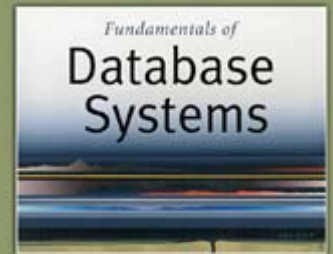


5th Edition

Elmasri / Navathe

# Chapter 10

## Functional Dependencies and Normalization for Relational Databases



5th Edition

Elmasri / Navathe



# Chapter Outline

- Informal Design Guidelines for Relational Databases
- Functional Dependencies (FDs)
- Normalization of Relations and Different Normal Forms

# 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)

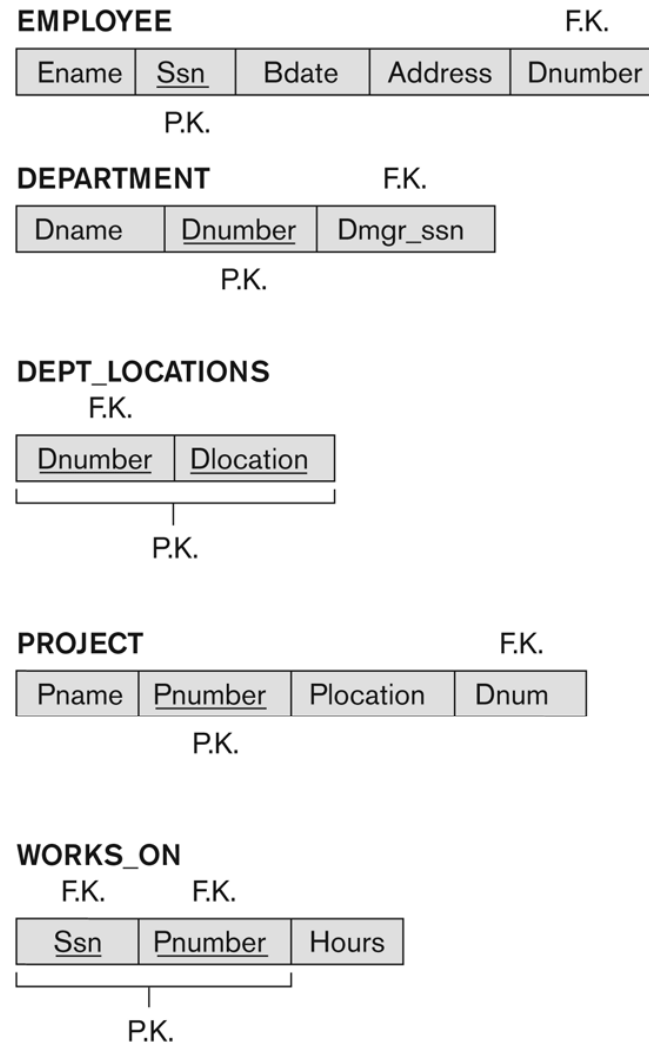
- We first discuss informal guidelines for good relational design
- Then we discuss formal concepts of functional dependencies and normal forms
  - - 1NF (First Normal Form)
  - - 2NF (Second Normal Form)
  - - 3NF (Third Normal Form)
  - - BCNF (Boyce-Codd Normal Form)



# Semantics of the Relation Attributes

- GUIDELINE 1: Informally, each tuple in a relation should represent one entity or relationship instance.
- 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.*

# A simplified COMPANY relational database schema



**Figure 10.1**  
A simplified COMPANY  
relational database  
schema.

# Redundant Information in Tuples and Update Anomalies

- Redundant Information causes:
  - storage wastage
  - problems with update anomalies
    - Insertion anomalies
    - Deletion anomalies
    - Modification anomalies



# Two relation schemas suffering from storage wastage and update anomalies

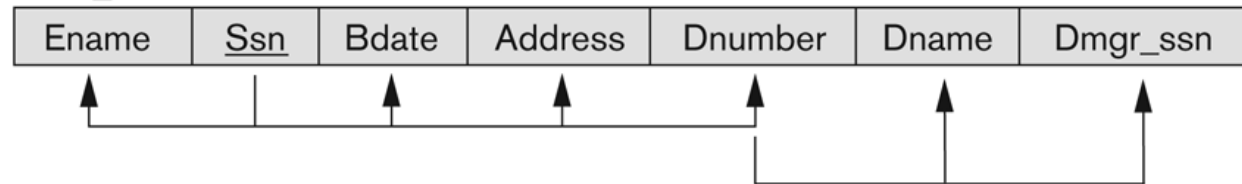
**Figure 10.3**

Two relation schemas suffering from update anomalies.

(a) EMP\_DEPT and  
(b) EMP\_PROJ.

