DB DESIGN THEORY

Chapter 10 (part 2):

NORMAL FORMS

Outline of Chapter 11

1 Introduction.

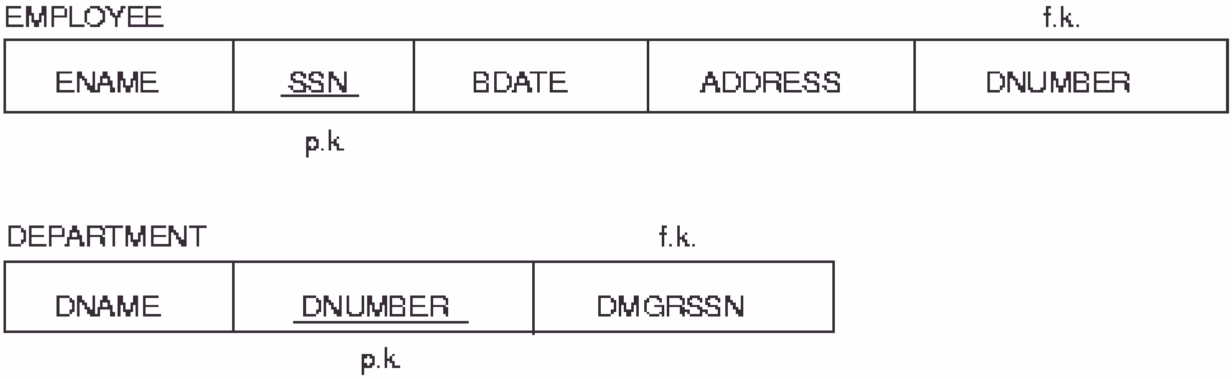
2 Informal Design Guidelines for Relational Databases.

3 Functional Dependencies (FDs)

4 General Normal Form Definitions

4.1. Redundant Information in Tuples (1)

- Grouping together attributes from different entities in a relation schema, has a significant effect on storage space.
- Example (Waste of storage space): Consider the following database schemas and instances:

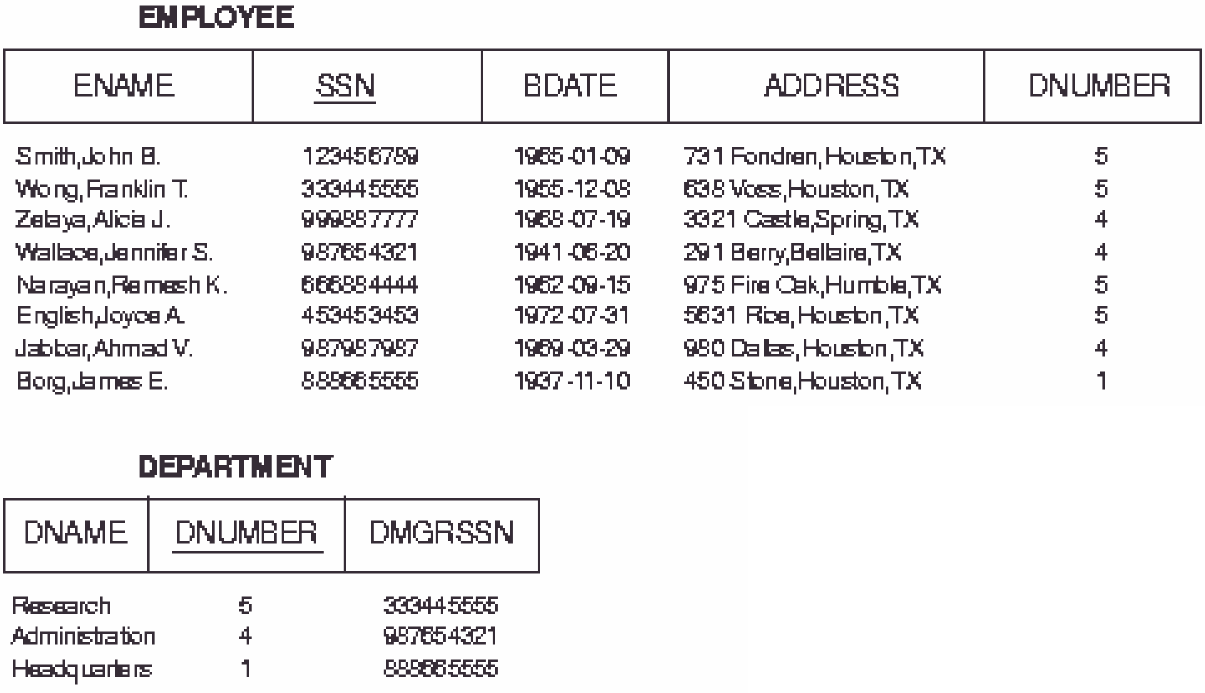


EMPLOYEE

				f.k.
ENAME	SSN	BDATE	ADDRESS	DNUMBER
	p.k.			

