CHAPTER 8

Relational Database Design

This chapter presents the principles of relational database design. Undergraduates frequently find this chapter difficult. It is acceptable to cover only Sections 8.1 and 8.3 for classes that find the material particularly difficult. However, a careful study of data dependencies and normalization is a good way to introduce students to the formal aspects of relational database theory.

There are many ways of stating the definitions of the normal forms. We have chosen a style which we think is the easiest to present and which most clearly conveys the intuition of the normal forms.

Exercises

8.19 Give a lossless-join decomposition into BCNF of schema R of Exercise 8.1.

Answer: From Exercise 8.6, we know that $B \rightarrow D$ is nontrivial and the left hand side is not a superkey. By the algorithm of Figure 8.11 we derive the relations $\{(A, B, C, E), (B, D)\}$. This is in BCNF.

8.20 Give a lossless-join, dependency-preserving decomposition into 3NF of schema R of Practice Exercise 8.1.

Answer: First we note that the dependencies given in Practice Exercise 8.1 form a canonical cover. Generating the schema from the algorithm of Figure 8.12 we get

$$R' = \{(A, B, C), (C, D, E), (B, D), (E, A)\}.$$

Schema (A, B, C) contains a candidate key. Therefore R' is a third normal form dependency-preserving lossless-join decomposition.

Note that the original schema $R = (A, B, C, D, E)$ is already in 3NF. Thus, it was not necessary to apply the algorithm as we have done above. The single original schema is trivially a lossless join, dependency-preserving decomposition.

8.21 Normalize the following schema, with given constraints, to 4NF.

$$
\begin{aligned}
&\textit{books}(\textit{accessionno}, \textit{isbn}, \textit{title}, \textit{author}, \textit{publisher})\\
&\textit{users}(\textit{userid}, \textit{name}, \textit{deptid}, \textit{deptname})\\
&\textit{accessionno} \rightarrow \textit{isbn}\\
&\textit{isbn} \rightarrow \textit{title}\\
&\textit{isbn} \rightarrow \textit{publisher}\\
&\textit{isbn} \twoheadrightarrow \textit{author}\\
&\textit{userid} \rightarrow \textit{name}\\
&\textit{userid} \rightarrow \textit{deptid}\\
&\textit{deptid} \rightarrow \textit{deptname}
\end{aligned}
$$

Answer: In *books*, we see that

$$\textit{isbn} \twoheadrightarrow \textit{title}, \textit{publisher}, \textit{author}$$

and yet, *isbn* is not a super key. Thus, we break *books* into

$$
\begin{aligned}
&\textit{books\_accnno}(\textit{accessionno}, \textit{isbn})\\
&\textit{books\_details}(\textit{isbn}, \textit{title}, \textit{publisher}, \textit{author})
\end{aligned}
$$

After this, we still have

$$\textit{isbn} \twoheadrightarrow \textit{author}$$

but neither is *isbn* a primary key of *book_details*, nor are the attributes of *book_details* equal to $\{\textit{isbn}\} \cup \{\textit{author}\}$. Therefore we decompose *book_details* again into

$$
\begin{aligned}
&\textit{books\_details1}(\textit{isbn}, \textit{title}, \textit{publisher})\\
&\textit{books\_authors}(\textit{isbn}, \textit{author})
\end{aligned}
$$

Similarly, in *users*,

$$\textit{deptid} \rightarrow \textit{deptname}$$

and yet, *deptid* is not a super key. Hence, we break *users* to

$$
\begin{aligned}
&\textit{users}(\textit{userid}, \textit{name}, \textit{deptid})\\
&\textit{departments}(\textit{deptid}, \textit{deptname})
\end{aligned}
$$

We verify that there are no further functional or multivalued dependencies that cause violation of 4NF, so the final set of relations are:

books_accnno(*accessionno*, *isbn*)
books_details1(*isbn*, *title*, *publisher*)
books_authors(*isbn*, *author*) *users*(*userid*, *name*, *deptid*)
departments(*deptid*, *deptname*)

8.22 Explain what is meant by *repetition of information* and *inability to represent information*. Explain why each of these properties may indicate a bad relational-database design.

