DEF
  $\mathcal{R}_2 = ABCDEF$ $\quad F_2 = \{ABD \to EC,\ EC \to BF\}$ $\quad$ Schlüssel: $ABC, ACDE$

  $\mathcal{R}_1$ ist in BCNF, $\mathcal{R}_2$ muss jedoch noch weiter zerlegt werden, da $EC \to BF$ die BCNF verletzt:

  $\mathcal{R}_{21} = BCEF$ $\quad F_{21} = \{EC \to BF\}$ $\quad$ Schlüssel: EC
  $\mathcal{R}_{22} = ACDE$ $\quad F_{22} = \{\}$ $\quad$ Schlüssel: $ACDE$

  Damit erhalten wir die Zerlegung

  $\mathcal{R}_1 = DEFG$ $\quad F_1 = \{DEF \to G\}$ $\quad$ Schlüssel: DEF
  $\mathcal{R}_{21} = BCEF$ $\quad F_{21} = \{EC \to BF\}$ $\quad$ Schlüssel: EC
  $\mathcal{R}_{22} = ACDE$ $\quad F_{22} = \{\}$ $\quad$ Schlüssel: $ACDE$

  Diese Zerlegung ist nicht abhängigkeitserhaltend, da $ABD \to EC$ verloren geht.

- Alternative Lösung wenn $EC \to BF$ ausgewählt wird:

$\mathcal{R}_1 = ECBF$	$F_1 = \{EC \to BF\}$	Schlüssel: EC
$\mathcal{R}_2 = ACDEG$	$F_2 = \{\}$	Schlüssel: $ACDEG$

  $\mathcal{R}_1$ und $\mathcal{R}_2$ sind beide in BCNF und die Zerlegung ist nicht abhängigkeitserhaltend, da $ABD \to EC$ und $DEF \to G$ verloren gehen.