



សូមអ្នកចិត្ត

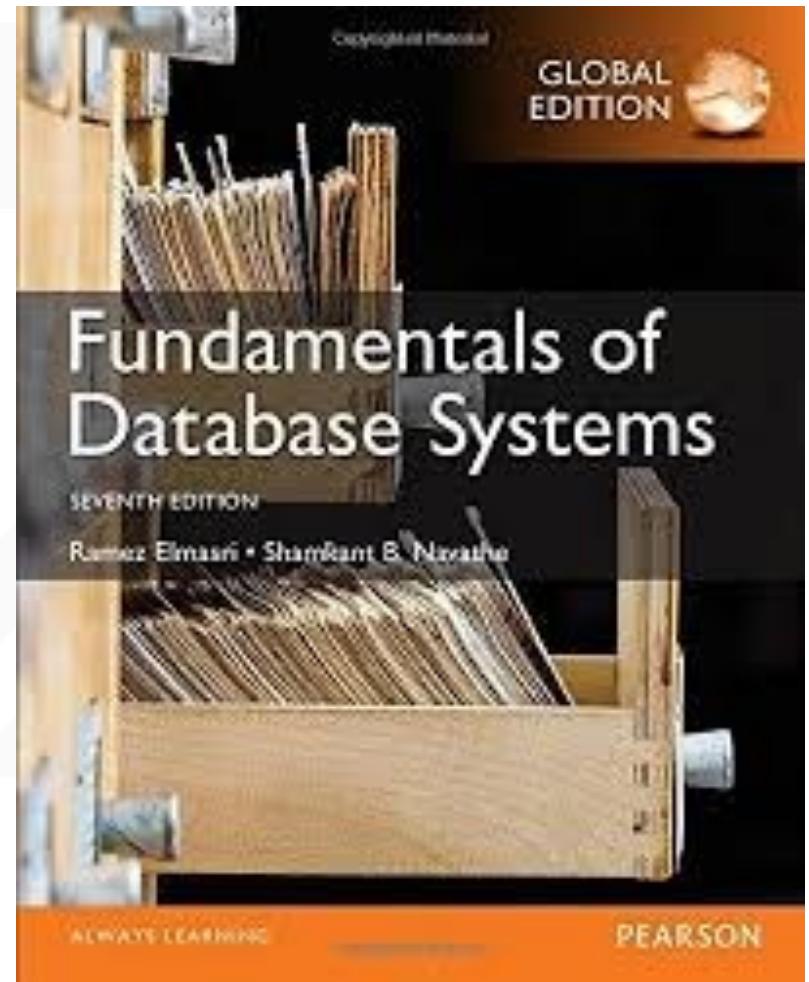
# Database Systems

Program in Computer Engineering  
Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang

# Text

- Ramez Elmasri and Shamkant B. Navathe.  
“Fundamentals of Database Systems”  
7<sup>th</sup> Edition., Pearson, 2017



# Chapter 14

## Basics of Functional Dependencies and Normalization for Relational Database

# Outline

1. Informal Design Guidelines for Relational Databases
2. Functional Dependencies (FDs)
3. Normal Forms Based on Primary Keys
4. General Normal Form Definitions for 2NF and 3NF (For Multiple Candidate Keys)  
*(To be continued)*
5. BCNF (Boyce-Codd Normal Form)
6. Multivalued Dependency and Fourth Normal Form
7. Join Dependencies and Fifth Normal Form

# 1. Informal Design Guidelines for Relational Databases

- 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?

ທີ່ໄດ້ເຮັດວຽກ → ແຕກມສໍາຜົນຜະລາຍດີ

ແລະ attr ທີ່ເກີ່າງຫຼື່ນຈິບ

ແລະ ຕາມສ່ວນຈຳນວນ ອົດລາຍ

↓  
logical

schema (logical  
view)

↓

ຕາມໄລຍະຂອງ

ລົງລູກເກີ່າງຂອງ (base  
relation)

↓

ຂອງລົງລູກທີ່ອະນາຍາມໄດ້ແພຸ່ນ?

ສິ່ງແນວ? ຂຶ້ນ?

