

CS2005 Database Systems

Fall 2021

National University of Computer and Emerging Sciences

CHAPTER 14

Basics of Functional Dependencies and Normalization for Relational Databases

Normalization || Data Normalization

- Normalization is the process of reorganizing/restructuring data in a database with a series of so called **normal-forms**, so that it meets two basic requirements:
 1. There is no redundancy of data (all data is stored in only one place), and
 2. Data dependencies are logical (all related data items are stored together)

- Normalization is important for many reasons, but chiefly because it allows databases to take up as little disk space as possible, resulting in increased performance.

1. Informal Design Guidelines for Relational Databases (1)

- What is relational database design?
 - The grouping of attributes to form "**good**" relation schemas
- Two levels of relation schemas
 - The logical "user view" level
 - The storage "base relation" level
- Design is concerned mainly with base relations
- What are the criteria for "good" base relations?

Informal Design Guidelines for Relational Databases (2)

- We first discuss informal guidelines for good relational design
- Then we discuss formal concepts of functional dependencies and normal forms
 - – 1NF (First Normal Form)
 - – 2NF (Second Normal Form)
 - – 3NF (Third Normal Form)
 - – BCNF (Boyce-Codd Normal Form)
- Additional types of dependencies, further normal forms, relational design algorithms by synthesis are discussed in Chapter 15

1.1 Semantics of the Relational Attributes must be clear

- GUIDELINE 1: Informally, each tuple in a relation should represent one entity or relationship instance. (Applies to individual relations and their attributes).
 - Attributes of different entities (EMPLOYEEs, DEPARTMENTs, PROJECTs) should not be mixed in the same relation
 - Only foreign keys should be used to refer to other entities
 - Entity and relationship attributes should be kept apart as much as possible.

- Bottom Line: *Design a schema that can be explained easily relation by relation. The semantics of attributes should be easy to interpret.*

EMPLOYEE

F.K.

Ename	<u>Ssn</u>	Bdate	Address	Dnumber
-------	------------	-------	---------	---------

P.K.

DEPARTMENT

F.K.

Dname	<u>Dnumber</u>	Dmgr_ssn
-------	----------------	----------

P.K.

DEPT_LOCATIONS

F.K.

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

P.K.

PROJECT

F.K.

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

P.K.

WORKS_ON

F.K.

F.K.

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

P.K.

Figure 14.1 A simplified COMPANY relational database schema

1.2 Redundant Information in Tuples and Update Anomalies

- Information is stored redundantly
 - Wastes storage
 - Causes problems with update anomalies
 - Insertion Anomalies
 - Deletion Anomalies
 - Modification Anomalies

EXAMPLE OF AN UPDATE ANOMALY

□ Consider the relation:

- EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)

□ Update Anomaly:

- Changing the name of project number P1 from “Billing” to “Customer-Accounting” may cause this update to be made for all 100 employees working on project P1.

EXAMPLE OF AN INSERT ANOMALY

□ Consider the relation:

- EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)

□ Insert Anomaly:

- Cannot insert a project unless an employee is assigned to it.

□ Conversely

- Cannot insert an employee unless an he/she is assigned to a project.

EXAMPLE OF A DELETE ANOMALY

□ Consider the relation:

- EMP_PROJ(Emp#, Proj#, Ename, Pname, No_hours)

□ Delete Anomaly:

- When a project is deleted, it will result in deleting all the employees who work on that project.
- Alternately, if an employee is the sole employee on a project, deleting that employee would result in deleting the corresponding project.

(a)

EMP_DEPT



(b)

EMP_PROJ

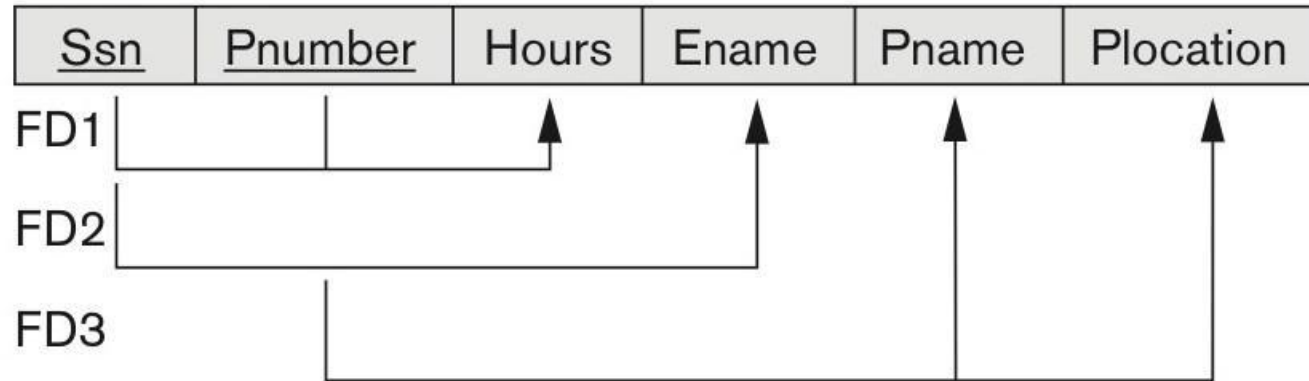


Figure 14.3

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

					Redundancy	
EMP_DEPT						
Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn
Smith, John B.	123456789	1965-01-09	731 Fondren, Houston, TX	5	Research	333445555
Wong, Franklin T.	333445555	1955-12-08	638 Voss, Houston, TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle, Spring, TX	4	Administration	987654321
Wallace, Jennifer S.	987654321	1941-06-20	291 Berry, Bellaire, TX	4	Administration	987654321
Narayan, Ramesh K.	666884444	1962-09-15	975 FireOak, Humble, TX	5	Research	333445555
English, Joyce A.	453453453	1972-07-31	5631 Rice, Houston, TX	5	Research	333445555
Jabbar, Ahmad V.	987987987	1969-03-29	980 Dallas, Houston, TX	4	Administration	987654321
Borg, James E.	888665555	1937-11-10	450 Stone, Houston, TX	1	Headquarters	888665555