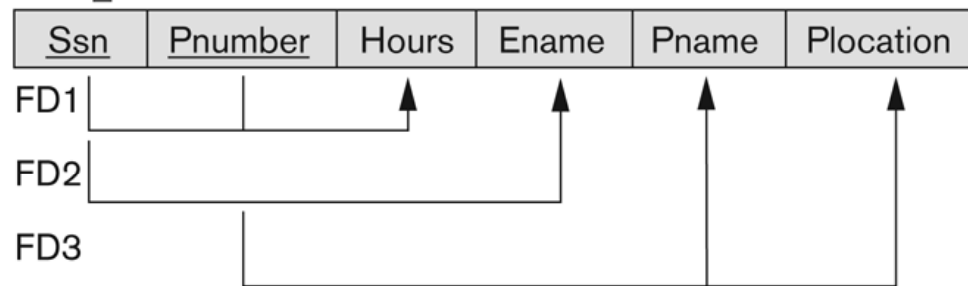
(a)

**EMP\_DEPT**



(b)

**EMP\_PROJ**



# Example States for EMP\_DEPT and EMP\_PROJ

**Figure 10.4**  
Example states for EMP\_DEPT and EMP\_PROJ resulting from applying NATURAL JOIN to the relations in Figure 10.2. These may be stored as base relations for performance reasons.

EMP_DEPT							Redundancy	
Ename	Ssn	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		

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				

# 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 he/she is assigned to a project.



# EXAMPLE OF AN 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 to 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.

# 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)
- Nulls are problematic in joins and aggregate functions
- Many interpretations for nulls:
  - Attribute not applicable or invalid
  - Attribute value unknown (may exist)
  - Value known to exist, but unavailable

# Spurious Tuples

- Bad designs for a relational database may result in erroneous results for certain JOIN operations
- GUIDELINE 4:
  - Design relation schemas so that they can be joined with equality conditions on attributes that are (primary key, foreign key) pairs in a way that guarantees that no spurious tuples are generated.
  - No spurious tuples should be generated by doing a natural-join of any relations.



(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.	Surgarland
Narayan, Ramesh K.	Houston
English, Joyce A.	Bellaire
English, Joyce A.	Surgarland
Wong, Franklin T.	Surgarland
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 10.5**

Particularly poor design for the EMP\_PROJ relation of Figure 10.3(b).  
(a) The two relation schemas EMP\_LOCS and EMP\_PROJ1. (b) The result of projecting the extension of EMP\_PROJ from Figure 10.4 onto the relations EMP\_LOCS and EMP\_PROJ1.

	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 10.6**

Result of applying NATURAL JOIN to the tuples above the dotted lines in EMP\_PROJ1 and EMP\_LOCS of Figure 10.5. Generated spurious tuples are marked by asterisks.

# Functional Dependencies (1)

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

# Functional Dependencies (2)

- $X \rightarrow Y$  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:  $\rightarrow$ ).
- FDs are derived from the real-world constraints on the attributes
- If  $K$  is a key of  $R$ , then  $K$  functionally determines all attributes in  $R$ 
  - since we never have two distinct tuples with  $t1[K]=t2[K]$



# Examples of FD constraints

- 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
  - $\{SSN, PNUMBER\} \rightarrow HOURS$

# Inference Rules for FDs

- Given a set of FDs  $F$ , we can **infer** additional FDs that hold whenever the FDs in  $F$  hold
- Armstrong's inference rules:
  - IR1. (**Reflexive**) If  $Y \text{ subset-of } X$ , then  $X \rightarrow Y$
  - IR2. (**Augmentation**) If  $X \rightarrow Y$ , then  $XZ \rightarrow YZ$ 
    - (Notation:  $XZ$  stands for  $X \cup Z$ )
  - IR3. (**Transitive**) If  $X \rightarrow Y$  and  $Y \rightarrow Z$ , then  $X \rightarrow Z$
- Some additional inference rules that are useful:
  - **Decomposition:** If  $X \rightarrow YZ$ , then  $X \rightarrow Y$  and  $X \rightarrow Z$
  - **Union:** If  $X \rightarrow Y$  and  $X \rightarrow Z$ , then  $X \rightarrow YZ$
  - **Pseudotransitivity:** If  $X \rightarrow Y$  and  $WY \rightarrow Z$ , then  $WX \rightarrow Z$
- The last three inference rules, as well as any other inference rules, can be deduced from IR1, IR2, and IR3 (completeness property)

# Normalization of Relations

- **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
  - Normal form of a relation is the highest NF condition that it meets, and hence indicates the degree to which it has been normalized
- **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

# 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.

# 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 relation

# Normalization into 1NF

(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 10.8**

Normalization into 1NF.