Answer:

- Repetition of information is a condition in a relational database where the values of one attribute are determined by the values of another attribute in the same relation, and both values are repeated throughout the relation. This is a bad relational database design because it increases the storage required for the relation and it makes updating the relation more difficult, and can lead to inconsistent data if updates are done to one copy of the value, but not to another.
- Inability to represent information is a condition where a relationship exists among only a proper subset of the attributes in a relation. This is bad relational database design because all the unrelated attributes must be filled with null values otherwise a tuple without the unrelated information cannot be inserted into the relation.
- Inability to represent information can also occur because of loss of information which results from the decomposition of one relation into two relations, which cannot be combined to recreate the original relation. Such a lossy decomposition may happen implicitly, even without explicitly carrying out decomposition, if the initial relational schema itself corresponds to the decomposition.

8.23 Why are certain functional dependencies called *trivial* functional dependencies?

Answer: Certain functional dependencies are called trivial functional dependencies because they are satisfied by all relations.

8.24 Use the definition of functional dependency to argue that each of Armstrong's axioms (reflexivity, augmentation, and transitivity) is sound.

Answer: The definition of functional dependency is: $\alpha \rightarrow \beta$ holds on R if in any legal relation $r(R)$, for all pairs of tuples t_1 and t_2 in r such that $t_1[\alpha] = t_2[\alpha]$, it is also the case that $t_1[\beta] = t_2[\beta]$.

Reflexivity rule: if α is a set of attributes, and $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$.
Assume $\exists\, t_1$ and t_2 such that $t_1[\alpha] = t_2[\alpha]$

$t_1[\beta] = t_2[\beta]$	since $\beta \subseteq \alpha$
$\alpha \rightarrow \beta$	definition of FD

Augmentation rule: if $\alpha \rightarrow \beta$, and γ is a set of attributes, then $\gamma\,\alpha \rightarrow \gamma\,\beta$.
Assume $\exists\, t_1,\ t_2$ such that $t_1[\gamma\,\alpha] = t_2[\gamma\,\alpha]$

$t_1[\gamma] = t_2[\gamma]$	$\gamma \subseteq \gamma\,\alpha$
$t_1[\alpha] = t_2[\alpha]$	$\alpha \subseteq \gamma\,\alpha$
$t_1[\beta] = t_2[\beta]$	definition of $\alpha \rightarrow \beta$
$t_1[\gamma\,\beta] = t_2[\gamma\,\beta]$	$\gamma\,\beta = \gamma \cup \beta$
$\gamma\,\alpha \rightarrow \gamma\,\beta$	definition of FD

Transitivity rule: if $\alpha \rightarrow \beta$ and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$.
Assume $\exists\, t_1,\ t_2$ such that $t_1[\alpha] = t_2[\alpha]$

$t_1[\beta] = t_2[\beta]$	definition of $\alpha \rightarrow \beta$
$t_1[\gamma] = t_2[\gamma]$	definition of $\beta \rightarrow \gamma$
$\alpha \rightarrow \gamma$	definition of FD

8.25 Consider the following proposed rule for functional dependencies: If $\alpha \rightarrow \beta$ and $\gamma \rightarrow \beta$, then $\alpha \rightarrow \gamma$. Prove that this rule is *not* sound by showing a relation r that satisfies $\alpha \rightarrow \beta$ and $\gamma \rightarrow \beta$, but does not satisfy $\alpha \rightarrow \gamma$.

Answer: Consider the following rule: if $A \rightarrow B$ and $C \rightarrow B$, then $A \rightarrow C$. That is, $\alpha = A$, $\beta = B$, $\gamma = C$. The following relation r is a counterexample to the rule.

r:

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2

Note: $A \rightarrow B$ and $C \rightarrow B$, (since no 2 tuples have the same C value, $C \rightarrow B$ is true trivially). However, it is not the case that $A \rightarrow C$ since the same A value is in two tuples, but the C value in those tuples disagree.

