BT5110: Tutorial 9

Pratik Karmakar

School of Computing,
National University of Singapore

November, 2024

Question

Your company, Apasaja Private Limited, is commissioned by an online company offering several services to design the relational schema the management of their users' profiles. A service is fully described and identified by its name. Each user can register to one or more services. A user is uniquely identified by her email as well as by her mobile number. Each user has both a postal address and a country of residence. The postal address, however, unambiguously identifies the country in which it is located. There can be several users with the same address.
However, we are only given an abstract schema for this application.
Consider the relations $R = \{A, B, C, D, E\}$ with the set of functional dependencies $\Sigma = \{\{A\} \rightarrow \{A, B, C\}, \{A, B\} \rightarrow \{A\}, \{B, C\} \rightarrow \{A, D\}, \{B\} \rightarrow \{A, B\}, \{C\} \rightarrow \{D\}\}$.

Question

1. Normal Forms
   (a) Is R with Σ in 3NF?
   (b) Is R with Σ in BCNF?
2. Normalisation
   (a) Synthesise R with Σ into a 3NF decomposition using the algorithm from the lecture.
   (b) Is the result lossless?
   (c) Is the result dependency preserving?
   (d) Is the result in BCNF?
   (e) Decompose R with Σ into a BCNF decomposition using the algorithm from the lecture.
   (f) Is the result lossless?
   (g) Is the result dependency preserving?

Question 1.a. Is R with Σ in 3NF?

Question 1.a. Is R with Σ in 3NF?

Criteria for 3NF

If $X \rightarrow \{A\} \in \Sigma$:

- $X \rightarrow \{A\}$ is trivial or
- X is a superkey or
- A is a prime attribute

Question 1.a. Is R with Σ in 3NF?

Criteria for 3NF

If $X \rightarrow \{A\} \in \Sigma$:

- $X \rightarrow \{A\}$ is trivial or
- X is a superkey or
- A is a prime attribute

Compact minimal cover: $\{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
Candidate keys: $\{A, E\}$ and {B,E}
Prime attributes: A, B, E

Question 1.a. Is R with Σ in 3NF?

Criteria for 3NF

If $X \rightarrow \{A\} \in \Sigma$:

- $X \rightarrow \{A\}$ is trivial or
- X is a superkey or
- A is a prime attribute

Compact minimal cover: $\{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
Candidate keys: $\{A, E\}$ and {B,E}
Prime attributes: A, B, E
We notice that $\{A\} \rightarrow \{C\}$ is non-trivial, A is not a superkey and C is not a prime attribute

Question 1.a. Is R with Σ in 3NF?

Criteria for 3NF

If $X \rightarrow \{A\} \in \Sigma$:

- $X \rightarrow \{A\}$ is trivial or
- X is a superkey or
- A is a prime attribute

Compact minimal cover: $\{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
Candidate keys: $\{A, E\}$ and {B,E}
Prime attributes: A, B, E
We notice that $\{A\} \rightarrow \{C\}$ is non-trivial, A is not a superkey and C is not a prime attribute $\implies$ R with Σ is **not in** 3NF.

Question 1.b. Is R with Σ in BCNF?

Question 1.b. Is R with Σ in BCNF?

Criteria for 3NF

If $X \rightarrow \{A\} \in \Sigma$:

- $X \rightarrow \{A\}$ is trivial or
- X is a superkey or
- A is a prime attribute

Criteria for BCNF

If $X \rightarrow \{A\} \in \Sigma$:

- $X \rightarrow \{A\}$ is trivial or
- X is a superkey

Question 1.b. Is R with Σ in BCNF?

Criteria for 3NF

If $X \rightarrow \{A\} \in \Sigma$:

- $X \rightarrow \{A\}$ is trivial or
- X is a superkey or
- A is a prime attribute

Criteria for BCNF

If $X \rightarrow \{A\} \in \Sigma$:

- $X \rightarrow \{A\}$ is trivial or
- X is a superkey

$\Longrightarrow$ The criteria of being in BCNF are stricter than those of being in 3NF.

Question 1.b. Is R with Σ in BCNF?

Criteria for 3NF

If $X \rightarrow \{A\} \in \Sigma$:

- $X \rightarrow \{A\}$ is trivial or
- X is a superkey or
- A is a prime attribute

Criteria for BCNF

If $X \rightarrow \{A\} \in \Sigma$:

- $X \rightarrow \{A\}$ is trivial or
- X is a superkey

$\Longrightarrow$ The criteria of being in BCNF are stricter than those of being in 3NF.
As R with Σ is not in 3NF, it **cannot be** in BCNF.

