# CS540: Assignment 2

**Ga Young Lee**
Department of Electrical Engineering and Computer Science
Oregon State University
leegay@oregonstate.edu

**Jeffrey M. Young**
Department of Electrical Engineering and Computer Science
Oregon State University
youngjef@oregonstate.edu

# 1 Problem 1

Problem Description:
Consider a relation R with five attributes A, B, C, D, and E. You are given the following functional dependencies: A $\rightarrow$ B, BC$\rightarrow$E, and ED$\rightarrow$A.

### 1.a

Problem Description:
List all keys for R

For these problems we'll use the notation $R$ to refer to the relation, and $\Sigma$ to refer to the set of functional dependency. To refer to a specific functional dependency we use a subscript $i \in \mathbb{N}$, such that $f_0$ = $A \rightarrow B$, $f_1$ = $BC \rightarrow C$, $f_2 = ED \rightarrow A$. We can compute the attribute closure to identify the candidate keys:

$$
\begin{aligned}
A^+ &= AB \\
B^+ &= B \\
C^+ &= C \\
D^+ &= D \\
E^+ &= E \\
ED^+ &= EDAB \\
BC^+ &= BCE
\end{aligned}
$$

We see that the with these functional dependencies no candidate keys are observed. Notice that the codomain (the result) of each functional dependency does not include $CD$. We proceed by augmenting the functional dependencies to add these attributes. Specifically, we augment $f_1$ to yield $f_1$' = $BCD \rightarrow ED$, and $f_0$ with $C$ to yield $f_0$' = $AC \rightarrow BC$, taken with $f_1$' this yields $\Sigma$ = $\{A \rightarrow B, AC \rightarrow BC, BC \rightarrow E$ $BCD \rightarrow ED, ED \rightarrow A\}$, with attribute closures:

$$
\begin{aligned}
A^+ &= AB \\
B^+ &= B \\
C^+ &= C \\
D^+ &= D \\
E^+ &= E \\
ED^+ &= EDAB \\
AC^+ &= ACBE \\
ACD^+ &= ACDEB \\
BCD^+ &= BCDEA \\
ECD^+ &= ECDAB
\end{aligned}
$$

We can see that there are three candidate keys for $R$: $BCD, ACD, ECD$

## 1.b

Problem Description:
Is R in BCNF? If it is not, decompose it into a collection of BCNF relations.

A relation $R$ is in Boyce—Codd Normal Form (BCNF), w.r.t., a set of functional dependencies $\Sigma^+$ if $\forall f. f \in \Sigma^+$, where $f : \alpha \rightarrow \beta$, either of the following hold:

1. $\alpha \rightarrow \beta$ is a trivial functional dependency
2. $\alpha$ is a superkey of $R$

Clearly $R$ is not in BCNF. Consider $f_0$ = $A \rightarrow B$, $f_0$ is not trivial i.e., $f_0$ is not tautological and $\beta \not\subseteq \alpha$, hence (1) does not follow, and $A$ is not a superkey of $R$ and thus (2) also does not follow. Similarly for $f_1$, $f_2$ is in BCNF as $\alpha_2$ is a candidate key for $R$.

We begin the decomposition of $R$ over $f_2$. For each non-BCNF functional dependency, and for each decomposition we create sub-relations indexed over the functional dependency index: $i$, thus $R_0$ is the sub-relation constructed through decomposition over $f_0$. We use the notation $R_{\{2,3\}}$ to refer to the remaining relation $R$ whose $\Sigma$ are those untouched by the decomposition, or in this example $\Sigma$ = $\{f_2, f_3\}$. In general, $R_i$ = $\alpha \cup \beta$, and $R_{\{\Sigma - f_i\}}$ = $R - (\beta - \alpha)$, with $\Sigma$ = $\Sigma$ - $f_i$.

