Database Systems

Lecture #13

Sang-Wook Kim
Hanyang University

Objectives

- To learn normal forms and normalization
  - Concepts of normalization
  - Normal forms
    - 1NF, 2NF, 3NF, BCNF

Outline

- Normalization
- First Normal Form
- Second Normal Form
- Third Normal Form
- Boyce-Codd Normal Form

Normalization

- Normal forms
  - Definition of desirable forms of relations
- Normalization
  - To divide a relation schema to smaller and more desirable relations

Normalization

- Properties that a desirable set of relation schemas should have:
  - Lossless join property
    - Solution for the spurious tuple problem
  - Dependency preservation property

Prime Attribute

- An attribute of relation schema R that is a member of *some candidate key*
- Example
  - In WORKS_ON (Ssn, Pnumber, Hours) relation:
    - {Ssn, Pnumber}: candidate key
    - Both Ssn and Pnumber are prime attributes
    - Hours is non-prime

First Normal Form

- Definition of 1NF
  - Only attribute values permitted are a *single atomic value*
    - Each attribute of a tuple has only one value from the given domain
    - Multiple values are not allowed
  - Part of the formal definition of a relation

First Normal Form

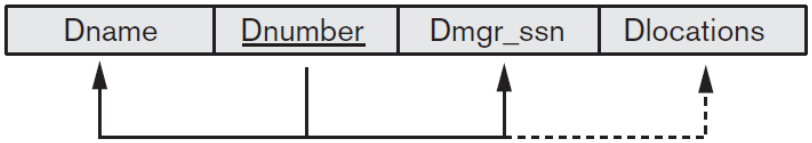
- Characteristics
  - Not allow following non-atomic attributes:
    - Composite attributes
    - Multi-value attributes
    - Nested relations

First Normal Form

- Example

(a)

DEPARTMENT



(b)

DEPARTMENT

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

(c)

DEPARTMENT

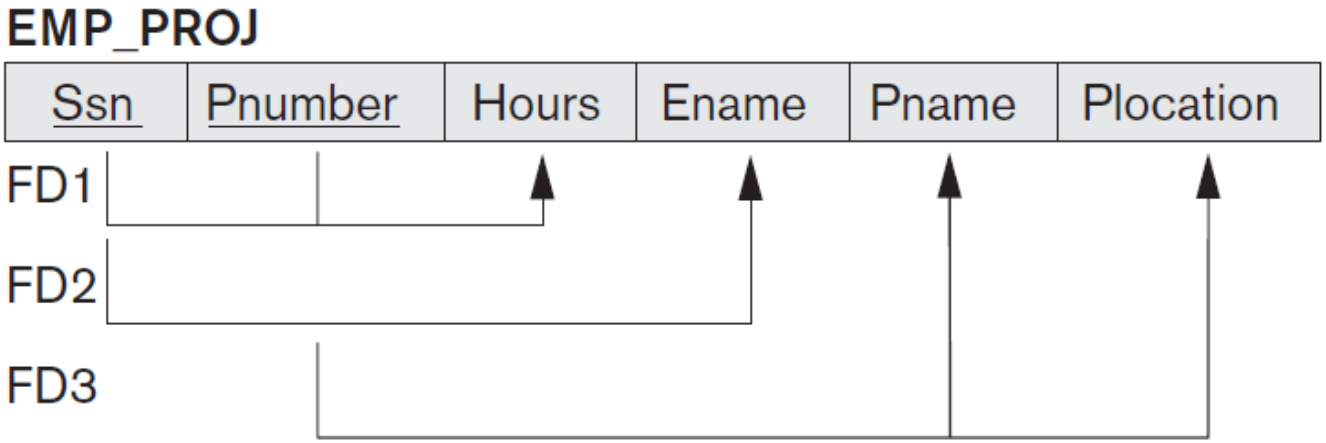
Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Second Normal Form

