Exercise Sheet 2 (WS 2021) – 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 7 tasks and at most 15 points that can be achieved on the entire sheet.

Deadlines

until 08.11.	**12:00 pm**:	Solve the SQL tasks with the online-tool
until 17.11.	**12:00**:	Upload your solution to TUWEL
from 29.11.	13:00:	Evaluation and feedback is available in TUWEL

SQL Exercise

Note that the **mandatory** SQL exercise takes place in parallel to this exercise sheet. The SQL exercise takes place in its own online tool and is not part of the exercise sheet.

You can access the online tool in TUWEL: Select **eSQL Tool** in the section "SQL exercise". 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.

Caution!
Separate deadline for finishing the SQL tasks:
Monday, November 08, 2021, 12:00 pm!

Further questions – TUWEL forum

If you have any further questions concerning the contents or organization, do not hestitate to ask them on TUWEL forum. **Under no circumstances should you post (partial) solutions on the forum!**

Changes due to COVID-19

Due to the ongoing situation with COVID-19 we will not offer in-person office hours for the exercise sheets. If you have technical issues, trouble understanding the tasks on this sheet, or other questions please use the TUWEL forum.

We also recommend that you get involved in the forum and actively discuss with your colleagues on the forum. From experience we believe that this helps all parties in the discussion greatly to improve their understanding of the material.

Normal Forms

Task 1 (Functional Dependencies) [1 point]

Consider the following relational schema

`Diving (Name, Certificate, Day, Time, Spot, Guide)`

with the following instance:

Diving					
Name	**Certificate**	**Day**	**Time**	**Spot**	**Guide**
Kurt	OWD	Monday	morning	Lavafinger	yes
Kurt	OWD	Monday	afternoon	Arena	no
Alan	OWD	Tuesday	afternoon	Lavafinger	yes
Alan	OWD	Wednesday	afternoon	Arena	yes
Ada	AOWD	Monday	morning	Lavafinger	yes
Ada	AOWD	Monday	afternoon	Arena	no
Alonzo	AOWD	Tuesday	morning	Boot	yes
Alonzo	AOWD	Wednesday	afternoon	Arena	no
Kurt	OWD	Wednesday	afternoon	Arena	no
Alan	OWD	Tuesday	afternoon	Lavafinger	yes
Ada	AOWD	Tuesday	morning	Boot	no
Alonzo	AOWD	Monday	morning	Lavafinger	yes

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) `Name` $\rightarrow$ `Certificate`. **Yes**

Every tuple with an existing name but a different certificate would work: for instance a tuple (Kurt, $AOWD, \dots$), which breaks the FD because the tuple (Kurt, $OWD, \dots$) already exists.

(b) `Certificate` $\rightarrow$ `Guide` **No**

This FD is violated, for instance by the first two tuples with `Certificate` *OWD*.

(c) `Day, Time` $\rightarrow$ `Spot`.

Yes

We can break this FD with the following tuple

$$(\star, \star, Monday, morning, Boot, \star)$$

where $\star$ serves as a placeholder symbol for an arbitrary attribute.

(d) `Day, Spot` $\rightarrow$ `Guide`. **No**

This FD is violated for instance by the two tuples with `Spot` *Boot*. The relevant part of the FD violations is:

$$(\star, \star, Tuesday, \star, Boot, yes)$$
$$(\star, \star, Tuesday, \star, Boot, no)$$

(e) `Name, Spot` $\rightarrow$ `Guide`.