- Relational database design ultimately produces a set of relations.
- The implicit goals of the design activity are *information preservation* and *minimum redundancy*.

↓  
(2) မြန်မာစွဲနှင့် ပုဂ္ဂနိုင်ရေး လတေသန  
ရုံး

insert  
update  
delete

①  
(က) ဆုတေသနပုဂ္ဂနိုင် (ပို့ဆောင်)  
လုပ်, atomic  
နောက်ခံပေါ်စွဲများ ပေါ်လာဖို့  
ယူဝါယာ

- 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) 3,5 NF

good design

# 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	Ssn	Bdate	Address	Dnumber	

P.K.

DEPARTMENT			F.K.
Dname	Dnumber	Dmgr_ssn	

P.K.

DEPT_LOCATIONS		F.K.
Dnumber	Dlocation	

P.K.

PROJECT				F.K.
Pname	Pnumber	Plocation	Dnum	

P.K.

WORKS_ON		
F.K.	F.K.	
Ssn	Pnumber	Hours

P.K.

**Figure 14.1** A simplified COMPANY relational database schema.

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

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

WORKS_ON			
Ssn	Pnumber	Hours	
123456789	1	32.5	
123456789	2	7.5	
666884444	3	40.0	
453453453	1	20.0	
453453453	2	20.0	
333445555	2	10.0	
333445555	3	10.0	
333445555	10	10.0	
333445555	20	10.0	
999887777	30	30.0	
999887777	10	10.0	
987987987	10	35.0	
987987987	30	5.0	
987654321	30	20.0	
987654321	20	15.0	
888665555	20	Null	

PROJECT			
Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

# 1.2 Redundant Information in Tuples and Update Anomalies

- Information is stored redundantly

- Wastes storage ↗  
မျက်လျှော်စွဲ
  - Causes problems with update anomalies
    - Insertion anomalies
    - Deletion anomalies
    - Modification anomaliesမျှေးဆုံး

# 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



各项目名 → 一个tuple 表示

# 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.

24/2015 11:00 8pm - 2

# 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.

ลบข้อมูลทั้งหมด

(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 14.3**

Two relation schemas suffering from update anomalies. (a) EMP\_DEPT and (b) EMP\_PROJ.

**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

**EMP\_PROJ**

<u>Ssn</u>	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

**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:

๑ ចំណាំទាមទារ ឱ្យបានការពារ និងការលើកឡើង និងការសម្រេចការណ៍ ក្នុងការប្រើប្រាស់បច្ចេកទេស