Question 2.a. Synthesise R with Σ into a 3NF decomposition using the algorithm from the lecture.

Question 2.a. Synthesise R with Σ into a 3NF decomposition using the algorithm from the lecture.

Compact minimal cover: $\{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
Candidate keys: $\{A, E\}, \{B, E\}$

Question 2.a. Synthesise R with Σ into a 3NF decomposition using the algorithm from the lecture.

Compact minimal cover: $\{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
Candidate keys: $\{A, E\}, \{B, E\}$
Fragments: $R_1 = \{A, B, C\}, R_2 = \{A, B\}, R_3 = \{C, D\}$

Question 2.a. Synthesise R with Σ into a 3NF decomposition using the algorithm from the lecture.

Compact minimal cover: $\{\{A\} \to \{B, C\}, \{B\} \to \{A\}, \{C\} \to \{D\}\}$
Candidate keys: $\{A, E\}, \{B, E\}$
Fragments: $R_1 = \{A, B, C\}, R_2 = \{A, B\}, R_3 = \{C, D\}$
R_2 is subsumed by R_1. Thus we remove R_2.

Question 2.a. Synthehsise R with Σ into a 3NF decomposition using the algorithm from the lecture.

Compact minimal cover: $\{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
Candidate keys: $\{A, E\}, \{B, E\}$
Fragments: $R_1 = \{A, B, C\}, R_2 = \{A, B\}, R_3 = \{C, D\}$
R_2 is subsumed by R_1. Thus we remove R_2.
Now we have: $R_1 = \{A, B, C\}, R_3 = \{C, D\}$

Question 2.a. Synthehsise R with Σ into a 3NF decomposition using the algorithm from the lecture.

Compact minimal cover: $\{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
Candidate keys: $\{A, E\}, \{B, E\}$
Fragments: $R_1 = \{A, B, C\}, R_2 = \{A, B\}, R_3 = \{C, D\}$
R_2 is subsumed by R_1. Thus we remove R_2.
Now we have: $R_1 = \{A, B, C\}, R_3 = \{C, D\}$
None of the fragments contain a candidate key. Thus we choose $\{A, E\}$ (we could choose $\{B, E\}$ as well) to add as another fragment $R_4 = \{A, E\}$.

Question 2.a. Syntehsise R with Σ into a 3NF decomposition using the algorithm from the lecture.

Compact minimal cover: $\{\{A\} \rightarrow \{B, C\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
Candidate keys: $\{A, E\}, \{B, E\}$
Fragments: $R_1 = \{A, B, C\}, R_2 = \{A, B\}, R_3 = \{C, D\}$
R_2 is subsumed by R_1. Thus we remove R_2.
Now we have: $R_1 = \{A, B, C\}, R_3 = \{C, D\}$
None of the fragments contain a candidate key. Thus we choose $\{A, E\}$ (we could choose $\{B, E\}$ as well) to add as another fragment $R_4 = \{A, E\}$.
Finally we have: $R_1 = \{\underline{A}, \underline{B}, C\}, R_3 = \{\underline{C}, D\}, R_4 = \{\underline{A, E}\}$

Question 2.b. Is the result lossless?

Question 2.b. Is the result lossless?

Yes. It is guaranteed to be lossless by the algorithm.

Question 2.c. Is the result dependency preserving?

Question 2.c. Is the result dependency preserving?

Yes. It is guaranteed to be dependency preserving by the algorithm.

Question 2.d. Is the result in BCNF?

[1] Dependency projection algorithm: Algorithm 1

Question 2.d. Is the result in BCNF?

$R_1 = \{A, B, C\}$
$\Sigma_1 = \{\{A\} \rightarrow \{A, B, C\}, \{A, B\} \rightarrow \{A\}, \{B, C\} \rightarrow \{A\}, \{B\} \rightarrow \{A, B\}\}$
Candidate keys: $\{A\}, \{B\}$

[1] Dependency projection algorithm: Algorithm 1

Question 2.d. Is the result in BCNF?

$R_1 = \{A, B, C\}$
$\Sigma_1 = \{\{A\} \rightarrow \{A, B, C\}, \{A, B\} \rightarrow \{A\}, \{B, C\} \rightarrow \{A\}, \{B\} \rightarrow \{A, B\}\}$
Candidate keys: $\{A\}, \{B\}$

$R_3 = \{C, D\}$
$\Sigma_3 = \{\{C\} \rightarrow \{D\}\}$
Candidate keys: $\{C\}$

[1] Dependency projection algorithm: Algorithm 1

Question 2.d. Is the result in BCNF?

