Exercise Sheet 2 (WS 2019) – Sample Solution

3.0 VU Datenmodellierung / 6.0 VU Datenbanksysteme

About the exercises

General information

In this part of the exercises you are asked to solve tasks in the areas of SQL, functional dependencies, and relational normal forms.

We recommend you to solve the problems **on your own** (it is a good preparation for the exam – and also for your possible future job – to carry out the tasks autonomously). Please note that if we detect duplicate submissions or any plagiarism, both the "original" and the "copy" will be awarded 0 points.

Your submission must consist of a single, typeset PDF document (max. 5MB). We do not accept PDF files with handwritten solutions.

In total there are 8 tasks and at most 15 points that can be achieved on the entire sheet.

Deadlines

until 06.11.	**12:00 pm**:	Solve the SQL tasks with the online-tool
until 13.11.	**12:00 pm**:	Upload your solution to TUWEL
from 22.11.	1:00 pm:	Evaluation and feedback is available in TUWEL

Consultation Hours (optional)

In the week before the deadline there are consultation hours held by tutors. If you have problems understanding the topics of the exercise sheet or questions about the exercises, you are welcome to just drop in at these consultation hours. The tutors will gladly answer your questions and help you understand the subjects.

The goal of these consultation hours is to help you with **understanding the topics and specific tasks** of the exercise sheet. The tutors will not solve your exercises or check your answers before you hand them in.

Participation is completely voluntary — dates and locations of the consultation hours can be found in TUWEL.

Debriefing (optional)

A few days after you received your feedback and grading of this exercise sheet, there is the opportunity to go through the tasks in small groups (max. 25 persons). The (relative) small group size is intended to enable an active participation. Each of these groups will be held by an assistant. The specific agenda of these meetings will depend on the interests and question of the participants (i.e., you). The main objectives are answering your open questions and resolving your remaining issues regarding the exercises. Therefore, please have a look at your feedback and evaluation to identify such problems before you attend the class. When participating, dare to ask your questions – no question has a (negative) impact on your grade.

Participation is absolutely voluntary. Registration in TUWEL is required to keep the size of the groups small. Dates and locations can be found in TUWEL.

Further Questions – TUWEL Forum

If you have any further questions concerning the contents or organization, do not hesitate to ask them on TUWEL forum.

SQL

Task 1 (eSQL)	[5 points]

Solve **the first 10 SQL tasks (Task 1-10)** in our SQL e-learning tool (eSQL-tool) from the current course (first course in the list). (*Pleas note:* We usually upload more than ten tasks for practicing. You are free to solve these as well – however, only the first 10 tasks will be considered for the grading of this exercise sheet.)

You can access the online tool in TUWEL: Select **eSQL Tool** in the section "2.Übungsblatt". No additional password is needed, you are authenticated via TUWEL. *Please note:* Unfortunately, the interface of the online tool is currently available only in German. However, task descriptions and the description of the database are available in English.

The mandatory SQL test will also take place on exactly the same platform. We therefore recommend practicing also using the exercises from previous coursed (for two more courses, English translations are available).

Caution!
Separate deadline for finishing the SQL tasks:
Wednesday, November 6, 2019, 12:00 pm!

Normal Forms

Task 2 (Functional Dependencies) [0.4 points]

Consider the following relational schema

`Offer (Coffee, Bean, Country, Roasting, Processing, Type)`

with the following instance (sorted by `Coffee`):

Offer					
Coffee	Bean	Country	Roasting	Processing	Type
Alice	Arabica	Brazil	plus	natural	pure
Bob	Arabica	Ecuador	plus	washed	pure
Carol	Arabica	Peru	full	washed	mixed
Carol	Arabica	Honduras	full	natural	mixed
Carol	Robusta	India	plus	washed	mixed
Dan	Arabica	Brazil	full	washed	mixed
Dan	Robusta	India	full	natural	mixed
Eve	Arabica	Indonesia	full	washed	pure
Faythe	Arabica	Ethiopia	plus	natural	mixed
Faythe	Arabica	Guatemala	plus	washed	mixed
Grace	Arabica	Ethiopia	normal	washed	pure
Rupert	Robusta	India	full	washed	pure

Check for each functional dependency below whether it is satisfied by the given instance or not. For each FD, provide an answer (yes/no). If a FD is not satisfied, provide also a counter example. If a FD is satisfied, provide a tuple that, by adding it to the instance, would lead to a violation of the FD.

(a) `Country` → `Bean` **Yes**

A possible tuple would be

- (New, Robusta, Indonesien, full, natur, mischung),

(b) `Bean` → `Country`. **No**

Two (out of several) counter examples are

- (Alice, Arabica, Brasilien, plus, natur, rein) and
  (Bob, Arabica, Ecuador, plus, nass, rein):
  Arabica = *Arabica* but *Brasilien* ≠ *Ecuador*.
- (Eve, Arabica, Indonesien, full, nass, rein) and
  (Faythe, Arabica, Äthiopien, plus, natur, mischung):
  Arabica = *Arabica* but *Indonesien* ≠ *Äthiopien*.

