

Recaps of Relational Algebra

- Relational Algebra consists of several groups of operations
 - Unary Relational Operations
 - SELECT (symbol: σ (sigma))
 - PROJECT (symbol: π (pi))
 - RENAME (symbol: ρ (rho))
 - Relational Algebra Operations From Set Theory
 - UNION (\cup), INTERSECTION (\cap), DIFFERENCE (or MINUS, $-$)
 - CARTESIAN PRODUCT (\times)
 - Binary Relational Operations
 - JOIN (several variations of JOIN exist)
 - DIVISION
 - Additional Relational Operations
 - OUTER JOINS, OUTER UNION
 - AGGREGATE FUNCTIONS (These compute summary of information: for example, SUM, COUNT, AVG, MIN, MAX)

CHAPTER 14

Basics of Functional Dependencies and Normalization for Relational Databases

Chapter Outline

- 1 Informal Design Guidelines for Relational Databases
 - 1.1 Semantics of the Relation Attributes
 - 1.2 Redundant Information in Tuples and Update Anomalies
 - 1.3 Null Values in Tuples
 - 1.4 Spurious Tuples
- 2 Functional Dependencies (FDs)
 - 2.1 Definition of Functional Dependency

Chapter Outline

- 3 Normal Forms Based on Primary Keys
 - 3.1 Normalization of Relations
 - 3.2 Practical Use of Normal Forms
 - 3.3 Definitions of Keys and Attributes Participating in Keys
 - 3.4 First Normal Form
 - 3.5 Second Normal Form
 - 3.6 Third Normal Form
- 4 General Normal Form Definitions for 2NF and 3NF (For Multiple Candidate Keys)

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)

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.*

Figure 14.1 A simplified COMPANY relational database schema

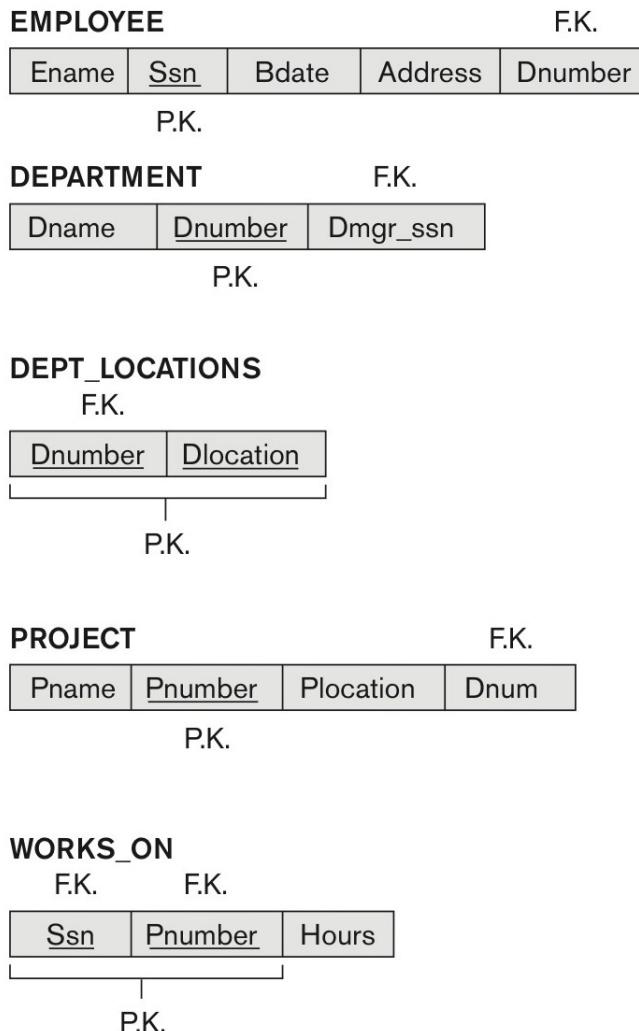


Figure 14.1 A simplified COMPANY relational database schema.

