Tutorial: Normal Forms and Normalisation

1. Consider $R = \{A, B, C, D, E\}$ with $\Sigma = \{\{A\} \rightarrow \{A, B, C\}, \{A, B\} \rightarrow \{A\}, \{B, C\} \rightarrow \{A, D\}, \{B\} \rightarrow \{A, B\}, \{C\} \rightarrow \{D\}\}$

   (a) What preliminary work is needed to study the normalisation of R with Σ?

Solution: We need to prepare the work by doing the following.
We compute the attribute closure.
$\{A\}^+ = \{A, B, C, D\}$
$\{B\}^+ = \{A, B, C, D\}$
$\{C\}^+ = \{C, D\}$
$\{D\}^+ = \{D\}$
$\{E\}^+ = \{E\}$
$\{A, B\}^+ = \{A, B, C, D\}$
$\{A, C\}^+ = \{A, B, C, D\}$
$\{A, D\}^+ = \{A, B, C, D\}$
$\underline{\{A, E\}^+} = \{A, B, C, D, E\}$
$\{B, C\}^+ = \{A, B, C, D\}$
$\{B, D\}^+ = \{A, B, C, D\}$
$\underline{\{B, E\}^+} = \{A, B, C, D, E\}$
$\{C, D\}^+ = \{C, D\}$
$\{C, E\}^+ = \{C, D, E\}$
$\{D, E\}^+ = \{D, E\}$
Other attribute closures need not be computed
The candidate keys are $\{A, E\}$ and $\{B, E\}$.
The prime attributes are A, B and E.

Solution: At this point or later (for 3NF synthesis) we should also compute a minmal cover and a compact minimal cover
We start from Σ:
$\{A\} \rightarrow \{A, B, C\}$
$\{A, B\} \rightarrow \{A\}$
$\{B, C\} \rightarrow \{A, D\}$
$\{B\} \rightarrow \{A, B\}$
$\{C\} \rightarrow \{D\}$
Step 1, we simplify the right-hand sides:
$\{A\} \rightarrow \{A\}$
$\{A\} \rightarrow \{B\}$
$\{A\} \rightarrow \{C\}$
$\{A, B\} \rightarrow \{A\}$
$\{B, C\} \rightarrow \{A\}$
$\{B, C\} \rightarrow \{D\}$
$\{B\} \rightarrow \{A\}$
$\{B\} \rightarrow \{B\}$
$\{C\} \rightarrow \{D\}$
Step 2, we simplify the left-hand sides:
$\{A\} \rightarrow \{A\}$.
$\{A\} \rightarrow \{B\}$.
$\{A\} \rightarrow \{C\}$.

$\{A, \cancel{B}\} \rightarrow \{A\}$ because $\{A\} \rightarrow \{A\}$.
$\{B, \cancel{C}\} \rightarrow \{A\}$ because $\{B\} \rightarrow \{A\}$.
$\{B, \cancel{C}\} \rightarrow \{D\}$ because $\{B\} \rightarrow \{D\}$ (we could also do $\{\cancel{B}, C\} \rightarrow \{D\}$ because $\{C\} \rightarrow \{D\}$).
Note that we know that $\{B\} \rightarrow \{D\}$ because $\{B\}^+ = \{A, B, C, D\}$.
$\{B\} \rightarrow \{A\}$.
$\{B\} \rightarrow \{B\}$.
$\{C\} \rightarrow \{D\}$.
Step 3, we simplify the set:
$\cancel{\{A\} \rightarrow \{A\}}$ because it is trivial.
$\{A\} \rightarrow \{B\}$.
$\{A\} \rightarrow \{C\}$.
$\{B\} \rightarrow \{A\}$.
$\cancel{\{B\} \rightarrow \{D\}}$ because it can be derived from the others.
$\cancel{\{B\} \rightarrow \{B\}}$ $\cancel{\{A\} \rightarrow \{A\}}$
$\{C\} \rightarrow \{D\}$.
The result is:
$\{A\} \rightarrow \{B\}$
$\{A\} \rightarrow \{C\}$
$\{B\} \rightarrow \{A\}$
$\{C\} \rightarrow \{D\}$
Note that there can be other minimal covers that the algorithm can compute by considering the functional dependencies in a different order at each step of the algorithm. This is not the case in the example.
However, there is a minimal cover that the algorithm cannot compute:
$\{A\} \rightarrow \{B\}$
$\{B\} \rightarrow \{A\}$
$\{B\} \rightarrow \{C\}$
$\{C\} \rightarrow \{D\}$
If the algorithm starts from Σ^+, then it can find all minimal covers.
A compact minimal cover is:
$\{A\} \rightarrow \{B, C\}$
$\{B\} \rightarrow \{A\}$
$\{C\} \rightarrow \{D\}$

(b) Is R with Σ in 3NF?