(c) `Coffee, Bean` → `Country`. **No**

The only counter example are

- (Carol, Arabica, Peru, full, nass, mischung) und
  (Carol, Arabica, Honduras, full, natur, mischung):
  Carol, Arabica = *Carol, Arabica*, aber *Peru* ≠ *Honduras*.

(d) `Country` → `Bean, Processing`. **No**

Two out of several counter examples are

- (Alice, Arabica, Brasilien, plus, natur, rein) und
  (Dan, Arabica, Brasilien, full, nass, mischung):
  Brasilien = *Brasilien* aber *Arabica, natur* ≠ *Arabica, nass.*
- (Dan, Robusta, Indien, full, natur, mischung) und
  (Rubert, Robusta, Indien, full, nass, rein):
  Indien = *Indien* aber *Arabica, natur* ≠ *Arabica, nass.*

(e) `Bean, Country, Roasting` → `Coffee, Processing`. **Yes**

- (Bob, Arabica, Brasilien, plus, natur, rein)

  would lead to the FD not being satisfied because of the already existing tuple

  (Alice, Arabica, Brasilien, plus, natur, rein):

  Arabica, Brasilien, plus = *Arabica, Brasilien, plus*, aber
  Bob, natur ≠ *Alice, natur.*

Task 3 (Equivalence of Functional Dependencies) [0.6 points]

(a) Consider the following relational schema $GHIJKL$ and two sets F_1 and F_2 of functional dependencies.

$$F_1 = \{HI \to GJ, G \to KL, IL \to JK, GK \to I\}$$
$$F_2 = \{HI \to GL, G \to KLI, IL \to JK, GKH \to H\}$$

Are F_1 and F_2 equivalent? Please explain your answer using the closures of F_1 and F_2 and show your reasoning.

Lösung:
Yes, $F_1 \equiv F_2$**.** Dies folgt aus $F_1^+ = F_2^+$ (d.h. die Hüllen der beiden Mengen an FDs sind identisch). Um die Gleichheit $F_1^+ = F_2^+$ zu zeigen werden wir nicht die beiden Hüllen F_1^+ und F_2^+ berechnen, sondern nur zeigen dass für jede FD $F \in F_1$ auch $F \in F_2^+$ gilt, und umgekehrt für für jede FD $F \in F_2$ auch $F \in F_1^+$ gilt. Daraus folgt sofort $F_1^+ = F_2^+$.

Um wiederum zu zeigen, dass eine FD $\alpha \to \beta$ in F_i^+ (für $i \in \{1, 2\}$) liegt, berechnen wir die Attributhülle γ von α unter F_i. Das Ergebnis ergibt sich dann aus der Beziehung $F_i \models \alpha \to \mathit{AttrH}(\alpha, F_i)$.

Man könnte natürlich auch anders vorgehen, und $F_i \models \alpha \to \beta$ auf andere Arten zeigen, z.B. mittels der Armstrong Axiome.

$F_1^+ \subseteq F_2^+$:

$HI \to GJ$ Wir berechnen $\mathit{AttrH}(HI, F_2)$.

a) Auf Grund der FD $HI \to GL$ erhalten wir $HIGL \in \mathit{AttrH}(HI, F_2)$.

b) Auf Grund der FD $IL \to JK$ erhalten wir $HIGLJK \in \mathit{AttrH}(HI, F_2)$.

Nachdem die Attributhülle nun bereits sämtliche Attribute umfasst können wir die Berechnung abbrechen und erhalten

$\mathit{AttrH}(HI, F_2) = GHIJKL$. Daher ist $GJ \subseteq \mathit{AttrH}(HI, F_2)$, und daraus folgt $F_2 \models HI \to GJ$, d.h. $HI \to GJ \in F_2^*$.

$G \to KL$ Nachdem F_2 die FD $G \to KLI$ enthält folgt $G \to KL \in F_2^+$ sofort.

$IL \to JK$ Die FD befindest sich direkt in F_2, es bleibt daher nichts zu zeigen.

$GK \to I$ $\mathit{AttrH}(GK, F_2) = GKLIJ$. Da $I \in \mathit{AttrH}(GK, F_2)$ gilt $GK \to I \in F_2^*$.

(Zur Berechnung der Attributhülle: Aus GK folgt wegen $G \to KLI$ dass $KLI \in \mathit{AttrH}(GK, F_2)$. Man könnte an dieser Stelle bereits abbrechen, da I bereits als Teil der Attributhülle identifiziert wurde. Berechnet man die komplette Hülle erhält man noch $J \in \mathit{AttrH}(GK, F_2)$ auf Grund der FD $IL \to JK$ und der Tatsache dass $IL \in \mathit{AttrH}(GK, F_2)$.)

Da jede FD aus F_1 in der Hülle von F_2 enthalten ist, erhalten wir $F_1^* \subseteq F_2^*$.

$F_2^+ \subseteq F_1^+$:

$HI \to GL$ Wir berechnen $\mathit{AttrH}(HI, F_1)$.

a) Auf Grund der FD $HI \to GJ$ erhalten wir $HIGJ \in \mathit{AttrH}(HI, F_1)$.

b) Auf Grund der FD $G \to KL$ erhalten wir $HIGJKL \in \mathit{AttrH}(HI, F_1)$.

