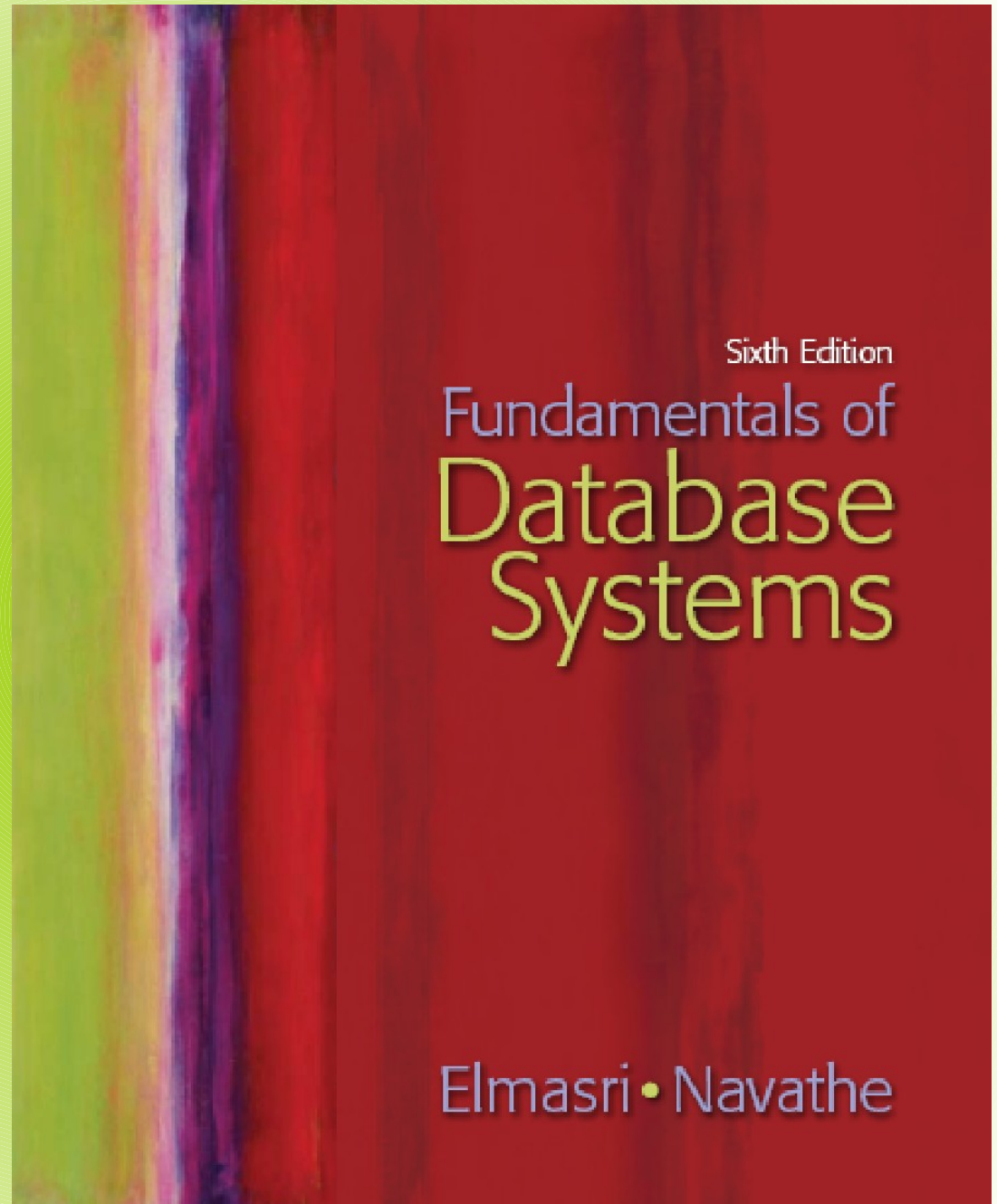


Chapter 15

**Basics of
Functional
Dependencies
and
Normalization
for
Relational
Databases**



Chapter 15 Outline

Informal Design Guidelines for Relation Schemas

Functional Dependencies

Normal Forms Based on Primary Keys

General Definitions of Second and Third Normal Forms

Boyce-Codd Normal Form

Introduction

Levels at which we can discuss *goodness* of relation schemas

- Logical (or conceptual) level

- Implementation (or physical storage) level

Approaches to database design:

