LECTURE 3

Unit 3

Normalization of Relations

Normalization of data can be looked upon as a process of analyzing the given relation schemas based on their FDs and primary keys to achieve the desirable properties of

Minimizing redundancy

Minimizing the insertion, deletion, and update anomalies

Terms to know (1/4)

Superkey

A superkey of a relation schema $R = \{A1, A2, \ldots, An\}$ is a set of attributes $S \subseteq R$ with the property that no two tuples t1 and t2 in any legal relation state r of R will have t1[S]=t2[S].

Key

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.

Terms to know (2/4)

Difference between key and superkey

Key has to be *minimal i.e.* , if we have a key K = {A1, A2, ... , Ak} of R, then K - {Ai} is not a key of R for any Ai, 1<=i<=k

Example: {SSN} is a key for EMPLOYEE, whereas, {SSN}, {SSN, ENAME} and {SSN, ENAME, BDATE} and any set of attributes that includes SSN are all superkeys

Terms to know (3/4)

Candidate key / Secondary keys

If a relation schema has one or more keys

Primary key

One of the candidate keys arbitrarily designated to be the main key of the relation

Terms to know (4/4)

Prime attribute

An attribute of relation schema R is called a prime attribute of R if it is a member of *some candidate key* of R.

Nonprime attribute

An attribute is called nonprime if it is not a prime attribute-that is, if it is not a member of any candidate key

Normal forms

First normal form

Second normal form

Third normal form

Boyce-Codd normal form

Fourth normal form

Fifth normal form

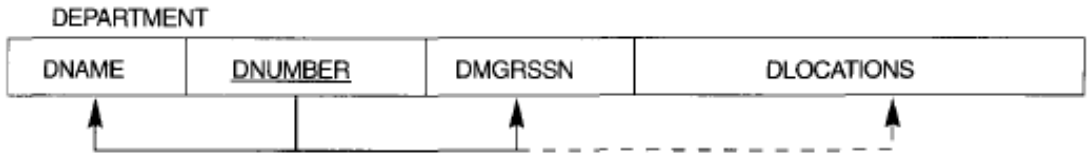
First Normal Form – 1NF (1/4)

No multivalued attributes or composite attributes

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

FIRST NORMAL FORM – 1NF (2/4)

(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

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 same relation with redundancy.

FIRST NORMAL FORM – 1NF (3/4)

(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

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

First Normal Form – 1NF (4/4)

{ } multivalued and list the component attributes that between ( )

The primary key of the new relation will combine the partial key with the primary key of the original relation

Second Normal Form – 2NF (1/5)

Second normal form (2NF) is based on the concept of *full functional dependency*.

A functional dependency X → *Y* is a full functional dependency if removal of any attribute A from X means that the dependency does not hold any more i.e. for any attribute A ∈ X, (X - {*A*}) does *not* functionally determine *Y*

A functional dependency X → *Y* is a partial dependency if some attribute A ∈ X can be removed from X and the dependency still holds; that is, for some A ∈ X, (X-{*A*}) → *Y*

Second Normal Form – 2NF (2/5)

Example

{SSN, PNumber} → Hours is a full FD

(neither SSN→Hours *nor* Pnumber→Hours hold)

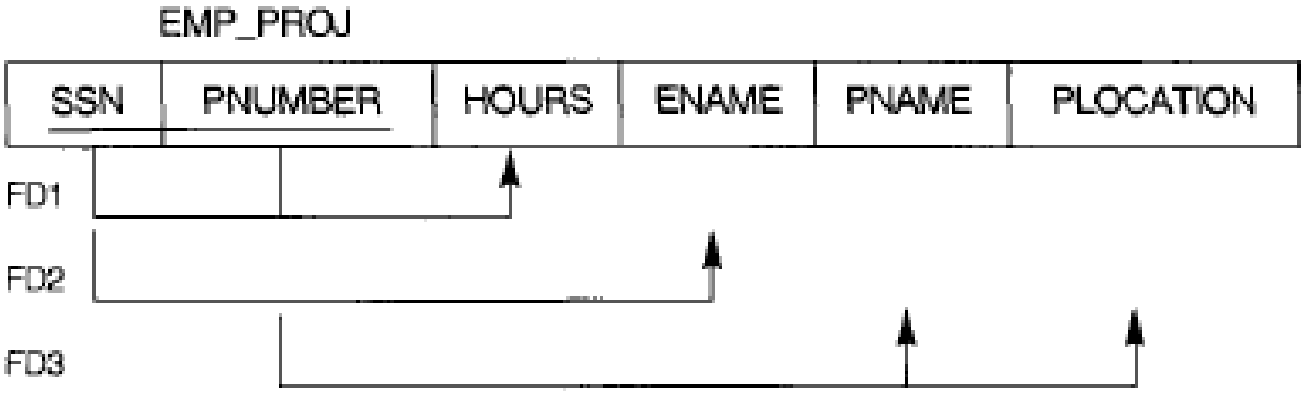
{SSN, Pname} → Ename is partial because SSN→Ename holds

