Functional Dependencies and Normalization for Relational Databases

Dr.M.Venkatesan

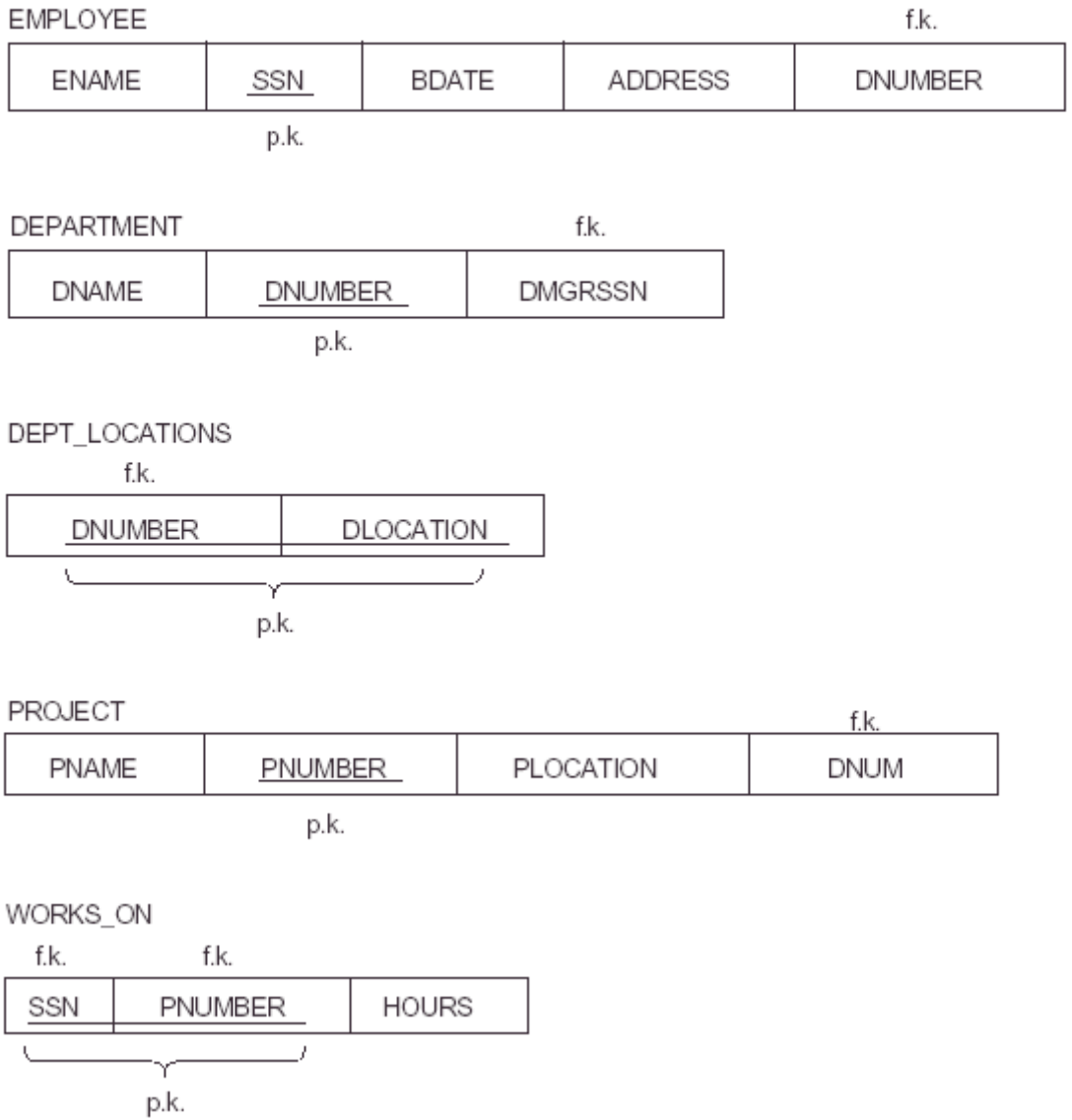
Informal Design Guidelines for Relational Databases

- 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
- Criteria for "good" base relations:
  - Discuss informal guidelines for good relational design
  - Discuss formal concepts of functional dependencies and normal forms 1NF 2NF 3NF BCNF

Semantics of the Relation Attributes

- Each tuple in a relation should represent one entity or relationship instance
  - Only foreign keys should be used to refer to other entities
  - Entity and relationship attributes should be kept apart as much as possible
  - Design a schema that can be explained easily relation by relation. The semantics of attributes should be easy to interpret.

