

Database Design with the Relational Normalization Theory

Reasons for studying normalization theory. Database design helps us model the real world in that it gives us a way to represent precisely the data/information pertinent to a company or an enterprise; we then can make use of the data we have stored. The design methodology that we learned in the previous chapter is adequate in the sense that it allows us to complete the task.

At the beginning of the database area, having the database and the tools for updating, inserting, and deleting data is good enough. It turns out that something more than that is required for databases. The most important issue in database is the maintainance of consistency of the database, especially when updates are made to the database. A good design can contribute — in several ways — to this process. It provides us the tools/theories for achieving schema that do not contain redundancy. Normalizations do help us remove redundancy from database (which is a source for data inconsistency.)

The note will be organized in the following way: first, I list the important definitions; second, the theorems and axioms; third, the algorithms; fourth, a list of questions — which, I believe, that if you can answer them — you are done with the chapter; last, the examples and precise definitions.

Important Definitions. functional dependency (FD); keys; superkeys; 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 Theorem and Axioms. Armstrong axioms (trivial/reflexive, augmentation, transitivity) plus their derivations (union, decomposition); soundness and completeness of Armstrong axioms; Use of Armstrong axioms in checking for entailment; in deriving new FDs; Similar axioms for MVDs;

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

Questions

1. Why normalization? (Address update-, insertion-, and delete- anomaly.)
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 Armstrong 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 closure $\bar{X}_{\mathcal{F}}^+$? 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.

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

Computing the closure:

Input: \mathcal{F} and \bar{X}

Output: $\bar{X}_{\mathcal{F}}^+$

```

Init:      closure =  $\bar{X}$ 
while     there exists some FD  $\bar{X} \rightarrow \bar{Y}$  in  $\mathcal{F}$  such that
            $\bar{X} \subset \text{closure}$  and  $\bar{Y} \setminus \text{closure} \neq \emptyset$  do
           closure = closure  $\cup$   $\bar{Y}$ 
endwhile
return  closure
    
```

Non-trivial Functional Dependency. $\bar{X} \rightarrow \bar{Y}$ is nontrivial if at least one of the attributes 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} and a set of attributes \bar{X} , the projection of \mathcal{F} onto \bar{X} , denoted by $\pi_{\bar{X}}(\mathcal{F})$ is defined by

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

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 $\bar{X} \rightarrow \bar{Y}$ 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 = {R}
while     there exists some schema S = ( $\bar{S}, \mathcal{F}$ ) in decomposition which is not in BCNF
do
    find  $\bar{X} \rightarrow \bar{Y}$  in  $\mathcal{F}$  which violates the BCNF condition
    replace S with S1 = ( $\bar{S}_1, \mathcal{F}'_1$ ) and S2 = ( $\bar{S}_2, \mathcal{F}'_2$ ) where
         $\bar{S}_1 = \bar{X} \cup \bar{Y}$ ,  $\mathcal{F}'_1 = \pi_{\bar{X}\bar{Y}}(\mathcal{F})$ 
         $\bar{S}_2 = \bar{X} \cup (\bar{R} - \bar{Y})$ ,  $\mathcal{F}'_2 = \pi_{\bar{X} \cup (\bar{R} - \bar{Y})}(\mathcal{F})$ 
    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 $\bar{X} \rightarrow \bar{Y}$ 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.

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

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 $\mathbf{R}_1 = (\bar{R}_1, \mathcal{F}_1), \dots, \mathbf{R}_n = (\bar{R}_n, \mathcal{F}_n)$ 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

$$\mathbf{r}_1 = \pi_{\bar{R}_1}(\mathbf{r}), \dots, \mathbf{r}_n = \pi_{\bar{R}_n}(\mathbf{r})$$

Lossless Decomposition A decomposition of $\mathbf{R} = (\bar{R}, \mathcal{F})$ into $\mathbf{R}_1 = (\bar{R}_1, \mathcal{F}_1), \dots, \mathbf{R}_n = (\bar{R}_n, \mathcal{F}_n)$ is *lossless* if for every valid instance \mathbf{r} of \mathbf{R} ,

$$\mathbf{r} = \mathbf{r}_1 \bowtie \mathbf{r}_2 \bowtie \dots \bowtie \mathbf{r}_n$$

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

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

A decomposition which is not lossless is *lossy*.