Second Normal Form – 2NF (3/5)

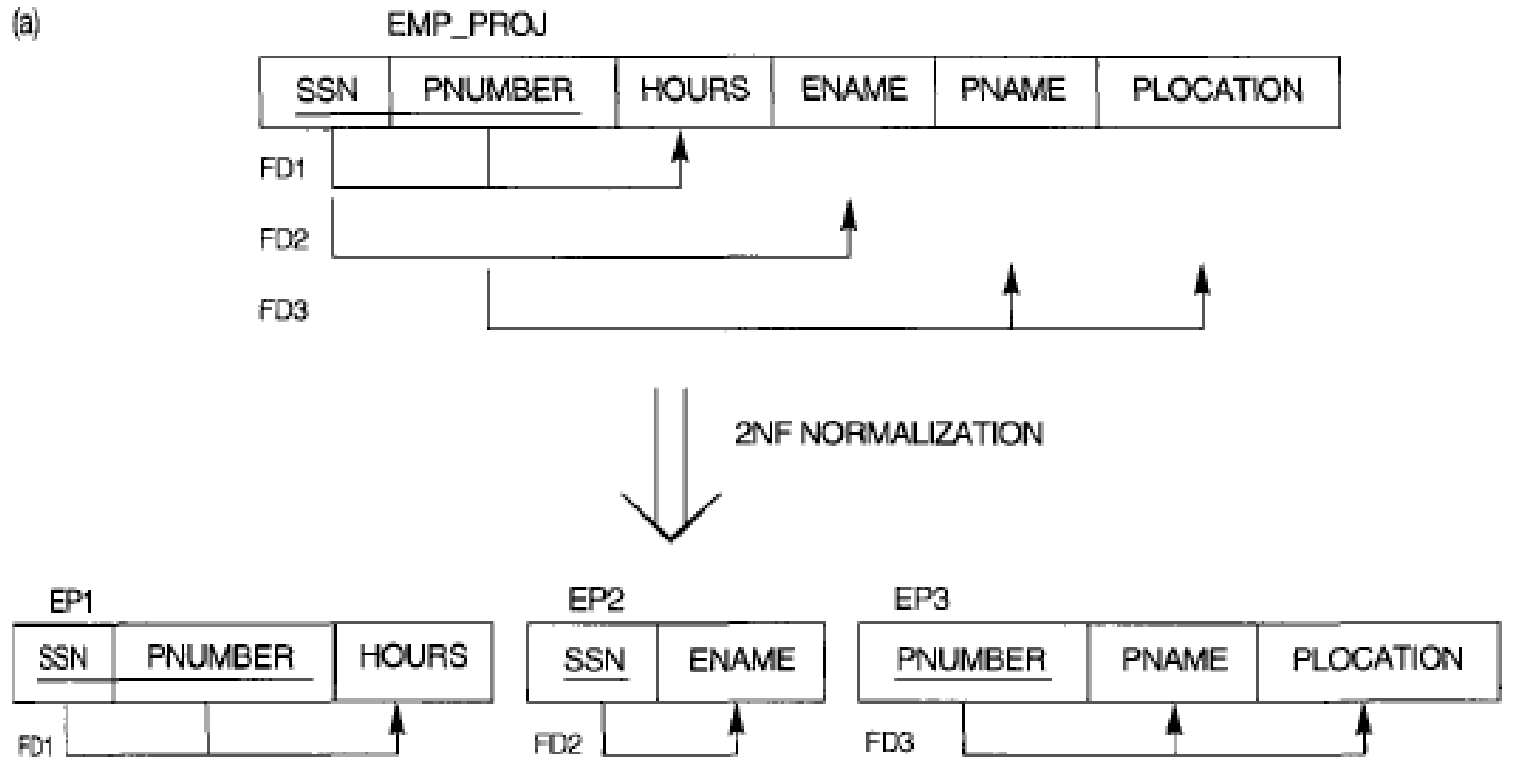
A relation schema R is in 2NF if every nonprime attribute A in R is *fully functionally dependent* on the primary key of R.