DEPARTMENT

		f.k.
DNAME	DNUMBER	DMGRSSN
	p.k.	



EMPLOYEE

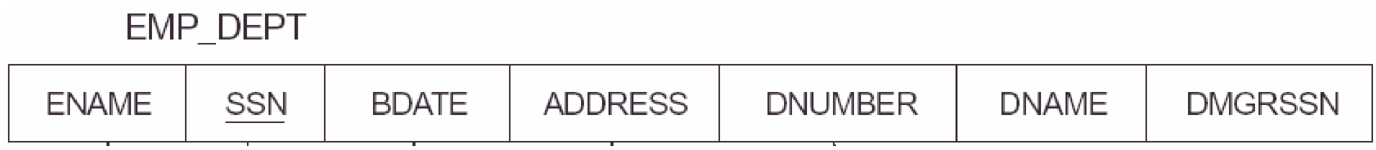
ENAME	SSN	BDATE	ADDRESS	DNUMBER
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5
Zelaya,Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4
Wallace,Jennifer S.	987654321	1941-06-20	291 Berry,Bellaire,TX	4
Narayan,Ramesh K.	666884444	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	DMGRSSN
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

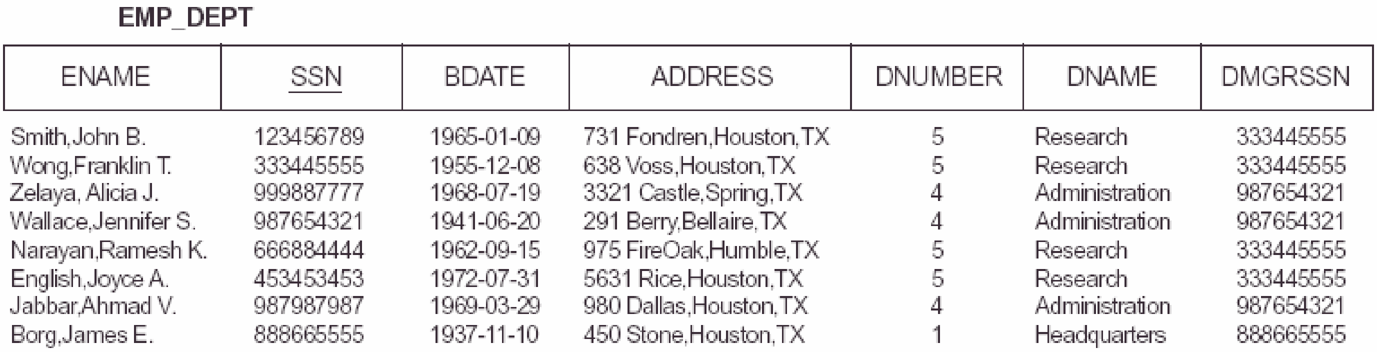
4.1. Redundant Information in Tuples (2)

- Example (Waste of storage space – cont.): Consider the alternative database schema and instance where all the attributes are grouped in one relation



EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN



EMP_DEPT

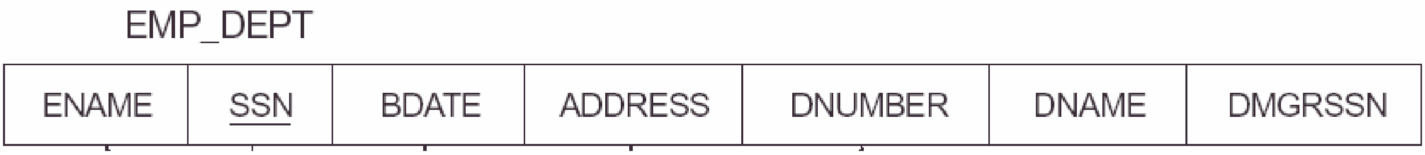
ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN
Smith,John B.	123456789	1965-01-09	731 Fondren,Houston,TX	5	Research	333445555
Wong,Franklin T.	333445555	1955-12-08	638 Voss,Houston,TX	5	Research	333445555
Zelaya, Alicia J.	999887777	1968-07-19	3321 Castle,Spring,TX	4	Administration	987654321
Wallace,Jennifer 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

- Clearly when attributes from both entities are group together in a relation schema, information is stored redundantly wasting storage space.

