Show that this decomposition is a lossless decomposition if the following set F of functional dependencies holds:

$$
\begin{aligned}
A &\rightarrow BC \\
CD &\rightarrow E \\
B &\rightarrow D \\
E &\rightarrow A
\end{aligned}
$$

8.2 List all functional dependencies satisfied by the relation of Figure 8.17.

8.3 Explain how functional dependencies can be used to indicate the following:

- A one-to-one relationship set exists between entity sets *student* and *instructor*.
- A many-to-one relationship set exists between entity sets *student* and *instructor*.

8.4 Use Armstrong's axioms to prove the soundness of the union rule. (*Hint*: Use the augmentation rule to show that, if $\alpha \rightarrow \beta$, then $\alpha \rightarrow \alpha\beta$. Apply the augmentation rule again, using $\alpha \rightarrow \gamma$, and then apply the transitivity rule.)

8.5 Use Armstrong's axioms to prove the soundness of the pseudotransitivity rule.

8.6 Compute the closure of the following set F of functional dependencies for relation schema $r(A, B, C, D, E)$.

$$
\begin{aligned}
A &\rightarrow BC \\
CD &\rightarrow E \\
B &\rightarrow D \\
E &\rightarrow A
\end{aligned}
$$

List the candidate keys for R.

8.7 Using the functional dependencies of Practice Exercise 8.6, compute the canonical cover F_c.

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_2	b_1	c_1
a_2	b_1	c_3

Figure 8.17 Relation of Practice Exercise 8.2.

8.8 Consider the algorithm in Figure 8.18 to compute α^+. Show that this algorithm is more efficient than the one presented in Figure 8.8 (Section 8.4.2) and that it computes α^+ correctly.

8.9 Given the database schema $R(a, b, c)$, and a relation r on the schema R, write an SQL query to test whether the functional dependency $b \rightarrow c$ holds on relation r. Also write an SQL assertion that enforces the functional dependency; assume that no null values are present. (Although part of the SQL standard, such assertions are not supported by any database implementation currently.)

8.10 Our discussion of lossless-join decomposition implicitly assumed that attributes on the left-hand side of a functional dependency cannot take on null values. What could go wrong on decomposition, if this property is violated?

8.11 In the BCNF decomposition algorithm, suppose you use a functional dependency $\alpha \rightarrow \beta$ to decompose a relation schema $r(\alpha, \beta, \gamma)$ into $r_1(\alpha, \beta)$ and $r_2(\alpha, \gamma)$.

a. What primary and foreign-key constraint do you expect to hold on the decomposed relations?

b. Give an example of an inconsistency that can arise due to an erroneous update, if the foreign-key constraint were not enforced on the decomposed relations above.

c. When a relation is decomposed into 3NF using the algorithm in Section 8.5.2, what primary and foreign key dependencies would you expect will hold on the decomposed schema?

8.12 Let $R_1, R_2, \ldots, R_n$ be a decomposition of schema U. Let $u(U)$ be a relation, and let $r_i = \Pi_{R_i}(u)$. Show that

$$u \subseteq r_1 \bowtie r_2 \bowtie \cdots \bowtie r_n$$

8.13 Show that the decomposition in Practice Exercise 8.1 is not a dependency-preserving decomposition.

8.14 Show that it is possible to ensure that a dependency-preserving decomposition into 3NF is a lossless decomposition by guaranteeing that at least one schema contains a candidate key for the schema being decomposed. (*Hint*: Show that the join of all the projections onto the schemas of the decomposition cannot have more tuples than the original relation.)