Yes

To break this FD we could add the tuple

$$(Alonzo, \star, \star, \star, Lavafinger, no)$$

(f) `Spot, Guide` $\rightarrow$ `Certificate, Day`. **No**

This FD is violated, for instance, by the first and third tuple. The relevant part is:

$$(\star, OWD, Monday, \star, Lavafinger, yes)$$
$$(\star, OWD, Tuesday, \star, Lavafinger, yes)$$

Task 2 (Equivalence of Functional Dependencies) [2 points]

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

$$F_1 = \{B \rightarrow AC, EF \rightarrow BD, D \rightarrow G, A \rightarrow BE, G \rightarrow A, E \rightarrow F\}$$
$$F_2 = \{B \rightarrow AC, EF \rightarrow BD, D \rightarrow G, A \rightarrow BE, G \rightarrow EF\}$$

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:

The equivalence of the sets holds exactly when: $F_1^+ \subseteq F_2^+$ and $F_2^+ \subseteq F_1^+$

Since the sets partially overlap it holds: $F_1^+ \subseteq F_2^+$ iff. $G \rightarrow A \in F_2^+$ and $E \rightarrow F \in F_2^+$ as well as $F_2^+ \subseteq F_1^+$ iff. $G \rightarrow EF \in F_1^+$.

It can be shown that $G \rightarrow A \in F_2^+$ and $G \rightarrow EF \in F_1^+$. However $E \rightarrow F \notin F_2^+$, since $attrHuelle(F_2^+, \{E\}) \cap \{F\} = \emptyset$

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

Lösung:

- $D \to G, G \to A \models D \to A$ (Transitivity)
- $D \to A, A \to BE \models D \to BE$ (Transitivity)
- $D \to BE \models D \to E$ (Decomposition)
- $D \to E, E \to F \models D \to F$ (Transitivity)
- $D \to BE, D \to F \models D \to BEF$ (Union)

Task 3 (Minimal Cover) [2 points]

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

(a) $\mathcal{F}_1 = \{A \to D, AD \to DFG, AG \to BC, B \to AEF, CF \to G, F \to BD, F \to BF\}$

(b) $\mathcal{F}_2 = \{ABG \to D, AFG \to DEF, B \to G, BC \to D, DF \to G, F \to B\}$

Lösung:

(a) A canonical cover can be computed using the following four steps (the solution is not unique)

- **decomposition:**

  with decomposition we obtain
  $\{A \to D, AD \to D, AD \to F, AD \to G, AG \to B, AG \to C, B \to A, B \to E, B \to F, CF \to G, F \to B, F \to D, F \to B, F \to F\}$, which is equivalent to (as it is a set)
  $\{A \to D, AD \to D, AD \to F, AD \to G, AG \to B, AG \to C, B \to A, B \to E, B \to F, CF \to G, F \to B, F \to D, F \to F\}$.

- **left reduction:**

  We have to check for each FD whether an attribute on the left side is superfluous and therefore can be omitted such that the resulting set of FDs is equivalent to the original set of FDs.

  Already left reduced are $\{A \to D, B \to A, B \to E, B \to F, F \to B, F \to D, F \to F\}$ and therefore have not to be considered. We focus on the set $\{AD \to D, AD \to F, AD \to G, AG \to B, AG \to C, CF \to G\}$.

  $AD \to D$ can be reduced to the trivial $D \to D$, even without constructing the attribute closure. This FD is trivial and therefore can be cancelled here already. $AD \to F$ can be reduced to $A \to F$, as F is in the attribute closure of $(\mathcal{F}_1, A)$ (attention: we have to consider the current - reduced - set here). $AD \to G$ can be reduced to $A \to G$, as G is in the attribute closure of $(\mathcal{F}_1, A)$. $AG \to B$ can be reduced to $A \to B$, as B is in the attribute closure of $(\mathcal{F}_1, A)$. $AG \to C$ can be reduced to $A \to C$, as C is in the attribtute closure of A. $CF \to G$ can be reduced to $F \to G$, as G is in the attribute closure of $(\mathcal{G}_1, F)$.

  The set F_r of FDs after the left reduction is
  $\{A \to B, A \to C, A \to D, A \to F, A \to G, B \to A, B \to E, B \to F, D \to D, F \to B, F \to D, F \to F, F \to G\}$.

- right reduction:

  Here we have to check for every FD whether it can be deleted such that the resulting set of FDs is equivalent to the original one. Therefore we have to check whether the corresponding FD can be derived from the remaining FDs or not.