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

# Normalization of nested relations into 1NF

(a)

EMP\_PROJ

Ssn	Ename	Projs	
		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, AliciaJ.	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

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

EMP\_PROJ2

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

**Figure 10.9**

Normalizing nested relations into 1NF. (a) Schema of the EMP\_PROJ relation with a *nested relation* attribute PROJS. (b) Example 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.



# 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

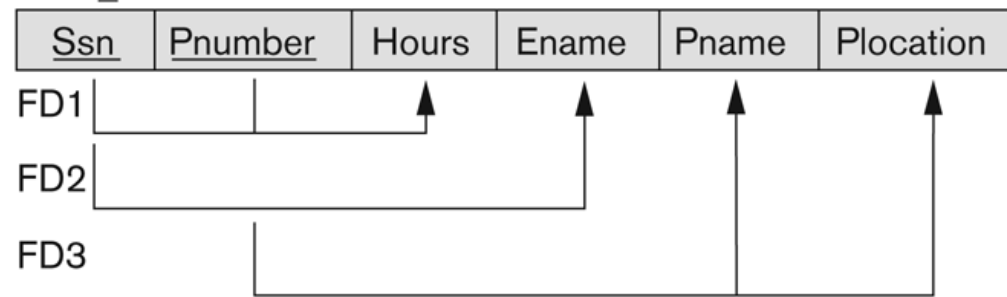
## Second Normal Form (2)

- 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

# Normalizing into 2NF

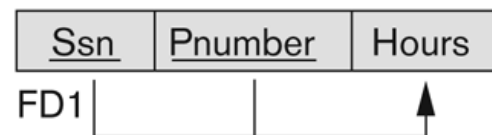
(a)

**EMP\_PROJ**

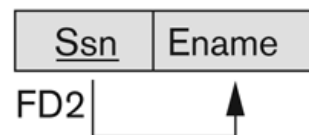


**2NF Normalization**

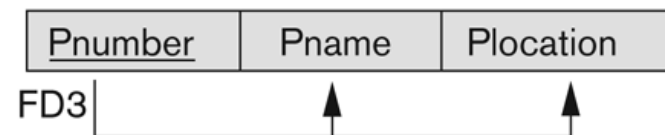
**EP1**



**EP2**



**EP3**



**Figure 10.10**

Normalizing into 2NF and 3NF.  
(a) Normalizing EMP\_PROJ into 2NF relations. (b) Normalizing EMP\_DEPT into 3NF relations.

# 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.

# Normalizing into 3NF

(b)

**EMP\_DEPT**

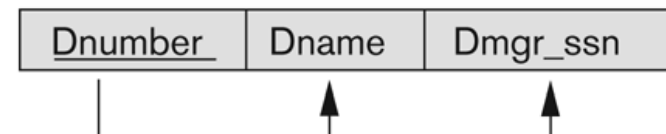


3NF Normalization

**ED1**



**ED2**



# Normal Forms Defined Informally

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

# General Normal Form Definitions (1)

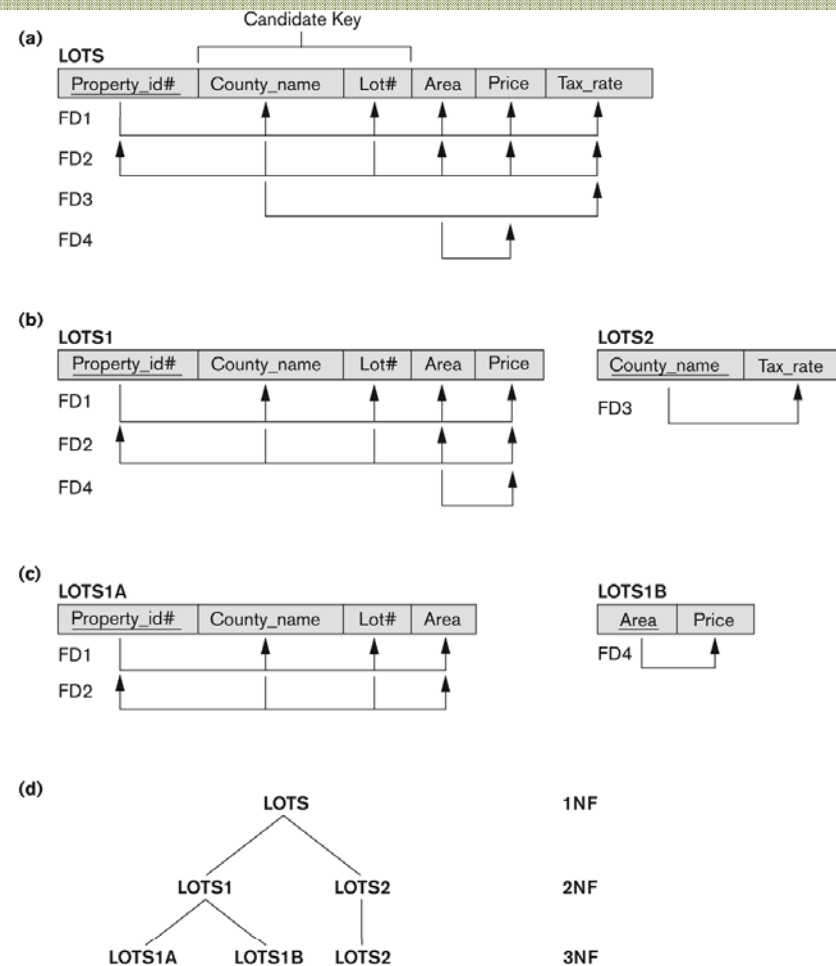
- The above definitions consider the primary key only
- The following more general definitions take into account relations with multiple candidate keys
- A relation schema  $R$  is in **second normal form (2NF)** if every non-prime attribute  $A$  in  $R$  is fully functionally dependent on *every* key of  $R$