Nachdem die Attributhülle nun bereits sämtliche Attribute umfasst können wir die Berechnung abbrechen und erhalten

$\mathit{AttrH}(HI, F_1) = GHIJKL$. Daher ist $GL \subseteq \mathit{AttrH}(HI, F_1)$, und daraus folgt $F_1 \models HI \to GL$, d.h. $HI \to GL \in F_1^*$.

$G \to KLI$ Wir berechnen $\mathit{AttrH}(G, F_1)$.

a) Auf Grund der FD $G \to KL$ erhalten wir $GKL \in \mathit{AttrH}(G, F_1)$.

b) Auf Grund der FD $GK \to I$ erhalten wir $GKLI \in \mathit{AttrH}(G, F_1)$.

Nachdem wir nun bereits gezeigt haben dass KLI in der Attributhülle enthalten sind, könnten wir die Berechnung bereits abbrechen. Wir berechnen trotzdem weiter die gesamte Hülle.

c) Auf Grund der FD $IL \to JK$ erhalten wir $GKLIJK \in \mathit{AttrH}(G, F_1)$.

Nachdem die Attributhülle nun bereits sämtliche Attribute umfasst können wir die Berechnung abbrechen.

Wir erhalten $\mathit{AttrH}(G, F_1) = GHIJKL$. Daher ist $KLI \subseteq \mathit{AttrH}(G, F_1)$, und daraus folgt $F_1 \models G \to KLI$, d.h. $G \to KLI \in F_1^*$.

$IL \to JK$ Die FD befindest sich direkt in F_1, es bleibt daher nichts zu zeigen.

$GKH \to H$ Hier handelt es sich um eine triviale FD (die rechte Seite ist eine Teilmenge der linken Seite). Triviale FDs sind immer erfüllt, und in jeder Hülle enthalten.

Da jede FD aus F_1 in der Hülle von F_2 enthalten ist, erhalten wir $F_1^* \subseteq F_2^*$.

Da sowohl $F_1^* \subseteq F_2^*$ gilt, als auch $F_2^* \subseteq F_1^*$, erhalten wir $F_1^* = F_2^*$, und daher sind F_1 und F_2 äquivalent.

(b) Consider the set F_1 of functional dependencies from task a). Please show using the Armstrong axioms that $F_1 \models \{HI \to GL\}$ holds (show your reasoning).

Lösung:
Um zu zeigen dass $F_1 \models \{HI \to GL\}$ gilt, müssen wir die FD $HI \to GL$ mit Hilfe der Armstrong-Axiome aus F_1 ableiten. Eine mögliche solche Ableitung ist

$$\dfrac{\text{Dekomposition}\;\dfrac{\overset{[\text{Gegeben}]}{HI \to GJ}}{HI \to G} \qquad \dfrac{\overset{[\text{Gegeben}]}{HI \to GJ} \qquad \dfrac{\dfrac{\overset{[\text{Gegeben}]}{G \to KL}}{GJ \to JKL}\;\text{Verstärkung}}{GJ \to L}\;\text{Dekomposition}}{HI \to L}\;\text{Transitivität}}{HI \to GL}\;\text{Vereinigung}$$

Task 4 (Minimal Cover)	[2 points]

Provide a minimal cover of the sets $\mathcal{F}_1, \mathcal{F}_2$ of functional dependencies over the relational schema $\mathcal{R} = HIJKLMN$ and show your reasoning.

(a) $\mathcal{F}_1 = \{KLN \to HI, I \to KL, H \to HKI, ILM \to LM, KH \to LM, MN \to IL\}$

(b) $\mathcal{F}_2 = \{JL \to HM, L \to I, HMI \to KN, JL \to KN, H \to L, HLM \to K, HM \to J, J \to N\}$

Lösung:

(a) Die Kanonische Überdeckung lässt sich mithilfe der folgenden vier Schritten berechnen: (Die Lösung ist nicht eindeutig.)

- **Dekomposition:**

  Mit der Regel der Dekomposition erhalten wir die Menge
  $\{KLN \to H, KLN \to I, I \to K, I \to L, H \to H, H \to K, H \to I, ILM \to L, ILM \to M, KH \to L, KH \to M, MN \to I, MN \to L\}$

- **Linksreduktion:**

  Wir müssen uns jetzt sukzessive die Frage stellen, ob für eine funktionale Abhängigkeit ein Attribut auf der linken Seite "überflüssig" ist, d.h. weggelassen werden kann so dass sich die resultierende Menge an FDs äquivalent zur ursprünglichn Menge an FDs ist.

  Bei den FDs $KLN \to H$ und $KLN \to I$ ist dies nicht der Fall, hier kann daher nichts reduziert werden. Die nächsten FDs haben auf der linken Seite jeweils nur ein Attribut stehen, sind daher auch bereits linksreduziert. (Die FD $H \to H$ ist trivial und könnte daher sofort eliminiert werden; wir halten uns hier jedoch an den in der VO gebrachten Algorithmus und belassen die FD vorerst noch.)

  Die FD $ILM \to L$ ist ebenfalls eine triviale FD. Wir könnten Sie daher entweder sofort eliminieren, oder zumindest ergibt sich ohne dass man die Attributhülle berechnen müsste dass I und M auf der linken Seite überflüssig sind. Die FD kann daher zu $L \to L$ reduziert werden. Die analoge Argumentation gilt für $ILM \to M$, welche zu $M \to M$ reduziert wird.