Solution: Let us look at the non-trivial functional dependencies of the form $X \rightarrow \{A\}$ derived from Σ. Namely after removing the trivial functional dependencies after step 1 of the minimal cover algorithm. Equivalently, we could use a minimal cover.
$\{A\} \rightarrow \{C\}$ is non-trivial, $\{A\}$ is not a superkey and $\{C\}$ is not a prime attribute. This functional dependency violates the three conditions of the 3NF definition. R with Σ is not in 3NF.
Incidentally, several other functional dependencies also violate the 3NF definition:
$\{A\} \rightarrow \{B\}$ is non-trivial, $\{A\}$ is not a superkey and $\{B\}$ is not a prime attribute.
$\{B, C\} \rightarrow \{D\}$ is non-trivial, $\{B, C\}$ is not a superkey and $\{D\}$ is not a prime attribute.
$\{B\} \rightarrow \{C\}$ is non-trivial, $\{B\}$ is not a superkey and $\{C\}$ is not a prime attribute.
$\{C\} \rightarrow \{D\}$ is non-trivial, $\{C\}$ is not a superkey and $\{D\}$ is not a prime attribute.
This one does not (one condition is met):
$\{B, C\} \rightarrow \{A\}$ $\{A\}$ is a prime attribute.

(c) Is R with Σ in BCNF?

Solution: R with Σ is not in 3NF, then it cannot be in BCNF.

Let us look at the non-trivial functional dependencies of the form $X \rightarrow \{A\}$ derived from Σ. Namely after removing the trivial functional dependencies after step 1 of the minimal cover algorithm. Equivalently, we could use a minimal cover.

$\{A\} \rightarrow \{C\}$ is non-trivial and $\{A\}$ is not a superkey . This functional dependency violates the two conditions of the BCNF definition. R with Σ is not in BCNF.

Incidentally, all the other functional dependencies also violate the BCNF definition:

$\{A\} \rightarrow \{B\}$ is non-trivial and $\{A\}$ is not a superkey.

$\{B, C\} \rightarrow \{D\}$ is non-trivial and $\{B, C\}$ is not a superkey.

$\{B\} \rightarrow \{C\}$ is non-trivial and $\{B\}$ is not a superkey.

$\{C\} \rightarrow \{D\}$ is non-trivial and $\{C\}$ is not a superkey .

$\{B, C\} \rightarrow \{A\}$ $\{A\}$ is non-trivial and $\{B, C\}$ is not a superkey.

(d) Decompose R with Σ into a BCNF decomposition using the algorithm from the lecture.

Solution: We decompose R with its compact minimal cover $\{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$. The candidate keys are $\{A, E\}$ and $\{B, E\}$.

All functional dependencies violate BCNF. We can decompose according to any one.

We use $\{\{A\} \rightarrow \{B, C\}$

We get $R_1 = \{A, B, C, D\}$ with $\Sigma_1 = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$. The candidate keys of R_1 with Σ_1 are $\{A\}$ and $\{B\}$. It is not in BCNF.

We get $R_2 = \{A, E\}$ with $\Sigma_2 = \emptyset$. It is in BCNF.

We decompose $R_1 = \{A, B, C, D\}$ with $\Sigma_1 = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$.

$\{C\} \rightarrow \{D\}$ violates BCNF. We use it to decompose.

We get $R_{1.1} = \{C, D\}$ with $\Sigma_{1.1} = \{\{C\} \rightarrow \{D\}\}$. It is in BCNF.

We get $R_{1.2} = \{A, B, C\}$ with $\Sigma_{1.2} = \{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}\}$. It is in BCNF.

The decomposition is R_2, $R_{1.1}$ and $R_{1.2}$. There are other possible decompositions using the same algorithm.

(e) Is the results lossless?

Solution: Yes, the algorithm guarantees that the result is lossless.

(f) Is the results dependency preserving?

Solution: The decomposition R_2, $R_{1.1}$ and $R_{1.2}$ is dependency preserving.

(g) Synthesise R with Σ into a 3NF decomposition using the algorithm from the lecture.

Solution: First compute a compact minimal cover:

$\{A\} \rightarrow \{B, C\}$

$\{B\} \rightarrow \{A\}$

$\{C\} \rightarrow \{D\}$

For each functional dependency create a fragment:

$R_1 = \{A, B, C\}$

$R_2 = \{B, A\}$, this fragment is not kept: it is subsumed by R_1.

$R_3 = \{C, D\}$.

None of the fragments contain a candidate key. We choose one: say $\{A, E\}$, and add it as fragment:

$R_4 = \{A, E\}$.
The result is: $R_1 = \{\underline{A}, \underline{B}, C\}$ $R_3 = \{\underline{C}, D\}$. $R_4 = \{\underline{A, E}\}$.
The algorithm always works. It is guaranteed to find a decomposition in 3NF. Note that using a different minimal cover or compact minimal cover may give a different (but equally correct) result.

(h) Is the results lossless?

Solution: Yes, the algorithm guarantees that the result is lossless.

(i) Is the results dependency preserving?

Solution: Yes, the algorithm guarantees that the result is dependency preserving.

(j) Is the results in BCNF?

Solution: The algorithm guarantees that the result is in 3NF. In this case, it is also in BCNF! (you can check). It is not always but often the case.