  $A \to B$ can be reduced, as B is in the attribute closure of (F_r', A), where $F_r' = F_r \setminus \{A \to B\}$. We obtain $F_r = F_r'$. $A \to C$ cannot be reduced, as this is the only FD which has C on the right side. $A \to D$, $A \to F$ and $A \to G$ cannot be reduced as well.

  $B \to A$ and $B \to E$ cannot be reduced. $B \to F$ can be reduced, as F is in the attribute closure of (F_r'', B), where $F_r'' = F_r \setminus \{B \to F\}$. We get $F_r = F_r''$.

  $D \to D$ is trivial and can therefore be reduced.

  $F \to B$ cannot be reduced, but $F \to D$ can, as D is in the attribute closure of (F_r''', F), where $F_r''' = F_r \setminus \{F \to D\}$. We obtain $F_r = F_r'''$.

  $F \to F$ is trivial and can therefore be reduced.

  $F \to G$ can be reduced, as G is in the attribute closure of (F_r'''', F), where $F_r'''' = F_r \setminus \{F \to G\}$. We obtain $F_r = F_r''''$.

  The set of FDs after right reduction is

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

- **union:**

  The last step is to combine FDs with identical left sides with union. We obtain:

  $$\{B \to AE, F \to B, A \to CDFG\}.$$

(b) We consider the four steps to construct a canonical cover (the solution is not unique).

- **decomposition:**

  We obtain:
  $\{AFG \to D, AFG \to E, AFG \to F, ABG \to D, B \to G, BC \to D, DE \to G, F \to B\}$.

- **left reduction:**

  We have to check for each FD whether an attribute on the left side is superfluous and therefore can be omitted such that the resulting set of FDs is equivalent to the original set of FDs.

  For $BC \to D$ this is not the case. $AFG \to D$ can be reduced to $AF \to D$, as D is in the attribute closure of AF. $AFG \to E$ can be reduced to $AF \to E$, as E is in the attribute closure of AF. $AFG \to F$ can be reduced to the trivial FD $F \to F$ without constructing the attribute closure. $DF \to G$ can be reduced to $F \to G$, as G is in the attribute closure of F. $ABG \to D$ can be reduced to $AB \to D$, as D is in the attribute closure of AB.

  The set of FDs after left reduction is

  $$F_r = \{AB \to D, AF \to D, AF \to E, B \to G, BC \to D, F \to G, F \to F, F \to B\}.$$

- right reduction:

  Here we have to check for every FD whether it can be deleted such that the resulting set of FDs is equivalent to the original one. Therefore we have to check whether the corresponding FD can be derived from the remaining FDs or not.

  $AF \to D$ can be reduced, as D is in the attribute closure of $(F_r \backslash (AF \to D), AF)$. $AF \to E$ cannot be reduced, but the FD $F \to F$ is trivial and therefore can be reduced. $F \to G$ can also be reduces, because G is in the attribute closure of F. All the remaining FDs cannot be reduced.

  Therefore, the set of FDs after right reduction is given by

  $$\{AB \to D, AF \to E, B \to G, BC \to D, F \to B\}.$$

- **union:**

  The last step is to combine FDs with identical left sides with union. We obtain:

  $$\{AB \to D, AF \to E, B \to G, BC \to D, DE \to G, F \to B\}.$$

Task 4 (Identifying Keys and Superkeys)	[2 points]

For the following relational schemata with their functional dependencies, find *all keys* and *all superkeys*.

(a) $\mathcal{R} = ABCDEF$
$F = \{A \to BCD, C \to AE, DE \to F\}$

Lösung:
The keys are A and C. The set of superkeys is { A, AB, ABC, $ABCD$, $ABCDE$, $ABCDEF$, $ABCDF$, $ABCE$, $ABCEF$, $ABCF$, ABD, $ABDE$, $ABDEF$, $ABDF$, ABE, $ABEF$, ABF, AC, ACD, $ACDE$, $ACDEF$, $ACDF$, ACE, $ACEF$, ACF, AD, ADE, $ADEF$, ADF, AE, AEF, AF, BC, BCD, $BCDE$, $BCDEF$, $BCDF$, BCE, $BCEF$, BCF, C, CD, CDE, $CDEF$, CDF, CE, CEF, CF }