Figure 14.3 Two relation schemas

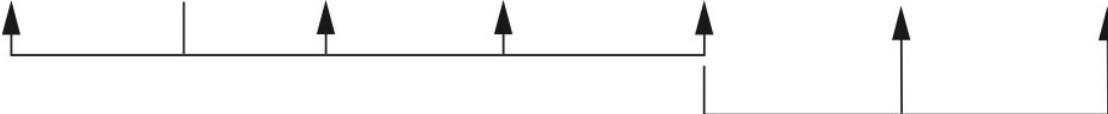
Figure 14.3

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

(a)

EMP_DEPT

Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn



(b)

EMP_PROJ

<u>Ssn</u>	Pnumber	Hours	Ename	Pname	Plocation
FD1					
FD2					
FD3					

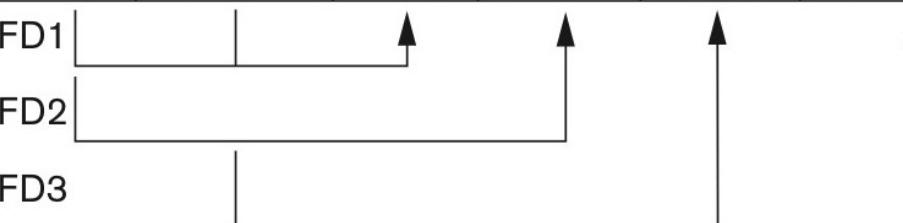


Figure 14.4 Sample states for EMP_DEPT and EMP_PROJ

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.

EMP_DEPT							Redundancy	
EMP_PROJ		Redundancy			Redundancy			
Ssn	Pnumber	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			

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

Figure 14.4 Sample states for EMP_DEPT and EMP_PROJ

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.

EMP_PROJ				Redundancy	
Ssn	Pnumber	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

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.

Figure 14.4 Sample states for EMP_DEPT and EMP_PROJ

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.

EMP_PROJ				Redundancy	
Ssn	Pnumber	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

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 he/she is assigned to a project.

Figure 14.4 Sample states for EMP_DEPT and EMP_PROJ

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.

EMP_PROJ				Redundancy	
Ssn	Pnumber	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

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.

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
 - Attribute value unknown (may exist)
 - Value known to exist, but unavailable

1.4 Generation of Spurious Tuples – avoid at any cost

- 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).

Figure 14.5

(a)

EMP_LOCS

Ename	Plocation
-------	-----------

P.K.

EMP_PROJ1

Ssn	Pnumber	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

Figure 14.6

	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.

*

*

*

2. Functional Dependencies

- Functional dependencies (FDs)
 - Are used to specify *formal measures* of the "goodness" of relational designs
 - Used to define **normal forms** for relation schemas
 - 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

- $X \rightarrow Y$ holds if whenever two tuples have the same value for X , they *must have the same value for Y*
 - For any two tuples t_1 and t_2 in any relation instance $r(R)$: If $t_1[X]=t_2[X]$, *then* $t_1[Y]=t_2[Y]$
- $X \rightarrow Y$ 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

Figure 14.3 Two relation schemas suffering from update anomalies

Figure 14.3

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

(a)

EMP_DEPT

Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn

The diagram shows the primary key of the EMP_DEPT relation, which is the column labeled 'Ssn' (underlined), pointing to all other columns: Ename, Bdate, Address, Dnumber, Dname, and Dmgr_ssn. This indicates that any update to the primary key 'Ssn' will affect all other attributes in the row.

(b)

EMP_PROJ

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

Figure 14.3 Two relation schemas suffering from update anomalies

Figure 14.3

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

(a)

EMP_DEPT

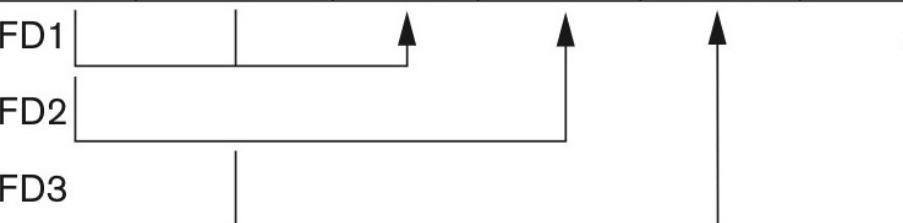
Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn



(b)

EMP_PROJ

<u>Ssn</u>	Pnumber	Hours	Ename	Pname	Plocation
FD1					
FD2					
FD3					



Examples of FD constraints (1)

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

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 $t1[K]=t2[K]$)

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: $\text{Text} \rightarrow \text{Course}$ may exist. However, the FDs $\text{Teacher} \rightarrow \text{Course}$, $\text{Teacher} \rightarrow \text{Text}$ and $\text{Course} \rightarrow \text{Text}$ are ruled out.