			Redundancy		Redundancy	
EMP_PROJ						
<u>Ssn</u>	<u>Pnumber</u>	Hours	Ename	Pname	Plocation	
123456789	1	32.5	Smith, John B.	ProductX	Bellaire	
123456789	2	7.5	Smith, John B.	ProductY	Sugarland	
666884444	3	40.0	Narayan, Ramesh K.	ProductZ	Houston	
453453453	1	20.0	English, Joyce A.	ProductX	Bellaire	
453453453	2	20.0	English, Joyce A.	ProductY	Sugarland	
333445555	2	10.0	Wong, Franklin T.	ProductY	Sugarland	
333445555	3	10.0	Wong, Franklin T.	ProductZ	Houston	
333445555	10	10.0	Wong, Franklin T.	Computerization	Stafford	
333445555	20	10.0	Wong, Franklin T.	Reorganization	Houston	
999887777	30	30.0	Zelaya, Alicia J.	Newbenefits	Stafford	
999887777	10	10.0	Zelaya, Alicia J.	Computerization	Stafford	
987987987	10	35.0	Jabbar, Ahmad V.	Computerization	Stafford	
987987987	30	5.0	Jabbar, Ahmad V.	Newbenefits	Stafford	
987654321	30	20.0	Wallace, Jennifer S.	Newbenefits	Stafford	
987654321	20	15.0	Wallace, Jennifer S.	Reorganization	Houston	
888665555	20	Null	Borg, James E.	Reorganization	Houston	

Figure 14.4
Sample states for EMP_DEPT and EMP_PROJ resulting from applying NATURAL JOIN to the relations in Figure 14.2. These may be stored as base relations for performance reasons.

Guideline for Redundant Information in Tuples and Update Anomalies

□ GUIDELINE 2:

- Design a schema that does not suffer from the insertion, deletion and update anomalies.
- If there are any anomalies present, then note them so that applications can be made to take them into account.

1.3 Null Values in Tuples

□ GUIDELINE 3

- Relations should be designed such that their tuples will have as few NULL values as possible
- Attributes that are NULL frequently could be placed in separate relations (with the primary key)

□ Reasons for nulls:

- ***Attribute not applicable or invalid*** (e.g. Visa_Status may not apply to local students)
- ***Attribute value unknown (may exist)*** (e.g. Date_of_birth may be unknown for an employee)
- ***Value known to exist, but unavailable*** (e.g. Home_Phone_Number for an employee may exist, but may not be available and recorded yet.)

(a)

EMP_LOCS

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

P.K.

EMP_PROJ1

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

P.K.

(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

Select *

from EMP_LOCS natural join EMP_PROJ1

where SSN = 123456789;

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.

(a)

EMP_LOCS

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

P.K.

Select *

from EMP_LOCS natural join EMP_PROJ1;

EMP_PROJ1

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

P.K.

(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
123456789	1	32.5	ProductX
123456789	2	7.5	ProductY
666884444	3	40.0	ProductZ
453453453	1	20.0	ProductX
453453453	2	20.0	ProductY
333445555	2	10.0	ProductY
333445555	3	10.0	ProductZ
333445555	10	10.0	Computerization
333445555	20	10.0	Reorganization
999887777	30	30.0	Newbenefits
999887777	10	10.0	Computerization
987987987	10	35.0	Computerization
987987987	30	5.0	Newbenefits
987654321	30	20.0	Newbenefits
987654321	20	15.0	Reorganization
888665555	20	NULL	Reorganization

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.

1.4 Generation of Spurious Tuples – avoid at any cost

- Design relation schemas so that they can be joined with equality conditions on attributes that are appropriately related (primary key, foreign key) pairs in a way that guarantees that no spurious tuples are generated.
- Avoid relations that contain matching attributes that are not (foreign key, primary key) combinations because joining on such attributes may produce spurious tuples.
- Bad designs for a relational database may result in erroneous results for certain JOIN operations
- The "lossless join" property is used to guarantee meaningful results for join operations
- **GUIDELINE 4:**
 - The relations should be designed to satisfy the lossless join condition.
 - No spurious tuples should be generated by doing a natural-join of any relations.

Spurious Tuples (2)

- There are two important properties of decompositions:
 - a) Non-additive or losslessness of the corresponding join
 - b) Preservation of the functional dependencies.

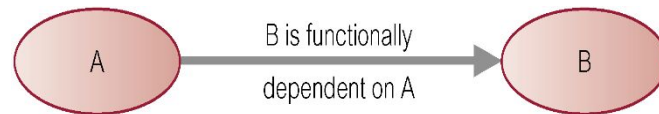
- Note that:
 - Property (a) is extremely important and cannot be sacrificed.
 - Property (b) is less stringent and may be sacrificed. (See Chapter 15).

2. Functional Dependencies

- Functional dependencies (FDs)
 - Are used to specify *formal measures* of the "goodness" of relational designs
 - And keys are used to define **normal forms** for relations
 - Are **constraints** that are derived from the *meaning* and *interrelationships* of the data attributes
- A set of attributes X *functionally determines* a set of attributes Y if the value of X determines a unique value for Y

2.1 Defining Functional Dependencies

- If A and B are attributes of relation R, B is functionally dependent on A (denoted $A \rightarrow B$), if each value of A in R is associated with exactly one value of B in R.



- $X \rightarrow Y$ holds if whenever two tuples have the same value for X, they *must have* the same value for Y
 - For any two tuples t1 and t2 in any relation instance r(R): If $t1[X]=t2[X]$, then $t1[Y]=t2[Y]$
- $X \rightarrow Y$ in R specifies a *constraint* on all relation instances r(R)
- Written as $X \rightarrow Y$; can be displayed graphically on a relation schema as in Figures. (denoted by the arrow:).
- FDs are derived from the real-world constraints on the attributes

Examples of FD constraints (1)

- Social security number determines employee name
 - $SSN \rightarrow ENAME$
- Project number determines project name and location
 - $PNUMBER \rightarrow \{PNAME, PLOCATION\}$
- Employee ssn and project number determines the hours per week that the employee works on the project
 - $\{SSN, PNUMBER\} \rightarrow HOURS$

Examples of FD constraints (2)

- An FD is a property of the attributes in the schema R
- The constraint must hold on *every* relation instance $r(R)$
- If K is a key of R , then K functionally determines all attributes in R
 - (since we never have two distinct tuples with $t_1[K]=t_2[K]$)

Defining FDs from instances

- Note that in order to define the FDs, we need to understand the meaning of the attributes involved and the relationship between them.
- An FD is a property of the attributes in the schema R
- Given the instance (population) of a relation, all we can conclude is that an FD may exist between certain attributes.
- What we can definitely conclude is – that certain FDs do not exist because there are tuples that show a violation of those dependencies.