(b) $\mathcal{R} = ABCDEFG$
$F = \{CD \to EF, E \to A, A \to B, G \to C\}$

Lösung:
The set of keys is

$$\{GD\}.$$

The set of super keys is

{ $ABCDEFG$, $ABCDEG$, $ABCDFG$, $ABCDG$, $ABDEFG$, $ABDEG$, $ABDFG$, $ABDG$, $ACDEFG$, $ACDEG$, $ACDFG$, $ACDG$, $ADEFG$, $ADEG$, $ADFG$, ADG, $BCDEFG$, $BCDEG$, $BCDFG$, $BCDG$, $BDEFG$, $BDEG$, $BDFG$, BDG, $CDEFG$, $CDEG$, $CDFG$, CDG, $DEFG$, DEG, DFG, DG

}

Task 5 (Normal Forms)	[2 points]

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.

(a) $\mathcal{R} = UVWXYZ$
$F = \{UV \rightarrow W, UVW \rightarrow X, XY \rightarrow Z, Z \rightarrow U, XYZ \rightarrow XYU\}$

Lösung:
The keys are given as UVY and XYV.

The three properties of the form $\alpha \rightarrow B$ which decide whether a set of FDs is in 3NF or BCNF are as follows:

1. $B \in \alpha$, i.e., the FD is trivial;
2. α is superkey of $\mathcal{R}$;
3. the attribute B is contained in a key of $\mathcal{R}$.

A schema $\mathcal{R}$ is in BCNF when for each FD conditions 1 or 2 are fulfilled, and it is in 3NF if conditions 1, 2 or 3 are fulfilled.

The keys are given as UVY and XYV.

We first apply decomposition, then we mark which conditions each FD fulfills:

$$\mathcal{F} = \{ \underbrace{UV \rightarrow W}, \underbrace{UVW \rightarrow X}_{3}, \underbrace{XY \rightarrow Z}, \underbrace{XYZ \rightarrow W}, \underbrace{XYZ \rightarrow X}_{1,3}, \underbrace{XYZ \rightarrow Y}_{1,3}, \underbrace{XYZ \rightarrow U}_{3}$$

Since some FDs fulfill no conditions, the schema is neither in 3NF nor BCNF.

(b) $\mathcal{R} = UVWXYZ$
$F = \{UV \rightarrow WXY, UVW \rightarrow YZ\}$

Lösung:
The keys are given as UV.

We apply decomposition first, then check which conditions the FDs fulfill:

$$\mathcal{F} = \{ \underbrace{UV \rightarrow W}_{2}, \underbrace{UV \rightarrow X}_{2}, \underbrace{UV \rightarrow Y}_{2}, \underbrace{UVW \rightarrow Y}_{2}, \underbrace{UVW \rightarrow Z}_{2}$$

Since each FD fulfills condition 3, the schema is in BCNF as well as 3NF.

Task 6 (Synthesis Algorithm) [3 points]

Consider the following relational schema and its functional dependencies:

$\mathcal{R} = QWERTYX$

$\mathcal{F} = \{WE \rightarrow Q, Q \rightarrow R, QW \rightarrow YX, X \rightarrow R, R \rightarrow T, W \rightarrow X, XRW \rightarrow Q, QR \rightarrow E\}$

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. Compute the canonical cover:

$$\mathcal{F}_c = \{Q \rightarrow R, QW \rightarrow Y, X \rightarrow R, R \rightarrow T, W \rightarrow X, XRW \rightarrow Q, QR \rightarrow E\}$$