TEACH

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

3 Normal Forms Based on Primary Keys

- 3.1 Normalization of Relations
- 3.2 Practical Use of Normal Forms
- 3.3 Definitions of Keys and Attributes
Participating in Keys
- 3.4 First Normal Form
- 3.5 Second Normal Form
- 3.6 Third Normal Form

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; see Chapter 15)

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
 - **nested relations**; attributes whose values for an *individual tuple* are non-atomic
- Considered to be part of the definition of a relation

Figure 14.9 Normalization into 1NF

(a)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations



(b)

DEPARTMENT

Dname	Dnumber	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.

Figure 14.9 Normalization into 1NF – Best Solution

(a)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations



(b)

DEPARTMENT

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

DEPARTMENT

Dname	Dnumber	Dmgr_ssn
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	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.

Figure 14.9 Normalization into 1NF - Solution 2

(a)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations



(b)

DEPARTMENT

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

(c)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	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.

Figure 14.10 Normalizing nested relations into 1NF

(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 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.

Figure 14.10 Normalizing nested relations into 1NF

(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
		4	20.0
453453453	English, Joyce A.	5	20.0
		6	20.0
333445555	Wong, Franklin T.	7	10.0
		8	10.0
999887777	Zelaya, Alicia J.	9	10.0
		10	10.0
987987987	Jabbar, Ahmad V.	11	35.0
		12	5.0
987654321	Wallace, Jennifer S.	13	20.0
		14	15.0
888665555	Borg, James E.	15	NULL

(c)

EMP_PROJ1

Ssn	Ename
-----	-------

EMP_PROJ2

Ssn	Pnumber	Hours
-----	---------	-------

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
- Are the following examples full FD?
 - $\{\text{SSN, PNUMBER}\} \rightarrow \text{HOURS}$
 - $\{\text{SSN, PNUMBER}\} \rightarrow \text{ENAME}$

Figure 14.11 Normalizing into 2NF and 3NF

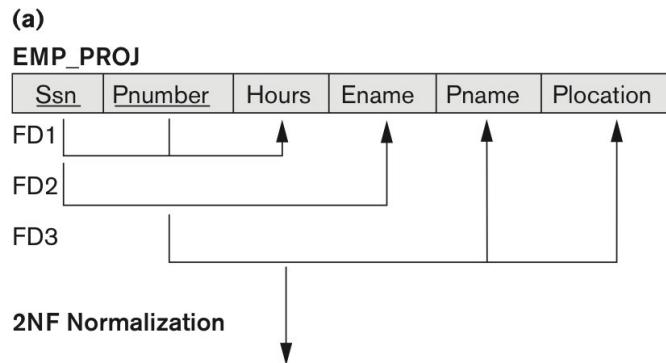


Figure 14.11
Normalizing into 2NF and 3NF.
(a) Normalizing EMP_PROJ into
2NF relations.

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

Figure 14.11 Normalizing into 2NF and 3NF

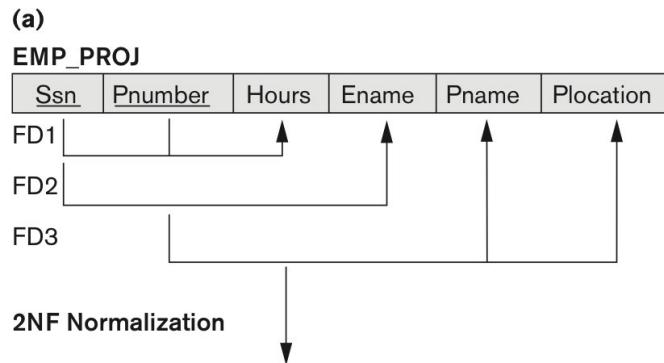


Figure 14.11
Normalizing into 2NF and 3NF.
(a) Normalizing EMP_PROJ into
2NF relations.