8.26 Use Armstrong's axioms to prove the soundness of the decomposition rule.

Answer: The decomposition rule, and its derivation from Armstrong's axioms are given below:

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

$\alpha \rightarrow \beta\gamma$	given
$\beta\gamma \rightarrow \beta$	reflexivity rule
$\alpha \rightarrow \beta$	transitivity rule
$\beta\gamma \rightarrow \gamma$	reflexive rule
$\alpha \rightarrow \gamma$	transitive rule

8.27 Using the functional dependencies of Practice Exercise 8.6, compute B^+.

Answer: Computing B^+ by the algorithm in Figure 8.8 we start with $result = \{B\}$. Considering FDs of the form $\beta \rightarrow \gamma$ in F, we find that the only dependencies satisfying $\beta \subseteq result$ are $B \rightarrow B$ and $B \rightarrow D$. Therefore $result = \{B, D\}$. No more dependencies in F apply now. Therefore $B^+ = \{B, D\}$

8.28 Show that the following decomposition of the schema R of Practice Exercise 8.1 is not a lossless-join decomposition:

(A, B, C)
(C, D, E).

Hint: Give an example of a relation r on schema R such that

$$\Pi_{A,B,C}(r) \bowtie \Pi_{C,D,E}(r) \neq r$$

Answer: Following the hint, use the following example of r:

A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_1	d_2	e_2

With $R_1 = (A, B, C)$, $R_2 = (C, D, E)$:

a. $\Pi_{R_1}(r)$ would be:

A	B	C
a_1	b_1	c_1
a_2	b_2	c_1

b. $\Pi_{R_2}(r)$ would be:

C	D	E
c_1	d_1	e_1
c_1	d_2	e_2

c. $\Pi_{R_1}(r) \bowtie \Pi_{R_2}(r)$ would be:

A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_1	b_1	c_1	d_2	e_2
a_2	b_2	c_1	d_1	e_1
a_2	b_2	c_1	d_2	e_2

Clearly, $\Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) \neq r$. Therefore, this is a lossy join.

8.29 Consider the following set F of functional dependencies on the relation schema $r(A, B, C, D, E, F)$:

$$A \rightarrow BCD$$
$$BC \rightarrow DE$$
$$B \rightarrow D$$
$$D \rightarrow A$$

a. Compute B^+.

b. Prove (using Armstrong's axioms) that AF is a superkey.

c. Compute a canonical cover for the above set of functional dependencies F; give each step of your derivation with an explanation.

d. Give a 3NF decomposition of r based on the canonical cover.

e. Give a BCNF decomposition of r using the original set of functional dependencies.

f. Can you get the same BCNF decomposition of r as above, using the canonical cover?

Answer:

a. $B \rightarrow BD$ (third dependency)
$BD \rightarrow ABD$ (fourth dependency)
$ABD \rightarrow ABCD$ (first dependency)
$ABCD \rightarrow ABCDE$ *(second dependency)*

Thus, $B^+ = ABCDE$

b. Prove (using Armstrong's axioms) that AF is a superkey.

$A \rightarrow BCD$ (Given)
$A \rightarrow ABCD$ (Augmentation with A)
$BC \rightarrow DE$ (Given)
$ABCD \rightarrow ABCDE$ (Augmentation with ABCD)
$A \rightarrow ABCDE$ (Transitivity)
$AF \rightarrow ABCDEF$ (Augmentation with F)

c. We see that D is extraneous in dep. 1 and 2, because of dep. 3. Removing these two, we get the new set of rules

$$A \rightarrow BC$$
$$BC \rightarrow E$$
$$B \rightarrow D$$
$$D \rightarrow A$$

Now notice that B^+ is $ABCDE$, and in particular, the FD $B \rightarrow E$ can be determined from this set. Thus, the attribute C is extraneous in

the third dependency. Removing this attribute, and combining with the third FD, we get the final canonical cover as :

$$A \rightarrow BC$$
$$B \rightarrow DE$$
$$D \rightarrow A$$

Here, no attribute is extraneous in any FD.

d. We see that there is no FD in the canonical cover such that the set of attributes is a subset of any other FD in the canonical cover. Thus, each each FD gives rise to its own relation, giving

$$r_1(A, B, C)$$
$$r_2(B, D, E)$$
$$r_3(D, A)$$

Now the attribute F is not dependent on any attribute. Thus, it must be a part of every superkey. Also, none of the relations in the above schema have F, and hence, none of them have a superkey. Thus, we need to add a new relation with a superkey.