Die $KH \to L$ kann zu $H \to L$ reduziert werden: Die Berechnung der Attributhülle unter der aktuellen Menge $F_{cur} = \{KLN \to H, KLN \to I, I \to K, I \to L, H \to H, H \to K, H \to I, L \to L, M \to M, KH \to L, KH \to M, MN \to I, MN \to L\}$ an funktionalen Abhängigkeiten ergibt: Beginnend mit $\{H\}$ erhalten wir auf Grund der FD $H \to K$ dass $\{H, K\} \subseteq \mathit{AttrH}(H, F_{cur})$. Nun erhalten wir durch die FD $KH \to L$ dass $L \in \mathit{AttrH}(H, F_{cur})$, welches die zu überprüfende Eigenschaft darstellt.

Für die FD $KH \to M$ ergibt die analoge Argumentation (nur am Ende unter Verwendung der FD $KH \to M$ an Stelle von $KH \to L$) dass ebenfalls das K auf der linken Seite entfernt werden kann.

Die beiden letzten FDs $MN \to I$ und $MN \to L$ können nicht weiter reduziert werden, da die Hülle sowohl von M als auch von N jeweils nur M bzw. N enthält.

Somit ist die Menge F_r der funktionalen Abhängigkeiten nach der Linksreduktion gegeben als
$\{KLN \to H, KLN \to I, I \to K, I \to L, H \to H, H \to K, H \to I, L \to L, M \to M, H \to L, H \to M, MN \to I, MN \to L\}$

- Rechtsreduktion:

  Bei der Rechtsreduktion muss man für jede funktionale Abhängigkeit überprüfen, ob das Entfernen der FD zu einer äquivalenten Menge an FDs führt, oder nicht. D.h. man muss überprüfen, ob die jeweilige FD aus den verbliebenen FDs folgt (abgeleitet werden kann) oder nicht.

  Für $KLN \to H$ ist dies nicht der Fall, da dies die einzige (nicht triviale) FD ist welche H auf der rechten Seite stehen hat. Die FD $KLN \to I$ ist jedoch überflüssig, da $I \in \mathit{AttrH}(KLN, F_c)$ (sei $F_c = F_r \setminus \{KLN \to I\}$) gilt: Auf Grund der FD $KLN \to H$ gilt $H \in \mathit{AttrH}(KLN, F_c)$, woraus sich auf Grund von $H \to I$ die gewünschte Eigenschaft $I \in \mathit{AttrH}(KLN, F_c)$ ergibt. Wir erhalten $F_r = F_c$.

  Die FD $I \to L$ ist nicht redundant, jedoch die triviale FD $H \to H$, welche entfernt wird.

  Die FD $H \to K$ ist ebenfalls redundant: Wegen der FDs $H \to I$ und $I \to K$ erhalten wir $K \in \mathit{AttrH}(H, F_c)$ (für $F_c = F_r \setminus \{H \to K\}$). Die FD wird daher entfernt, und $F_r = F_c$.

  Die FD $H \to I$ ist nicht redundant, jedoch die trivialen FDs $L \to L$ und $M \to M$, welche ebenfalls entfernt werden.

  Die FD $H \to L$ ist redundant: Für $F_c = F_r \setminus \{H \to L\}$ erhalten wir auf Grund von $H \to I$ und $I \to L$ dass $L \in \mathit{AttrH}(H, F_c)$ gilt. Daher wird $H \to L$ entfernt und $F_r = F_c$.

  Die FD $H \to M$ ist nicht redudant (die einzige FD mit M auf der rechten Seite), ebenso ist $MN \to I$ nicht redundant da ohne dieser FD die Hülle von MN nur MNL (aber nicht I) enthält.

  Die FD $MN \to L$ ist schlussendlich wieder redundant, und zwar auf Grund der FDs $MN \to I$ und $I \to L$.

Somit ist die Menge der funktionalen Abhängigkeiten nach der Rechtsreduktion gegeben als

$$\{KLN \to H, I \to K, I \to L, H \to I, H \to M, MN \to I\}$$

- **Zusammenfassen:**

  Als letzten Schritt müssen nur noch die Funktionalen Abhängigkeiten mit identischen linken Seiten zusammengefasst werden. Wir erhalten:

  $$\{KLN \to H, I \to KL, H \to IM, MN \to I\}$$

(b) Wir berechnen wiederum die vier Schritte um eine kanonische Überdeckung zu erhalten (Die Lösung ist nicht eindeutig).

- **Dekomposition:**

  Durch Anwendung der Dekompositionsregel erhalten wir die Menge $\{JL \to H, JL \to M, L \to I, HMI \to K, HMI \to N, JL \to K, JL \to N, H \to L, HLM \to K, HM \to J, J \to N\}$

- **Linksreduktion:**

  Wir müssen uns jetzt sukzessive die Frage stellen, ob für eine funktionale Abhängigkeit ein Attribut auf der linken Seite "überflüssig" ist, d.h. weggelassen werden kann so dass sich die resultierende Menge an FDs äquivalent zur ursprünglichn Menge an FDs ist.