8.15 Give an example of a relation schema R' and set F' of functional dependencies such that there are at least three distinct lossless decompositions of R' into BCNF.

```
result := ∅;
/* fdcount is an array whose ith element contains the number
   of attributes on the left side of the ith FD that are
   not yet known to be in α+ */
for i := 1 to |F| do
   begin
      let β → γ denote the ith FD;
      fdcount [i] := |β|;
   end
/* appears is an array with one entry for each attribute. The
   entry for attribute A is a list of integers. Each integer
   i on the list indicates that A appears on the left side
   of the ith FD */
for each attribute A do
   begin
      appears [A] := NIL;
      for i := 1 to |F| do
        begin
           let β → γ denote the ith FD;
           if A ∈ β then add i to appears [A];
        end
   end
addin (α);
return (result);

procedure addin (α);
for each attribute A in α do
   begin
      if A ∉ result then
        begin
           result := result ∪ {A};
           for each element i of appears[A] do
             begin
                fdcount [i] := fdcount [i] − 1;
                if fdcount [i] := 0 then
                  begin
                     let β → γ denote the ith FD;
                     addin (γ);
                  end
             end
        end
   end
```

Figure 8.18 An algorithm to compute α^+.

8.16 Let a **prime** attribute be one that appears in at least one candidate key. Let α and β be sets of attributes such that $\alpha \rightarrow \beta$ holds, but $\beta \rightarrow \alpha$ does not hold. Let A be an attribute that is not in α, is not in β, and for which $\beta \rightarrow A$ holds. We say that A is **transitively dependent** on α. We can restate our definition of 3NF as follows: *A relation schema R is in 3NF with respect to a set F of functional dependencies if there are no nonprime attributes A in R for which A is transitively dependent on a key for R*. Show that this new definition is equivalent to the original one.

8.17 A functional dependency $\alpha \rightarrow \beta$ is called a **partial dependency** if there is a proper subset γ of α such that $\gamma \rightarrow \beta$. We say that β is *partially dependent* on α. A relation schema R is in **second normal form** (2NF) if each attribute A in R meets one of the following criteria:

- It appears in a candidate key.
- It is not partially dependent on a candidate key.

Show that every 3NF schema is in 2NF. (*Hint*: Show that every partial dependency is a transitive dependency.)

8.18 Give an example of a relation schema R and a set of dependencies such that R is in BCNF but is not in 4NF.

Exercises

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

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

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

books(*accessionno*, *isbn*, *title*, *author*, *publisher*)
users(*userid*, *name*, *deptid*, *deptname*)
accessionno $\rightarrow$ *isbn*
isbn $\rightarrow$ *title*
isbn $\rightarrow$ *publisher*
isbn $\rightarrow\rightarrow$ *author*
userid $\rightarrow$ *name*
userid $\rightarrow$ *deptid*
deptid $\rightarrow$ *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.

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

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

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 $\alpha \rightarrow \gamma$.

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

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

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

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

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

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

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?

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

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

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.)

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.

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

Bibliographical Notes

The first discussion of relational database design theory appeared in an early paper by Codd [1970]. In that paper, Codd also introduced functional dependencies and first, second, and third normal forms.

Armstrong's axioms were introduced in Armstrong [1974]. Significant development of relational database theory occurred in the late 1970s. These results are collected in several texts on database theory including Maier [1983], Atzeni and Antonellis [1993], and Abiteboul et al. [1995].

BCNF was introduced in Codd [1972]. Biskup et al. [1979] give the algorithm we used to find a lossless dependency-preserving decomposition into 3NF. Fundamental results on the lossless decomposition property appear in Aho et al. [1979a].

Beeri et al. [1977] gives a set of axioms for multivalued dependencies, and proves that the authors' axioms are sound and complete. The notions of 4NF, PJNF, and DKNF are from Fagin [1977], Fagin [1979], and Fagin [1981], respectively. See the bibliographical notes of Appendix C for further references to literature on normalization.

Jensen et al. [1994] presents a glossary of temporal-database concepts. A survey of extensions to E-R modeling to handle temporal data is presented by Gregersen and Jensen [1999]. Tansel et al. [1993] covers temporal database theory, design, and implementation. Jensen et al. [1996] describes extensions of dependency theory to temporal data.