Figure 14.7 Ruling Out FDs

Note that given the state of the TEACH relation, we can say that the FD: Text \rightarrow Course may exist. However, the FDs Teacher \rightarrow Course, Teacher \rightarrow Text and Course \rightarrow Text are ruled out.

TEACH

Teacher	Course	Text
Smith	Data Structures	Bartram
Smith	Data Management	Martin
Hall	Compilers	Hoffman
Brown	Data Structures	Horowitz

Figure 14.8 What FDs may exist?

- A relation $R(A, B, C, D)$ with its extension.
- Which FDs may exist in this relation?

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

FD holds

$B \rightarrow C$

$C \rightarrow B$

$\{A, B\} \rightarrow C$

$\{A, B\} \rightarrow D$

$\{A, C\} \rightarrow B$

$\{A, C\} \rightarrow D$

$\{A, D\} \rightarrow B$

$\{A, D\} \rightarrow C$

$\{B, D\} \rightarrow A$

$\{B, D\} \rightarrow C$

$\{C, D\} \rightarrow A$

$\{C, D\} \rightarrow B$

$\{A, B, C\} \rightarrow D$

$\{A, B, D\} \rightarrow C$

$\{B, C, D\} \rightarrow A$

FD not holds

$A \rightarrow B$

$A \rightarrow C$

$A \rightarrow D$

$B \rightarrow A$

$B \rightarrow D$

$C \rightarrow A$

$C \rightarrow D$

$D \rightarrow A$

$D \rightarrow B$

$D \rightarrow C$

$\{B, C\} \rightarrow A$

$\{B, C\} \rightarrow D$

Important Definitions

□ Determinant

- Refers to the attribute, or group of attributes, on the left-hand side of the arrow of a functional dependency.

□ Full Functional dependency:

- Indicates that if A and B are attributes of a relation, B is fully functionally dependent on A if B is functionally dependent on A, but not on any proper subset of A.
 - {StaffNo, StaffName} □ BranchNo
 - StaffNo □ BranchNo

□ Transitive Dependency

- A condition where A, B, and C are attributes of a relation such that if $A \rightarrow B$ and $B \rightarrow C$, then C is transitively dependent on A via B (provided that A is not functionally dependent on B or C).

Example Transitive Dependency

StaffBranch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

- Consider functional dependencies in the StaffBranch relation

staffNo → sName, position, salary, branchNo, bAddress

branchNo → bAddress

- Transitive dependency,

- branchNo → bAddress exists on staffNo via branchNo

3.1 Normalization of Relations (1)

□ Normalization:

- The process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations

□ Normal form:

- Condition using keys and FDs of a relation to certify whether a relation schema is in a particular normal form

Normalization of Relations (2)

□ 2NF, 3NF, BCNF

- based on keys and FDs of a relation schema

□ 4NF

- based on keys, multi-valued dependencies : MVDs;

□ 5NF

- based on keys, join dependencies : JDs

- Additional properties may be needed to ensure a good relational design (lossless join, dependency preservation)

3.2 Practical Use of Normal Forms

- **Normalization** is carried out in practice so that the resulting designs are of high quality and meet the desirable properties
- The practical utility of these normal forms becomes questionable when the constraints on which they are based are *hard to understand* or to *detect*
- The database designers *need not* normalize to the highest possible normal form
 - (usually up to 3NF and BCNF. 4NF rarely used in practice.)
- **Denormalization:**
 - The process of storing the join of higher normal form relations as a base relation—which is in a lower normal form

3.3 Definitions of Keys and Attributes Participating in Keys (1)

- A **superkey** of a relation schema $R = \{A_1, A_2, \dots, A_n\}$ is a set of attributes S *subset-of* R with the property that no two tuples t_1 and t_2 in any legal relation state r of R will have $t_1[S] = t_2[S]$
- A **key** K is a **superkey** with the *additional property* that removal of any attribute from K will cause K not to be a superkey any more.

Definitions of Keys and Attributes Participating in Keys (2)

- If a relation schema has more than one key, each is called a **candidate** key.
 - One of the candidate keys is *arbitrarily* designated to be the **primary key**, and the others are called **secondary keys**.
- A **Prime attribute** must be a member of *some* candidate key
- A **Nonprime attribute** is not a prime attribute—that is, it is not a member of any candidate key

3.4 First Normal Form

□ Disallows

- composite attributes
- multivalued attributes (no repeating group)
- **nested relations**; attributes whose values for an *individual tuple* are non-atomic (Composite)

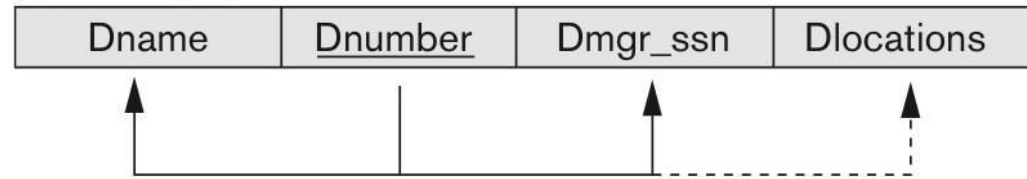
title	author_list	date	keyword_list
		day month year	
salesplan	{Smith, Jones}	1 April 89	{profit, strategy}
stat. report	{Jones, Frick}	17 July 94	{profit, personnel}

□ Considered to be part of the definition of a relation

□ Most RDBMSs allow only those relations to be defined that are in First Normal Form

(a)

DEPARTMENT



(b)

DEPARTMENT

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

Figure 14.9

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.

(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

(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

(c)

EMP_PROJ1

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

EMP_PROJ2

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

Figure 14.10

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.