↑  
if much data

- 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:**

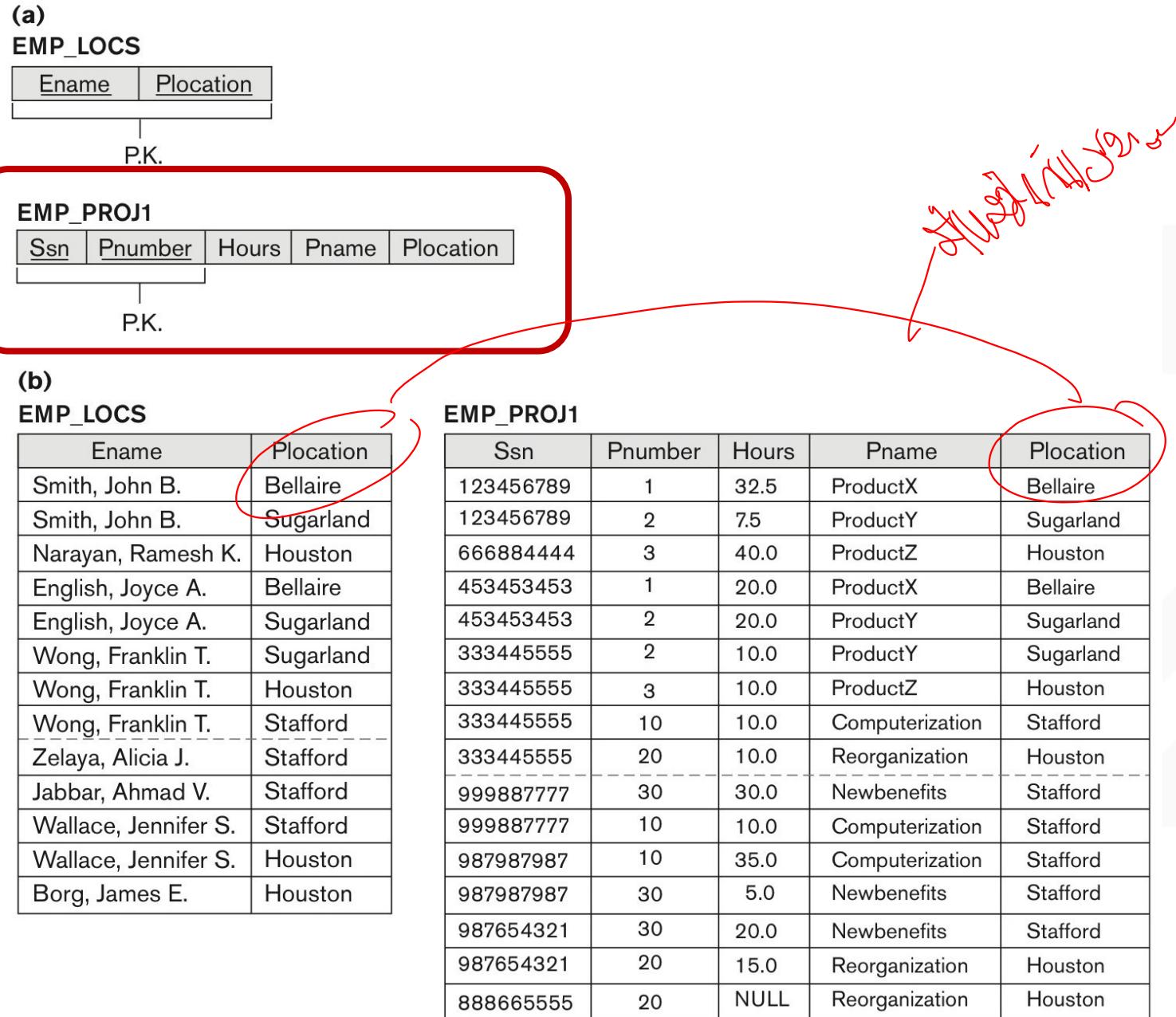
- Relations should be designed such that their tuples will have as few NULL values as possible
- Attributes that are NULL frequently could be placed in separate relations (with the primary key)
- Reasons for nulls:
  - Attribute not applicable or invalid
  - Attribute value unknown (may exist)
  - Value known to exist, but unavailable

冗長な NULL  
冗長な → ハブテーブル

# 1.4 Generation of Spurious Tuples – avoid at any cost

Join លេខទី២

- Bad designs for a relational database may result in erroneous results for certain JOIN operations



**Figure 14.5** Particularly poor design for the EMP\_PROJ relation in Figure 14.3(b). (a) The two relation schemas EMP\_LOCS and EMP\_PROJ1. (b) The result of projecting the extension of EMP\_PROJ from Figure 14.4 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	English, Joyce A.
*	123456789	2	7.5	ProductY	Sugarland
*	123456789	2	7.5	ProductY	Sugarland
*	123456789	2	7.5	ProductY	Wong, Franklin T.
666884444	3	40.0	ProductZ	Houston	Narayan, Ramesh K.
*	666884444	3	40.0	ProductZ	Houston
*	453453453	1	20.0	ProductX	Bellaire
453453453	1	20.0	ProductX	Bellaire	English, Joyce A.
*	453453453	2	20.0	ProductY	Sugarland
453453453	2	20.0	ProductY	Sugarland	Smith, John B.
*	453453453	2	20.0	ProductY	English, Joyce A.
*	453453453	2	20.0	ProductY	Sugarland
*	333445555	2	10.0	ProductY	Sugarland
*	333445555	2	10.0	ProductY	Smith, John B.
333445555	2	10.0	ProductY	Sugarland	English, Joyce A.
*	333445555	2	10.0	ProductY	Wong, Franklin T.
*	333445555	3	10.0	ProductZ	Houston
333445555	3	10.0	ProductZ	Houston	Narayan, Ramesh K.
*	333445555	10	10.0	Computerization	Stafford
*	333445555	20	10.0	Reorganization	Houston
333445555	20	10.0	Reorganization	Houston	Narayan, Ramesh K.
*					
*					
*					