Figure 14.11 Normalizing into 2NF and 3NF

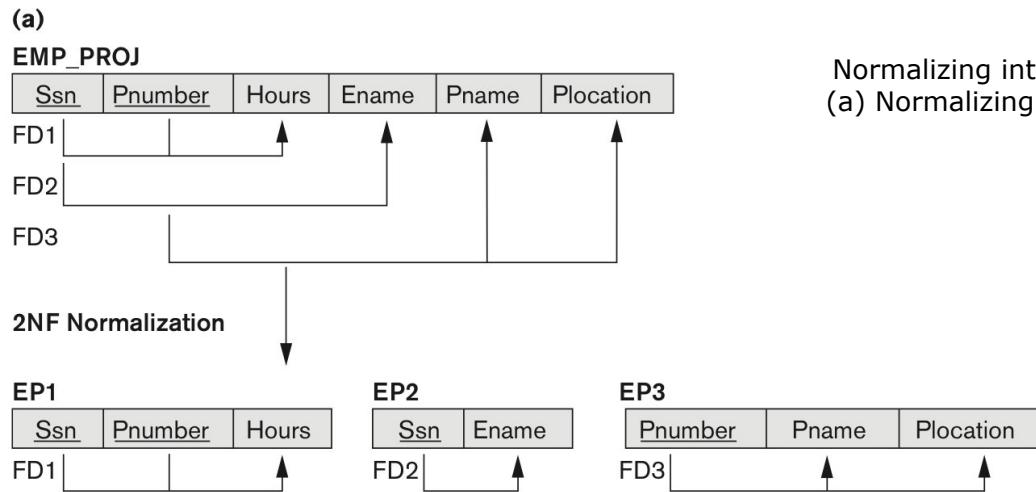


Figure 14.11
Normalizing into 2NF and 3NF.
(a) Normalizing **EMP_PROJ** into
2NF relations.

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
 - SSN \rightarrow ENAME

Figure 14.11 Normalizing into 2NF and 3NF

(b)

EMP_DEPT

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

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

Figure 14.11

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

(b)

EMP_DEPT

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

Figure 14.11 Normalizing into 2NF and 3NF

Figure 14.11

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

(b)

EMP_DEPT

Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn

The diagram illustrates the decomposition of the primary key for the EMP_DEPT relation. The primary key consists of the columns Ssn and Dnumber. Arrows point from the Ename, Bdate, Address, and Dmgr_ssn columns to a vertical line, indicating they are non-key attributes. Arrows also point from the Ssn and Dnumber columns to another vertical line, indicating they are part of the primary key.

Figure 14.11 Normalizing into 2NF and 3NF

Figure 14.11

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

(b)

EMP_DEPT

Ename	<u>Ssn</u>	Bdate	Address	Dnumber	Dname	Dmgr_ssn

```
graph TD; Ename --> Ename; Ssn --> Ssn; Bdate --> Bdate; Address --> Address; Dmgr_ssn --> Dmgr_ssn; Dnumber[Dnumber] --> Dnumber; Dname[Dname] --> Dname; Dmgr_ssn[Dmgr_ssn] --> Dmgr_ssn; Dnumber --> Dname; Dmgr_ssn --> Dname; Dmgr_ssn --> Dmgr_ssn; Ename --- Ssn --- Bdate --- Address --- Dmgr_ssn --- Dnumber --- Dname --- Dmgr_ssn; Dnumber --- Dname; Dmgr_ssn --- Dname; Dmgr_ssn --- Dmgr_ssn; Dmgr_ssn --> Dmgr_ssn;
```

3NF Normalization

ED1

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

```
graph TD; Ename --> Ename; Ssn --> Ssn; Bdate --> Bdate; Address --> Address; Dnumber --> Dnumber; Ename --- Ssn --- Bdate --- Address --- Dnumber;
```