- Bottom-up or top-down

Informal Design Guidelines for Relation Schemas

Measures of quality

- Making sure attribute semantics are clear

- Reducing redundant information in tuples

- Reducing NULL values in tuples

- Disallowing possibility of generating spurious tuples

Imparting Clear Semantics to Attributes in Relations

Semantics of a relation

Meaning resulting from interpretation of attribute values in a tuple

Easier to explain semantics of relation

Indicates better schema design

Guideline 1

Design relation schema so that it is easy to explain its meaning

Do not combine attributes from multiple entity types and relationship types into a single relation

Example of violating Guideline 1: Figure 15.3

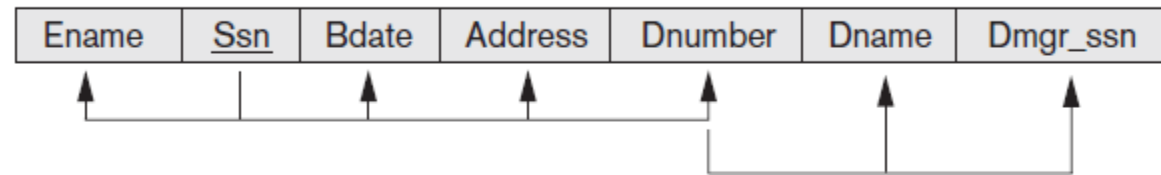
Guideline 1 (cont'd.)

Figure 15.3

Two relation schemas suffering from update anomalies. (a) EMP_DEPT and (b) EMP_PROJ.

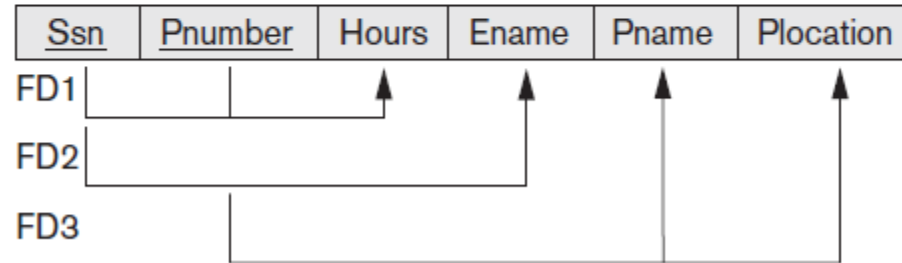
(a)

EMP_DEPT



(b)

EMP_PROJ



Redundant Information in Tuples and Update Anomalies

Grouping attributes into relation schemas

Significant effect on storage space

Storing natural joins of base relations
leads to **update anomalies**

Types of update anomalies:

Insertion

Deletion

Modification

Guideline 2

Design base relation schemas so that no update anomalies are present in the relations

If any anomalies are present:

- Note them clearly

- Make sure that the programs that update the database will operate correctly

NULL Values in Tuples

May group many attributes together into a “fat” relation

Can end up with many NULLs

Problems with NULLs

Wasted storage space

Problems understanding meaning

Guideline 3

Avoid placing attributes in a base relation whose values may frequently be NULL

If NULLs are unavoidable:

Make sure that they apply in exceptional cases only, not to a majority of tuples

Generation of Spurious Tuples

Figure 15.5(a)

Relation schemas EMP_LOCS and
EMP_PROJ1

NATURAL JOIN

Result produces many more tuples than the
original set of tuples in EMP_PROJ

Called **spurious tuples**

Represent spurious information that is not
valid

(a)

EMP_LOCS

<u>Ename</u>	<u>Plocation</u>
--------------	------------------

P.K.