- Concepts
  - Full functional dependency
    - $X \rightarrow Y$ is a *full functional dependency* if the removal of any attribute A from X means that the dependency does not hold any more
  - Partial functional dependency
    - $X \rightarrow Y$ is a *partial functional dependency* even when some attribute A from X is removed from X, the dependency still holds

Second Normal Form

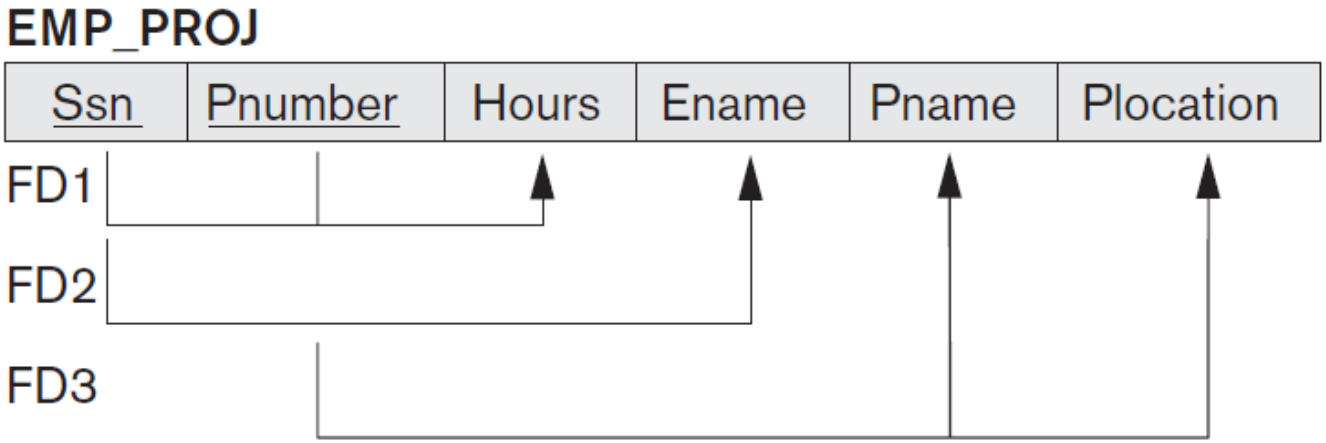
- Example



- {Ssn, Pnumber} → Hours
  - FD 'Ssn → Hours' and 'Pnumber → Hours' do not hold
  - Full functional dependency

Second Normal Form

- Example



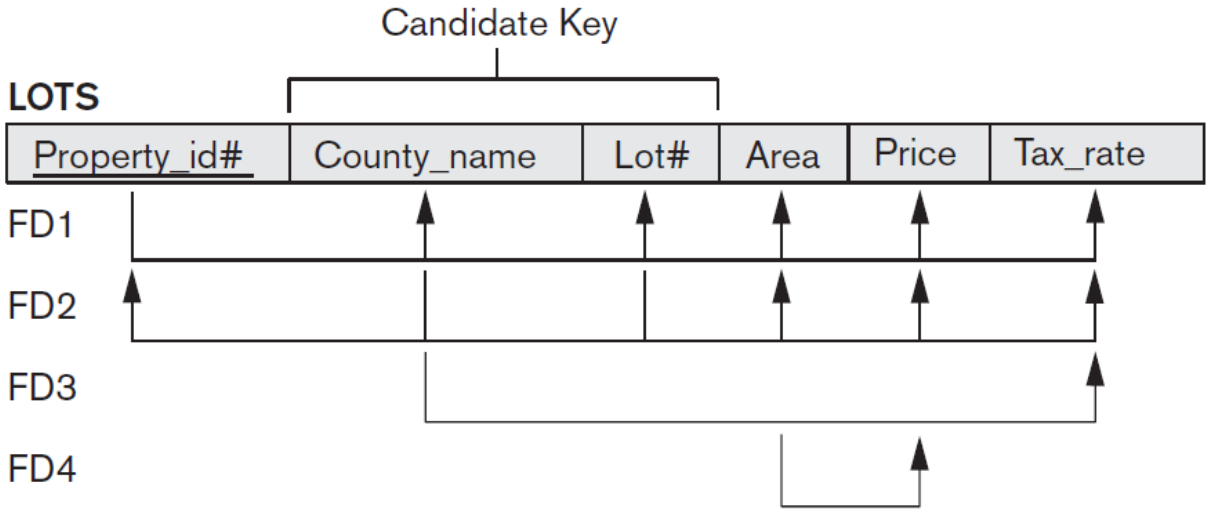
  - {Ssn, Pnumber} → Ename
    - FD 'Ssn → Ename' does hold
    - Partial functional dependency