# General Normal Form Definitions (2)

- 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
- NOTE: Boyce-Codd normal form disallows condition (b) above

# Normalization into General 2NF and 3NF



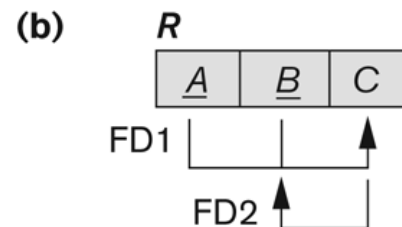
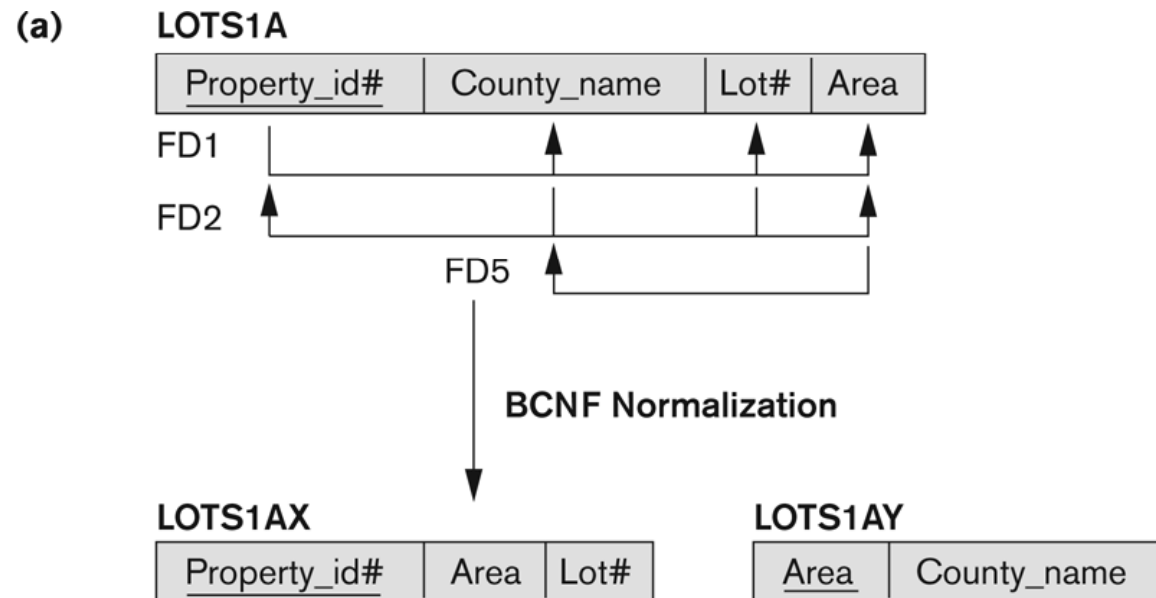
**Figure 10.11**

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.

# 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
- The goal is to have each relation in BCNF (or 3NF)

# Boyce-Codd normal form



**Figure 10.12**

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.



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

fd1: { student, course} -> instructor  
fd2: instructor -> course

**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

**Figure 10.13**  
A relation TEACH that  
is in 3NF but not  
BCNF.

# Achieving the BCNF by Decomposition

- Two FDs exist in the relation TEACH:
  - fd1: { student, course } -> instructor
  - fd2: instructor -> course
- {student, course} is a candidate key for this relation and that the dependencies shown follow the pattern in Figure 10.12 (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.
  - {instructor, course } and {instructor, student}