Database Design with the Relational Normalization Theory

Important Definitions. functional dependency (FD); satisfaction of FD by a relation; unsatisfaction of FD by a relation; properties of FDs; entailment of a FD by a set of FDs; equivalence between two sets of FDs; normal forms (BCNF, 3NF, 4NF); decompositions (lossless and lossy, dependency preserving); join dependency JD); multivalued dependency (MVD); properties of MVDs; projection of FDs

Important Theoresm and Axioms. Amstrong axioms (trivial/reflexive, augmentation, transitivety) plus their derivations (union, decomposition); soundness and completeness of Amstrong axioms; Use of Amstrong axioms in checking for entailment; in deriving new FDs; Similar axioms for MVDs;

Important Algorithms. attribute closure computation; BCNF, 4NF decomposition; (we skip the algorithm for 3NF);

Questions

1. Why normalization? (Address update-, insertion-, and delete- anormaly.)
2. What is an FD? (an expression of the form $\bar{X} \rightarrow \bar{Y}$)
3. When does a relation satisfy a FD $\bar{X} \rightarrow \bar{Y}$?
4. What are the Amstrong axioms? What properties do they have?
5. When does a set of FDs $\mathcal{F}$ entail a FD f? How can we check whether or not $\mathcal{F}$ entails f?
6. When are two set of FDs $\mathcal{F}$ and $\mathcal{G}$ equivalent?
7. What is the attribute closre $\bar{X}^+_{\mathcal{F}}$? How to compute $\bar{X}^+_{\mathcal{F}}$? Use of $\bar{X}^+_{\mathcal{F}}$ for checking of entailment and equivalence?
8. What is the projection of FDs onto a set of attributes? How to compute it?
9. What is BCNF, 3NF, or 4NF? How can we check whether a relation is in BCNF, 3NF, or 4NF?
10. What is lossless and lossy decomposition? Why should a decomposition be lossless?
11. How do we decompose a relation into BCNF, 3NF, or 4NF? (We will skip 3NF decomposition)

Functional Dependencies. A functional dependency is of the form

$$\bar{X} \rightarrow \bar{Y}$$

(Remember that $\bar{X}$ is the notation that we use to denote a *set of attributes*.) It states that tuples agree on the values of the attributes $\bar{X}$ must also agree on the value of $\bar{Y}$.

Keys. A set of attributes $\bar{X}$ is a *key* of a relation R if

- it functionally determines all other attributes, and
- none of its proper subsets functionally determines all other attributes (i.e., it is *minimal*)

A set of attributes that contains a key is a *superkey*.

Rules to derive new functional dependencies

- Reflexivity: If $\bar{Y} \subseteq \bar{X}$ then $\bar{X} \rightarrow \bar{Y}$.
- Augmentation: If $\bar{X} \rightarrow \bar{Y}$ then $\overline{XZ} \rightarrow \overline{YZ}$.
- Transitivity: If $\bar{X} \rightarrow \bar{Y}$ and $\bar{Y} \rightarrow \bar{Z}$ then $\bar{X} \rightarrow \bar{Z}$.

Closure of Attributes with respect to a set of FDs.

$$\bar{X}^+_{\mathcal{F}} = \{\bar{X} \to \bar{Y} \mid \bar{X} \to \bar{Y} \text{ is entailed by } \mathcal{F}\}.$$

Computing the closure:

Input: $\mathcal{F}$ and $\bar{X}$
Output: $\bar{X}^+_{\mathcal{F}}$
 Init: $closure = \bar{X}$
 while there exists some FD $\bar{X} \to \bar{Y}$ in $\mathcal{F}$ such that
 $\bar{X} \subset closure$ and $\bar{Y} \setminus closure \neq \emptyset$ do
 $closure = closre \cup \bar{Y}$
 endwhile
 return $closure$

Non-trivial Functional Dependency. $\bar{X} \to \bar{Y}$ in nontrivial if at least on of the attribute in $\bar{Y}$ is not among the attributes in $\bar{X}$; it is completely nontrivial if none of the attributes in $\bar{X}$ is an attribute in $\bar{X}$.

Projection of FDs onto a set of attributes. Given $\mathcal{F}$ ann a set of attributes $\bar{X}$, the projection of $\mathcal{F}$ onto $\bar{X}$, denoted by $\pi_{\bar{X}}(\mathcal{F})$ is defined by

$$\pi_{\bar{X}}(\mathcal{F}) = \{\bar{Y} \to \bar{Z} \mid \bar{Y} \cup \bar{Z} \subseteq \bar{X}, \bar{Y} \to \bar{Z} \in \mathcal{F}^+\}$$

i.e, $\pi_{\bar{X}}(\mathcal{F})$ is the set of FDs entailed by $\mathcal{F}$ whose left hand side and right hand side are attributes belonging to $\bar{X}$.