$R_1 = \{A, B, C\}$
$\Sigma_1 = \{\{A\} \rightarrow \{A, B, C\}, \{A, B\} \rightarrow \{A\}, \{B, C\} \rightarrow \{A\}, \{B\} \rightarrow \{A, B\}\}$
Candidate keys: $\{A\}, \{B\}$

$R_3 = \{C, D\}$
$\Sigma_3 = \{\{C\} \rightarrow \{D\}\}$
Candidate keys: $\{C\}$

$R_4 = \{A, E\}$
$\Sigma_4 = \emptyset$
Candidate keys: {A,E}

[1] Dependency projection algorithm: Algorithm 1

Question 2.d. Is the result in BCNF?

$R_1 = \{A, B, C\}$
$\Sigma_1 = \{\{A\} \rightarrow \{A, B, C\}, \{A, B\} \rightarrow \{A\}, \{B, C\} \rightarrow \{A\}, \{B\} \rightarrow \{A, B\}\}$
Candidate keys: $\{A\}, \{B\}$

$R_3 = \{C, D\}$
$\Sigma_3 = \{\{C\} \rightarrow \{D\}\}$
Candidate keys: $\{C\}$

$R_4 = \{A, E\}$
$\Sigma_4 = \emptyset$
Candidate keys: {A,E}

Σ_1, Σ_3 and Σ_4 are projected dependencies. We obtain these by projecting Σ on R_1, R_3 and R_4 respectively.[1]

[1] Dependency projection algorithm: Algorithm 1

Question 2.d. Is the result in BCNF?

$R_1 = \{A, B, C\}$
$\Sigma_1 = \{\{A\} \to \{A, B, C\}, \{A, B\} \to \{A\}, \{B, C\} \to \{A\}, \{B\} \to \{A, B\}\}$
Candidate keys: $\{A\}, \{B\}$

$R_3 = \{C, D\}$
$\Sigma_3 = \{\{C\} \to \{D\}\}$
Candidate keys: $\{C\}$

$R_4 = \{A, E\}$
$\Sigma_4 = \emptyset$
Candidate keys: {A,E}

Σ_1, Σ_3 and Σ_4 are projected dependencies. We obtain these by projecting Σ on R_1, R_3 and R_4 respectively.[1]

R_1 with Σ_1, R_3 with Σ_3 and R_4 with Σ_4 satisfy the BCNF criteria individually.

[1] Dependency projection algorithm: Algorithm 1

Question 2.d. Is the result in BCNF?

$R_1 = \{A, B, C\}$
$\Sigma_1 = \{\{A\} \rightarrow \{A, B, C\}, \{A, B\} \rightarrow \{A\}, \{B, C\} \rightarrow \{A\}, \{B\} \rightarrow \{A, B\}\}$
Candidate keys: $\{A\}, \{B\}$

$R_3 = \{C, D\}$
$\Sigma_3 = \{\{C\} \rightarrow \{D\}\}$
Candidate keys: $\{C\}$

$R_4 = \{A, E\}$
$\Sigma_4 = \emptyset$
Candidate keys: {A,E}

Σ_1, Σ_3 and Σ_4 are projected dependencies. We obtain these by projecting Σ on R_1, R_3 and R_4 respectively.[1]

R_1 with Σ_1, R_3 with Σ_3 and R_4 with Σ_4 satisfy the BCNF criteria individually.
Thus the result is in BCNF. (This is not a general case though. There is no guarantee that synthesising a 3NF will yield a BCNF too.)

[1] Dependency projection algorithm: Algorithm 1

Question 2.e. Decompose R with Σ into a BCNF decomposition using the algorithm from the lecture.

We found that $\{A\} \rightarrow \{C\}$ violates the BCNF condition. (A is not a superkey)

Question 2.e. Decompose R with Σ into a BCNF decomposition using the algorithm from the lecture.

We found that $\{A\} \rightarrow \{C\}$ violates the BCNF condition. (A is not a superkey)
So, we decompose R with Σ as below:
$R_1 = \{A\}^+ = \{A, B, C, D\}$ with $\Sigma_1 = \{\{A\} \rightarrow \{C\}, \{A\} \rightarrow \{B\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
$R_2 = \{A, E\}$ with $\Sigma_2 = \emptyset$

Question 2.e. Decompose R with Σ into a BCNF decomposition using the algorithm from the lecture.