  Die erste FD welche auf der linken Seite redundante Attribute enthält ist $HMI \to K$, und zwar kann I entfernt werden. Die Berechnung der Attributhülle von HM unter den FDs ergibt folgendes: Auf Grund der FD $HM \to J$ ist J Teil der Hülle, und auf Grund von $H \to L$ ist L in der Hülle enthalten. Durch JL und die FD $JL \to K$ ergibt sich nun, dass K in der Hülle von HM enthalten ist, und daher ist I redundant und kann entfernt werden.

  Das selbe Argument (nur mit $JL \to N$ anstatt $JL \to K$ als letzter FD) ergibt auch dass $HMI \to N$ zu $HM \to N$ reduziert werden kann.

  Auf Grund der FD $J \to N$ kann auch die FD $JL \to N$ zu $J \to N$ reduziert werden. Ebenso kann $HLM \to K$ wegen $HM \to K$ zu $HM \to K$ reduziert werden. (Da wir mit einer *Menge* and funktionalen Abhängigkeiten arbeiten können keine Duplikate auftreten, d.h. die FDs $HLM \to K$ und $JL \to N$ werden effektiv gelöscht.)

  Nachdem keine der verbleibenden FDs weiter reduziert werden kann ist die Menge der funktionalen Abhängigkeiten nach der Linksreduktion gegeben als

  $$\{JL \to H, JL \to M, L \to I, HM \to K, HM \to N, JL \to K, J \to N, H \to L, HM \to J\}$$

- Rechtsreduktion:

  Bei der Rechtsreduktion muss man für jede funktionale Abhängigkeit überprüfen, ob das Entfernen der FD zu einer äquivalenten Menge an FDs führt, oder nicht. D.h. man muss überprüfen, ob die jeweilige FD aus den verbliebenen FDs folgt (abgeleitet werden kann) oder nicht.

$JL \to H$ ist die einzige FD mit H auf der rechten Seite, und muss daher erhalten bleiben. Genauso $JL \to M$ und $L \to I$ für M bzw. I auf der rechten Seite.

Die FD $HM \to K$ hingegen ist redundant: Mittels $HM \to J$ (HMJ), $H \to L$ ($HMJL$) und $JL \to K$ ($HMJLK$) erhalten wir, dass K auch ohne der FD $HM \to K$ Teil der Hülle von HM ist.

Die FD $HM \to N$ ist redundant wegen $HM \to J$ und $J \to N$.

Die verbleibende Menge an FDs kann nicht weiter reduziert werden (die rechte Seite jeder FD ist eindeutig).

Somit ist die Menge der funktionalen Abhängigkeiten nach der Rechtsreduktion gegeben als

$$\{JL \to H, JL \to M, L \to I, JL \to K, J \to N, H \to L, HM \to J\}$$

- **Zusammenfassen:**

  Als letzten Schritt müssen nur noch die Funktionalen Abhängigkeiten mit identischen linken Seiten zusammengefasst werden:

  $$\{JL \to KHM, L \to I, J \to N, H \to L, HM \to J\}$$

  Eine mögliche alternative Lösung ist

  $$\{JL \to HM, L \to I, J \to N, H \to L, HM \to JK\}$$

Task 5 (Identifying Keys and Superkeys)	[2 points]

For the following relational schemata with their functional dependencies, find *all keys* and *all superkeys*. (*Please note:* For task a), please write down all superkeys. For task b) you **don't need to list all superkeys**; it is sufficient, if the reader can clearly infer the total set of superkeys.)

(a) $\mathcal{R} = FGHIJ$
$F = \{FG \to GI, FJ \to I, F \to G, FI \to J, H \to J, I \to F\}$

Lösung:
Die Schlüssel sind FH und HI. Die Menge der Superschlüssel ist

$$\{FH, HI, FGH, FHI, FHJ, GHI, HIJ, FGHI, FHIJ, FGHJ, GHIJ, FGHIJ\}$$

(b) $\mathcal{R} = EFGHIJK$
$F = \{GI \to EF, E \to K, FJ \to GE, FE \to I, IK \to KJH\}$

Lösung:
Die Menge der Schlüssel ist

$$\{GI, FJ, EF, FIK\}.$$

Die Menge der Superschlüssel ist

$\{EF, EFG, EFGH, EFGHI, EFGHIJ, EFGHIJK, EFGHIK, EFGHJ, EFGHJK, EFGHK, EFGI, EFGIJ, EFGIJK, EFGIK, EFGJ, EFGJK, EFGK, EFH, EFHI, EFHIJ, EFHIJK, EFHIK, EFHJ, EFHJK, EFHK, EFI, EFIJ, EFIJK, EFIK, EFJ, EFJK, EFK, EGHI, EGHIJ, EGHIJK, EGHIK, EGI, EGIJ, EGIJK, EGIK, FGHI, FGHIJ, FGHIJK, FGHIK, FGHJ, FGHJK, FGI, FGIJ, FGIJK, FGIK, FGJ, FGJK, FHIJ, FHIJK, FHIK, FHJ, FHJK, FIJ, FIJK, FIK, FJ, FJK, GHI, GHIJ, GHIJK, GHIK, GI, GIJ, GIJK, GIK\}$

Task 6 (Normal Forms) [1 point]

For each subtask, assume a relational schema $\mathcal{R}$ with its set $\mathcal{F}$ of functional dependencies. Please check, whether $\mathcal{R}$

- is in third normal form,
- in Boyce-Codd normal form,

and justify your answer.