Second Normal Form

- Definition of 2NF
  - A relation schema *R* is in 2NF
    - If every nonprime attribute *A* in *R* is *fully functional dependent* on the key of *R*
  - *R* is not in 2NF
    - If any nonprime attribute *A* in *R* is *partially functional dependent* on any key of *R*

Second Normal Form

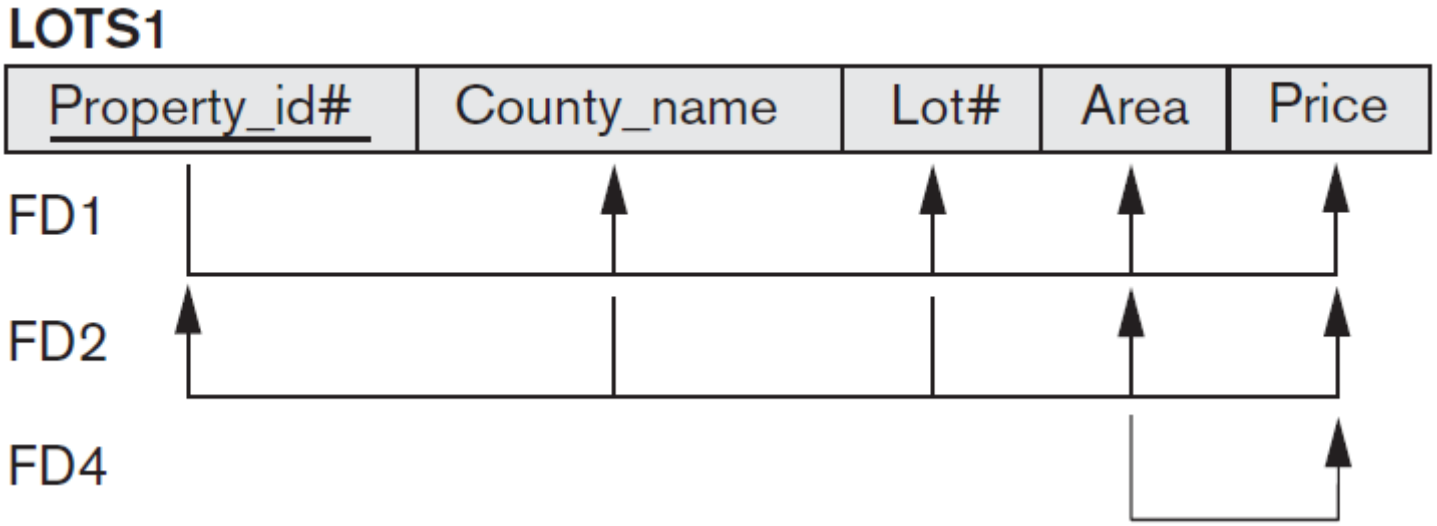
- Example



- Keys: Property_id, {Country_name, Lot#}
- FD3 is partially dependent on {Country_name, Lot#}
- Not in 2NF