Figure 14.1 Simplified version of the COMPANY relational database schema.



Redundant Information in Tuples and Update Anomalies

- Mixing attributes of multiple entities may cause problems
  - Information is stored redundantly wasting storage
  - 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
- **Insert Anomaly**
  - Cannot insert a project unless an employee is assigned to .
  - Inversely- Cannot insert an employee unless he/she is assigned to a project.

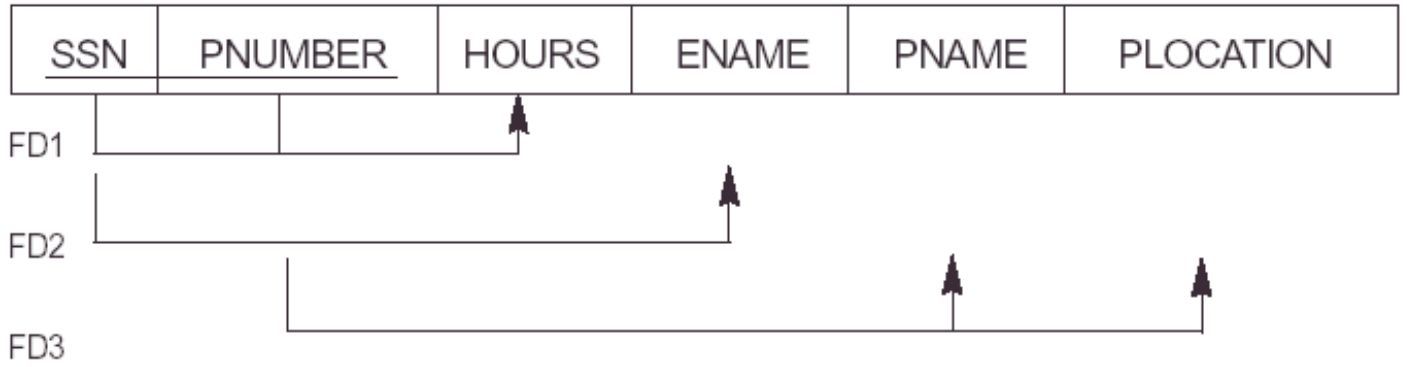
EXAMPLE OF AN UPDATE ANOMALY (2)

- **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.

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

(a) EMP_DEPT

(b) EMP_PROJ



Null Values in Tuples

- 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:
  - a. attribute not applicable or invalid
  - b. attribute value unkown (may exist)
  - c. value known to exist, but unavailable

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
- 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

(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	Product X	Bellaire
123456789	2	7.5	Product Y	Sugarland
666884444	3	40.0	Product Z	Houston
453453453	1	20.0	Product X	Bellaire
453453453	2	20.0	Product Y	Sugarland
333445555	2	10.0	Product Y	Sugarland
333445555	3	10.0	Product Z	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

Functional Dependencies

- Functional dependencies (FDs) are used to specify *formal measures* of the "goodness" of relational designs
- FDs and keys are used to define **normal forms** for relations
- FDs are **constraints** that are derived from the *meaning* and *interrelationships* of the data attributes

Functional Dependencies (2)

- A set of attributes X *functionally determines* a set of attributes Y if the value of X determines a unique value for Y
- X → Y holds if whenever two tuples have the same value for X, they *must have* the same value for Y

  If t1[X]=t2[X], *then* t1[Y]=t2[Y] in any relation instance r(R)
- X → Y in R specifies a *constraint* on all relation instances r(R)
- FDs are derived from the real-world constraints on the attributes

Examples of FD constraints

- Social Security Number determines employee name

  SSN → ENAME
- Project Number determines project name and location

  PNUMBER → {PNAME, PLOCATION}
- Employee SSN and project number determines the hours per week that the employee works on the project