Lösung:
Die drei Eigenschaften, mit welcher für eine Menge funktionaler Abhängigkeiten der Form $\alpha \to B$ entschieden werden kann, ob sich das Relationenschema $\mathcal{R}$ in dritter Normalform bzw. Boyce-Codd Normalform befindet, 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 Relationenschema $\mathcal{R}$ in Boyce-Codd Normalform, wenn für jede Funktionale Abhängigkeit Bedingungen 1 oder 2 erfüllt ist, und $\mathcal{R}$ ist in dritter Normalform wenn für jede Funktionale Abhängigkeit eine der Bedingungen 1, 2 oder 3 erfüllt ist.

(a) $\mathcal{R} = ABCD$,
$\mathcal{F} = \{AB \to C, C \to A, B \to DBA, AC \to BD\}$

Lösung:
Wir bestimmen zuerst die Schlüssel von $\mathcal{R}$. Diese sind: B und C.

Um die Bedingungen zu untersuchen wenden wir zuerst die Dekomposition an, und kennzeichnen anschließend für jede FD, welche Bedingungen sie erfüllt:

$$\mathcal{F} = \{\underbrace{AB \to C}_{2,3}, \underbrace{C \to A}_{2}, \underbrace{B \to D}_{2}, \underbrace{B \to B}_{1,2,3}, \underbrace{B \to A}_{2}, \underbrace{AC \to B}_{2,3}, \underbrace{AC \to D}_{2}\}.$$

Nachdem jede FD entweder Eigenschaft 1 oder 2 erfüllt, befindet sich $\mathcal{R}$ in BCNF (und daher auch in 3NF).

(b) $\mathcal{R} = ABCDEF$
$F = \{ACD \to BC, BDF \to CE, EF \to ABD, AEB \to CD, ABC \to BF,$
$AD \to AC, ACF \to CF\}$
The keys are given as EF, BDF, ABE, and DA.

Lösung:
Um die Bedingungen zu untersuchen wenden wir die Dekomposition an, und kennzeichnen für jede FD, welche Bedingungen sie erfüllt:

$$\begin{aligned}\mathcal{F} = \{ & \underbrace{ACD \to B}_{2,3}, \underbrace{ACD \to C}_{2}, \underbrace{BDF \to C}_{2}, \underbrace{BDF \to E}_{2,3}, \underbrace{EF \to A}_{2,3}, \\ & \underbrace{EF \to B}_{2,3}, \underbrace{EF \to D}_{2,3}, \underbrace{ABE \to C}_{2}, \underbrace{ABE \to D}_{2,3}, \underbrace{ABC \to B}_{3}, \\ & \underbrace{ABC \to F}_{3}, \underbrace{AD \to A}_{1,2}, \underbrace{AD \to C}_{2}, \underbrace{ACF \to C}_{1}, \underbrace{ACF \to F}_{3}\}.\end{aligned}$$

Nachdem jede FD eine der drei Eigenschaften erfüllt, einige FDs jedoch nur Eigenschaft (3) befindet sich das Schema im 3NF, jedoch nicht in BCNF.

Task 7 (Synthesis Algorithm) [2 points]

Consider the following relational schema and its functional dependencies:

$$\mathcal{R} = ABCDEFGH$$
$$\mathcal{F} = \{BCE \to AG, AB \to FH, BG \to C, A \to CH, AF \to B, D \to EF\}$$

We are looking for a lossless and dependency preserving decomposition in third normal form. Please apply the synthesis algorithm and show the results after every single step. Compute all keys of $\mathcal{R}$ and all relations of the decomposition.

Lösung:

1. Find a minimal cover:

$$\mathcal{F}_c = \{BCE \to AG, AB \to F, BG \to C, A \to CH, AF \to B, D \to EF\}$$

2. Identify all keys of $\mathcal{R}$ wrt. $\mathcal{F}_c$:

$$\{BDG, AD, BCD\}.$$

3. Create one schema for each FD in $\mathcal{F}_c$:

Relationenschema	Geltende FDs
$\mathcal{R}_1 = ABCGE$	$\mathcal{F}_1 = \{BCE \to AG, BG \to C, A \to C\}$
$\mathcal{R}_2 = ABF$	$\mathcal{F}_2 = \{AB \to F, AF \to B\}$
$\mathcal{R}_3 = BCG$	$\mathcal{F}_3 = \{BG \to C\}$
$\mathcal{R}_4 = ACH$	$\mathcal{F}_4 = \{A \to CH\}$
$\mathcal{R}_5 = ABF$	$\mathcal{F}_5 = \{AF \to B, AB \to F\}$
$\mathcal{R}_6 = DEF$	$\mathcal{F}_6 = \{D \to EF\}$

4. Eliminate contained schemas:

   Das Schema $\mathcal{R}_3$ ist im Schema $\mathcal{R}_1$ enthalten und muss daher eliminiert werden. Die Schemata $\mathcal{R}_2$ und $\mathcal{R}_5$ sind identisch, daher wird $\mathcal{R}_5$ ebenfalls eliminiert.