BCNF. $\mathbf{R} = (\bar{R}, \mathcal{F})$ is in BCNF if for every nontrivial functional dependency $\bar{X} \to \bar{Y}$ in $\mathcal{F}$, $\bar{X}$ is a superkey for R.

Decomposing into BCNF:
Input: $\mathbf{R} = (\bar{R}, \mathcal{F})$
Output: a lossless decomposition of $\mathbf{R}$
 Init: $decomposition = \{\mathbf{R}\}$
 while there exists some schema $\mathbf{S} = (\bar{S}, \mathcal{F}')$ in $decomposition$ which is not in BCNF do
 find $\bar{X} \to \bar{Y}$ in $\mathcal{F}'$ which violates the BCNF condition
 replace $\mathbf{S}$ with $\mathbf{S}_1 = (\bar{S}_1, \mathcal{F}'_1)$ and $\mathbf{S}_2 = (\bar{S}_2, \mathcal{F}'_2)$ where
 $\bar{S}_1 = \bar{X} \cup \bar{Y}$, $\mathcal{F}'_1 = \pi_{\overline{XY}}(\mathcal{F})$
 $\bar{S}_2 = \bar{X} \cup (\bar{R} - \bar{Y})$, $\mathcal{F}'_2 = \pi_{\bar{X} \cup (\bar{R} - \bar{Y})}(\mathcal{F})$
 endwhile
return $decomposition$

Third Normal Forms. (or 3NF) $\mathbf{R} = (\bar{R}, \mathcal{F})$ is in 3NF if for every nontrivial functional dependency functional dependency $\bar{X} \to \bar{Y}$ in $\mathcal{F}$, $\bar{X}$ is a superkey or every $B \in \bar{Y} - \bar{X}$ is part of some key.

Decomposing into 3NF: Algorithm in the book. We skip this algorithm.

Projection of a set of FD onto a set of attributes. For a relation $\mathbf{R} = (\bar{R}, \mathcal{F})$ and a set of attributes $\bar{S} \subseteq \bar{R}$, the projection of $\mathcal{F}$ on $\bar{S}$ is a set of FDs, denoted by $\pi_{\bar{S}}(\mathcal{F})$, and is defined as follows.

$$\pi_{\bar{S}}(\mathcal{F}) = \{\bar{X} \to \bar{Y} \mid \bar{X} \to \bar{Y} \in \mathcal{F}^+ \text{ and } \bar{X} \cup \bar{Y} \subseteq \bar{S}\}.$$

This set can be computed by

- Compute $\bar{X}^+$ for each $\bar{X} \subseteq \bar{S}$
- Determine the FDs deriveable from $\bar{X}^+$

Decomposition. A decomposition of a relation schema $\mathbf{R} = (\bar{R}, \mathcal{F})$ is a collection of schemas $\mathbf{R_1} = (\bar{R}_1, \mathcal{F}_\infty)$,..., $\mathbf{R_n} = (\bar{R}_n, \mathcal{F}_\backslash)$ such that

1. $\mathbf{R_i} \neq \mathbf{R_j}$ for $i \neq j$
2. $\bar{R} = \bigcup_{i=1}^{n} \bar{R}_i$
3. $\mathcal{F}$ entails $\mathcal{F}_i$

This leads to a decomposition of a relation $\mathbf{r}$ (of the schema $\mathbf{R}$ into a set of relations

$$\mathbf{r_1} = \pi_{\bar{R}_1}(\mathbf{r}), \ldots, \mathbf{r_n} = \pi_{\bar{R}_n}(\mathbf{r})$$

Lossless Decomposition A decompostion of $\mathbf{R} = (\bar{R}, \mathcal{F})$ into $\mathbf{R_1} = (\bar{R}_1, \mathcal{F}_\infty)$,..., $\mathbf{R_n} = (\bar{R}_n, \mathcal{F}_\backslash)$ is *lossless* if for every valid instance $\mathbf{r}$ of $\mathbf{R}$,

$$\mathbf{r} = \mathbf{r_1} \bowtie \mathbf{r_2} \bowtie \ldots \bowtie \mathbf{r_n}$$

A binary decomposition of $\mathbf{R} = (\bar{R}, \mathcal{F})$ into $\mathbf{R_1} = (\bar{R}_1, \mathcal{F}_\infty)$ and $\mathbf{R_2} = (\bar{R}_2, \mathcal{F}_\in)$ is lossless if

- $\bar{R}_1 \cap \bar{R}_2 \rightarrow \bar{R}_1 \in \mathcal{F}^+$ or
- $\bar{R}_1 \cap \bar{R}_2 \rightarrow \bar{R}_2 \in \mathcal{F}^+$

Multivalued Dependencies. A multivalued dependency (MVD) is of the form

$$\bar{X} \twoheadrightarrow \bar{Y}$$

It states that for each pair of tuples t and u of an instance $\mathbf{r}$ that agree on the values of the attributes $\bar{X}$ we can find in $\mathbf{r}$ some tuple r that agrees

1. with both t and u on $\bar{X}$,
2. with t on $\bar{Y}$, and
3. with u on the attributes that are not among the $\bar{X}$'s or $\bar{Y}$'s.