3.5 Second Normal Form (1)

□ Uses the concepts of **FDs**, **primary key**

□ Definitions

- **Prime attribute:** An attribute that is member of the primary key K
- **Full functional dependency:** a FD $Y \rightarrow Z$ where removal of any attribute from Y means the FD does not hold any more

□ Examples:

- $\{SSN, PNUMBER\} \rightarrow HOURS$ is a full FD since neither $SSN \rightarrow HOURS$ nor $PNUMBER \rightarrow HOURS$ hold
- $\{SSN, PNUMBER\} \rightarrow ENAME$ is not a full FD (it is called a partial dependency) since $SSN \rightarrow ENAME$ also holds

Second Normal Form (2)

- A relation schema R is in **second normal form (2NF)** if every non-prime attribute A in R is fully functionally dependent on the primary key
- R can be decomposed into 2NF relations via the process of 2NF normalization or “second normalization”

(a)

EMP_PROJ

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



2NF Normalization

EP1

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



EP2

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



EP3

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



Figure 14.11

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

3.6 Third Normal Form (1)

□ Definition:

- **Transitive functional dependency:** a FD $X \rightarrow Z$ that can be derived from two FDs $X \rightarrow Y$ and $Y \rightarrow Z$

□ Examples:

- $SSN \rightarrow DMGRSSN$ is a **transitive** FD
 - Since $SSN \rightarrow DNUMBER$ and $DNUMBER \rightarrow DMGRSSN$ hold
- $SSN \rightarrow ENAME$ is **non-transitive**
 - Since there is no set of attributes X where $SSN \rightarrow X$ and $X \rightarrow ENAME$

Third Normal Form (2)

- A relation schema R is in **third normal form (3NF)** if it is in 2NF *and* no non-prime attribute A in R is *transitively dependent* on the primary key
- R can be decomposed into 3NF relations via the process of 3NF normalization
- NOTE:
 - In $X \rightarrow Y$ and $Y \rightarrow Z$, with X as the primary key, we consider this a problem only if Y is not a candidate key.
 - When Y is a candidate key, there is no problem with the transitive dependency .
 - E.g., Consider EMP (SSN, Emp#, Salary).
 - Here, $SSN \rightarrow Emp\# \rightarrow Salary$ and $Emp\#$ is a candidate key.

Figure 14.11
Normalizing into 2NF and 3NF.
(b) Normalizing EMP_DEPT into
3NF relations.

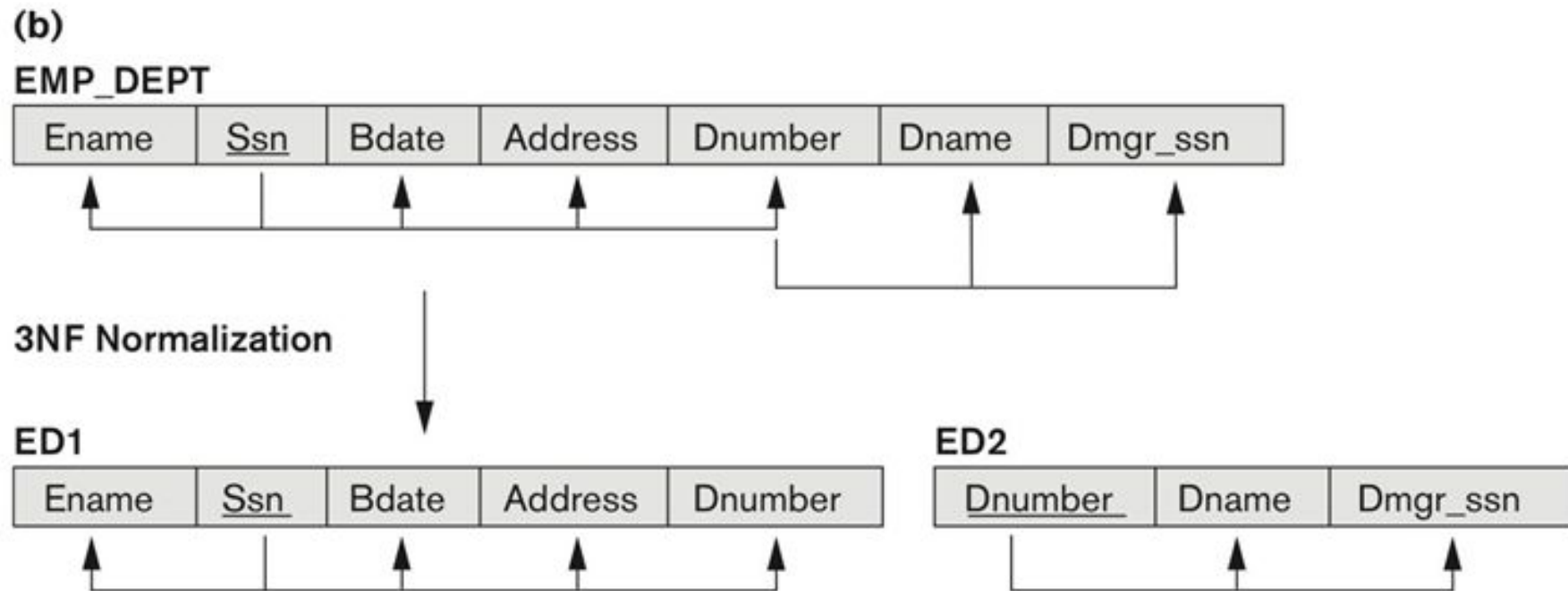
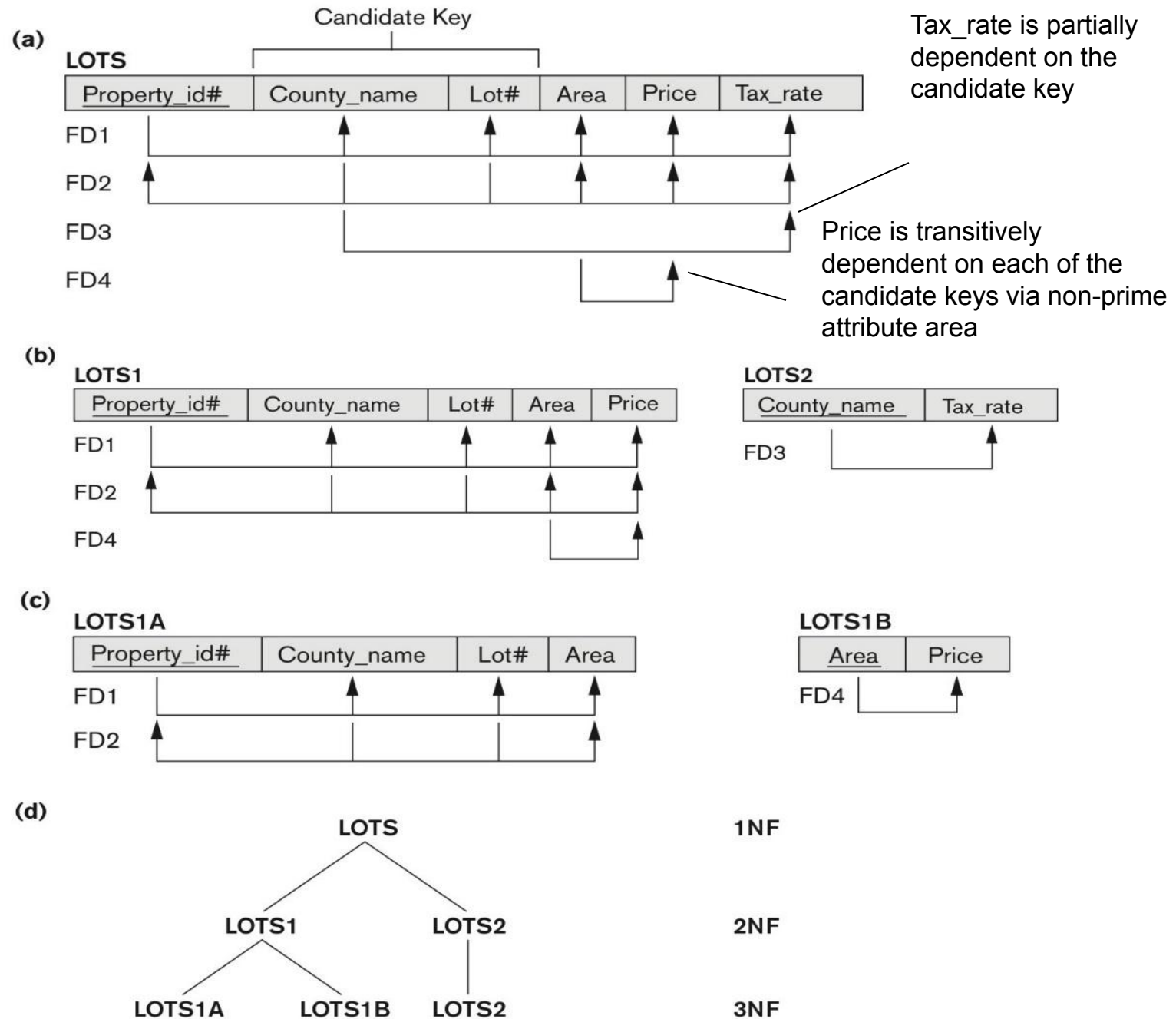