ED2

Dnumber	Dname	Dmgr_ssn

```
graph TD; Dnumber --> Dnumber; Dname --> Dname; Dmgr_ssn --> Dmgr_ssn; Dnumber --- Dname --- Dmgr_ssn;
```

Normal Forms Defined Informally

- 1st normal form
 - All attributes depend on **the key**
 - ***Atomic attributes***
- 2nd normal form
 - All attributes depend on **the whole key**
 - ***No partial dependencies***
- 3rd normal form
 - All attributes depend on **nothing but the key**
 - ***No transitive dependencies***

Relation

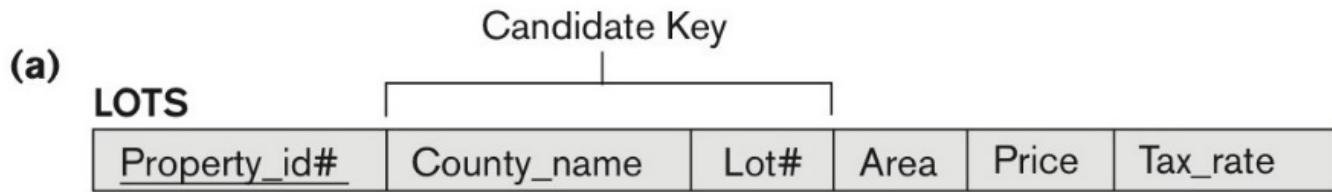
- ✖ A relation is a named, two-dimensional table of data.
- ✖ A table consists of rows (records) and columns (attribute or field).
- ✖ Requirements for a table to qualify as a relation:
 - + It must have a unique name.
 - + Every attribute value must be atomic (not multivalued, not composite).
 - + Every row must be unique (can't have two rows with exactly the same values for all their fields).
 - + Attributes (columns) in tables must have unique names.
 - + The order of the columns must be irrelevant.
 - + The order of the rows must be irrelevant.

NOTE: All *relations* are in *1st Normal form*.

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*.

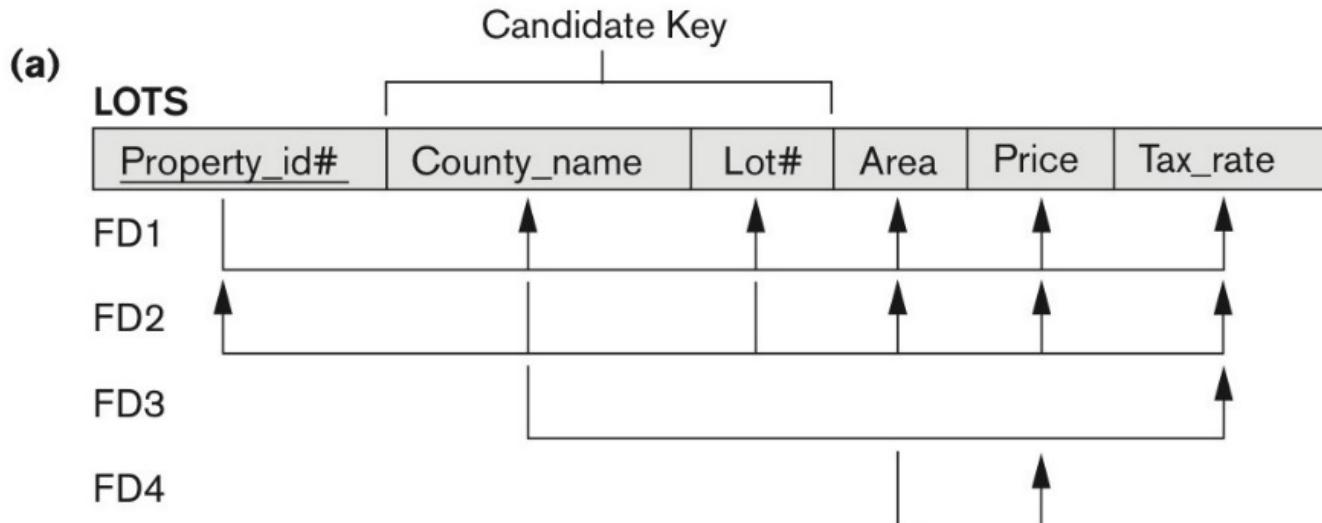
Figure 14.12 Normalization into 2NF and 3NF



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
- **Is the relation in Figure 14.12 in 2NF?**

Figure 14.12 Normalization into 2NF and 3NF



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
 $\text{County_name} \rightarrow \text{Tax_rate}$ violates 2NF.