$$r_4(A, F)$$

e. We start with

$$r(A, B, C, D, E, F)$$

We see that the relation is not in BCNF because of the first FD. Hence, we decompose it accordingly to get

$$r_1(A, B, C, D)\ r_2(A, E, F)$$

Now we notice that $A \rightarrow E$ is an FD in F^+, and causes r_2 to violate BCNF. Once again, decomposing r_2 gives

$$r_1(A, B, C, D)\ r_2(A, F)\ r_3(A, E)$$

This schema is now in BCNF.

f. Can you get the same BCNF decomposition of r as above, using the canonical cover?
If we use the functional dependencies in the preceding canonical cover directly, we cannot get the above decomposition. However, we can infer the original dependencies from the canonical cover, and if we use those for BCNF decomposition, we would be able to get the same decomposition.

8.30 List the three design goals for relational databases, and explain why each is desirable.

Answer: The three design goals are lossless-join decompositions, dependency preserving decompositions, and minimization of repetition of

information. They are desirable so we can maintain an accurate database, check correctness of updates quickly, and use the smallest amount of space possible.

8.31 In designing a relational database, why might we choose a non-BCNF design?

Answer: BCNF is not always dependency preserving. Therefore, we may want to choose another normal form (specifically, 3NF) in order to make checking dependencies easier during updates. This would avoid joins to check dependencies and increase system performance.

8.32 Given the three goals of relational-database design, is there any reason to design a database schema that is in 2NF, but is in no higher-order normal form? (See Practice Exercise 8.17 for the definition of 2NF.)

Answer: The three design goals of relational databases are to avoid

- Repetition of information
- Inability to represent information
- Loss of information.

2NF does not prohibit as much repetition of information since the schema (A, B, C) with dependencies $A \rightarrow B$ and $B \rightarrow C$ is allowed under 2NF, although the same (B, C) pair could be associated with many A values, needlessly duplicating C values. To avoid this we must go to 3NF. Repetition of information is allowed in 3NF in some but not all of the cases where it is allowed in 2NF. Thus, in general, 3NF reduces repetition of information. Since we can always achieve a lossless join 3NF decomposition, there is no loss of information needed in going from 2NF to 3NF.

Note that the decomposition $\{(A, B), (B, C)\}$ is a dependency-preserving and lossless-loin 3NF decomposition of the schema (A, B, C). However, in case we choose this decomposition, retrieving information about the relationship between A, B and C requires a join of two relations, which is avoided in the corresponding 2NF decomposition.

Thus, the decision of which normal form to choose depends upon how the cost of dependency checking compares with the cost of the joins. Usually, the 3NF would be preferred. Dependency checks need to be made with *every* insert or update to the instances of a 2NF schema, whereas, only some queries will require the join of instances of a 3NF schema.

8.33 Given a relational schema $r(A, B, C, D)$, does $A \twoheadrightarrow BC$ logically imply $A \twoheadrightarrow B$ and $A \twoheadrightarrow C$? If yes prove it, else give a counter example.

Answer: $A \twoheadrightarrow BC$ holds on the following table:

r :

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_2
a_1	b_1	c_1	d_2
a_1	b_2	c_2	d_1

If $A \twoheadrightarrow B$, then we know that there exists t_1 and t_3 such that $t_1[B] = t_3[B]$. Thus, we must choose one of the following for t_1 and t_3:

- $t_1 = r_1$ and $t_3 = r_3$, or $t_1 = r_3$ and $t_3 = r_1$:
  Choosing either $t_2 = r_2$ or $t_2 = r_4$, $t_3[C] \neq t_2[C]$.
- $t_1 = r_2$ and $t_3 = r_4$, or $t_1 = r_4$ and $t_3 = r_2$:
  Choosing either $t_2 = r_1$ or $t_2 = r_3$, $t_3[C] \neq t_2[C]$.

Therefore, the condition $t_3[C] = t_2[C]$ can not be satisfied, so the conjecture is false.

8.34 Explain why 4NF is a normal form more desirable than BCNF.

Answer: 4NF is more desirable than BCNF because it reduces the repetition of information. If we consider a BCNF schema not in 4NF (see Practice Exercise 7.16), we observe that decomposition into 4NF does not lose information provided that a lossless join decomposition is used, yet redundancy is reduced.