Figure 14.12 Normalization into 2NF and 3NF. (a) The LOTS relation with its functional dependencies FD1 through FD4. (b) Decomposing into the 2NF relations LOTS1 and LOTS2. (c) Decomposing LOTS1 into the 3NF relations LOTS1A and LOTS1B. (d) Progressive normalization of LOTS into a 3NF design.



Normal Forms Defined Informally

- 1st normal form
 - All attributes depend on **the key**
- 2nd normal form
 - All attributes depend on **the whole key**
- 3rd normal form
 - All attributes depend on **nothing but the key**

Table 14.1 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).

□ 3NF

- No Non-Prime attribute is transitively dependent on any candidate key

□ BCNF (Boyce-Codd Normal Form)

- Every determinant is a candidate key
- every relation in BCNF is also in 3NF; however, a relation in 3NF is *not necessarily* in BCNF

(a)

LOTS1A

<u>Property_id#</u>	County_name	Lot#	Area
---------------------	-------------	------	------



BCNF Normalization

LOTS1AX

<u>Property_id#</u>	Area	Lot#
---------------------	------	------

LOTS1AY

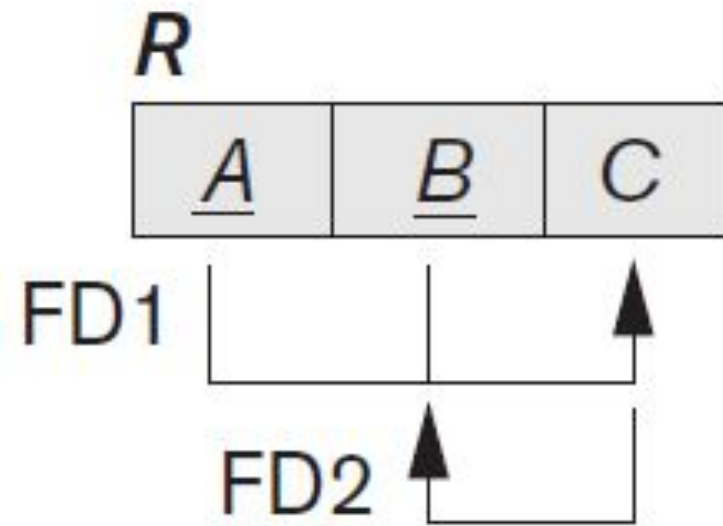
<u>Area</u>	County_name
-------------	-------------

Suppose that we have thousands of lots in the relation, but the lots are from only two counties: Punjab and Sindh.

Suppose also that lot sizes in Punjab County are only 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0 acres, whereas

lot sizes in Sindh County are restricted to 1.1, 1.2, ... , 1.9, and 2.0 acres.

In such a situation we would have the additional functional dependency FD5: Area → County_name.



The table above is in 3NF but not in BCNF, because of the $C \twoheadrightarrow B$