  {SSN, PNUMBER} → HOURS

Functional Dependencies (3)

- 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])

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
  - A1. (Reflexive) If Y subset-of X, then X → Y
  - A2. (Augmentation) If X → Y, then XZ → YZ
    (Notation: XZ stands for X ∪ Z)
  - A3. (Transitive) If X → Y and Y → Z, then X → Z
- A1, A2, A3 form a *sound* and *complete* set of inference rules

Additional Useful Inference Rules

- Decomposition
  - If X → YZ, then X → Y and X → Z
- Union
  - If X → Y and X → Z, then X → YZ
- Psuedotransitivity
  - If X → Y and WY → Z, then WX → Z
- **Closure** of a set F of FDs is the set F+ of all FDs that can be inferred from F

Full functional Dependency

- *An attribute B of a relation R is fully functionally dependent on attribute A of R if it is functionally dependent on A & not functionally dependent on any proper subset of*
- Report(S#,C#,Title,Lname,Room#,Marks)
- S#, C# $\longrightarrow$ Marks .

Partial Functional Dependency

- *An attribute B of a relation R is partially dependent on attribute A of R if it is functionally dependent on any proper subset of A.*
- Report(S#,C#,Title,Lname,Room#,Marks)
- C# $\longrightarrow$ Title
- C# $\longrightarrow$ LName

Transitive Functional Dependency

- *An attribute B of a relation R is transitively dependent on attribute A of R if it is functionally dependent on an attribute C*
- *Which in turn is functionally dependent on A or any proper subset of A.*
- Report(S#,C#,Title,Lname,Room#,Marks)
- C# $\longrightarrow$ LName Lname $\longrightarrow$ Room#

  C# $\longrightarrow$ Room#

Introduction to Normalization

- **Normalization**: 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
  - 2NF, 3NF, BCNF based on keys and FDs of a relation schema
  - 4NF based on keys, multi-valued dependencies

First Normal Form

- Disallows composite attributes, multivalued attributes, and **nested relations**; attributes whose values *for an individual tuple* are non-atomic
- Considered to be part of the definition of relation

(a)

(b)

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATIONS
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATION
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

(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,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