Decomposition on $R$ with $f_2$ yields:

$$\begin{aligned} \alpha &= ED && \text{domain of } f_2 \\ \beta &= A && \text{codomain of } f_2 \\ R_2 &= R(E, D, A) && \text{with } \Sigma_2 = \{f_2\} \\ R_{\{0,1\}} &= R(B, C, D, E) && \text{with } \Sigma = \{f_0, f_1\} \end{aligned}$$

We can now check BCNF over $R_2$. With only a single functional dependency, $R_2$ is in BCNF with candidate key $ED$. However, $R_{\{0,1\}}$ is still not in BCNF with the candidate keys from section 1, $f_0$ is trivially in BCNF because $R_{\{0,1\}}$ does not contain an $A$ and thus $f_0$ does not apply. $f_1$ applies to $R_{\{0,1\}}$ but is not in BCNF because $BC$ is not a candidate key. We continue with decomposition over $f_1$:

$$\begin{aligned} \alpha &= BC && \text{domain of } f_1 \\ \beta &= E && \text{codomain of } f_1 \\ R_1 &= R(B, C, E) && \text{with } \Sigma_1 = \{f_1\} \\ R_{\{0\}} &= R(B, C, D) && \text{with } \Sigma = \{f_0\} \end{aligned}$$

This completes the decomposition, $R_2$ is in BCNF with candidate keys $ED$ and $\Sigma_2$; $R_1$ is in BCNF with candidate keys $BC$ and $\Sigma_1$; and $R_{\{0\}}$ is trivially in BCNF because $f_0$ does not apply to it.

Therefore, R can be decomposed in BCNF as $R_2$(E,D,A), $R_1$(B,C,E), and $R_{\{0\}}$(B,C,D).

### 1.c

Problem Description:
Is R in 3NF? If it is not, convert it into a collection of 3NF relations.

To show that $R$, with $\Sigma$ is in third normal form (3NF) we need to show either of the following:

1. $R$ w.r.t. $\Sigma$ is in BCNF.
2. $\forall A \in (\beta - \alpha).\ A \in$ a candidate key of $R$.

For this problem we have $R$ with attributes $A, B, C, D, E$, with $\Sigma$ = $\{A \to B, BC \to E, ED \to A\}$. From section 1 we know that the candidate keys for $R$ are $\{BCD, ACD, ECD\}$ and that $R$ is not in BCNF w.r.t., $\Sigma$. Thus we check item 2 for each $f \in \Sigma$:

$$(\beta - \alpha)_{f_0} = B,\ B \in BCD$$
$$(\beta - \alpha)_{f_1} = E,\ E \in ECD$$
$$(\beta - \alpha)_{f_2} = A,\ A \in ACD$$

We see that item 2 holds for each functional dependency, hence $R$ is in 3NF.

## 2 Problem 2

Problem Description:
Consider a relation R with five attributes A, B, C, and D. You are given the following functional dependencies: AB→C, AC→B, B→D, and BC→A.

### 2.a

Problem Description:
List all keys for R

To identify (candidate) keys for R, we followed the following steps:

First, find the set of attributes that appear on the RHS but not on the LHS of the given functional dependencies, which is {D}. This set of attributes must not be in a candidate key.

Then, find the closure set of attributes that are not on the RHS, which is { }.

Based on the findings, add one attribute from the set, R-{D}-{ } to check whether the result is a superkey. If yes, check if it's a candidate key to see if it's the minimal set of superkey. If not, add more attributes until we have checked all possibilities.

- {A} is not a superkey.
- {B} is not a superkey.
- {C} is not a superkey.
- {A,B} is a superkey and candidate key since it's the minimal set of superkey.
- {A,C} is a superkey and candidate key since it's the minimal set of superkey.
- {B,C} is a superkey and candidate key since it's the minimal set of superkey.
- {A,B,C} is a superkey but not candidate key since it's not the minimal set of superkey.

Therefore, we can see that the three candidate keys for R are AB, AC, and BC.

### 2.b

Problem Description:
Is R in BCNF? If it is not, decompose it into a collection of BCNF relations.

R is clearly not in the BCNF form since it violates every functional dependency.
Therefore, let's decompose R to meet the above condition for BCNF.

- $f_0$: AB $\rightarrow$ C
- $f_1$: AC $\rightarrow$ B
- $f_2$: B $\rightarrow$ D
- $f_3$: BC $\rightarrow$ A

Consider $f_2$ to decompose R: it can be decomposed into $R_1$: {B,D} and $R_2$:{B,A,C}.
$R_1$: {B,D} is in BCNF, since B from the LHS of $FD_3$ is a key for $R_1$.
$R_2$: {B,A,C} is also in BCNF as it complies with applicable functional dependencies. Specifically, BC from the LHS of $f_3$ is a key for $R_2$. Additionally, $f_0$ holds as AB is a key in $R_2$ as well as $f_1$ holds since AC is a key in this case.