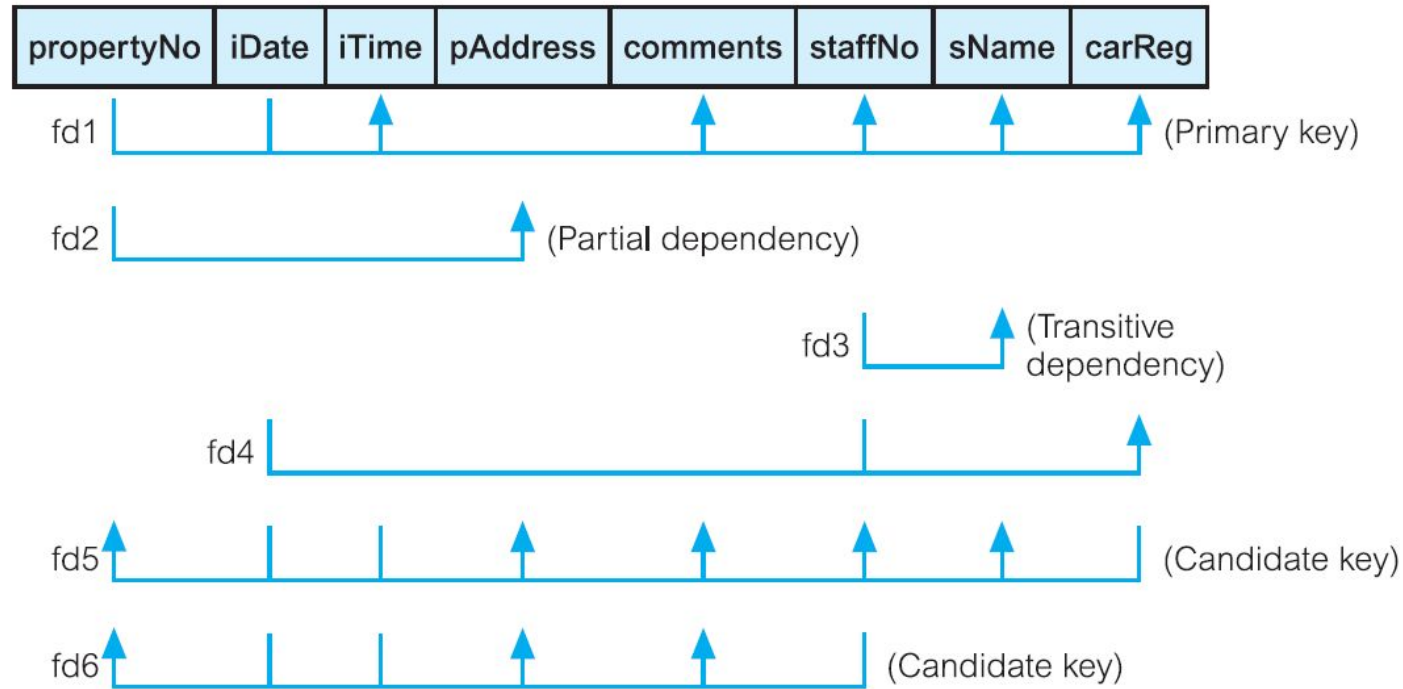
StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg
PG4	18-Oct-12	10.00	6 Lawrence St, Glasgow	Need to replace crockery	SG37	Ann Beech	M231 JGR
PG4	22-Apr-13	09.00	6 Lawrence St, Glasgow	In good order	SG14	David Ford	M533 HDR
PG4	1-Oct-13	12.00	6 Lawrence St, Glasgow	Damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	22-Apr-13	13.00	5 Novar Dr, Glasgow	Replace living room carpet	SG14	David Ford	M533 HDR
PG16	24-Oct-13	14.00	5 Novar Dr, Glasgow	Good condition	SG37	Ann Beech	N721 HFR

The First Normal Form(1NF) StaffPropertyInspection relation.

StaffPropertyInspection (propertyNo, iDate, iTime, pAddress, comments, staffNo, sName, carReg)

StaffPropertyInspection



- fd1 propertyNo, iDate \rightarrow iTime, comments, staffNo, sName, carReg
(Primary key)
- fd2 propertyNo \rightarrow pAddress
(Partial dependency)
- fd3 staffNo \rightarrow sName
(Transitive dependency)
- fd4 staffNo, iDate \rightarrow carReg
- fd5 carReg, iDate, iTime \rightarrow propertyNo, pAddress, comments, staffNo, sName
(Candidate key)
- fd6 staffNo, iDate, iTime \rightarrow propertyNo, pAddress, comments
(Candidate key)

Second Normal Form

Property (propertyNo, pAddress)

PropertyInspection (propertyNo, iDate, iTime, comments, staffNo, sName, carReg)

Third Normal Form (3NF)

Property (propertyNo, pAddress)

Staff (staffNo, sName)

PropertyInspect (propertyNo, iDate, iTime, comments, staffNo, carReg)

Boyce–Codd Normal Form (BCNF)

StaffCar (staffNo, iDate, carReg)

Inspection (propertyNo, iDate, iTime, comments, staffNo)

