Course 4. Normal forms

Redundancy is at the root of several problems associated with relational schemas: ***redundant storage, insert/delete/update anomalies.***

Integrity constraints, in particular *functional dependencies*, can be used to identify schemas with such problems and to suggest refinements. The main refinement technique is *decomposition*, but it should be used judiciously. Each time we want to decompose a relation we should think if:

- Is there reason to decompose a relation?
- Are there any problems the decomposition cause?

Normal forms

If a relation is in a certain *normal form*, it is known that certain kinds of problems are avoided /minimized. This can be used to help us decide whether decomposing the relation will help.

The normal forms based on FDs are *first normal form (1NF), second normal form (2NF), third normal form (3NF)* and *Boyce-Codd normal form (BCNF).*

$$\{BCNF \subseteq 3NF,\ 3NF \subseteq 2NF,\ 2NF \subseteq 1NF\}$$

Functional dependencies have a crucial role in detecting redundancy.

Having a relation R with 3 attributes, ABC:

- If no functional dependency holds -> there is no redundancy;
- Given $A \rightarrow B$: several tuples could have the same A value, and if so, they'll all have the same B value!

Definition. A relation is in *First Normal Form* (1NF) if every fields contains only atomic values, that is, not lists or sets (this requirement is implicit in our definition of the relational model).

A partial functional dependency occurs when the value in a non-prime attribute of a table is dependent on the value of some part of the table's primary key (but not all of it).

Definition. A relation is in *Second Normal Form* (2NF) if there are no partial dependencies.

The *Second Normal Form* is mainly of historical interest

Definition. Relation R with FDs *F* is in *Boyce-Codd Normal Form* BCNF if, for all $\alpha \rightarrow A$ in F^+

- $A \in \alpha$ (*trivial* FD), or
- α contains a key for R (α is a superkey).

In other words, R is in BCNF if the only non-trivial functional dependencies that hold over R are key constraints. No dependency in R that can be predicted using FDs alone.

Definition. Relation R with FDs *F* is in 3NF if, for all $\alpha \rightarrow A$ in F^+

- $A \in \alpha$ (*trivial* FD), or
- α is a superkey for R, or
- A is part of some key for R (i.e. A is a prime attribute).

Minimality of a key is crucial in third condition above! If R is in BCNF, obviously in 3NF. If R is in 3NF, some redundancy is possible. It is a compromise, used when BCNF not achievable (e.g., no ``good'' decomposition, or performance considerations).

Lossless-join, dependency-preserving decomposition of R into a collection of 3NF relations is always possible.

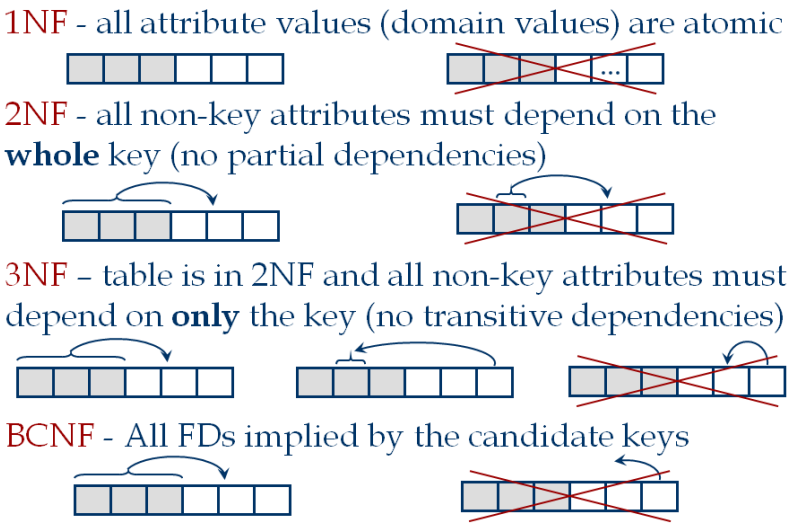

Figure 4.1. Summary of normal forms based on functional dependencies

Examples of normal forms violations:

- 2NF - all non-key attributes must depend on the **whole** key

  Exam (***Student***, ***Course***, Teacher, Grade)

- 3NF - all non-key attributes must depend on **only** the key

  Dissertation(***Student***, Title, Teacher, Department)

- BCNF - all functional dependencies are implied by the candidate keys

Schedule (***Day, Route, Bus,*** Driver)

Consider relation R with FDs F. If $\alpha \rightarrow A$ violates BCNF, decompose R into R - A and αA. Repeated application of this idea will give us a collection of relations that are in BCNF, lossless join decomposition, and guaranteed to terminate.

