Technische Hochschule Rosenheim

Exercise 06: Relational database design and normal forms

Task 1: Normal forms

Given the following relations R and S in the first normal form with functional dependencies F :

$$\begin{aligned} R &= (A, B, C, D, E, F)\ mit \\ A, B &\rightarrow C, D, E \\ D &\rightarrow F \\ A, B, C &\rightarrow D, E \end{aligned} \qquad \begin{aligned} S &= (V, W, X, Y, Z)\ mit \\ V &\rightarrow W, X, Y, Z \\ W, Z &\rightarrow V, X, Y \\ Y &\rightarrow Z. \end{aligned}$$

1. First, use the `COVER` algorithm to simplify the set of functional dependencies.
2. Determine the keys of the relations.
3. Do the relations correspond with the second normal form (2NF)?
4. Do the relations correspond with the third normal form (3NF)?
5. Do the relations correspond with the Boyce-Codd normal form (BCNF)?

COVER for R:

1. $F_C = F$
2. $F_C = SPLITTING(F_C) =$

$$\begin{array}{rcl} A,B & \rightarrow & C \\ A,B & \rightarrow & D \\ A,B & \rightarrow & E \\ D & \rightarrow & F \\ A,B,C & \rightarrow & D \\ A,B,C & \rightarrow & E \end{array}$$

3. Minimise left sides

$$\begin{array}{lll} C & \notin & \{A\}^+_F = \{A\} \\ C & \notin & \{B\}^+_F = \{B\} \\ D & \notin & \{A\}^+_F = \{A\} \\ D & \notin & \{B\}^+_F = \{B\} \\ E & \notin & \{A\}^+_F = \{A\} \\ E & \notin & \{B\}^+_F = \{B\} \\ D & \notin & \{C\}^+_F = \{C\} \\ E & \notin & \{C\}^+_F = \{C\} \\ D & \notin & \{B,C\}^+_F = \{B,C\} \\ E & \notin & \{B,C\}^+_F = \{B,C\} \\ D & \notin & \{A,C\}^+_F = \{A,C\} \\ E & \notin & \{A,C\}^+_F = \{A,C\} \\ D & \in & \{A,B\}^+_F = \{A,B,C,D,E,F\} \\ & \Rightarrow & F_C = F_C - (A,B,C \rightarrow D) \cup (A,B \rightarrow D) \\ E & \in & \{A,B\}^+_F = \{A,B,C,D,E,F\} \\ & \Rightarrow & F_C = F_C - (A,B,C \rightarrow E) \cup (A,B \rightarrow E) \end{array}$$

4. Remove unnecessary FDs

$$\begin{array}{lll} C & \notin & \{A,B\}^+_{F_C-(A,B\rightarrow C)} = \{A,B,D,E,F\} \\ D & \notin & \{A,B\}^+_{F_C-(A,B\rightarrow D)} = \{A,B,C,E,F\} \\ E & \notin & \{A,B\}^+_{F_C-(A,B\rightarrow E)} = \{A,B,C,D,F\} \\ F & \notin & \{D\}^+_{F_C-(D\rightarrow F)} = \{D\} \end{array}$$

5. Summarise left sides

$$\begin{array}{rcl} A,B & \rightarrow & C,D,E \\ D & \rightarrow & F \end{array}$$

COVER for S:

1. $F_C = F$
2. $F_C = SPLITTING(F_C) =$

$$\begin{array}{rcl} V & \rightarrow & W \\ V & \rightarrow & X \\ V & \rightarrow & Y \\ V & \rightarrow & Z \\ W,Z & \rightarrow & V \\ W,Z & \rightarrow & X \\ W,Z & \rightarrow & Y \\ Y & \rightarrow & Z \end{array}$$

3. Minimise left sides

$$\begin{array}{lll} V & \notin & \{W\}^+_F = \{W\} \\ V & \notin & \{Z\}^+_F = \{Z\} \\ X & \notin & \{W\}^+_F = \{W\} \\ X & \notin & \{Z\}^+_F = \{Z\} \\ Y & \notin & \{W\}^+_F = \{W\} \\ Y & \notin & \{Z\}^+_F = \{Z\} \end{array}$$