Thus, R can be transformed into a collection of BCNF relations, $R_1$: {B,D} and $R_2$: {B,A,C}.

## 2.c

Problem Description:
Is R in 3NF? If it is not, convert it into a collection of 3NF relations.

To show that $R$, with $\Sigma$ is in 3NF we need to show either of the following:

1. $R$ w.r.t. $\Sigma$ is in BCNF.
2. $\forall A \in (\beta - \alpha)$. $A \in$ a candidate key of $R$.

For this problem we have $R$ with attributes $A, B, C, D$, with $\Sigma = \{AB \to C, AC \to B, B \to D\ BC \to A\}$. From section 2 we know that the candidate keys for $R$ are $\{AB, AC, BC\}$ and that $R$ is not in BCNF w.r.t., $\Sigma$. Thus we check item 2 for each $f \in \Sigma$:

$$(\beta - \alpha)_{f_0} = C,\ C \in AC$$
$$(\beta - \alpha)_{f_1} = B,\ B \in AB$$
$$(\beta - \alpha)_{f_2} = D,\ D \notin AB$$
$$(\beta - \alpha)_{f_3} = A,\ A \in AB$$

We see that item 2 does not hold for $f_2$, hence $R$ is not in 3NF. Therefore, R should be decomposed into 3NF sub-relations.

Given $f_2 : B \to D$, $R$ can be decomposed into $R_1$(B,D) and $R_2$(B,A,C).

$R_1$(B,D) is in 3NF, since B from the LHS of $f_2$ is a key for $R_1$ and $B \in AB, BC$.
$R_2$(B,A,C) is also in 3NF, since since BC from the LHS of $f_3$ is a key for $R_2$ and B,A,C are part of the candidate keys, $AB, AB, BC$.

Thus, R can be transformed into a collection of 3NF relations, $R_1$(B,D) and $R_2$(B,A,C).

# 3 Problem 3

### 3.a

Problem Description:
Given that X, Y, W, Z are attributes in a relation, using Armstrong's axioms, prove that if we have X→Y and Y W→Z, then XW→Z.

Armstrong's axioms are:

**Reflexivity** if $\alpha$ is a set of attributes, and $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$ holds.

**Augmentation** if $\alpha \rightarrow \beta$ holds and $\gamma$ is a set of attributes, then $\gamma\alpha \rightarrow \gamma\beta$ holds.

**Transitivity** if $\alpha \rightarrow \beta$, and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$.

To prove that with $X \rightarrow Y$, and $YW \rightarrow Z$, then $XW \rightarrow Z$, we must show that $XW \rightarrow Z$ can be derived from $X \rightarrow Y$ and $YW \rightarrow Z$:

$$X \rightarrow Y \quad \text{axiom} \tag{1a}$$
$$YW \rightarrow Z \quad \text{axiom} \tag{1b}$$
$$XW \rightarrow YW \quad \text{by augmentation with W} \tag{1c}$$
$$XW \rightarrow Z \quad \text{by transitivity of (1c) and (1d)} \tag{1d}$$

### 3.b

Problem Description:
Given that X, Y, Z are attributes in a relation, using Armstrong's axioms, prove that if we have X→Y and X→Z, then X→YZ

To prove that with $X \rightarrow Y$, and $X \rightarrow Z$, then $X \rightarrow YZ$, we must show that $X \rightarrow YZ$ can be derived from $X \rightarrow Y$ and $X \rightarrow Z$:

$$X \rightarrow Y \quad \text{axiom} \tag{2a}$$
$$X \rightarrow Z \quad \text{axiom} \tag{2b}$$
$$XX \rightarrow XY \quad \text{augmentation on (2a) with X} \tag{2c}$$
$$X \rightarrow XY \quad \text{set union over X} \tag{2d}$$
$$YX \rightarrow YZ \quad \text{augmentation on (2b) with Y} \tag{2e}$$
$$XY \rightarrow YZ \quad \text{commutivity of } \cup \tag{2f}$$
$$X \rightarrow YZ \quad \text{transitivity of (2d) over (2f)} \tag{2g}$$

### 3.c

Problem Description:
Prove that, if relation R has only one simple key, it is in BCNF if and only if it is in 3NF.