Relationenschema	Geltende FDs
$\mathcal{R}_1 = ABCGE$	$\mathcal{F}_1 = \{BCE \to AG, BG \to C, A \to C\}$
$\mathcal{R}_2 = ABF$	$\mathcal{F}_2 = \{AB \to F, AF \to B\}$
$\mathcal{R}_4 = ACH$	$\mathcal{F}_4 = \{A \to CH\}$
$\mathcal{R}_6 = DEF$	$\mathcal{F}_6 = \{D \to EF\}$

5. Check if at least one key is contained in one of the schemas:

   Keiner der Schlüssel ist in einem Teilschema anthalten. Es muss ein neues Relationenschema $\mathcal{R}_\mathcal{K}$ erstellt werden, wir wählen den Schlüssel AD:

Relationenschema	Geltende FDs
$\mathcal{R}_\mathcal{K} = AD$	$\mathcal{F}_\mathcal{K} = \emptyset$

6. Final result (one key per relation is underlined):

Relationenschema	Geltende FDs
$\mathcal{R}_1 = A\underline{BCE}G$	$\mathcal{F}_1 = \{BCE \to AG, BG \to C, A \to C\}$
$\mathcal{R}_2 = \underline{AB}F$	$\mathcal{F}_2 = \{AB \to F, AF \to B\}$
$\mathcal{R}_4 = \underline{A}CH$	$\mathcal{F}_4 = \{A \to CH\}$
$\mathcal{R}_6 = \underline{D}EF$	$\mathcal{F}_6 = \{D \to EF\}$
$\mathcal{R}_\mathcal{K} = \underline{AD}$	$\mathcal{F}_\mathcal{K} = \emptyset$

Task 8 (Decomposition Algorithm) [2 points]

Consider the following relational schema with its functional dependencies and the list of all its keys:

$$\mathcal{R} = ABCDEF$$
$$\mathcal{F} = \{F \to C, D \to C, BDE \to AC, ABD \to DEF, BD \to E, EF \to AEB, AD \to CEF\}$$

Keys: DEF, BD, AD

We are looking for a lossless decomposition into Boyce-Codd normal form. Please apply the decomposition algorithm and show the results after every single step. Compute all keys for all relations of the decomposition. Is the decomposition dependency preserving? If not, please provide the dependencies in $\mathcal{F}$ that got lost.

Hint: Compute for every decomposition the corresponding closures of FDs!

Lösung:

Die drei FDs

- $F \to C$,

- $D \rightarrow C$, und
- $EF \rightarrow AEB$

verletzen die BCNF (sind nicht trivial und die linke Seite beinhaltet keinen Schlüssel), die restlichen FDs erfüllen die BCNF. Es gibt daher drei Möglichkeiten für die Zerlegung. Wir zeigen hier alle drei Möglichkeiten auf.

Hinweis: Wir geben in jeder Hülle $\mathcal{F}_i^+[\mathcal{R}_j]$ nur die nicht-trivialen, linksreduzierten Abhängigkeiten an.

- Die FD $F \rightarrow C$ wird gewählt. Damit erhalten wir das Schema:

$$\begin{aligned}
\mathcal{R}_1 &= CF & \mathcal{F}_1 &= \mathcal{F}^+[\mathcal{R}_1] = \{F \rightarrow C\} \\
&& &\text{Schlüssel: } F \\
\mathcal{R}_2 &= ABDEF & \mathcal{F}_2 &= \mathcal{F}^+[\mathcal{R}_2] = \{AD \rightarrow EF, EF \rightarrow AB, BD \rightarrow A\} \\
&& &\text{Schlüssel: } AD, BD, DEF
\end{aligned}$$

  Das Schema $\mathcal{R}_1$ ist nun in BCNF, nicht jedoch das Schema $\mathcal{R}_2$, da die FD $EF \rightarrow AB$ wiederum die Bedingungen der BCNF verletzt. Daher muss $\mathcal{R}_2$ weiter zerlegt werden. Dabei wird nach der einzigen FD zerlegt, welche die BCNF verletzt.

  - Wir zerlegen nach $EF \rightarrow AB$ zu

$$\begin{array}{lll}
\mathcal{R}_1 = CF & \mathcal{F}_1 = \mathcal{F}^+[\mathcal{R}_1] = \{F \rightarrow C\} & \text{Schlüssel: } F \\
\mathcal{R}_{2,1} = ABEF & \mathcal{F}_{2,1} = \mathcal{F}_2^+[\mathcal{R}_{2,1}] = \{EF \rightarrow AB\} & \text{Schlüssel: } EF \\
\mathcal{R}_{2,2} = DEF & \mathcal{F}_{2,2} = \mathcal{F}_2^+[\mathcal{R}_{2,2}] = \{\} & \text{Schlüssel: } DEF
\end{array}$$

    Nun erfüllen alle Relationenschemata die BCNF, es ist daher kein weiteres zerlegen notwendig.

    Diese Zerlegung ist nicht abhängigkeitserhaltend. Von den gegebenen FDs gehen $D \rightarrow C$, $BDE \rightarrow AC$, $ABD \rightarrow DEF$, $BD \rightarrow E$ und $AD \rightarrow CEF$ verloren.