EMP_PROJ1

<u>Ssn</u>	<u>Pnumber</u>	Hours	Pname	Plocation
------------	----------------	-------	-------	-----------

P.K.

Figure 15.5

Particularly poor design for the EMP_PROJ relation in Figure 15.3(b). (a) The two relation schemas EMP_LOCS and EMP_PROJ1. (b) The result of projecting the extension of EMP_PROJ from Figure 15.4 onto the relations EMP_LOCS and EMP_PROJ1.

EMP_PROJ

<u>Ssn</u>	<u>Pnumber</u>	Hours	Ename	Pname	Plocation
------------	----------------	-------	-------	-------	-----------

(b)

EMP_LOCS

Ename	Plocation
Smith, John B.	Bellaire
Smith, John B.	Sugarland
Narayan, Ramesh K.	Houston
English, Joyce A.	Bellaire
English, Joyce A.	Sugarland
Wong, Franklin T.	Sugarland
Wong, Franklin T.	Houston
Wong, Franklin T.	Stafford
Zelaya, Alicia J.	Stafford
Jabbar, Ahmad V.	Stafford
Wallace, Jennifer S.	Stafford
Wallace, Jennifer S.	Houston
Borg, James E.	Houston

EMP_PROJ1

Ssn	Pnumber	Hours	Pname	Plocation
123456789	1	32.5	ProductX	Bellaire
123456789	2	7.5	ProductY	Sugarland
666884444	3	40.0	ProductZ	Houston
453453453	1	20.0	ProductX	Bellaire
453453453	2	20.0	ProductY	Sugarland
333445555	2	10.0	ProductY	Sugarland
333445555	3	10.0	ProductZ	Houston
333445555	10	10.0	Computerization	Stafford
333445555	20	10.0	Reorganization	Houston
999887777	30	30.0	Newbenefits	Stafford
999887777	10	10.0	Computerization	Stafford
987987987	10	35.0	Computerization	Stafford
987987987	30	5.0	Newbenefits	Stafford
987654321	30	20.0	Newbenefits	Stafford
987654321	20	15.0	Reorganization	Houston
888665555	20	NULL	Reorganization	Houston

Ssn	Pnumber	Hours	Pname	Plocation	Ename
123456789	1	32.5	ProductX	Bellaire	Smith, John B.
* 123456789	1	32.5	ProductX	Bellaire	English, Joyce A.
123456789	2	7.5	ProductY	Sugarland	Smith, John B.
* 123456789	2	7.5	ProductY	Sugarland	English, Joyce A.
* 123456789	2	7.5	ProductY	Sugarland	Wong, Franklin T.
666884444	3	40.0	ProductZ	Houston	Narayan, Ramesh K.
* 666884444	3	40.0	ProductZ	Houston	Wong, Franklin T.
* 453453453	1	20.0	ProductX	Bellaire	Smith, John B.
453453453	1	20.0	ProductX	Bellaire	English, Joyce A.
* 453453453	2	20.0	ProductY	Sugarland	Smith, John B.
453453453	2	20.0	ProductY	Sugarland	English, Joyce A.
* 453453453	2	20.0	ProductY	Sugarland	Wong, Franklin T.
* 333445555	2	10.0	ProductY	Sugarland	Smith, John B.
* 333445555	2	10.0	ProductY	Sugarland	English, Joyce A.
333445555	2	10.0	ProductY	Sugarland	Wong, Franklin T.
* 333445555	3	10.0	ProductZ	Houston	Narayan, Ramesh K.
333445555	3	10.0	ProductZ	Houston	Wong, Franklin T.
333445555	10	10.0	Computerization	Stafford	Wong, Franklin T.
* 333445555	20	10.0	Reorganization	Houston	Narayan, Ramesh K.
333445555	20	10.0	Reorganization	Houston	Wong, Franklin T.