Example. ***R***(**C**,S,J,D,P,Q,V), key C, {JP$\rightarrow$ C, SD$\rightarrow$ P, J$\rightarrow$ S}

To deal with SD$\rightarrow$ P, decompose into (**S,D**,P), (**C**,S,J,D,Q,V).

To deal with J$\rightarrow$ S, decompose (**C**,S,J,D,Q,V) into (**J**,S) and (**C**,J,D,Q,V)

In general, several dependencies may cause violation of BCNF. The order in which we ``deal with'' them could lead to very different sets of relations!

In general, there may not be a dependency preserving decomposition into BCNF.

- e.g., ***R***(C,S,Z), {CS$\rightarrow$ Z, Z$\rightarrow$ C}
- can't decompose while preserving 1st FD; not in BCNF.

Similarly, decomposition of ***R***(C, S, J, P, D, Q, V) into (S, D, P), (J, S) and (C, J, D, Q, V) is not dependency preserving (with respect to the FDs {JP$\rightarrow$ C, SD$\rightarrow$ P, J$\rightarrow$ S}). However, it is a lossless join decomposition.

In this case, adding JPC to the collection of relations gives us a dependency preserving decomposition.

- JPC tuples stored only for checking FD! (*Redundancy!*)

Obviously, the algorithm for lossless join decomposition into BCNF can be used to obtain a lossless join decomposition into 3NF (typically, can stop earlier).

One idea to ensure dependency preservation could be: if X$\rightarrow$Y is not preserved, add relation XY. The problem is that XY may violate 3NF! (e.g., consider the addition of CJP to `preserve' JP$\rightarrow$C. What if we also have J$\rightarrow$C ?). Instead of the given set of functional dependencies F, we should use a *minimal cover for F*.

Minimal cover

Definition. An attribute $A \in \alpha$ is redundant in the FD $\alpha \rightarrow B$ if

$$(F - \{\alpha \rightarrow B\}) \cup \{\alpha - A \rightarrow B\} \quad \equiv \quad F$$

To check if $A \in \alpha$ is redundant in $\alpha \rightarrow B$, it is enough to compute $(\alpha - A)^+$ w.r.t. F. Then $A \in \alpha$ is redundant in $\alpha \rightarrow B$ if $B \in (\alpha - A)^+$

Exercise: What are the redundant attributes in $AB \rightarrow C$ w.r.t. $\{AB \rightarrow C, A \rightarrow B, B \rightarrow A\}$?

Definition. A functional dependency $f \in F$ is redundant if $F - \{f\}$ is equivalent to F.

To check if $\alpha \rightarrow A$ is redundant in F, we should compute α^+ w.r.t. $F-\{\alpha \rightarrow A\}$. Then $\alpha \rightarrow A$ is redundant in F if $A \in \alpha^+$

Exercise: What are the redundant FDs in $\{A \rightarrow C, A \rightarrow B, B \rightarrow A, B \rightarrow C, C \rightarrow A\}$?

Definition. A minimal cover for a set F of functional dependencies is a set G of functional dependencies such that:

1. Every FD in G is of the form $\alpha \rightarrow A$
2. For each FD $\alpha \rightarrow A$ in G, α has no redundant attributes
3. There are no redundant FDs in G
4. G and F are equivalent

Observation. Each set of FDs has at least one minimal cover.

Algorithm to compute minimal cover of F:

1. Use decomposition to obtain FDs with 1 attr. on R.H.S.
2. Remove redundant attributes
3. Remove redundant FDs

Exercise. Consider $F = \{ABCD \rightarrow E, E \rightarrow D, A \rightarrow B, AC \rightarrow D\}$

Attributes BD in $ABCD \rightarrow E$ are redundant => $F = \{AC \rightarrow E, E \rightarrow D, A \rightarrow B, AC \rightarrow D\}$

$AC \rightarrow D$ is redundant => $F = \{AC \rightarrow E, E \rightarrow D, A \rightarrow B\}$, which is a minimal cover

Observation. Minimal covers may not be unique due to choice of redundant functional dependencies or redundant attributes.

Algorithm for Decomposition into 3NF (Synthesis Approach)

Input: Schema R with *F* which is a minimal cover

Output: A dependency-preserving, lossless-join 3NF decomposition of R

Initialize D = ∅

Apply *union rule* to combine FDs in F with same L.H.S. into a single FD

    For each FD $\alpha \rightarrow \beta$ in F do

        Insert the relation schema $\alpha\beta$ into D

    Insert δ into D, where δ is some key of R

Remove redundant relation schema from D as follows:

    delete R_i from D if $R_i \subseteq R_j$, where $R_j \in$ D

return D

Exercise. Consider again R(A,B,C,D,E) with functional dependencies:

$$F = \{ABCD \rightarrow E, E \rightarrow D, A \rightarrow B, AC \rightarrow D\}$$