4.2. Update Anomalies (1)

- Mixing attributes of multiple entities may cause update anomalies:
  - Insertion anomalies
  - Deletion anomalies
  - Modification anomalies.

- Example (Update anomalies):
  Consider the relation:



EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN

Insert Anomaly: Cannot insert a department unless an employee works for it (null is not allowed for the primary key attribute SSN).

Inversely - Cannot insert an employee unless he/she is assigned to a department (or nulls have to be included for the Department attributes).

4.2. Update Anomalies (2)

- Example (Update anomalies – cont.):

Delete Anomaly: When a department is deleted, it will result in deleting all the employees who work for that department.

Alternately, if an employee is the sole employee of a department, deleting that employee would result in deleting the corresponding department.

Modification Anomaly: Changing the manager of a department may cause this modification to be made for all 100 employees working in this department.

If we fail to update some tuples, the same department will be shown to have two different values for manager in different employee tuples.

4.3. Normal forms and Normalization (1)

- Given a relation schema we need to decide *whether it is a good design*. This decision must be guided by an understanding of what problems, if any, arise from the current schema.

- To this end, several **normal forms** have been proposed. If a relation schema is in one of these forms, we know that certain kinds of problems cannot arise.

- Initially, Codd proposed three normal forms (1NF, 2NF, and 3NF). A stronger NF (Boyce-Codd NF) was proposed later. All these NFs are based on the notion of FD.

- Later a 4NF (based on the concept of Multivalued dependency) and a 5NF (based on the concept of join dependency were proposed.

4.3. Normal forms and Normalization (2)

- The normal forms have *increasingly restrictive requirements*: Every relation in BCNF is also in 3NF, every relation in 3NF is also in 2NF etc.

  2NF is mainly of historical interest and we will not examine it.

- **Normalization of data** can be looked upon as a process of analyzing the given relation schemas based on their FDs to achieve the desirable properties of: (1) minimizing redundancy, and (2) minimizing the insertion, deletion and update anomalies.

- Note that sometimes, *database designers do not normalize to the highest possible normal form*. Relations may be left in a lower normalization status for performance reasons. The process of storing the join of higher normal form relations as a base relation – which is in a lower normal form – is known as **denormalization**.

4.4. First normal form

- **1NF** states that the domain of an attribute must include only *atomic (simple, indivisible) values*, and that the value of any attribute in a tuple must be a *single value* from the domain of that attribute.

- 1NF is now considered to be part of the formal definition of the *(flat) relational model*.

  This condition is removed in the *nested relational model* and in the *object-relational model*.

4.5. Second normal form

- **2NF:** A relation is in 2NF if, and only if, it is in 1NF and all non-key attributes are determined by the entire primary key. This is also known as a Partial Dependency.

- For example: A relation R(A, B, N, O, P) with the composite key (A,B) means that none of the non-key attributes N, O, or P can be determined by just A or just B.

If the following FDs hold for a relation, it would NOT be in 2NF

(StudentID, Activity) → ActivityFee
Activity → ActivityFee

4.6. Third normal form (1)

- An attribute A of a relation schema is a **prime attribute** if it is part of *any (candidate) key* of the relation.

- Attribute A must be part of a key (*any key*, if there are several). It is not enough for A to be part of a superkey because the latter condition is satisfied by each and every attribute.

- Example

  Consider the relation R(A, B, C) and the set of FDs F = {A → B, BC → A, B → C } on R.

  Which attributes of R are prime attributes?

  We need to compute all the keys of R.
  (Recall that *a key is a minimal superkey*)

  Keys of R: A and B.
  Prime attributes: A and B.

4.6. Third normal form (2)

- Let R be a relation schema, X be a subset of R and A be an attribute of R. R is in **3NF** if for every *non-trivial* FD X → A that holds over R, *either*:
  (a) X is a *superkey* of R, or
  (b) A is a *prime attribute*.

  Recall that X → A is *trivial* if A ∈ X.