⋮

Figure 15.6

Result of applying NATURAL JOIN to the tuples above the dashed lines in EMP_PROJ1 and EMP_LOCS of Figure 15.5. Generated spurious tuples are marked by asterisks.

Guideline 4

Design relation schemas to be joined with equality conditions on attributes that are appropriately related

Guarantees that no spurious tuples are generated

Avoid relations that contain matching attributes that are not (foreign key, primary key) combinations

Summary and Discussion of Design Guidelines

Anomalies cause redundant work to be done

Waste of storage space due to NULLs

Difficulty of performing operations and joins due to NULL values

Generation of invalid and spurious data during joins

Functional Dependencies

Formal tool for analysis of relational schemas

Enables us to detect and describe some of the above-mentioned problems in precise terms

Theory of functional dependency

Definition of Functional Dependency

Constraint between two sets of attributes from the database

Definition. A functional dependency, denoted by $X \rightarrow Y$, between two sets of attributes X and Y that are subsets of R specifies a *constraint* on the possible tuples that can form a relation state r of R . The constraint is that, for any two tuples t_1 and t_2 in r that have $t_1[X] = t_2[X]$, they must also have $t_1[Y] = t_2[Y]$.

Property of semantics or meaning of the attributes

Legal relation states

Satisfy the functional dependency constraints

$\{ \text{State}, \text{Driver_license_number} \} \rightarrow \text{Ssn}$

Definition of Functional Dependency (cont'd.)

Given a populated relation

Cannot determine which FDs hold and which do not

Unless meaning of and relationships among attributes known

Can state that FD does not hold if there are tuples that show violation of such an FD

A	B	C	D
a1	b1	c1	d1
a1	b2	c2	d2
a2	b2	c2	d3
a3	b3	c4	d3

Normal Forms Based on Primary Keys

Normalization process

Approaches for relational schema design

Perform a conceptual schema design using a conceptual model then map conceptual design into a set of relations

Design relations based on external knowledge derived from existing implementation of files or forms or reports

Normalization of Relations

Takes a relation schema through a series of tests

Certify whether it satisfies a certain normal form

Proceeds in a top-down fashion

Normal form tests

Definition. The normal form of a relation refers to the highest normal form condition that it meets, and hence indicates the degree to which it has been normalized.

Practical Use of Normal Forms

Normalization carried out in practice

Resulting designs are of high quality and meet the desirable properties stated previously

Pays particular attention to normalization only up to 3NF, BCNF, or at most 4NF

Do not need to normalize to the highest possible normal form

Definition. Denormalization is the process of storing the join of higher normal form relations as a base relation, which is in a lower normal form.

Definitions of Keys and Attributes Participating in Keys

Definition of **superkey** and **key**

Candidate key

If more than one key in a relation schema

- One is **primary key**
- Others are **secondary keys**

Definition. An attribute of relation schema R is called a **prime attribute** of R if it is a member of *some candidate key* of R . An attribute is called **nonprime** if it is not a prime attribute—that is, if it is not a member of any candidate key.

First Normal Form

Part of the formal definition of a relation in the basic (flat) relational model

Only attribute values permitted are single **atomic (or indivisible) values**

Techniques to achieve first normal form


- Remove attribute and place in separate relation

- Expand the key

- Use several atomic attributes

(a)

DEPARTMENT

Dname	<u>Dnumber</u>	Dmgr_ssn	Dlocations
			

(b)

DEPARTMENT

Dname	<u>Dnumber</u>	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT

Dname	<u>Dnumber</u>	Dmgr_ssn	<u>Dlocation</u>
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Figure 15.9

Normalization into 1NF. (a) A relation schema that is not in 1NF. (b) Sample state of relation DEPARTMENT. (c) 1NF version of the same relation with redundancy.