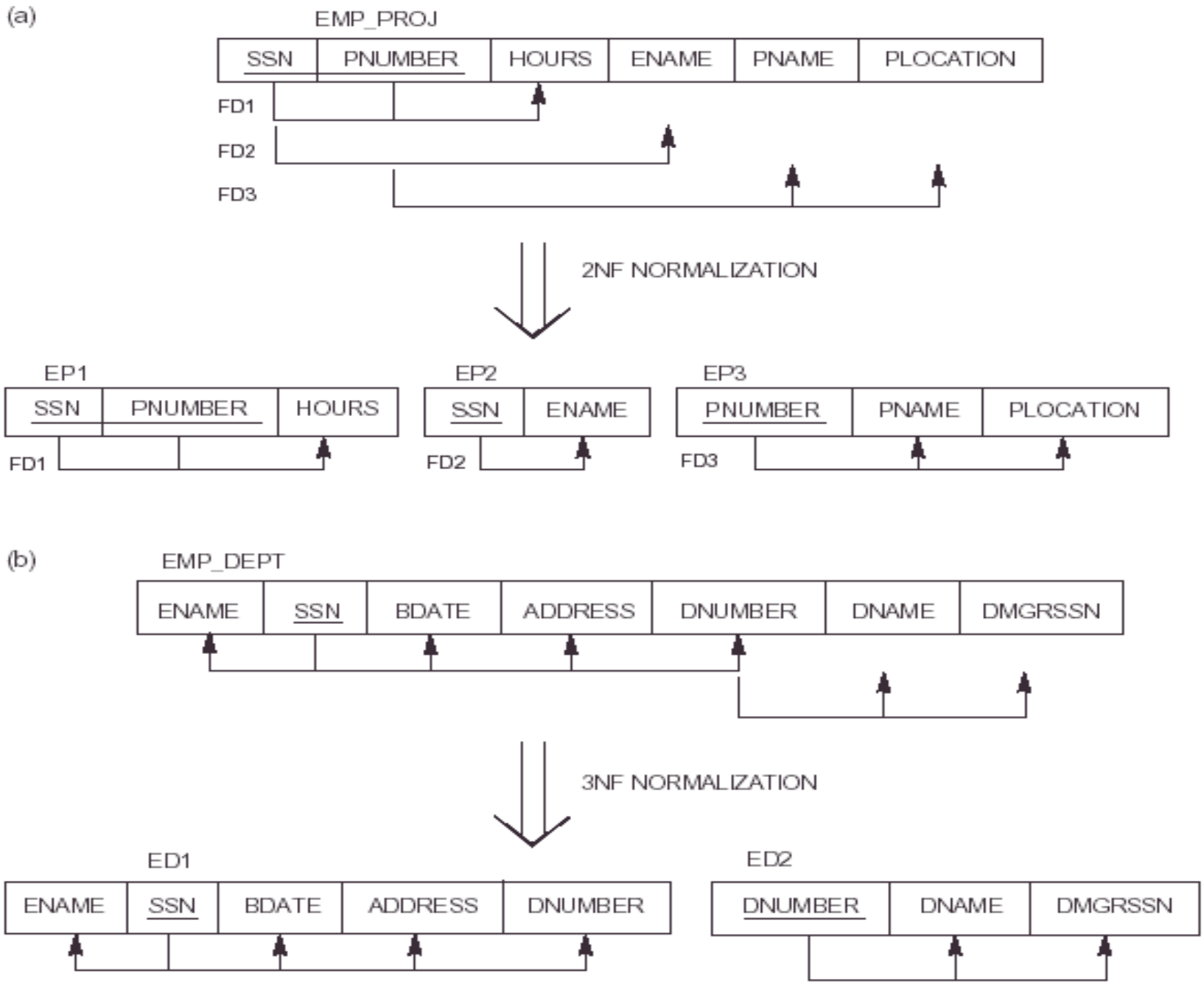
SSN	PNUMBER	HOURS

Second Normal Form

- Uses the concepts of FDs, primary key
- Definitions:
  - **Prime attribute** - attribute that is member of the primary key K
  - **Full functional dependency** - a FD Y → Z where removal of any attribute from Y means the FD does not hold any more