We found that $\{A\} \rightarrow \{C\}$ violates the BCNF condition. (A is not a superkey)
So, we decompose R with Σ as below:
$R_1 = \{A\}^+ = \{A, B, C, D\}$ with $\Sigma_1 = \{\{A\} \rightarrow \{C\}, \{A\} \rightarrow \{B\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
$R_2 = \{A, E\}$ with $\Sigma_2 = \emptyset$
R_2 with Σ_2 is in BCNF.
R_1 with Σ_1 is not in BCNF because $\{C\} \rightarrow \{D\}$ violates the BCNF condition.

Question 2.e. Decompose R with Σ into a BCNF decomposition using the algorithm from the lecture.

We found that $\{A\} \rightarrow \{C\}$ violates the BCNF condition. (A is not a superkey)
So, we decompose R with Σ as below:
$R_1 = \{A\}^+ = \{A, B, C, D\}$ with $\Sigma_1 = \{\{A\} \rightarrow \{C\}, \{A\} \rightarrow \{B\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
$R_2 = \{A, E\}$ with $\Sigma_2 = \emptyset$
R_2 with Σ_2 is in BCNF.
R_1 with Σ_1 is not in BCNF because $\{C\} \rightarrow \{D\}$ violates the BCNF condition.
We decompose it into two fragments:
$R_{1.1} = \{C\}^+ = \{C, D\}$ with $\Sigma_{1.1} = \{\{C\} \rightarrow \{D\}\}$
$R_{1.2} = \{A, B, C\}$ with $\Sigma_{1.2} = \{\{A\} \rightarrow \{B\}, \{A\} \rightarrow \{C\}, \{B\} \rightarrow \{A\}\}$
$R_{1.1}$ with $\Sigma_{1.1}$ is in BCNF.
$R_{1.2}$ with $\Sigma_{1.2}$ is in BCNF.

Question 2.e. Decompose R with Σ into a BCNF decomposition using the algorithm from the lecture.

We found that $\{A\} \rightarrow \{C\}$ violates the BCNF condition. (A is not a superkey)
So, we decompose R with Σ as below:
$R_1 = \{A\}^+ = \{A, B, C, D\}$ with $\Sigma_1 = \{\{A\} \rightarrow \{C\}, \{A\} \rightarrow \{B\}, \{B\} \rightarrow \{A\}, \{C\} \rightarrow \{D\}\}$
$R_2 = \{A, E\}$ with $\Sigma_2 = \emptyset$
R_2 with Σ_2 is in BCNF.
R_1 with Σ_1 is not in BCNF because $\{C\} \rightarrow \{D\}$ violates the BCNF condition.
We decompose it into two fragments:
$R_{1.1} = \{C\}^+ = \{C, D\}$ with $\Sigma_{1.1} = \{\{C\} \rightarrow \{D\}\}$
$R_{1.2} = \{A, B, C\}$ with $\Sigma_{1.2} = \{\{A\} \rightarrow \{B\}, \{A\} \rightarrow \{C\}, \{B\} \rightarrow \{A\}\}$
$R_{1.1}$ with $\Sigma_{1.1}$ is in BCNF.
$R_{1.2}$ with $\Sigma_{1.2}$ is in BCNF.
The final BCNF decomposition is: R_2, $R_{1.1}$, and $R_{1.2}$.

Question 2.f. Is the result lossless?

Question 2.f. Is the result lossless?

Yes. The result is guaranteed to be lossless by the algorithm.

Question 2.g. Is the result dependency preserving?

Question 2.g. Is the result dependency preserving?

Yes. This result is dependency preserving. (We have reached same result as the 3NF synthesis. Thus we can guarantee without checking.)

Question 2.g. Is the result dependency preserving?

Yes. This result is dependency preserving. (We have reached same result as the 3NF synthesis. Thus we can guarantee without checking.)
This is not guaranteed for BCNF decomposition. It may not always be dependency preserving.

Dependency Projection Algorithm

Algorithm 1 Computing FD Projections (closure-based)

Require: Σ : set of functional dependencies on schema R
Require: $R' \subseteq R$: attribute set to project onto
Ensure: Σ' : projection of Σ onto R'

1: $\Sigma' \leftarrow \varnothing$
2: **for all** $Y \subseteq R'$ **do**
3: $\quad T \leftarrow \textsc{Closure}(Y, \Sigma)$ // compute Y^+ w.r.t. Σ
4: $\quad H \leftarrow T \cap R'$
5: $\quad \Sigma' \leftarrow \Sigma' \cup \{ Y \rightarrow H \}$ // optionally emit unit FDs: $\{ Y \rightarrow A \mid A \in H \setminus Y \}$
6: **end for**
7: **return** Σ'