First Normal Form (cont'd.)

Does not allow **nested relations**

Each tuple can have a relation within it

EMP_PROJ(Ssn , Ename , { PROJS(Pnumber , Hours)})

To change to 1NF:

Remove nested relation attributes into a new relation

Propagate the primary key into it

Unnest relation into a set of 1NF relations

(a)

EMP_PROJ

		Projs	
Ssn	Ename	Pnumber	Hours

(b)

EMP_PROJ

Ssn	Ename	Pnumber	Hours
123456789	Smith, John B.	1	32.5
		2	7.5
666884444	Narayan, Ramesh K.	3	40.0
453453453	English, Joyce A.	1	20.0
		2	20.0
333445555	Wong, Franklin T.	2	10.0
		3	10.0
		10	10.0
		20	10.0
999887777	Zelaya, Alicia J.	30	30.0
		10	10.0
987987987	Jabbar, Ahmad V.	10	35.0
		30	5.0
987654321	Wallace, Jennifer S.	30	20.0
		20	15.0
888665555	Borg, James E.	20	NULL

Figure 15.10

Normalizing nested relations into 1NF. (a) Schema of the EMP_PROJ relation with a *nested relation* attribute PROJS. (b) Sample extension of the EMP_PROJ relation showing nested relations within each tuple. (c) Decomposition of EMP_PROJ into relations EMP_PROJ1 and EMP_PROJ2 by propagating the primary key.

(c)

EMP_PROJ1

<u>Ssn</u>	Ename
------------	-------

EMP_PROJ2

<u>Ssn</u>	<u>Pnumber</u>	Hours
------------	----------------	-------

Second Normal Form

Based on concept of **full functional dependency**

Versus **partial dependency**

Definition. A relation schema R is in 2NF if every nonprime attribute A in R is *fully functionally dependent* on the primary key of R .

Second normalize into a number of 2NF relations

Nonprime attributes are associated only with part of primary key on which they are fully functionally dependent