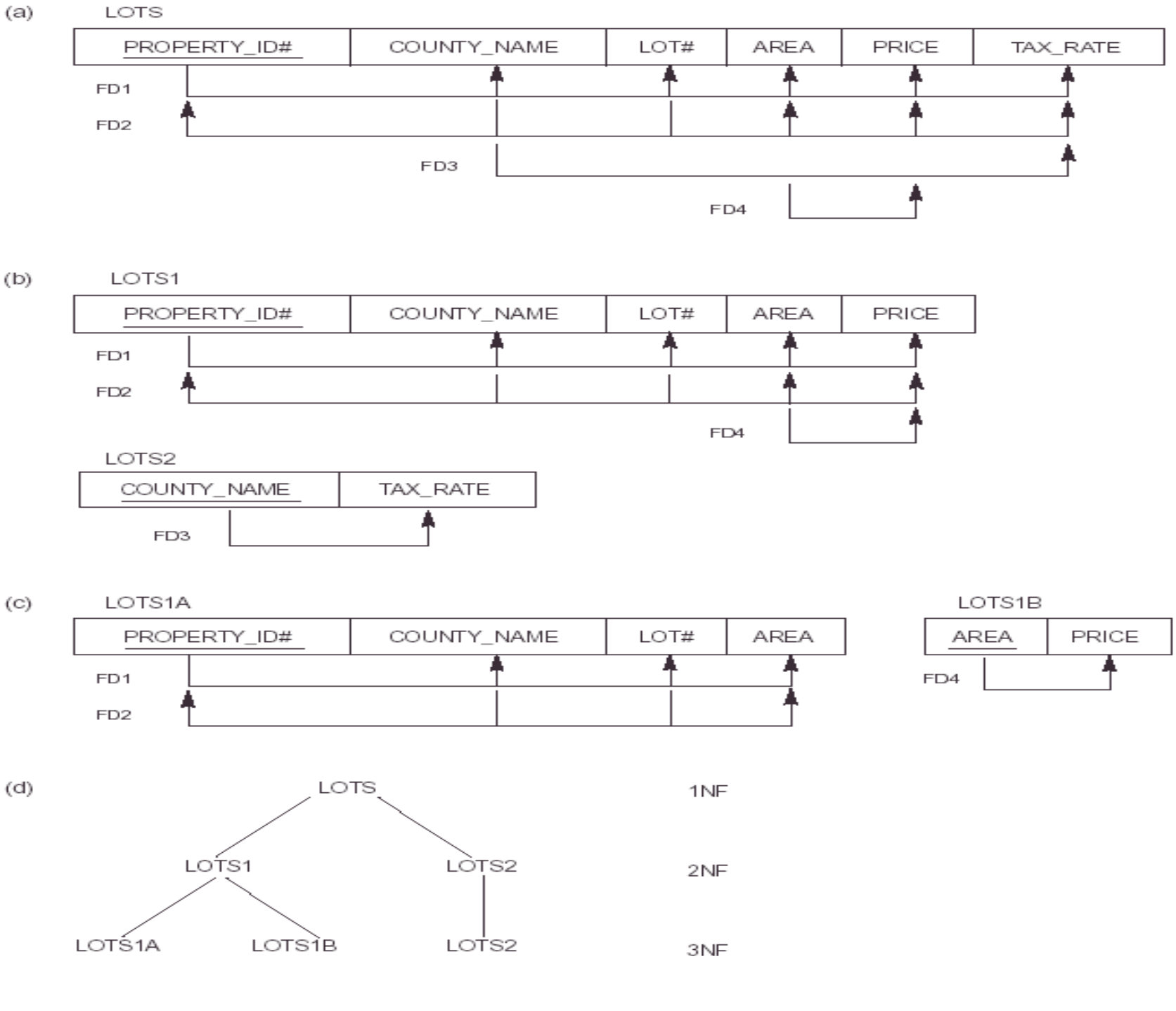
Examples
Second Normal Form

- {SSN, PNUMBER} → HOURS is a full FD since neither SSN → HOURS nor PNUMBER → HOURS hold
- {SSN, PNUMBER} → ENAME is *not* a full FD (it is called a *partial dependency*) since SSN → 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


(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
(b)
EMP_DEPT
ENAME
SSN
BDATE
ADDRESS
DNUMBER
DNAME
DMGRSSN
3NF NORMALIZATION
ED1
ENAME
SSN
BDATE
ADDRESS
DNUMBER
ED2
DNUMBER
DNAME
DMGRSSN

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 -> 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


(a)
LOTS
PROPERTY_ID#
COUNTY_NAME
LOT#
AREA
PRICE
TAX_RATE
FD1
FD2
FD3
FD4
(b)
LOTS1
PROPERTY_ID#
COUNTY_NAME
LOT#
AREA
PRICE
FD1
FD2
FD4
LOTS2
COUNTY_NAME
TAX_RATE
FD3
(c)
LOTS1A
PROPERTY_ID#
COUNTY_NAME
LOT#
AREA
FD1
FD2
LOTS1B
AREA
PRICE
FD4
(d)
LOTS
LOTS1
LOTS2
LOTS1A
LOTS1B
LOTS2
1NF
2NF
3NF

Third Normal Form

- Definition
  - **Transitive functional dependency** - a FD X → Z that can be derived from two FDs X → Y and Y
- Examples:
  - SSN → DMGRSSN is a transitive FD since