QWERTYUIOP  
\* = 1 hour

**Figure 14.6** Result of applying NATURAL JOIN to the tuples in EMP\_PROJ1 and EMP\_LOCS of Figure 14.5 just for employee with Ssn = "123456789". Generated spurious tuples are marked by asterisks.

- The "lossless join" property is used to guarantee meaningful results for join operations
- **GUIDELINE 4:** *→ ចងារ 4 នៃការបង្កើតរបៀបរាយការពិនិត្យ ដែលត្រូវបានសម្រេចឡើង → គិតជាការណ៍។*
  - 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

- 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

- $X \rightarrow Y$  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

# Examples of FD constraints

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

- 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

TEACH

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

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.

# 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

# 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

- **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

- **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

- 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

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

# Figure 14.9 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 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)

		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.	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	
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.

Figure 14.10

# 3.5 Second Normal Form

- 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

- 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**

Ssn	Pnumber	Hours	Ename	Pname	Plocation
FD1			↑		
FD2				↑	
FD3					↑

2NF Normalization

**EP1**

Ssn	Pnumber	Hours
FD1		↑

**EP2**

Ssn	Ename
FD2	↑

**EP3**

Pnumber	Pname	Plocation
FD3		↑

# Third Normal Form

- **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

- 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.

(b)