To complete the proof we must show both directions:

**BCNF→ 3NF** Given $R$ conforms to BCNF show that $R$ also conforms to 3NF. This follows directly from the definition of 3NF.

**3NF→ BCNF** Given $R$ conforms to 3NF show that $R$ also conforms to BCNF.

Thus, we only have to show 3NF→ BCNF. For this proof we'll use an alternative definition of 3NF (from the alice book definition 11.2.11):

**Definition 3.1** (3NF)**.** A functional dependency schema is in 3NF if whenever $\alpha \to \beta$ is a non-trivial functional dependency implied by $\Sigma$, then either $\alpha$ is a superkey or $\beta$ is a key (also called prime in this book) attribute. A functional dependency schema is then in 3NF if each of its components $f \in \Sigma$ w.r.t., $R$ is.

By assumption $R$ is in 3NF. This means that for each $(f : \alpha \to \beta) \in \Sigma$ we have two cases:

1. $\alpha$ is a super key
2. $\beta$ is a prime attribute. i.e., a member of a candidate key.

*Proof.* If item 1 holds then $R$ is in BCNF. For item 2 we proceed by contradiction. Consider a functional dependency, $f : \alpha \to \beta$ where $f \in \Sigma$ and assume that $\alpha$ is not a superkey. By assumption of 3NF every key $b \in \beta$ is also in a candidate key. But because every key of $R$ is simple, every candidate key, and $\alpha$ is a singleton set, thus $b$ itself is a candidate key. Because $b$ is a candidate key there must be functional dependencies of the form $b \to A - b \in \Sigma$, where $A$ is the set of attributes of $R$. Then by transitivity with $\alpha$ we have $\alpha \to A - b$, hence $\alpha$ is a superkey and we have a contradiction. Therefore, by both cases given by the definition of 3NF, $\alpha$ is a superkey and therefore satisfies the first case of the definition of BCNF, thus implying that $R$ is in BCNF. □

# 4 Problem 4

### 4.a

Problem Description:
Suppose you are given a relation R(A,B,C,D) with functional dependencies B→C and D→A.State whether the decomposition of R to $S_1$(B,C) and $S_2$(A,D) is lossless or dependency preserving and briefly explain why or why not.

#### Showing lossless decomposition

By the problem definition we have:

$$S_1(B, C),\ \Sigma_{S_1} = \{B \rightarrow C\}$$
$$S_1(A, D),\ \Sigma_{S_2} = \{A \rightarrow D\}$$

We use the following test for lossless decomposition:

**Definition 4.1** (Lossless test)**.** Given $R$, $R_1$, and $R_2$, and $\Sigma$. $R_1$ and $R_2$ form a lossless decomposition of $R$ if at least one of the following functional dependencies is in $\Sigma^+$:

1. $R_1 \cap R_2 \rightarrow R_1$
2. $R_1 \cap R_2 \rightarrow R_2$

But clearly $S_1$, and $S_2$ are disjoint, and thus $S_1 \cap S_2 = \emptyset$ and therefore the decomposition is lossy.

#### Dependency preservation

We use the following test for dependency preservation:

**Definition 4.2** (Dependency preservation)**.** Given a relation, $R$, a decomposition $D$ = { $R_0$, $R_1$, ... , $R_i$ } is dependency preserving with respect to a set of functional dependencies $\Sigma$ = { $f_0, f_1, \ldots\ f_n$ }, if $\Sigma$' = $\Sigma^+$, where $\Sigma$' is a restriction of $\Sigma$ with respect to $R_i$, for some $i$.

Rather than using the book algorithm, which will require exponential time to compute $\Sigma^+$, we use the following method based on canonical covers:

1. Find the canonical cover: $\Sigma_c$ of $\Sigma$.
2. Find $\Sigma_c$', where $\Sigma_c$' is the union of restrictions of $\Sigma_c$ for each sub-relation $R_i$.
3. check that $\Sigma_c == \Sigma_c$'

Clearly $\Sigma$ is already a canonical cover as there are no extraneous attributes and the domain for each functional dependency in $\Sigma$ is unique. This means that $\Sigma_c$' = $\{B \rightarrow C\} \cup \{A \rightarrow D\}$ and thus $\Sigma_c$' == $\Sigma_c$ == $\Sigma$. Hence we have a dependency preserving decomposition.