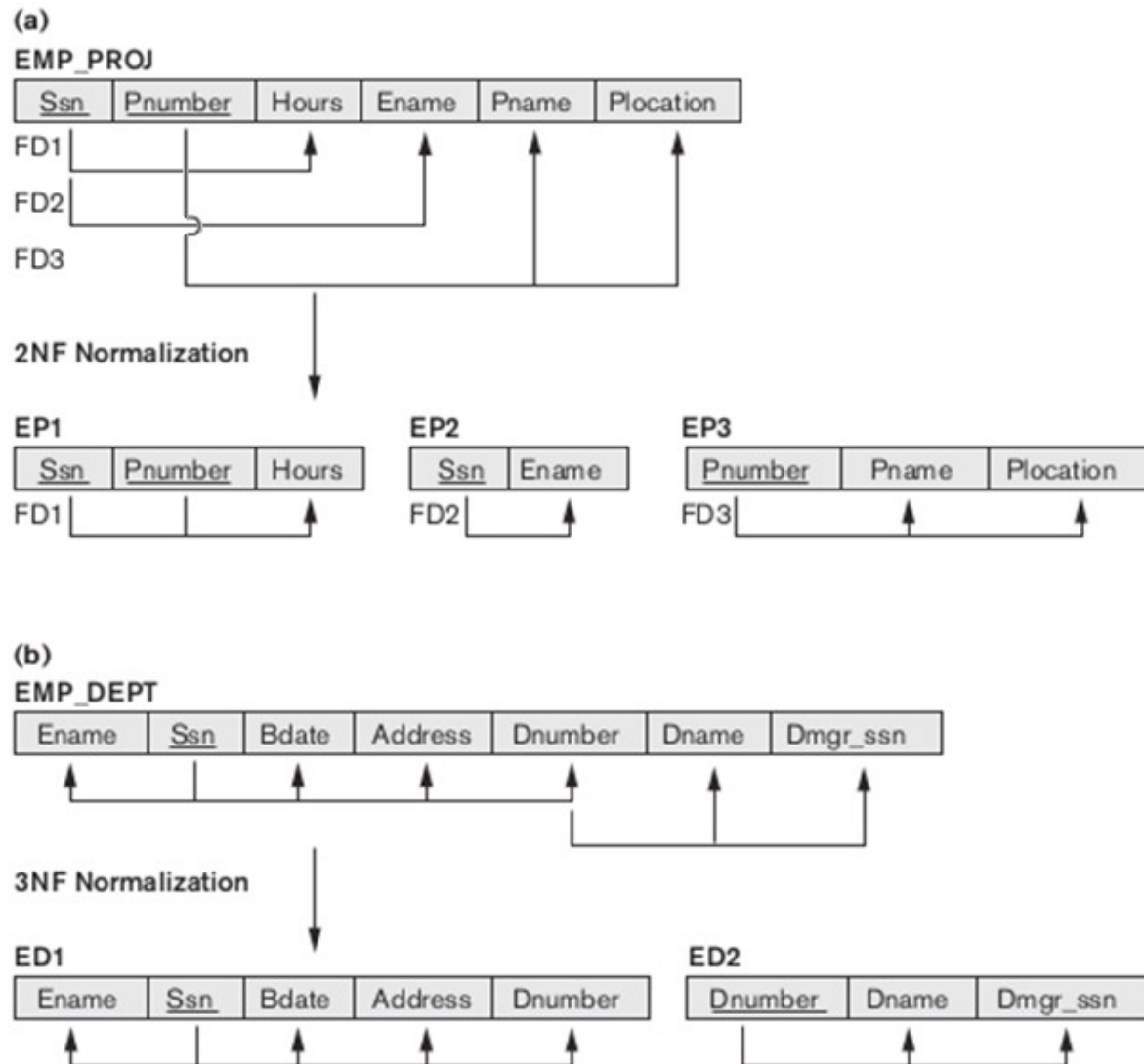


Figure 15.11

Normalizing into 2NF and 3NF. (a) Normalizing EMP_PROJ into 2NF relations. (b) Normalizing EMP_DEPT into 3NF relations.

Third Normal Form

Based on concept of transitive

Definition. According to Codd's original definition, a relation schema R is in 3NF if it satisfies 2NF *and* no nonprime attribute of R is transitively dependent on the primary key.

Problematic FD

Left-hand side is part of primary key

Left-hand side is a nonkey attribute

General Definitions of Second and Third Normal Forms

Table 15.1 Summary of Normal Forms Based on Primary Keys and Corresponding Normalization

Normal Form	Test	Remedy (Normalization)
First (1NF)	Relation should have no multivalued attributes or nested relations.	Form new relations for each multivalued attribute or nested relation.
Second (2NF)	For relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.	Decompose and set up a new relation for each partial key with its dependent attribute(s). Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it.
Third (3NF)	Relation should not have a nonkey attribute functionally determined by another nonkey attribute (or by a set of nonkey attributes). That is, there should be no transitive dependency of a nonkey attribute on the primary key.	Decompose and set up a relation that includes the nonkey attribute(s) that functionally determine(s) other nonkey attribute(s).

General Definitions of Second and Third Normal Forms (cont'd.)