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

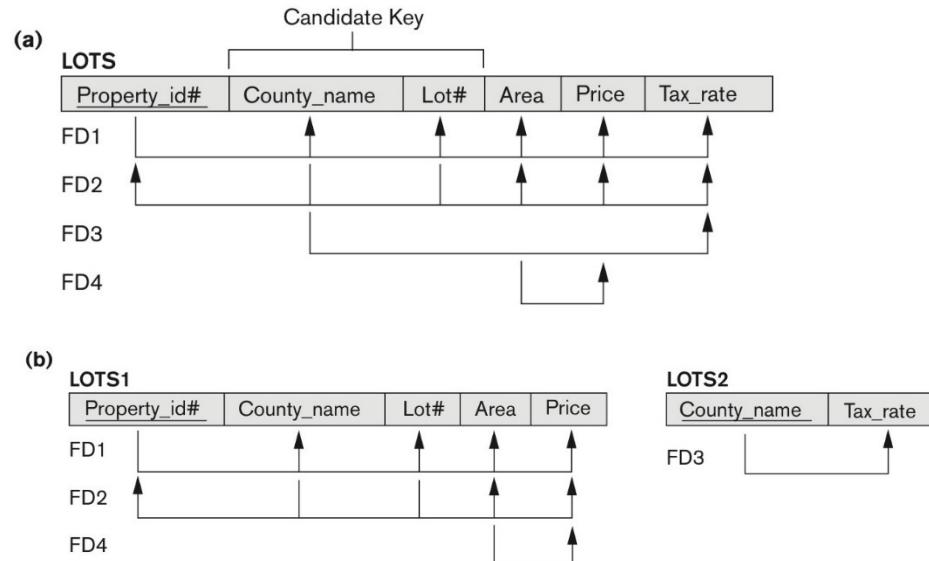
- A relation schema R is in **second normal form (2NF)** if every non-prime attribute A in R is fully functionally dependent on *every key* of R
- In Figure 14.12 the FD
 $\text{County_name} \rightarrow \text{Tax_rate}$ violates 2NF.

So second normalization converts LOTS into
LOTS1 (Property_id#, County_name, Lot#, Area, Price)
LOTS2 (County_name, Tax_rate)

Figure 14.12 Normalization into 2NF and 3NF

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.



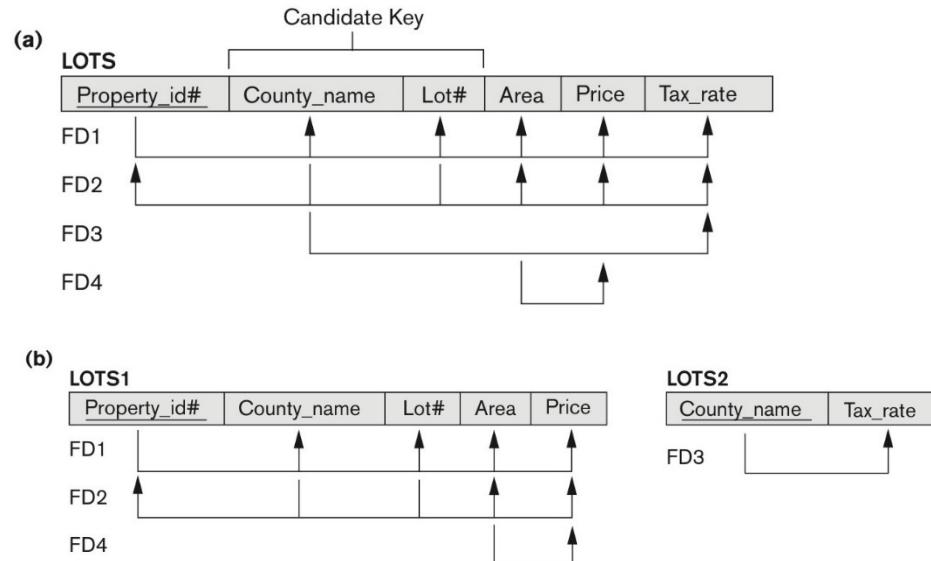
4.2 General Definition of Third Normal Form

- 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

Figure 14.12 Normalization into 2NF and 3NF

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.