2. Determine the candidate keys of $\mathcal{R}$ i.e. $\mathcal{F}_c$:

$$\{W\}.$$

3. Create schemata for each element of $\mathcal{F}_c$:

Schema	applying FDs
$\mathcal{R}_1 = QR$	$\mathcal{F}_1 = \{Q \rightarrow R\}$
$\mathcal{R}_2 = QWY$	$\mathcal{F}_2 = \{QW \rightarrow Y\}$
$\mathcal{R}_3 = XR$	$\mathcal{F}_3 = \{X \rightarrow R\}$
$\mathcal{R}_4 = RT$	$\mathcal{F}_4 = \{R \rightarrow T\}$
$\mathcal{R}_5 = WX$	$\mathcal{F}_5 = \{W \rightarrow X\}$
$\mathcal{R}_6 = XRWQ$	$\mathcal{F}_6 = \{XRW \rightarrow Q, Q \rightarrow R, X \rightarrow R, W \rightarrow X\}$
$\mathcal{R}_7 = QRE$	$\mathcal{F}_7 = \{QR \rightarrow E, Q \rightarrow R\}$

4. Eliminate schemata which are contained in other schemata.

   The schema R_1 is contained in schema R_7 as well as R_3 in R_6 and R_5 in R_6.

Schema	applying FDs
$\mathcal{R}_2 = QWY$	$\mathcal{F}_2 = \{QW \rightarrow Y\}$
$\mathcal{R}_4 = RT$	$\mathcal{F}_4 = \{R \rightarrow T\}$
$\mathcal{R}_6 = XRWQ$	$\mathcal{F}_6 = \{XRW \rightarrow Q, Q \rightarrow R, X \rightarrow R, W \rightarrow X\}$
$\mathcal{R}_7 = QRE$	$\mathcal{F}_7 = \{QR \rightarrow E, Q \rightarrow R\}$

5. Test whether a candidate key is in one of the schemata

   The key W is in schemata R_2 and R_6, so no schema needs to be added.

Task 7 (Decomposition Algorithm) [3 points]

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

$$\mathcal{R} = ABCDEF$$
$$\mathcal{F} = \{BC \to CDE, E \to AF, D \to BE\}$$

The keys are BC and CD.

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:

The first FD satisfies the BCNF, but the other two FDs do not, as they are not trivial and the left side contains no super key. There are two possibile ways to decompose:

Remark: we illustrate in every hull $\mathcal{F}_i^+[\mathcal{R}_j]$ only the non trivial FDs.

- We pick the FD $E \to AF$ and obtain:

  $\mathcal{R}_1 = AEF \qquad \mathcal{F}_1 = \mathcal{F}^+[\mathcal{R}_1] = \{E \to AF\}$
  key: E

  $\mathcal{R}_2 = BCDE \qquad \mathcal{F}_2 = \mathcal{F}^+[\mathcal{R}_2] = \{BC \to DE, D \to BE\}$
  key: BC, CD

  $\mathcal{R}_2$ is not in BCNF and therefore has to be further decomposed.

  $\mathcal{R}_1 = AEF \qquad \mathcal{F}_1 = \mathcal{F}^+[\mathcal{R}_1] = \{E \to AF\}$
  key: E

  $\mathcal{R}_{2,1} = BDE \qquad \mathcal{F}_{2,1} = \mathcal{F}_2^+[\mathcal{R}_{2,1}] = \{D \to BE\} \qquad$ key: D

  $\mathcal{R}_{2,2} = CD \qquad \mathcal{F}_{2,2} = \mathcal{F}_2^+[\mathcal{R}_{2,2}] = \emptyset \qquad$ key: CD

  Now all relation schemata satisfy the BCNF.

  The decomposition is not dependency preserving, as for instance the FG $CD \to AF$ is lost.

- We pick the FD $D \to BE$ and obtain:

  $\mathcal{R}_1 = BDE \qquad \mathcal{F}_1 = \mathcal{F}^+[\mathcal{R}_1] = \{D \to BE\}$
  Schlüssel: D

  $\mathcal{R}_2 = ACDF \qquad \mathcal{F}_2 = \mathcal{F}^+[\mathcal{R}_2] = \{CD \to AF\}$
  Schlüssel: CD

Both schemata satisfy the BCNF. The decomposition is not dependency preserving, as for instance the FG $E \rightarrow AF$ is lost.