Prime attribute

Part of any candidate key will be considered as prime

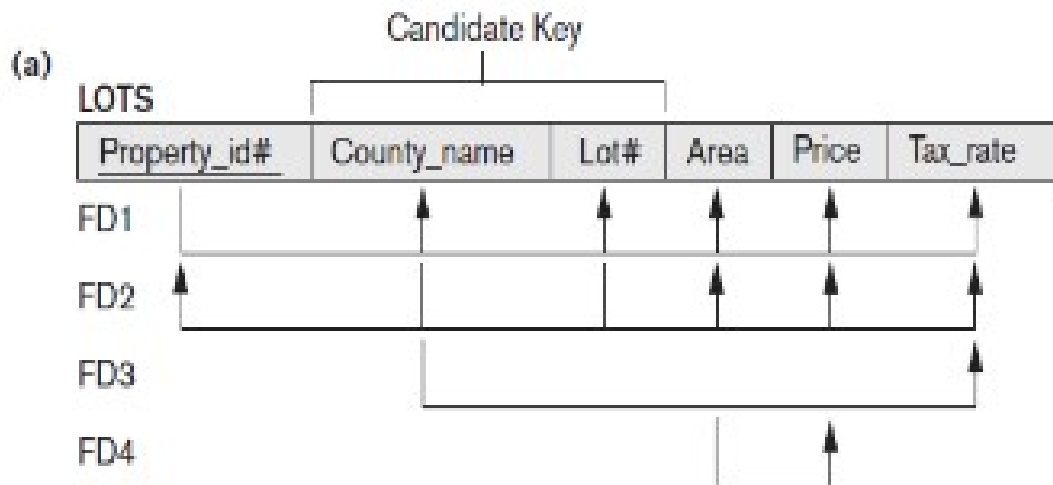
Consider partial, full functional, and transitive dependencies with respect to all candidate keys of a relation

General Definition of Second Normal Form

Definition. A relation schema R is in **second normal form (2NF)** if every non-prime attribute A in R is not partially dependent on *any* key of R .¹¹

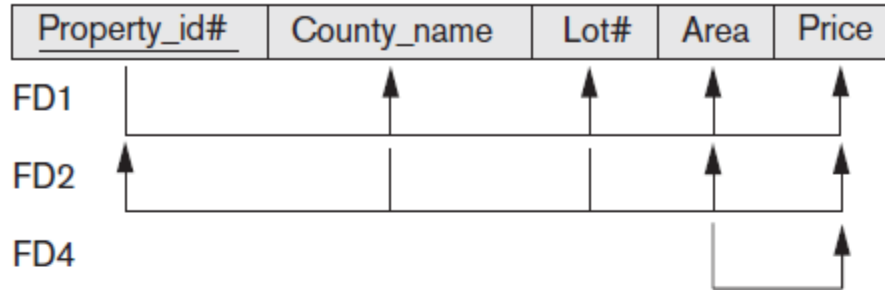
Figure 15.12

Normalization into 2NF and 3NF. (a) The LOTS relation with its functional dependencies FD1 through FD4. (b) Decomposing into the 2NF relations LOTS1 and LOTS2. (c) Decomposing LOTS1 into the 3NF relations LOTS1A and LOTS1B. (d) Summary of the progressive normalization of LOTS.

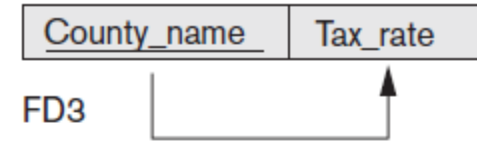


(b)

LOTS1

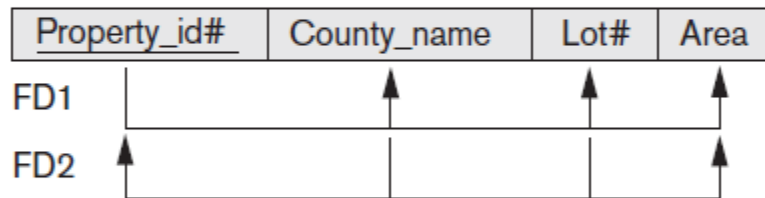


LOTS2

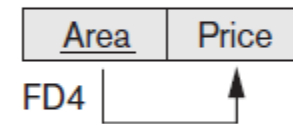


(c)