- Note that, according to the previous definition, we must consider each dependency in the closure F^+ (that is, every dependency that holds over R) to determine whether R is in 3NF.
  Fortunately, it has been proven that *it is sufficient to check only the FDs in F*.

4.6. Third normal form (3)

- Example

  Consider the relation schema R(A, B, C) and the set of FDs F = { A → B, BC → A, B → C } on R.

  The prime attributes of R are A and B.

  R is in 3NF w.r.t F because for the FDs in F:

  A → B : A is a superkey
  (B is also a prime attribute here)
  BC → A : BC is a superkey
  (A is also a prime attribute here)
  B → C : B is a superkey

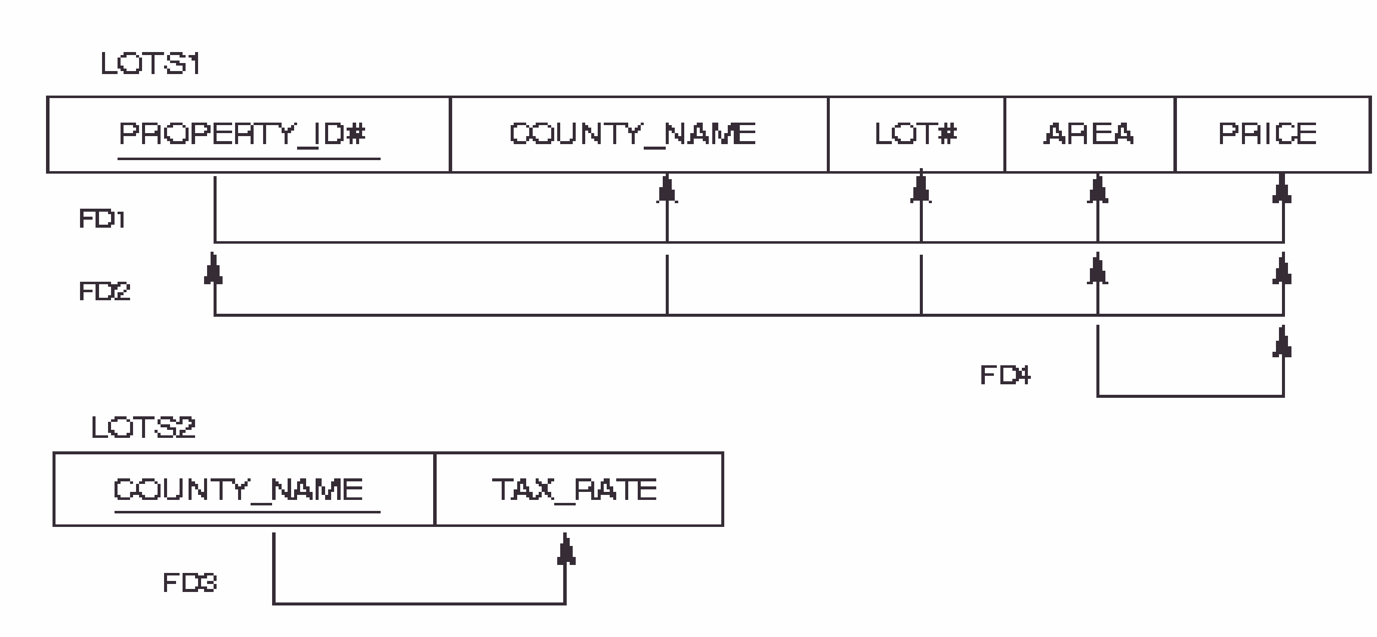
  Consider now the set of FDs F' = { B → A, C → A } on R.

  R is not in 3NF w.r.t. F' because for
  B → A : B is not a superkey and A is not prime.
  (A is not a prime attribute because BC is the only key of R w.r. t. F')

  We say that B → A *violates* the 3NF.

4.6. Third normal form (4)

- Example (more realistic)
  Consider the relations LOTS1 and LOTS2 describing parcels of land for sale in various counties of a state.