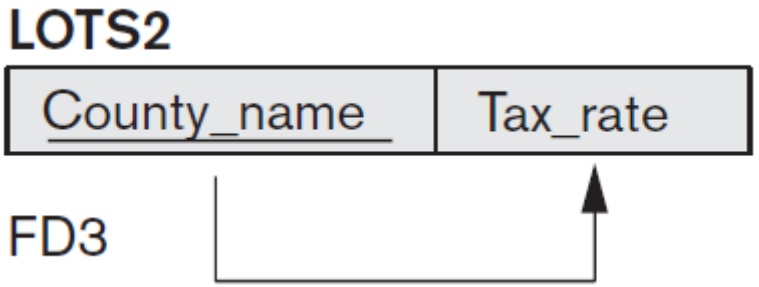
Second Normal Form

- Normalization into 2NF
  - Decompose it into two relations LOTS1 and LOTS2
    - LOTS1: remove Tax_rate from LOTS (FD3 is removed)



Second Normal Form

- Normalization into 2NF
  - Decompose it into two relations LOTS1 and LOTS2
    - LOTS2: create a new relation with attributes in FD3



Second Normal Form

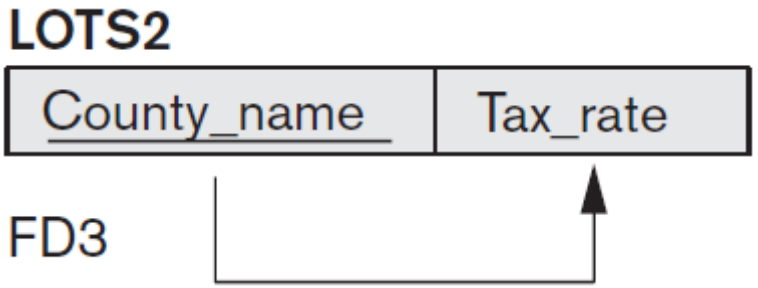
- Normalization into 2NF
  - Decompose it into two relations LOTS1 and LOTS2
    - LOTS1: remove Tax_rate from LOTS (FD3 is removed)
    - LOTS2: create a new relation with attributes in FD3
  - Any 1NF relation can be decomposed into 2NF relation by the *normalization process*

Third Normal Form

- Definition of 3NF
  - For every functional dependency $X \rightarrow A$ that holds in a relation schema R,
  - Either one of following condition holds:
    - (a) X is a superkey of R
    - (b) A is a prime attribute of R.