**EMP\_DEPT**

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

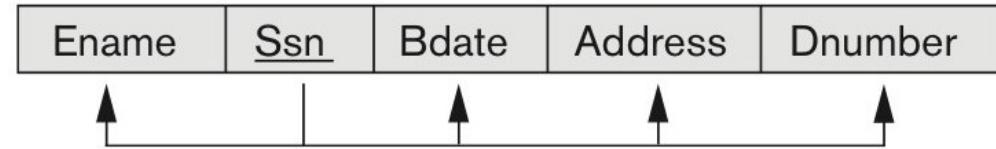


3NF Normalization



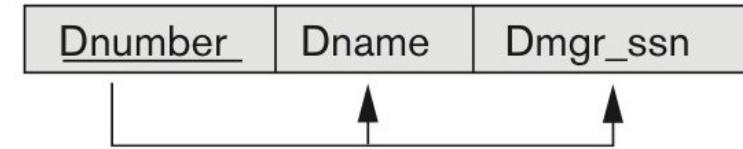
**ED1**

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

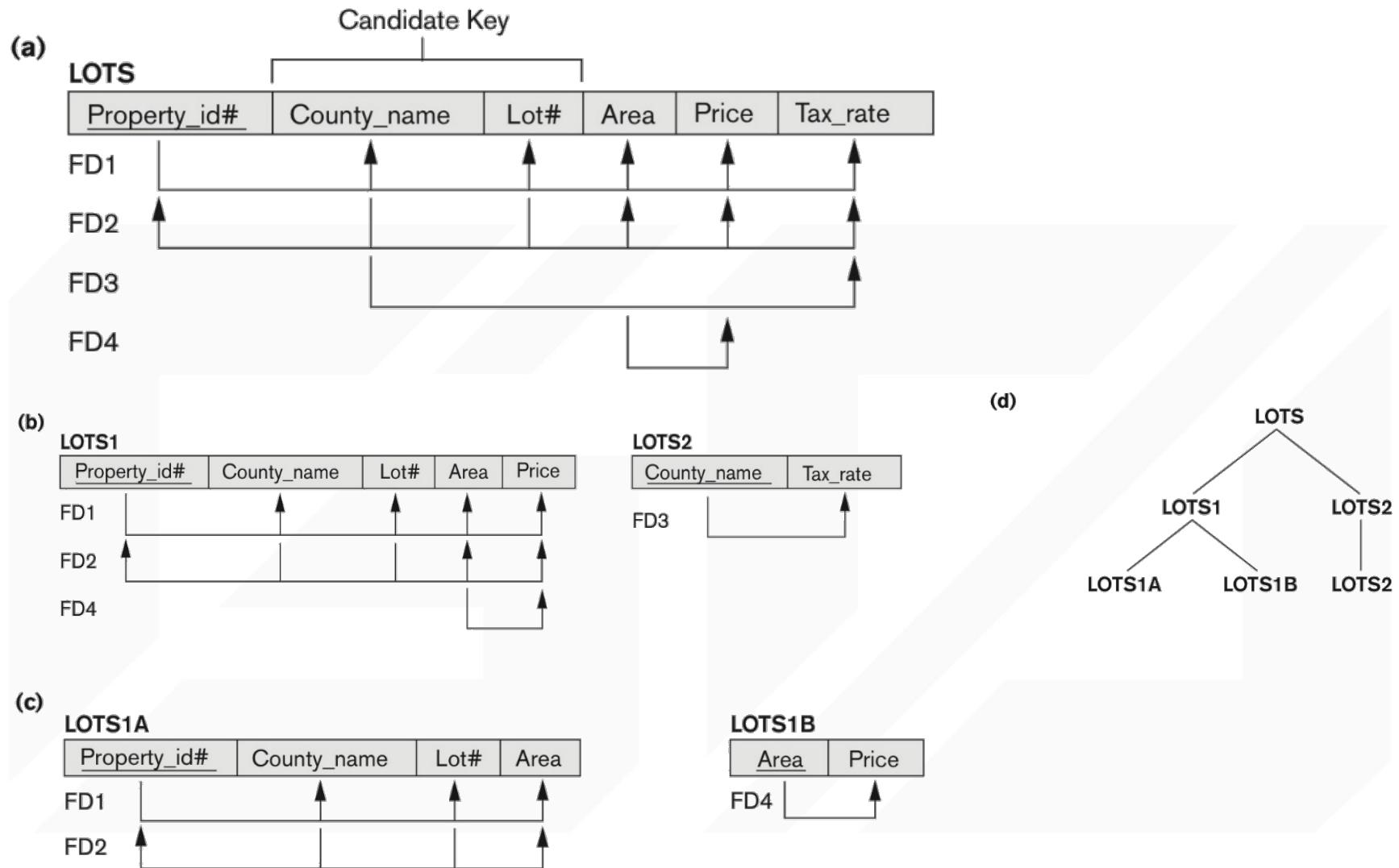


**ED2**

<u>Dnumber</u>	Dname	Dmgr_ssn



**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

- 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**

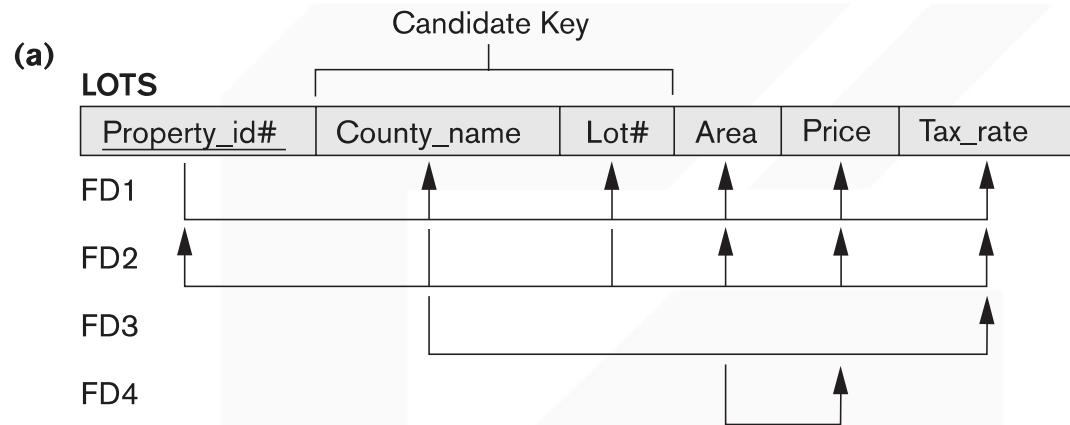
## 4. General Normal Form Definitions (For Multiple Keys)

- 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.