LOTS1

PROPERTY_ID#	COUNTY_NAME	LOT#	AREA	PRICE

LOTS2

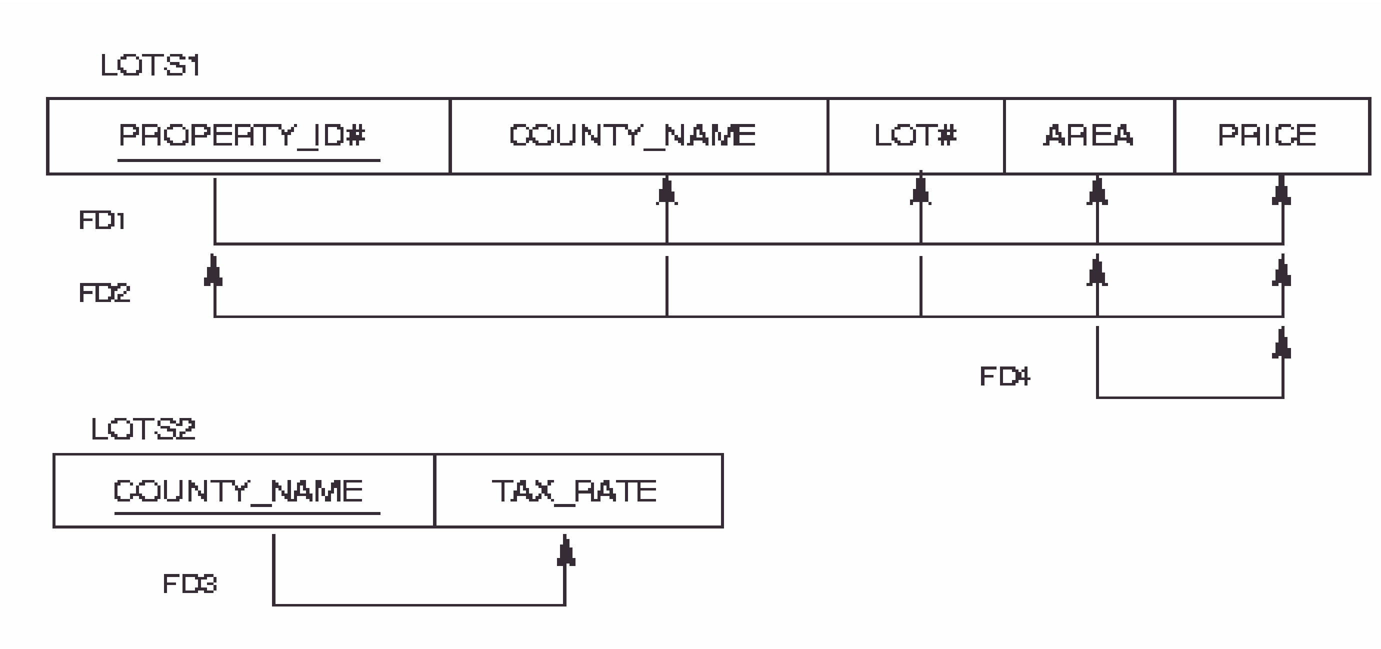
COUNTY_NAME	TAX_RATE

- The meaning of the FDs is as follows:
  FD1: PROPERTY_ID# uniquely identifies a parcel of land.
  FD2: COUNTY_NAME and LOT# together uniquely identifie a parcel of land.
  FD3: The tax rate is fixed for a given county.
  FD4: The price of a lot is determined by its area (regardless of the county it is in).

4.6. Third normal form (5)

- Example (cont.)
  The keys of LOTS1 are PROPERTY_ID#, and (COUNTY_NAME, LOT#).