Third Normal Form

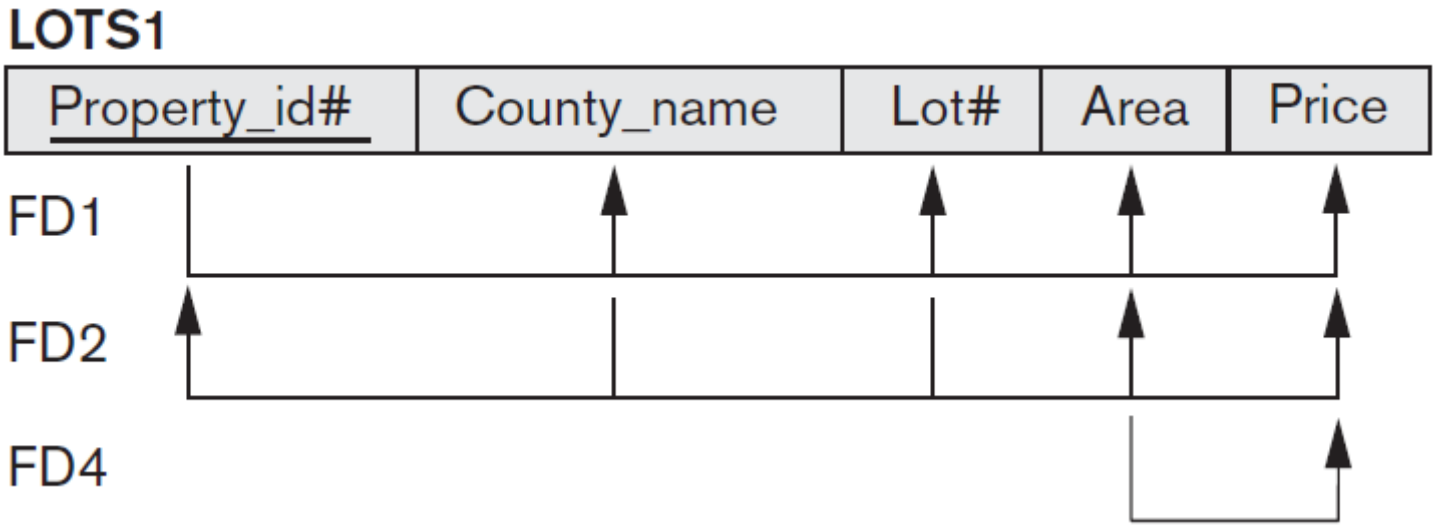
- Example



  - LOTS2 is in 3NF

Third Normal Form

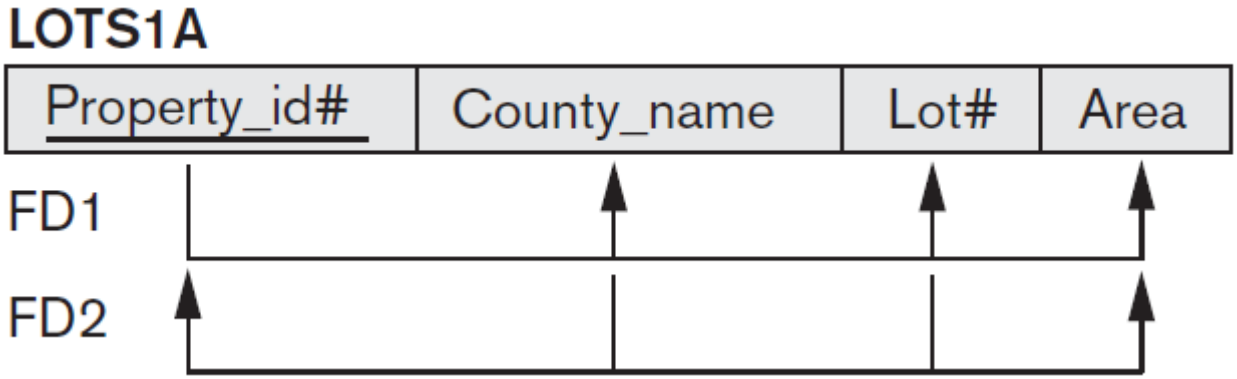
- Example



  - LOTS1 is not in 3NF
    - FD4
      - Area is not a superkey
      - Price is not a prime attribute

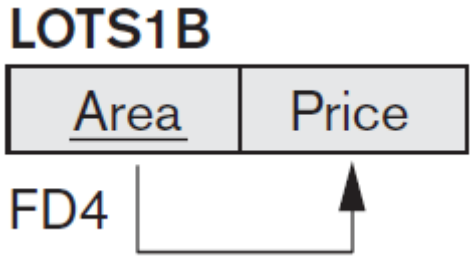
Third Normal Form

- Normalization into 3NF
  - Decompose it into two relations LOTS1A and LOTS1B
    - LOTS1A: remove Price from LOTS1 (FD4 is removed)



Third Normal Form

- Normalization into 3NF
  - Decompose it into two relations LOTS1A and LOTS1B
    - LOTS1B: create a new relation with attributes in FD4



Third Normal Form

- Normalization into 3NF
  - Decompose it into two relations LOTS1A and LOTS1B
    - LOTS1A: remove Price from LOTS1 (FD4 is removed)
    - LOTS1B: create a new relation with attributes in FD4
  - Any 2NF relation can be decomposed into 3NF relation by the *normalization process*