A decomposition of $\mathbf{R} = (\bar{R}, \mathcal{F})$ into $\mathbf{R}_1 = (\bar{R}_1, \mathcal{F}_1), \dots, \mathbf{R}_n = (\bar{R}_n, \mathcal{F}_n)$ is *dependency preserving* if \mathcal{F} and $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \dots \cup \mathcal{F}_n$ are equivalent.

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

Rules to derive new functional dependencies from multivalued dependencies:

- if $\bar{W} \subseteq \bar{Y}$ and $\bar{Y} \cap \bar{Z} = \emptyset$ then $\bar{X} \twoheadrightarrow \bar{Y}$ and $\bar{Z} \rightarrow \bar{W}$ entail $\bar{X} \rightarrow \bar{W}$.

Non-trivial multivalued Dependency. $\bar{X} \twoheadrightarrow \bar{Y}$ is 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. We can use the following rule to do this:

Let $\bar{V} = \bar{X} \cup \bar{Y}$ and $\bar{W} = (\bar{R} - \bar{Y}) \cup \bar{X}$. The projection of $\bar{X} \twoheadrightarrow \bar{Y}$ onto \bar{S} is undefined if $\bar{V} \cap \bar{W} \not\subseteq \bar{S}$; when it is defined, it is $\bar{P} \twoheadrightarrow \bar{Q}$ where $\bar{P} = (\bar{S} \cap \bar{V}) \cap (\bar{S} \cap \bar{W})$ and $\bar{Q} = (\bar{S} \cap \bar{Y})$.

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} .

Example of Computing Keys We wish to compute the keys of $\mathbf{R} = (ABCD, \mathcal{F})$ with $\mathcal{F} = \{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$.

We observe that the right handside of every FD in \mathcal{F} does not contain B . Thus, any key need to contain B . Because $B^+ = \{B\}$, B is not a key. So, any key will need to contain at least two attributes.

AB, BC, BD are three possible 2-attributes key of \mathbf{R} . We can check that

$$AB^+ = \{A, B, C, D\}$$

$$BC^+ = \{B, C, D, A\}$$

and

$$BD^+ = \{B, D, A, C\}$$

Thus, \mathbf{R} has three keys AB, BC, BD .

Note that we stop here since a key needs to be minimal and every set of attributes with more than two attributes will contain some key.

Projection of a set of FD onto a set of attributes Let $\mathcal{F} = \{AB \rightarrow D, AC \rightarrow E, BC \rightarrow D, D \rightarrow A, E \rightarrow B\}$. We would like to compute $\pi_{ABC}(\mathcal{F})$. This can be done in a systematic way as follows. First, we compute

$$\begin{aligned} A^+ &= \{A\} \\ B^+ &= \{B\} \\ C^+ &= \{C\} \\ AB^+ &= \{A, B, D\} \\ AC^+ &= \{A, C, E, B, D\} \\ BC^+ &= \{B, C, D, A, E\} \end{aligned}$$

This means that

$$\pi_{ABC}(\mathcal{F}) = \{AC \rightarrow B, BC \rightarrow A\}.$$

Note that we only list non-trivial FDs and FDs with only one attribute on the right hand side.

Decomposition into BCNF Consider $\mathbf{R} = (ABCD, \mathcal{F})$ with $\mathcal{F} = \{AB \rightarrow C, C \rightarrow D, D \rightarrow A\}$. We can see that the non-trivial FD $D \rightarrow A$ violates the BCNF condition (D is not a superkey since $D^+ = \{D, A\}$). Thus, \mathbf{R} is not in BCNF.

We use $D \rightarrow A$ to decompose \mathbf{R} . This results in $\mathbf{R}_1 = (AD, \mathcal{F}_1)$ and $\mathbf{R}_2 = (BCD, \mathcal{F}_2)$ with $\mathcal{F}_1 = \{D \rightarrow A\}$ and $\mathcal{F}_2 = \{C \rightarrow D\}$.

The first schema is in BCNF but the second one is not due to $C \rightarrow D$. Decompose the second one with respect to this FD, we have $\mathbf{R}_{21} = (CD, \{C \rightarrow D\})$ and $\mathbf{R}_{22} = (CB, \emptyset)$.