- Die FD $D \rightarrow C$ wird gewählt. Damit erhalten wir das Schema:

$$\begin{aligned}
\mathcal{R}_1 &= CD & \mathcal{F}_1 &= \mathcal{F}^+[\mathcal{R}_1] = \{D \rightarrow C\} \\
&& &\text{Schlüssel: } D \\
\mathcal{R}_2 &= ABDEF & \mathcal{F}_2 &= \mathcal{F}^+[\mathcal{R}_2] = \{AD \rightarrow EF, EF \rightarrow AB, BD \rightarrow A\} \\
&& &\text{Schlüssel: } AD, BD, DEF
\end{aligned}$$

  $\mathcal{R}_2$ ist nun ident zu $\mathcal{R}_2$ zuvor. Wir zerlegen $\mathcal{R}_2$ daher wie zuvor in $\mathcal{R}_{2,1}$ und $\mathcal{R}_{2,2}$ und erhalten das Schema

$$\begin{array}{lll}
\mathcal{R}_1 = CD & \mathcal{F}_1 = \mathcal{F}^+[\mathcal{R}_1] = \{D \rightarrow C\} & \text{Schlüssel: } D \\
\mathcal{R}_{2,1} = ABEF & \mathcal{F}_{2,1} = \mathcal{F}_2^+[\mathcal{R}_{2,1}] = \{EF \rightarrow AB\} & \text{Schlüssel: } EF \\
\mathcal{R}_{2,2} = DEF & \mathcal{F}_{2,2} = \mathcal{F}_2^+[\mathcal{R}_{2,2}] = \{\} & \text{Schlüssel: } DEF
\end{array}$$

  in BCNF.

  Die Zerlegung ist nicht abhängigkeitserhaltend. Von den gegebenen FDs gehen $F \rightarrow C$, $BDE \rightarrow AC$, $ABD \rightarrow EF$, $BD \rightarrow E$ und $AD \rightarrow CEF$ verloren.

- Die FD $EF \to AEB$ wird gewählt. Damit erhalten wir das Schema:

$$\begin{aligned} \mathcal{R}_1 &= ABEF & \mathcal{F}_1 &= \mathcal{F}^+[\mathcal{R}_1] = \{EF \to AB\} \\ & & &\text{Schlüssel: } EF \\ \mathcal{R}_2 &= CDEF & \mathcal{F}_2 &= \mathcal{F}^+[\mathcal{R}_2] = \{F \to C, D \to C\} \\ & & &\text{Schlüssel: } DEF \end{aligned}$$

  Das Schema $\mathcal{R}_1$ ist nun in BCNF, nicht jedoch das Schema $\mathcal{R}_2$, da beide FDs – $F \to C$ und $D \to C$ – die Bedingungen der BCNF nicht erfüllen. Das Schema muss also weiter zerlegt werden, wobei zwei Möglichkeiten zur Auswahl stehen.

  - Wir zerlegen nach $F \to C$ zu

$$\begin{aligned} \mathcal{R}_1 &= ABEF & \mathcal{F}_1 &= \mathcal{F}^+[\mathcal{R}_1] = \{EF \to AB\} & &\text{Schlüssel: } EF \\ \mathcal{R}_{2,1} &= CF & \mathcal{F}_{2,1} &= \mathcal{F}_2^+[\mathcal{R}_{2,1}] = \{F \to C\} & &\text{Schlüssel: } F \\ \mathcal{R}_{2,2} &= DEF & \mathcal{F}_{2,2} &= \mathcal{F}_2^+[\mathcal{R}_{2,2}] = \{\} & &\text{Schlüssel: } DEF \end{aligned}$$

    Nun erfüllen alle Relationenschemata die BCNF, es ist daher kein weiteres zerlegen notwendig.

    Diese Zerlegung ist nicht abhängigkeitserhaltend. Von den gegebenen FDs gehen $D \to C$, $BDE \to AC$, $ABD \to EF$, $BD \to E$ und $AD \to CEF$ verloren.

  - Wir zerlegen nach $D \to C$ zu

$$\begin{aligned} \mathcal{R}_1 &= ABEF & \mathcal{F}_1 &= \mathcal{F}^+[\mathcal{R}_1] = \{EF \to AB\} & &\text{Schlüssel: } EF \\ \mathcal{R}_{2,1} &= CD & \mathcal{F}_{2,1} &= \mathcal{F}_2^+[\mathcal{R}_{2,1}] = \{D \to C\} & &\text{Schlüssel: } D \\ \mathcal{R}_{2,2} &= DEF & \mathcal{F}_{2,2} &= \mathcal{F}_2^+[\mathcal{R}_{2,2}] = \{\} & &\text{Schlüssel: } DEF \end{aligned}$$

    Nun erfüllen alle Relationenschemata die BCNF, es ist daher kein weiteres zerlegen notwendig.

    Diese Zerlegung ist nicht abhängigkeitserhaltend. Von den gegebenen FDs gehen $F \to C$, $BDE \to AC$, $ABD \to EF$, $BD \to E$ und $AD \to CEF$ verloren.