Property Relation

fd2 $\text{propertyNo} \rightarrow \text{pAddress}$

Staff Relation

fd3 $\text{staffNo} \rightarrow \text{sName}$

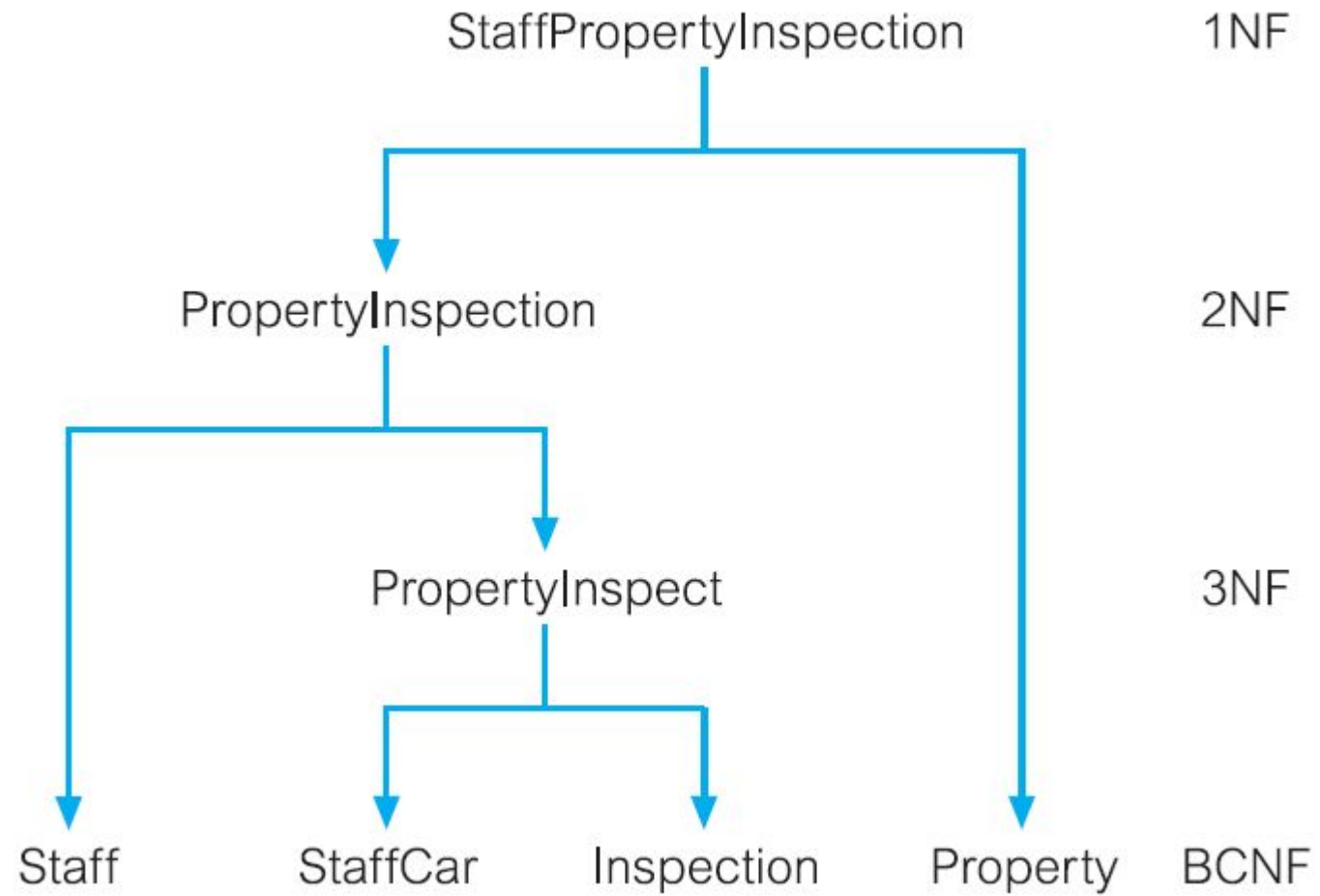
PropertyInspect Relation

fd1' $\text{propertyNo}, \text{iDate} \rightarrow \text{iTime}, \text{comments}, \text{staffNo}, \text{carReg}$

fd4 $\text{staffNo}, \text{iDate} \rightarrow \text{carReg}$

fd5' $\text{carReg}, \text{iDate}, \text{iTime} \rightarrow \text{propertyNo}, \text{comments}, \text{staffNo}$

fd6' $\text{staffNo}, \text{iDate}, \text{iTime} \rightarrow \text{propertyNo}, \text{comments}$



4. General Normal Form Definitions (For Multiple Keys) (1)

- The above definitions consider the primary key only
- The following more general definitions take into account relations with multiple candidate keys
- Any attribute involved in a candidate key is a prime attribute
- All other attributes are called non-prime attributes.

4.1 General Definition of 2NF (For Multiple Candidate Keys)

□ A relation schema R is in **second normal form (2NF)** if every non-prime attribute A in R is fully functionally dependent on *every* key of R

□ In Figure 14.12 the FD

County_name \rightarrow Tax_rate violates 2NF.

So second normalization converts LOTS into

LOTS1 (Property_id#, County_name, Lot#, Area, Price)

LOTS2 (County_name, Tax_rate)

4.2 General Definition of Third Normal Form

□ Definition:

- **Superkey** of relation schema R – a set of attributes S of R that contains a key of R
- A relation schema R is in **third normal form (3NF)** if whenever a FD $X \rightarrow A$ holds in R, then either:
 - (a) X is a superkey of R, or
 - (b) A is a prime attribute of R

□ LOTS1 relation violates 3NF because

Area \rightarrow Price ; and Area is not a superkey in LOTS1. (see Figure 14.12).

4.3 Interpreting the General Definition of Third Normal Form

- Consider the 2 conditions in the Definition of 3NF:

A relation schema R is in **third normal form (3NF)** if whenever a FD $X \rightarrow A$ holds in R , then either:

- (a) X is a superkey of R , or
- (b) A is a prime attribute of R

- Condition (a) catches two types of violations :

- one where a prime attribute functionally determines a non-prime attribute. This catches 2NF violations due to non-full functional dependencies
- second, where a non-prime attribute functionally determines a non-prime attribute. This catches 3NF violations due to a transitive dependency

4.3 Interpreting the General Definition of Third Normal Form (2)

□ **ALTERNATIVE DEFINITION** of 3NF: We can restate the definition as:

A relation schema R is in **third normal form (3NF)** if every non-prime attribute in R meets both of these conditions:

- It is fully functionally dependent on every key of R
- It is non-transitively dependent on every key of R

Note that stated this way, a relation in 3NF also meets the requirements for 2NF.

□ The condition (b) from the last slide takes care of the dependencies that “slip through” (are allowable to) 3NF but are “caught by” BCNF which we discuss next.

5. BCNF (Boyce–Codd Normal Form)

- A relation schema R is in **Boyce–Codd Normal Form (BCNF)** if whenever an **FD** $X \rightarrow A$ holds in R , then X is a **superkey** of R
- Each normal form is strictly stronger than the previous one
 - Every 2NF relation is in 1NF
 - Every 3NF relation is in 2NF
 - Every BCNF relation is in 3NF
- There exist relations that are in 3NF but not in BCNF
- Hence BCNF is considered a **stronger form of 3NF**
- The goal is to have each relation in BCNF (or 3NF)

Figure 14.14 A relation TEACH that is in 3NF but not in BCNF

Figure 14.14

A relation TEACH that is in 3NF but not BCNF.

TEACH

Student	Course	Instructor
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman
Wallace	Database	Mark
Wallace	Operating Systems	Ahamad
Wong	Database	Omiecinski
Zelaya	Database	Navathe
Narayan	Operating Systems	Ammar

Achieving the BCNF by Decomposition (1)

- Two FDs exist in the relation TEACH:
 - fd1: { student, course} \rightarrow instructor
 - fd2: instructor \rightarrow course
- {student, course} is a candidate key for this relation and that the dependencies shown follow the pattern in Figure 14.13 (b).
 - So this relation is in 3NF *but not in* BCNF
- A relation **NOT** in BCNF should be decomposed so as to meet this property, while possibly forgoing the preservation of all functional dependencies in the decomposed relations.
 - (See Algorithm 15.3)

Achieving the BCNF by Decomposition (2)

- Three possible decompositions for relation TEACH
 - D1: {student, instructor} and {student, course}
 - D2: {course, instructor } and {course, student}
 - D3: {instructor, course } and {instructor, student} ✓
- All three decompositions will lose fd1.
 - We have to settle for sacrificing the functional dependency preservation. But we cannot sacrifice the non-additivity property after decomposition.
- Out of the above three, only the 3rd decomposition will not generate spurious tuples after join.(and hence has the non-additivity property).
- A test to determine whether a binary decomposition (decomposition into two relations) is non-additive (lossless) is discussed under Property NJB on the next slide. We then show how the third decomposition above meets the property

Test for checking non-additivity of Binary Relational Decompositions

□ Testing Binary Decompositions for Lossless Join (Non-additive Join) Property

- **Binary Decomposition:** Decomposition of a relation R into two relations.
- **PROPERTY NJB (non-additive join test for binary decompositions):** A decomposition $D = \{R_1, R_2\}$ of R has the lossless join property with respect to a set of functional dependencies F on R *if and only if* either
 - The f.d. $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$ is in F^+ , or
 - The f.d. $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$ is in F^+ .