SSN → DNUMBER and DNUMBER → DMGRSSN hold

  - SSN → ENAME is *non-transitive* since there is no set of

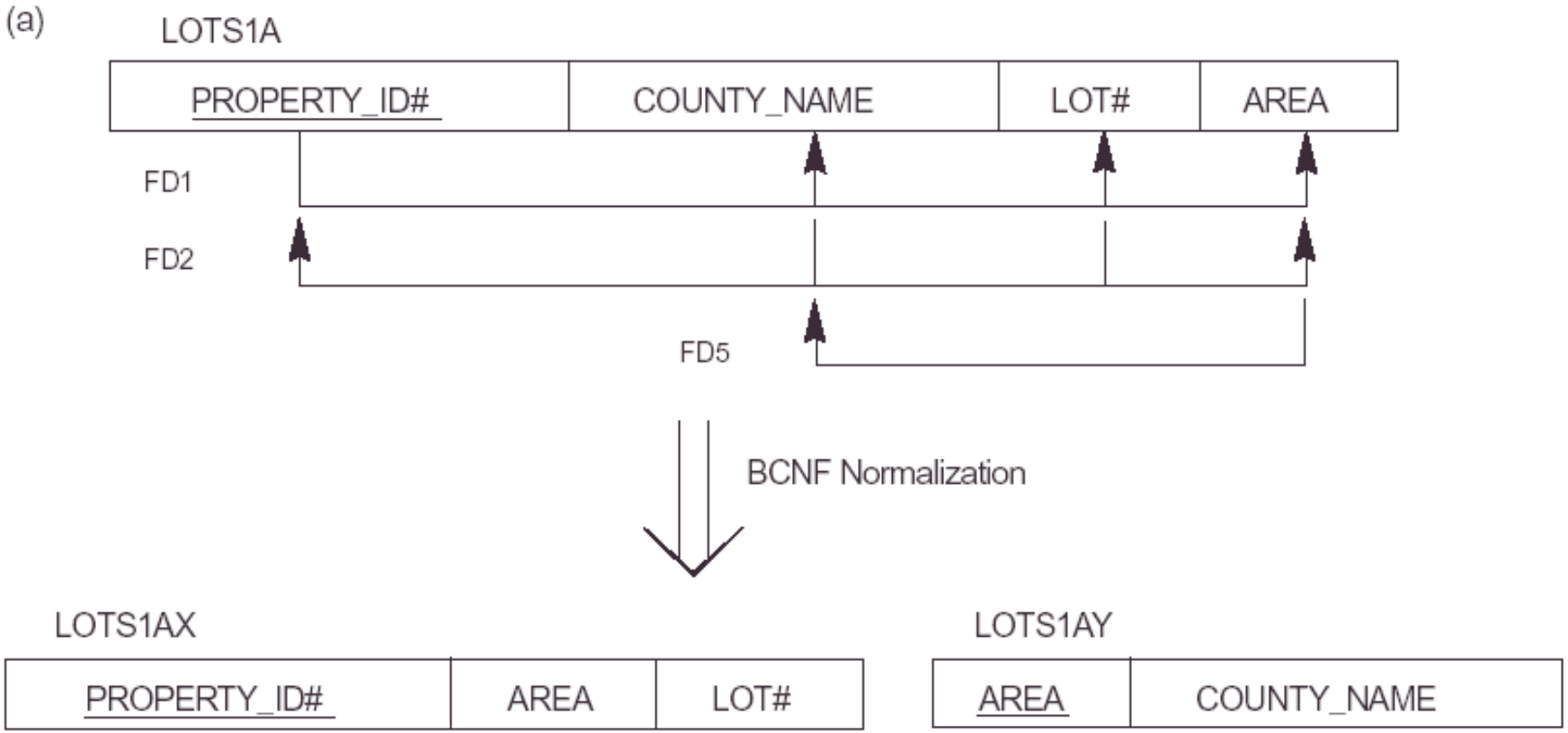
attributes X where SSN → X and X → ENAME

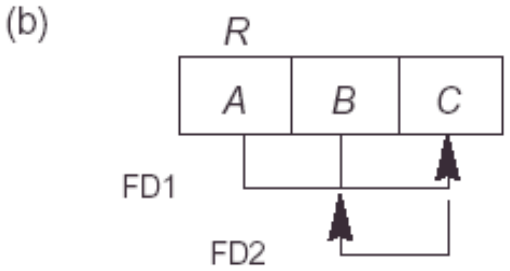
3rd Normal Form

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

BCNF (Boyce-Codd Normal Form)

- A relation schema R is in **Boyce-Codd Normal Form** (**BCNF**) if whenever an FD X → 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)


(a)
LOTS1A
PROPERTY_ID#
COUNTY_NAME
LOT#
AREA
FD1
FD2
FD5
BCNF Normalization
LOTS1AX
PROPERTY_ID#
AREA
LOT#
LOTS1AY
AREA
COUNTY_NAME


(b)
R
A
B
C
FD1
FD2

SUMMARY OF NORMAL FORMS based on Primary Keys

Table 10.1
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).