Boyce-Codd Normal Form

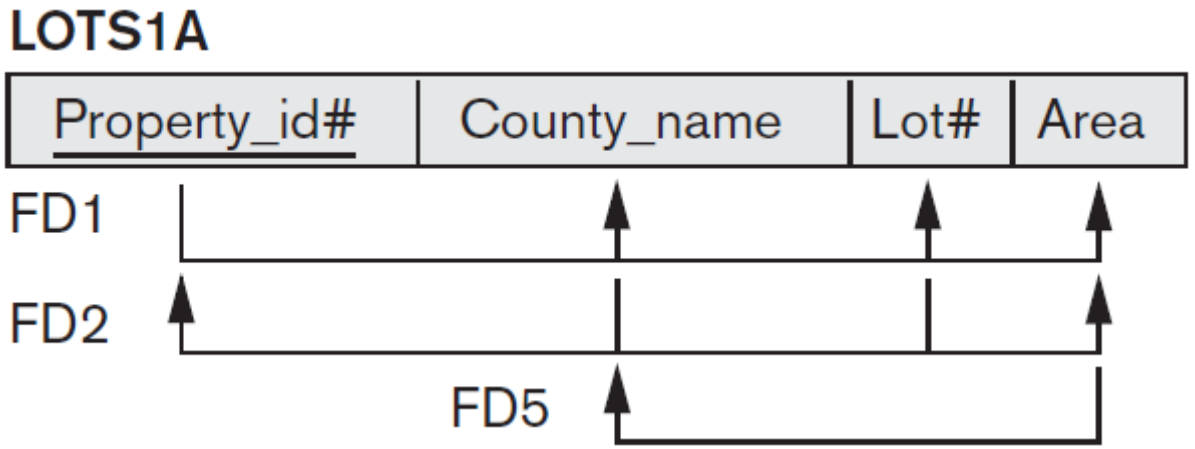
- Definition of BCNF
  - For every functional dependency $X \rightarrow A$ that holds in a relation schema R,
  - X is a superkey of R

Boyce-Codd Normal Form

- Note again the definition of 3NF
  - For every functional dependency $X \rightarrow A$ holds in a relation schema R,
  - Either one of following condition holds:
    - (a) X is a superkey of R
    - (b) A is a prime attribute of R (REMOVED!)

Boyce-Codd Normal Form

- Example



  - LOTS1A is not in BCNF
    - FD5
      - Area is not a superkey
      - Country_name is a prime attribute of LOTS1A (LOTS1A is in 3NF)

Boyce-Codd Normal Form

- Normalization into BCNF
  - Decompose it into LOTS1AX and LOTS1AY
    - LOTS1AX: remove Country_name from LOTS1A (FD5 is removed)
    - LOTS1AY: create a new relation with attributes in FD5

LOTS1AX

Property_id#	Area	Lot#

LOTS1AY

Area	County_name

Boyce-Codd Normal Form

- Normalization into BCNF
  - Decompose it into LOTS1AX and LOTS1AY
    - LOTS1AX: remove Country_name from LOTS1A (FD5 is removed)
    - LOTS1AY: create a new relation with attributes in FD5
  - Any 3NF relation can be decomposed into BCNF relation by the *normalization process*

Summary

- A normal form has a tighter condition than lower normal forms:
  - Every 2NF relation are also in 1NF
  - Every 3NF relation are also in 2NF
  - Every BCNF relation are also in 3NF
    - Some 3NF relations, however, are not in BCNF

Summary

- Eventual goal for good relational design
  - Design every relation in a database to be in BCNF or 3NF

Summary

- Additional properties for good relational design
  - Lossless join property
  - Dependency preservation property

References

1. Codd, Edgar F. "Recent Investigations in Relational Data Base Systems." *IFIP congress*. Vol. 74. 1974.
2. Ullman J. *Principles of Database and Knowledge-Base Systems*, Vol. 1, Computer Science Press, 1988.
3. Maier, David. *The theory of relational databases*. Vol. 11. Rockville: Computer science press, 1983.
4. Atzeni, Paolo, and Valeria De Antonellis. *Relational database theory*. Benjamin-Cummings Publishing Co., Inc., 1993.
5. Rustin, R., ed. Data Base Systems, Prantice-Hall, 1972.

Have a nice day!