## 4.b

Problem Description:
Prove that the 3NF synthesis algorithm produces a lossless-join decomposition of the relation containing all the original attributes.

The synthesis algorithm to obtain 3NF with preservation of functional dependencies is as follows:

---

**Algorithm 1:** 3NF synthesis algorithm

---

**Input:** A relation schema $(U, \Sigma)$, where $\Sigma$ is a set of functional dependency (FD) that is a minimal cover (also called canonical cover). We assume that each attribute of $U$ occurs in at least one functional dependency of $\Sigma$

**Result:** An FD schema $(R, \Gamma)$ in 3NF

1. If there is an $f \in \Sigma$, where $f : X \rightarrow A$ and $XA = U$, then output $(U, \Sigma)$
2. Otherwise
   (a) for each $(f : X \rightarrow A) \in \Sigma$, include the relational schema $(XA, \{X \rightarrow A\})$ in the output schema $(R, \Gamma)$
   (b) choose a key $X$ of $U$ under $\Sigma$, and include $X, \emptyset$ in the output.

(Reference: Alice Book Ch.11)

---

*Proof.* We proceed with a proof by induction over $\Sigma$. There are two base cases corresponding to item 1 and item 2. The first base case, given by item 1 is lossless simply because by the algorithm the input is returned as output. The second base case, given by item 2b is an edge case, where $\Sigma = \emptyset$, thus for any sub-relation, $U$, produced by item 2b when joined with any other sub-relation $U$' will form a cartesian product and thus is lossless. We prove progress over $\Sigma$ such that for each inductive step on $\Sigma$ w.r.t. $U$, either a decomposition occurs producing a $U' \subset U$, or $\Sigma = \emptyset$ and thus we arrive at a base case.

Consider an $f_i : \alpha \rightarrow \beta \in \Sigma$, we know by assumption that $\Sigma$ is a minimal cover over $U$, which implies that $\alpha$ is unique and $\alpha \notin \beta$. Based on the lossless decomposition test, 4.1, a decomposition is lossless if for two sub-relations $U_1$, and $U_2$ either $U_1 \cup U_2 \rightarrow U_1 \in \Sigma^+$ or $U_1 \cap U_2 \rightarrow U_2 \in \Sigma^+$.

Applying the decomposition algorithm yields two sub-relations: $U_i = \alpha \cup \beta$, and $U_{\{\Sigma'\}} = U - (\beta - \alpha)$, where $\Sigma' = \Sigma - f_i$, clearly $U_{\{\Sigma'\}} \subset U$ since $U_{\{\Sigma'\}} \neq U$ because $\beta \neq \emptyset$. Thus at each recursive step $U$ reduces until a base case, hence we have shown progress. We still must show that at each inductive step the decomposition is lossless. Thus, we must show that either $U_i \cap U_{\{\Sigma'\}} \rightarrow U_i \in \Sigma^+$ or $U_i \cap U_{\{\Sigma'\}} \rightarrow U_{\{\Sigma'\}} \in \Sigma^+$. By assumption of a minimal cover we know that $\alpha$ and $\beta$ are disjoint sets, thus $U_i \cap U_{\{\Sigma'\}} = \alpha$ and so we have either $\alpha \rightarrow U_i \in \Sigma^+$ or $\alpha \rightarrow U_{\{\Sigma'\}} \in \Sigma^+$. Recall that $U_i = \alpha \cup \beta$, thus we have $\alpha \rightarrow \alpha\beta$ which by armstrong's axioms is $\alpha \rightarrow \beta = f_i$ and therefore $f_i \in \Sigma \in \Sigma^+$.

In short, the 3NF synthesis algorithm is information lossless, since JOIN does not produce any bogus tuples. This is primarily due to the fact that the join attribute is a key. When we're decomposing a relation into sub-relations, one of the sub-relations contains the key of a corresponding functional dependency and the other contains the key and other attributes.

Thus, we have shown the decomposition is lossless for each inductive step and for the base cases. Hence the algorithm produces lossless decompositions. □