Boyce-Codd Normal Form

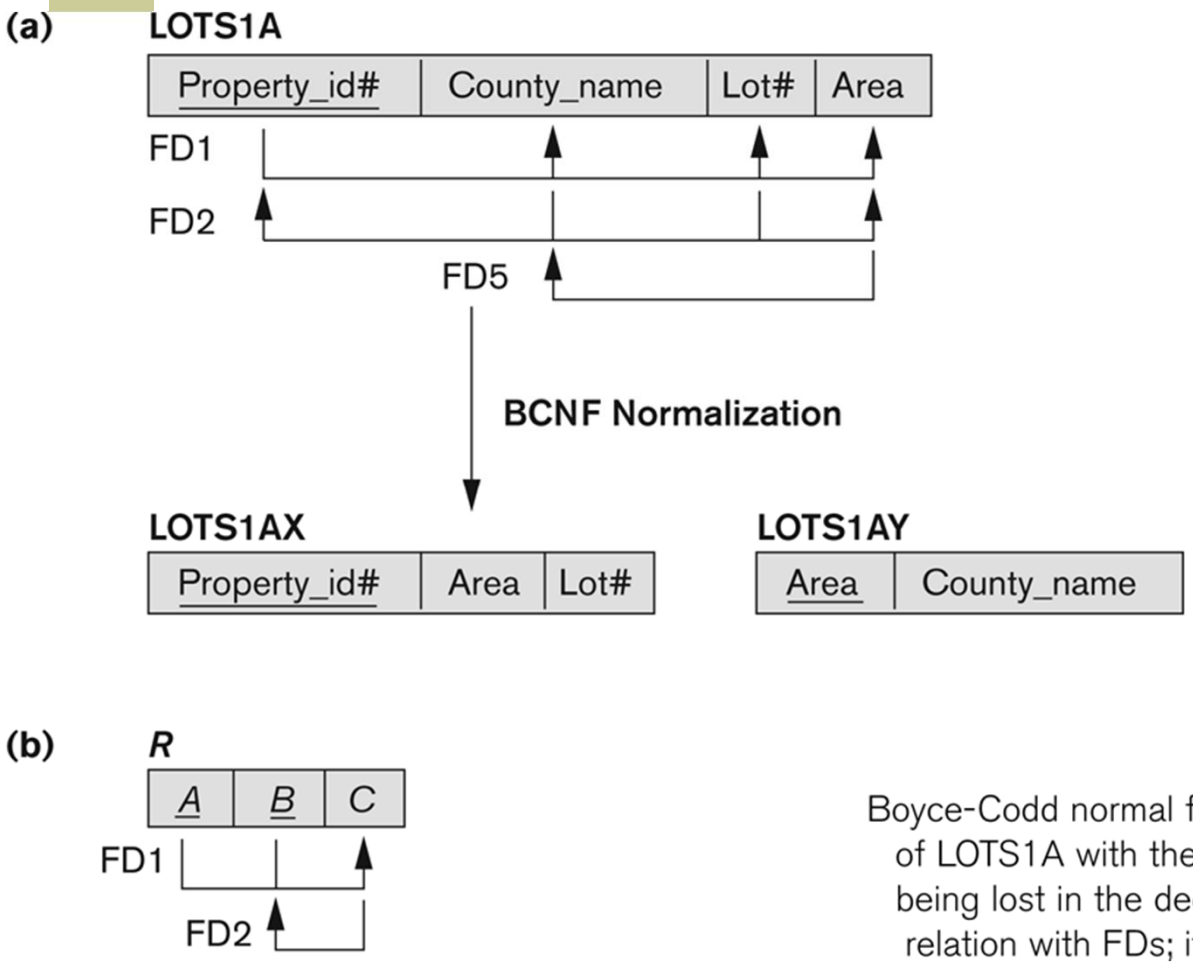


Figure 10.12
Boyce-Codd normal form. (a) BCNF normalization of LOTS1A with the functional dependency FD2 being lost in the decomposition. (b) A schematic relation with FDs; it is in 3NF, but not in BCNF.

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