Rules to derive new multivalued dependencies

- Trivial: If $\bar{X} \twoheadrightarrow \bar{Y}$ then $\bar{X} \twoheadrightarrow \bar{Z}$ where $\bar{Z} \subseteq \bar{X} \cup \bar{Y}$.
- Transitivity: If $\bar{X} \twoheadrightarrow \bar{Y}$ and $\bar{Y} \twoheadrightarrow \bar{Z}$ then $\bar{X} \twoheadrightarrow \bar{Z}$.
- Completementation: If $\bar{X} \twoheadrightarrow \bar{Y}$ then $\bar{X} \twoheadrightarrow \bar{X}$ where $\bar{Z} = \bar{R} \setminus (\bar{X} \cup \bar{Y})$.
- FD implication: If $\bar{X} \rightarrow \bar{Y}$ then $\bar{X} \twoheadrightarrow \bar{Y}$.

Non-trivial multivalued Dependency. $\bar{X} \twoheadrightarrow \bar{Y}$ in nontrivial if none of the attributes in $\bar{Y}$ is among the attributes in $\bar{X}$ and $\bar{X} \cup \bar{Y}$ is not the set of all attributes of $\mathbf{R}$.

Fourth Normal Form. $\mathbf{R} = (\bar{R}, \mathcal{D})$ – where $\mathcal{D}$ is a set of multivalued dependencies – is in 4NF if for every nontrivial multivalued dependency $\bar{X} \twoheadrightarrow \bar{Y}$ in $\mathcal{D}$ $\bar{X}$ is a superkey.

Decomposing into 4NF: The algorithm for decomposing a relation into a set of 4NF relations is similar to the BCNF decomposition. We also take a nontrivial MVD $\bar{X} \twoheadrightarrow \bar{Y}$ that violates the 4NF condition and split the relation into two, one with the set of attributes $\bar{X} \cup \bar{Y}$ and the other with the set of attributes $\bar{X} \cup (\bar{R} - \bar{Y})$. The main difficulty in this process is to compute the projection of MVDs onto a set of attributes.

Remark 1. The notation in this note follows the textbook. $\bar{X}$ stands for a set of attributes. Elsewhere, you will see $A_1 \dots A_n$ instead of $\bar{X}$. The MVD part is a special case of what is called *join dependency* in the book. Each join dependency $\bar{R} = \bar{V} \bowtie \bar{W}$ can be represented as $\bar{X} \twoheadrightarrow \bar{Y}$ where $\bar{X} = \bar{V} \cap \bar{W}$ and $\bar{X} \cup \bar{Y} = \bar{V}$ or $\bar{X} \cup \bar{Y} = \bar{W}$.

Remark 2. The definitions of BCNF, 3NF, 4NF omit the case where $\bar{Y} \subseteq \bar{X}$. This is the case of trivial dependency.

Remark 3. To determine whether or not a relation schema $\mathbf{R} = (\bar{R}, \mathcal{F})$ is in BCNF (or 3NF, 4NF), we need to determine if *any* FD (or MVD) in $\mathcal{F}$ satisfies the BCNF (or 3NF, 4NF) condition, respectively. If one does not satisfy the BCNF (or 3NF, 4NF) condition, we can stop and conclude that the schema is not in BCNF (or 3NF, 4NF). Only after all FDs (or MVD) satisfy the BCNF (or 3NF, 4NF) condition, we can say that the schema is in BCNF (or 3NF, 4NF).

Remark 4. To determine whether or not a relation schema $\mathbf{R} = (\bar{R}, \mathcal{F})$ is in BCNF, we do not really need to know the key of the relation. We just need to compute the closure of the left hand side of all the FDs in $\mathcal{F}$.

Remark 5. To determine whether or not a relation schema $\mathbf{R} = (\bar{R}, \mathcal{F})$ is in 3NF or 4NF, we need to know the key of the relation. Computing the key using brute-force search is not a good idea. We often can find the *must-be-presented* elements of the key by looking at the right hand side of the FDs in $\mathcal{F}$.