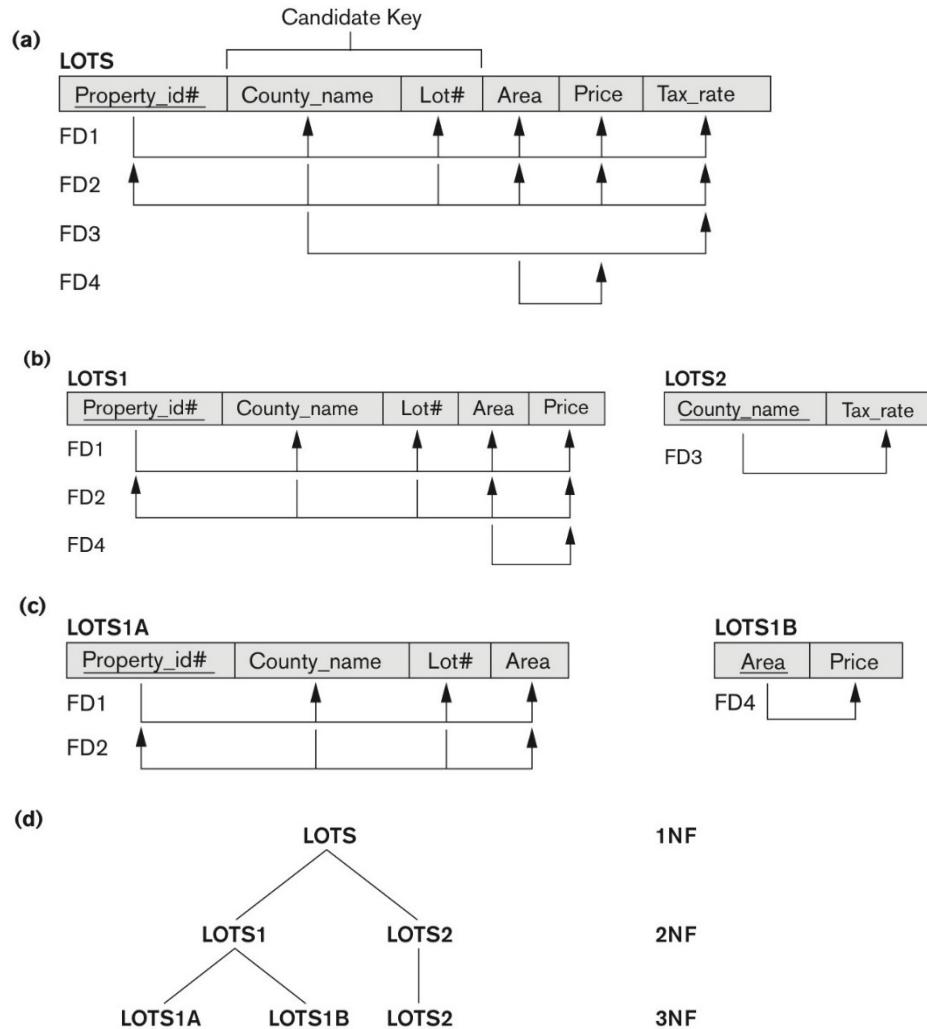
4.2 General Definition of Third Normal Form

- 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
 $\text{Area} \rightarrow \text{Price}$; and Area is not a superkey in LOTS1. (see Figure 14.12).

Figure 14.12 Normalization into 2NF and 3NF

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.



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.

Chapter Summary

- Informal Design Guidelines for Relational Databases
- Functional Dependencies (FDs)
- Normal Forms (1NF, 2NF, 3NF) Based on Primary Keys
- General Normal Form Definitions of 2NF and 3NF (For Multiple Keys)
- ***Reading for next week: Chapter 3!***