Test for checking non-additivity of Binary Relational Decompositions

If you apply the NJB test to the 3 decompositions of the TEACH relation:

□ D1 gives **Student** → Instructor or **Student** → Course, none of which is true.

□ D2 gives **Course** → Instructor or **Course** → Student, none of which is true.

□ However, in D3 we get **Instructor** → Course or **Instructor** → Student.

Since **Instructor** → Course is indeed true, the NJB property is satisfied and D3 is determined as a non-additive (good) decomposition.

General Procedure for achieving BCNF when a relation fails BCNF

Here we make use the algorithm from Chapter 15 (Algorithm 15.5):

- Let R be the relation not in BCNF, let X be a subset-of R , and let $X \rightarrow A$ be the FD that causes a violation of BCNF. Then R may be decomposed into two relations:
 - (i) $R - A$ and (ii) $X \cup A$.
- If either $R - A$ or $X \cup A$ is not in BCNF, repeat the process.
 - Note that the f.d. that violated BCNF in TEACH was $\text{Instructor} \rightarrow \text{Course}$. Hence its BCNF decomposition would be :

(TEACH – COURSE) and (Instructor \cup Course), which gives the relations: (Instructor, Student) and (Instructor, Course) that we obtained before in decomposition D3.

5. Multivalued Dependencies and Fourth Normal Form (1)

Definition:

- A **multivalued dependency (MVD)** $X \twoheadrightarrow Y$ specified on relation schema R , where X and Y are both subsets of R , specifies the following constraint on any relation state r of R : If two tuples t_1 and t_2 exist in r such that $t_1[X] = t_2[X]$, then two tuples t_3 and t_4 should also exist in r with the following properties, where we use Z to denote $(R - (X \cup Y))$:
 - $t_3[X] = t_4[X] = t_1[X] = t_2[X]$.
 - $t_3[Y] = t_1[Y]$ and $t_4[Y] = t_2[Y]$.
 - $t_3[Z] = t_2[Z]$ and $t_4[Z] = t_1[Z]$.
- An MVD $X \twoheadrightarrow Y$ in R is called a **trivial MVD** if (a) Y is a subset of X , or (b) $X \cup Y = R$.

Multivalued Dependencies and Fourth Normal Form (3)

Definition:

- A relation schema R is in **4NF** with respect to a set of dependencies F (that includes functional dependencies and multivalued dependencies) if, for every *nontrivial* multivalued dependency $X \twoheadrightarrow Y$ in F^+ , X is a superkey for R .
 - Note: F^+ is the (complete) set of all dependencies (functional or multivalued) that will hold in every relation state r of R that satisfies F . It is also called the **closure** of F .

Figure 14.15 Fourth and fifth normal forms.

(a) EMP

<u>Ename</u>	<u>Pname</u>	<u>Dname</u>
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

(c) SUPPLY

<u>Sname</u>	<u>Part_name</u>	<u>Proj_name</u>
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

(b) EMP_PROJECTS

<u>Ename</u>	<u>Pname</u>
Smith	X
Smith	Y

EMP_DEPENDENTS

<u>Ename</u>	<u>Dname</u>
Smith	John
Smith	Anna

(d) R_1

<u>Sname</u>	<u>Part_name</u>
Smith	Bolt
Smith	Nut
Adamsky	Bolt
Walton	Nut
Adamsky	Nail

R_2

<u>Sname</u>	<u>Proj_name</u>
Smith	ProjX
Smith	ProjY
Adamsky	ProjY
Walton	ProjZ
Adamsky	ProjX

R_3

<u>Part_name</u>	<u>Proj_name</u>
Bolt	ProjX
Nut	ProjY
Bolt	ProjY
Nut	ProjZ
Nail	ProjX

Figure 14.15

Fourth and fifth normal forms. (a) The EMP relation with two MVDs: $Ename \twoheadrightarrow Pname$ and $Ename \twoheadrightarrow Dname$. (b) Decomposing the EMP relation into two 4NF relations EMP_PROJECTS and EMP_DEPENDENTS. (c) The relation SUPPLY with no MVDs is in 4NF but not in 5NF if it has the JD(R_1, R_2, R_3). (d) Decomposing the relation SUPPLY into the 5NF relations R_1, R_2, R_3 .

6. Join Dependencies and Fifth Normal Form (1)

Definition:

- A **join dependency (JD)**, denoted by $JD(R_1, R_2, \dots, R_n)$, specified on relation schema R , specifies a constraint on the states r of R .
 - The constraint states that every legal state r of R should have a non-additive join decomposition into R_1, R_2, \dots, R_n ; that is, for every such r we have
 - $$* (\pi_{R_1}(r), \pi_{R_2}(r), \dots, \pi_{R_n}(r)) = r$$

Note: an MVD is a special case of a JD where $n = 2$.

- A join dependency $JD(R_1, R_2, \dots, R_n)$, specified on relation schema R , is a **trivial JD** if one of the relation schemas R_i in $JD(R_1, R_2, \dots, R_n)$ is equal to R .

Join Dependencies and Fifth Normal Form (2)

Definition:

- A relation schema R is in **fifth normal form (5NF)** (or **Project-Join Normal Form (PJNF)**) with respect to a set F of functional, multivalued, and join dependencies if,
 - for every nontrivial join dependency $JD(R_1, R_2, \dots, R_n)$ in F^+ (that is, implied by F),
 - every R_i is a superkey of R .
- Discovering join dependencies in practical databases with hundreds of relations is next to impossible. Therefore, 5NF is rarely used in practice.