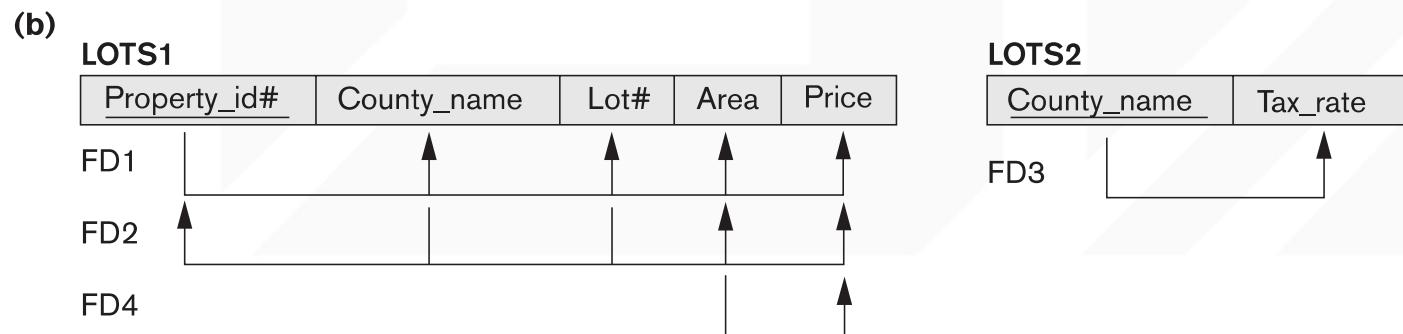
## 4.1 General Definition of 2NF (For Multiple Candidate Keys)

- 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  $\text{County\_name} \rightarrow \text{Tax\_rate}$  violates 2NF.



So second normalization converts LOTS into  
 $\text{LOTS1 } (\text{Property_id\#}, \text{County\_name}, \text{Lot\#}, \text{Area}, \text{Price})$   
 $\text{LOTS2 } (\text{County\_name}, \text{Tax\_rate})$

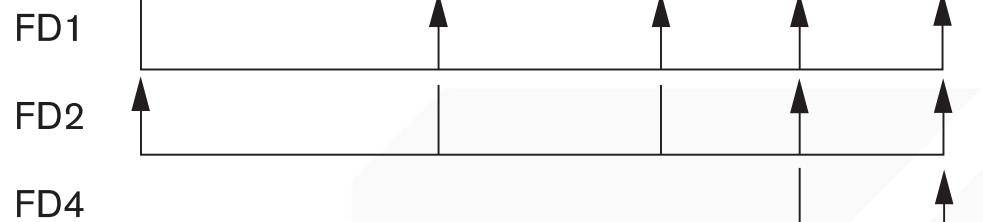


## 4.2 General Definition of Third Normal Form

- **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**

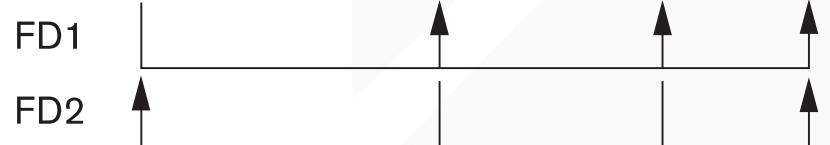
Property_id#	County_name	Lot#	Area	Price
--------------	-------------	------	------	-------



- LOTS1 relation violates 3NF because  $\text{Area} \rightarrow \text{Price}$  ; and **Area is not a superkey in LOTS1.** (see Figure 14.12).

**LOTS1A**

Property_id#	County_name	Lot#	Area
--------------	-------------	------	------



**LOTS1B**

Area	Price
------	-------



## 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.

## 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 previous slide takes care of the dependencies that “slip through” (are allowable to) 3NF but are “caught by” BCNF which we discuss next.