4. Remove unnecessary DFs

$$\begin{array}{lll} W & \notin & \{V\}^+_{F_C-(V\rightarrow W)} = \{V,X,Y,Z\} \\ X & \in & \{V\}^+_{F_C-(V\rightarrow X)} = \{V,W,X,Y,Z\} \\ & & \Rightarrow F_C = F_C - (V \rightarrow X) \\ Y & \in & \{V\}^+_{F_C-(V\rightarrow Y)} = \{V,W,X,Y,Z\} \\ & & \Rightarrow F_C = F_C - (V \rightarrow Y) \\ Z & \notin & \{V\}^+_{F_C-(V\rightarrow Z)} = \{V,W\} \\ V & \notin & \{W,Z\}^+_{F_C-(W,Z\rightarrow V)} = \{W,Z,X,Y\} \\ X & \notin & \{W,Z\}^+_{F_C-(W,Z\rightarrow X)} = \{W,Z,V,Y,Z\} \\ Y & \notin & \{W,Z\}^+_{F_C-(W,Z\rightarrow Y)} = \{W,Z,V,X\} \\ Z & \notin & \{Y\}^+_{F_C-(Y\rightarrow Z)} = \{Y\} \end{array}$$

5. Summarise left sides

$$\begin{array}{rcl} V & \rightarrow & W,Z \\ W,Z & \rightarrow & V,X,Y \\ Y & \rightarrow & Z \end{array}$$

If the rule set $W,Z \rightarrow V,X,Y$ is processed first when removing unnecessary functional dependencies, then you get a different solution, but this is also correct:

$$\begin{array}{rcl} V & \rightarrow & W,X,Y \\ W,Z & \rightarrow & V \\ Y & \rightarrow & Z \end{array}$$

Keys for R based on heuristics:

1. Attributes that do not occur in any functional dependency: none
2. Attributes that do not occur on any right side of a functional dependency: A, B
3. Test whether A, B is key: $\{A,B\}_F^+ = \{A,B,C,D,E,F\} = R$
4. A, B is the only key of R

Schlüssel für S nach Heuristik:

1. Attributes that do not occur in any functional dependency: none
2. Attributes that do not occur on any right side of a functional dependency: none
3. Test single element sets:

$$\begin{aligned}
\{V\}_F^+ &= \{V,W,X,Y,Z\} = S \\
\{W\}_F^+ &= \{W\} \\
\{X\}_F^+ &= \{X\} \\
\{Y\}_F^+ &= \{Y,Z\} \\
\{Z\}_F^+ &= \{Z\}
\end{aligned}$$

4. Test two element set without V

$$\begin{aligned}
\{W,X\}_F^+ &= \{W,X\} \\
\{W,Y\}_F^+ &= \{W,Y,Z,V,X\} = S \\
\{W,Z\}_F^+ &= \{W,Z,V,X,Y\} = S \\
\{X,Y\}_F^+ &= \{X,Y,Z\} \\
\{X,Z\}_F^+ &= \{X,Z\} \\
\{Y,Z\}_F^+ &= \{Y,Z\}
\end{aligned}$$

5. Test three element set without V, W, Z or W, Y

$$\{X,Y,Z\}_F^+ = \{X,Y,Z\}$$

6. V, W, Y and W, Z are the keys of S.

2NF, 3NF und BCNF für R:

- R corresponds with the 2NF, because every non-key attribute depends on the whole key. C, D, E depend directly on A, B and F is transitively dependent on A, B. If it is somehow possible to construct a functional dependency where a non-key attribute only depends on a part of the key, then this relation does not correspond with the 2NF.
- R does not correspond with the 3NF, because F is transitively dependent on A, B. If it is somehow possible to construct a transitive dependency in which a non-key attribute does not depend directly on the key, the 3NF is violated. Typically, a transitive dependency becomes visible through `COVER`, since functional dependencies such as A, B F are superfluous and are removed by the algorithm.
- Since R does not correspond with the 3NF, R also does not correspond with the BCNF.

2NF, 3NF and BCNF for S:

- S corresponds with the 2NF, since the remaining non-key attribute X depends on V, depends on W, Z and depends on W, Y. All non-key attributes depend on all keys, and all non-key attributes don't depend on a key but only on a subset.
- S corresponds with the 3NF, because the remaining non-key attributes are not transitively dependent on any of the keys. X depends directly on V and directly on W, Z. Y depends directly on V and directly on W, Z.
- S does not correspond with the BCNF, because not every attribute depends directly, i.e. not transitively, on a key. V depends on W, Z. But W, Z depends on V and X depends on V and on W, Z, just like Y. So far everything fits. But: Z depends on V but is unfortunately only transitively dependent on W, Z. The criterion for BCNF is thus violated. Or put more simply: Since $Y \rightarrow Z$, then Y would have to be a superkey.