  The single key of LOTS2 is COUNTY_NAME

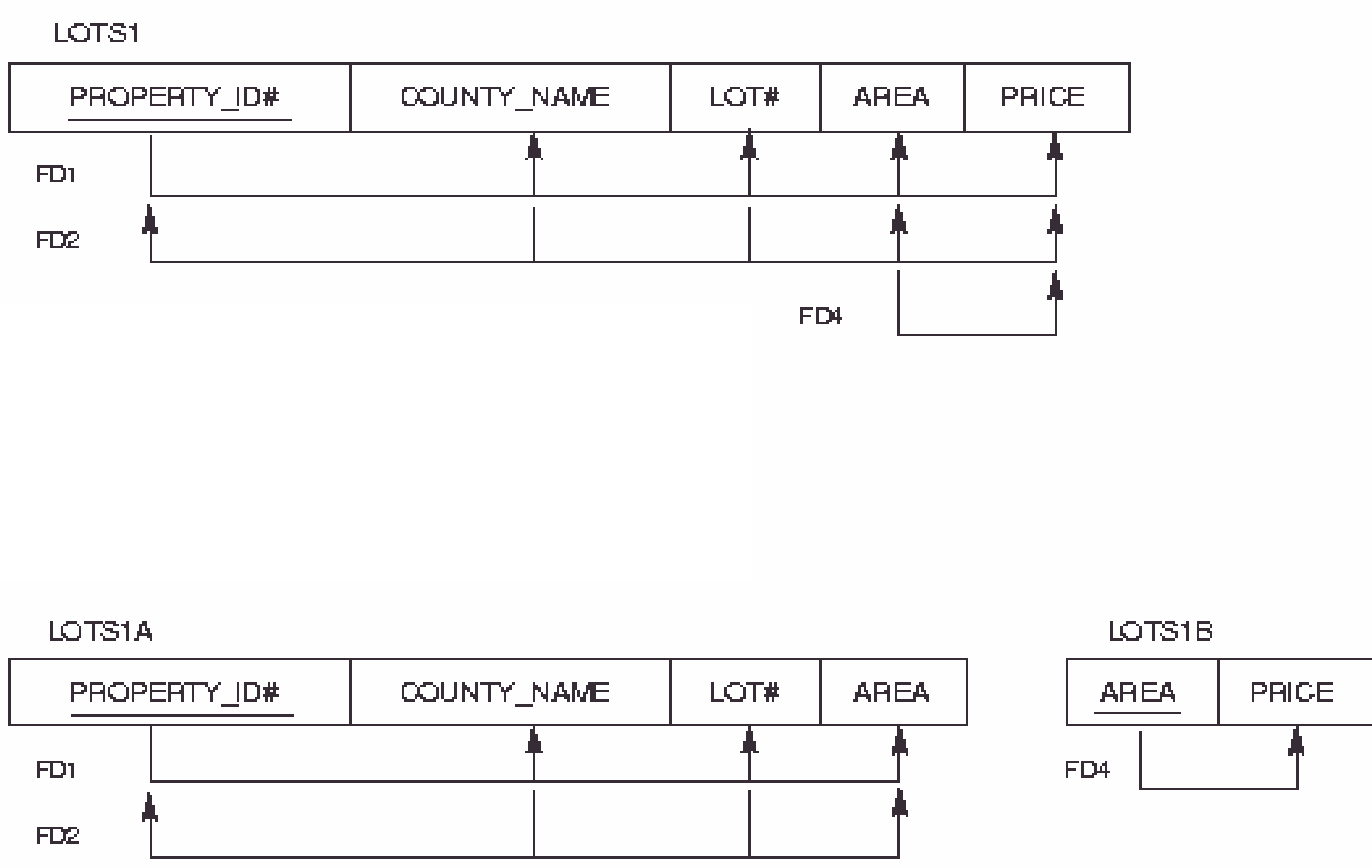


- Therefore:
  LOTS2 is in 3NF (why?)

  LOTS1 is not in 3NF:
  FD4 violates 3NF because AREA is not a superkey and PRICE is not a prime attribute in LOTS1.

4.6. Third normal form (6)

- Example (cont.)
  Suppose that we decompose LOTS1 into the relations LOTS1A and LOTS1B.



- Both LOTS1A and LOTS1B are in 3NF.

4.7. Boyce-Codd normal form (1)

- BCNF ensures that *no redundancy can be detected using FD information alone*.

  It is the most desirable normal form (from the point of view of redundancy).

4.7. Boyce-Codd normal form (2)

- Let R be a relation schema, X be a subset of R and A be an attribute of R. R is in **BCNF** if for every *non-trivial* FD X → A that holds over R, X is a *superkey* of R.

- Intuitively, in a BCNF relation the only non-trivial dependencies are those in which a *superkey determines some attributes*.

- As with 3NF *it is sufficient to consider only FDs in F*.

- Clearly, *a relation schema in BCNF is also in 3NF*. The opposite in not necessarily true.

- Example

  Consider the relation schema R(A, B, C) and the set of FDs F = { AB → C, C → B } on R.

  R is not in BCNF because C in C → B is not a superkey.

  Is R in 3NF?

4.7. Boyce-Codd normal form (3)

- The motivation for 3NF is rather technical: we will see that we can ensure that every relation schema can be decomposed into a collection of 3NF relations using decompositions that have the “dependency preservation” and “lossless-join” properties.

  These properties together are not guaranteed for a decomposition into a collection of BCNF relations.