- Recall that a min. cover of F is $\{AC \rightarrow E, E \rightarrow D, A \rightarrow B\}$ (v. previous exercise)
- The only key is AC
- R is not in 3NF because $A \rightarrow B$ violates 3NF
- 3NF decomposition of R:
  - Create a schema for each FD: R_1(A, C, E), R_2(E,D), and R_3(A,B)
  - Create a schema for a key of R: R_4(A, C)
  - Remove redundant schema: R_4 is redundant because $R_4 \subseteq R_1$
    $\Rightarrow$ 3NF decomposition is $\{R_1(A,C,E), R_2(E,D), R_3(A,B)\}$

Observation. 3NF decomposition may not be unique:

- Choice of *minimal cover* or
- Choice of *redundant relation schema to be removed*

Remarks on Decomposition

- Decomposition is a last resort to solve problems of redundancy & anomalies
- Too much decomposition can be harmful! *Example*: R = (Teacher, Department, Phone, Office) with the set of functional dependencies F = {Teacher → Dept Phone Office}. { R_1

= (Teacher,Dept), R_2 = (Teacher, Phone), R_3 = (Teacher, Office)} is a good decomposition for R, but it is not necessary.

- Might need to consider de-normalization for performance reasons

Multi-valued dependencies

Let's consider a relation with attributes course, teacher, book (CTB). (a teacher T can teach course C and the recommended book for the course is B). This relation is in BCNF

course	teacher	book
alg101	Green	Alg Basics
alg101	Green	Alg Theory
alg101	Brown	Alg Basics
alg101	Brown	Alg Theory
logic203	Green	Logic B.
logic203	Green	Logic F.
logic203	Green	Logic intro.

Figure 4.2. Relation sample with multi-valued dependencies

Definition. Let α, β be subsets of the attributes of R. The multi-valued dependency $\alpha \rightarrow\rightarrow \beta$ is said to hold over R if, in every legal instance r or R, each α value is associated with a set of β values and this set is independent of the values in other attributes.

Formally: If the MVD $\alpha \rightarrow\rightarrow \beta$ holds over R and γ=R-$\alpha\beta$, the following must be true for every legal instance of r of R:

$$t_1, t_2 \in r \text{ and } \pi_\alpha(t_1) = \pi_\alpha(t_2) \Rightarrow$$

$$\exists t_3 \in r \text{ such that } \pi_{\alpha\beta}(t_1) = \pi_{\alpha\beta}(t_3) \text{ and } \pi_\gamma(t_2) = \pi_\gamma(t_3)$$

As a consequence if we take t_2 and t_1 we can deduce that there is also $t_4 \in r$ such that $\pi_{\alpha\beta}(t_2) = \pi_{\alpha\beta}(t_4)$ and $\pi_\gamma(t_1) = \pi_\gamma(t_4)$.

Additional rules:

MVD Complementation: $X \rightarrow\rightarrow Y \Rightarrow X \rightarrow\rightarrow R - XY$

MVD Augmentation: $X \rightarrow\rightarrow Y, Z \subseteq W \Rightarrow WX \rightarrow\rightarrow YZ$

MVD Transitivity: $X \rightarrow\rightarrow Y, Y \rightarrow\rightarrow Z \Rightarrow X \rightarrow\rightarrow Z - Y$

Replication: $X \rightarrow Y \Rightarrow X \rightarrow\rightarrow Y$

Coalescence: $X \rightarrow\rightarrow Y, W \cap Y = \varnothing, W \rightarrow Z, Z \subseteq Y \Rightarrow X \rightarrow Z$

Definition. Let R be a relation schema. $X,Y \subseteq$ attributes(R), F={FDs $\cup$ MVDs}.

R is said to be in *Fourth Normal Form* (4NF) if $\forall\ X \rightarrow\rightarrow Y \in$ MVDs that holds over R, one of the following statements is true:

- $Y \subseteq X$ or
- XY = R or
- X is a super-key

For our example, the relation can be decomposed in: (Course, Teacher) and (Course, Book).

Join Dependency

Definition. A join dependency (JD) $\otimes\{R_1,\ldots,R_n\}$ is said to hold over a relation R if

$R_1, R_2,\ldots,R_n$ is a lossless-join decomposition of R.

An MVD $X \rightarrow\rightarrow Y$ over a relation R can be expressed as the join dependency $\otimes\{XY, X(R-Y)\}$.

Definition. A relational schema is said to be in *Fifth Normal Form* (*5NF*) if $\forall \otimes\{R_1,\ldots,R_n\}$ that holds over R, one of the following statements is true:

- R_i=R for some *i*, or
- the JD is implied by the set of those FDs over R in which the left side is a key for R