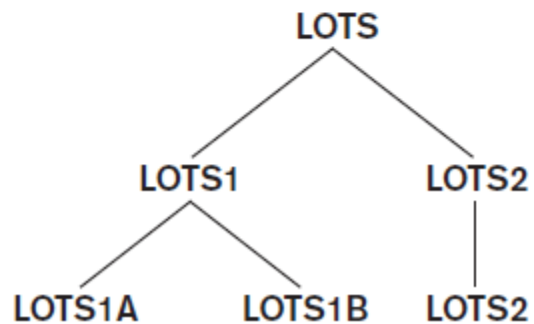
LOTS1A



LOTS1B



(d)



1NF

2NF

3NF

General Definition of Third Normal Form

Definition. A relation schema R is in **third normal form (3NF)** if, whenever a *nontrivial* functional dependency $X \rightarrow A$ holds in R , either (a) X is a superkey of R , or (b) A is a prime attribute of R .

Alternative Definition. A relation schema R is in 3NF if every nonprime attribute of R meets both of the following conditions:

- It is fully functionally dependent on every key of R .
- It is nontransitively dependent on every key of R .

Boyce-Codd Normal Form

Every relation in BCNF is also in 3NF

Relation in 3NF is not necessarily in BCNF

Definition. A relation schema R is in BCNF if whenever a *nontrivial* functional dependency $X \rightarrow A$ holds in R , then X is a superkey of R .

Difference:

Condition which allows A to be prime is absent from BCNF

Most relation schemas that are in 3NF are also in BCNF

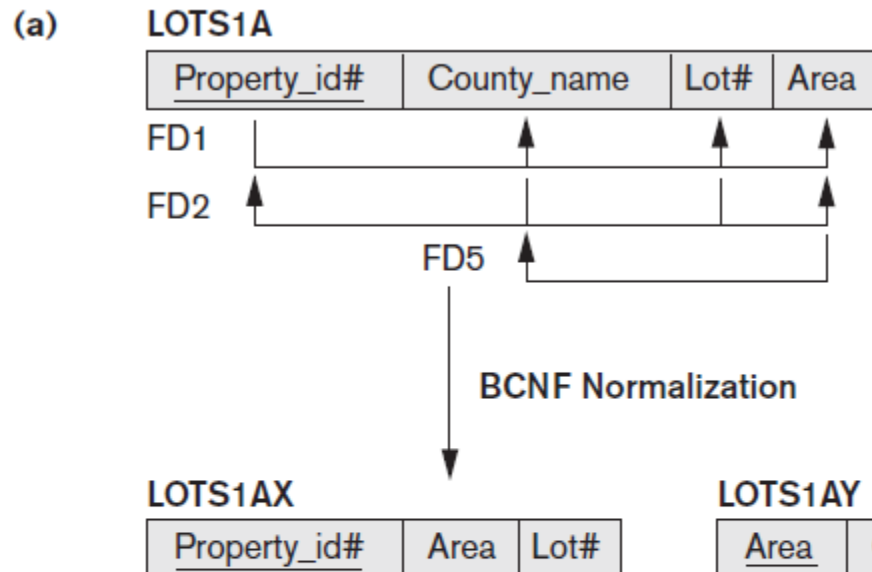
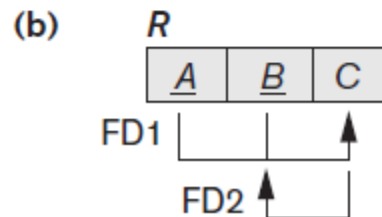


Figure 15.13

Boyce-Codd normal form. (a) BCNF normalization of LOTS1A with the functional dependency FD2 being lost in the decomposition. (b) A schematic relation with FDs; it is in 3NF, but not in BCNF.



Summary

Informal guidelines for good design

Functional dependency

Basic tool for analyzing relational schemas

Normalization:

1NF, 2NF, 3NF, BCNF, (4NF, 5NF)