If the primary key contains a single attribute, the test need not be applied at all.

SECOND NORMAL FORM – 2NF (4/5)



SECOND NORMAL FORM – 2NF (5/5)



Third Normal Form – 3NF (1/4)

Third normal form (3NF) is based on the concept of *transitive dependency.*

A functional dependency X → *Y* in a relation schema R is a transitive dependency if there is a set of attributes Z that is neither a candidate key nor a subset of any key of R, and both X → Z and Z →Y hold.

Third Normal Form – 3NF (2/4)

The dependency SSN → DMGRSSN is transitive through DNUMBER in EMP_DEPT because both the dependencies

SSN→DNUMBER

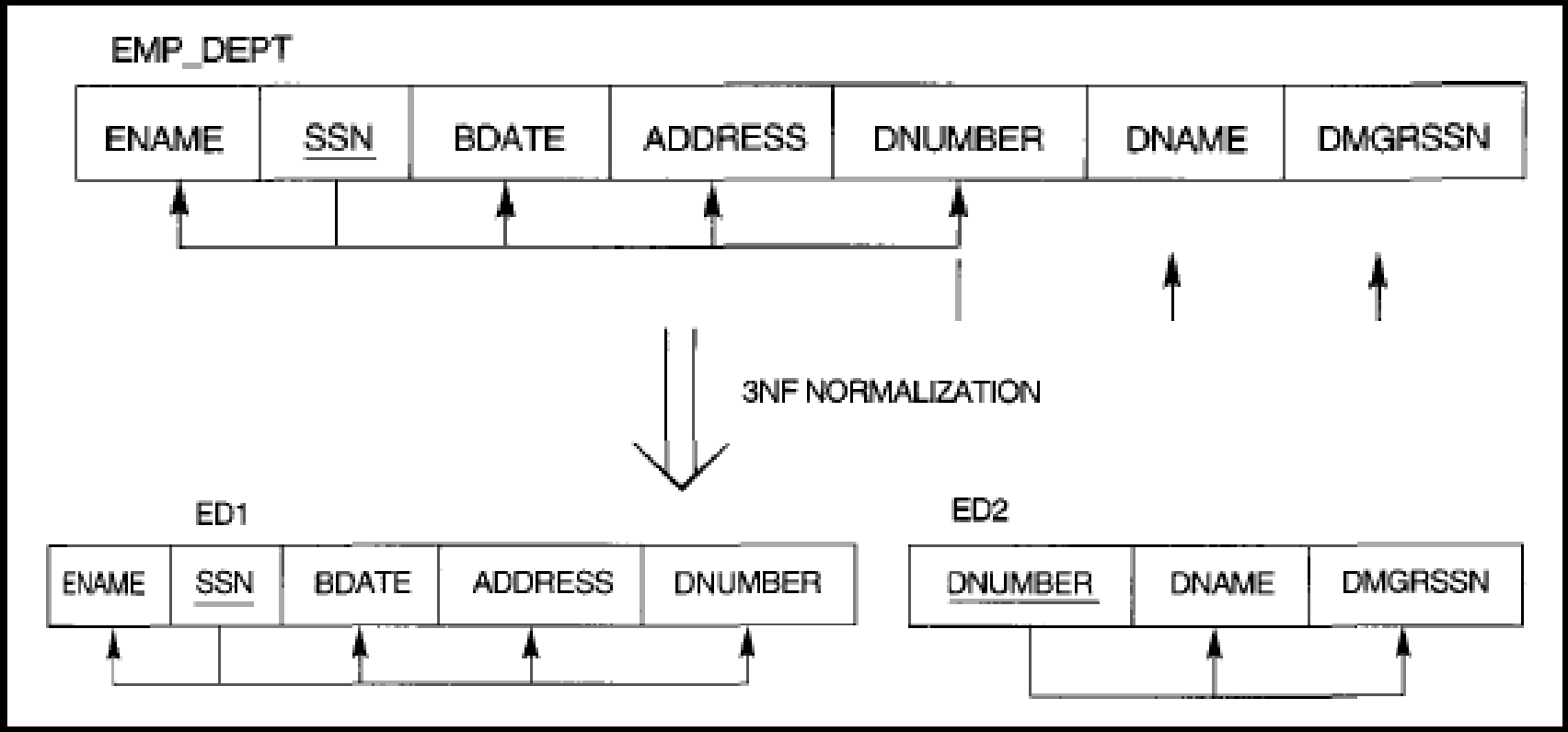
DNUMBER→DMGRSSN

hold *and* DNUMBER is neither a key itself nor a subset of the key of EMP_DEPT

Third Normal Form – 3NF (3/4)

A relation schema R is in 3NF if it satisfies 2NF *and* no nonprime attribute of R is transitively dependent on the primary key.

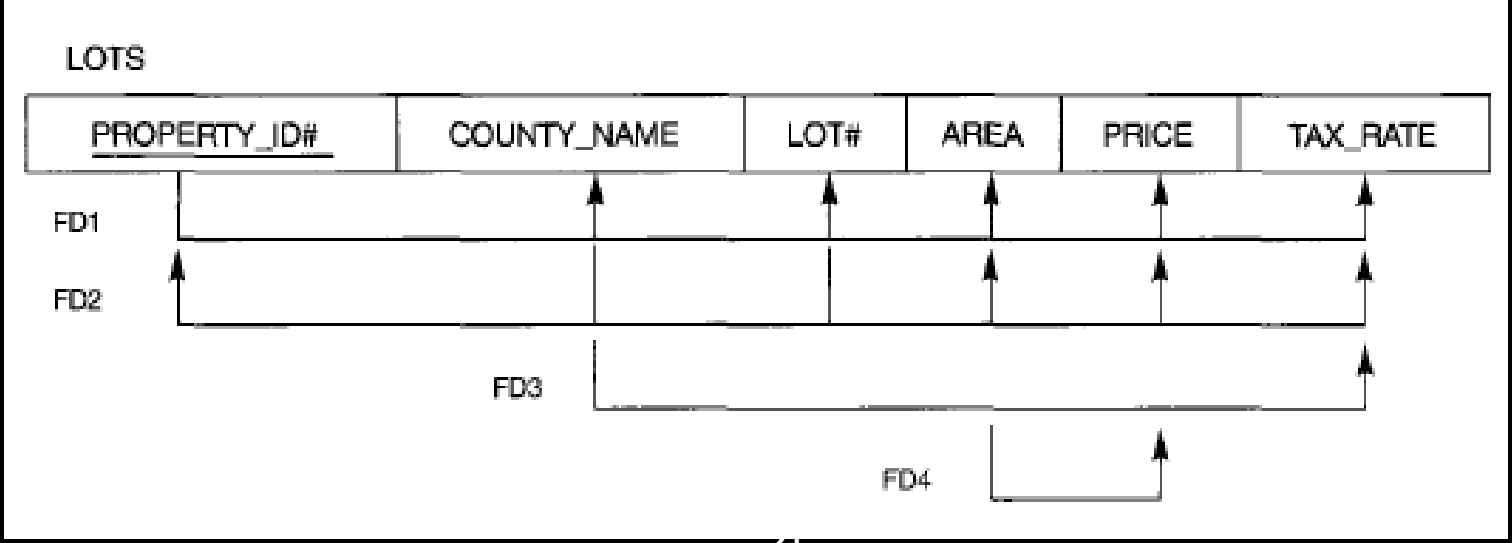
THIRD NORMAL FORM – 3NF (4/4)



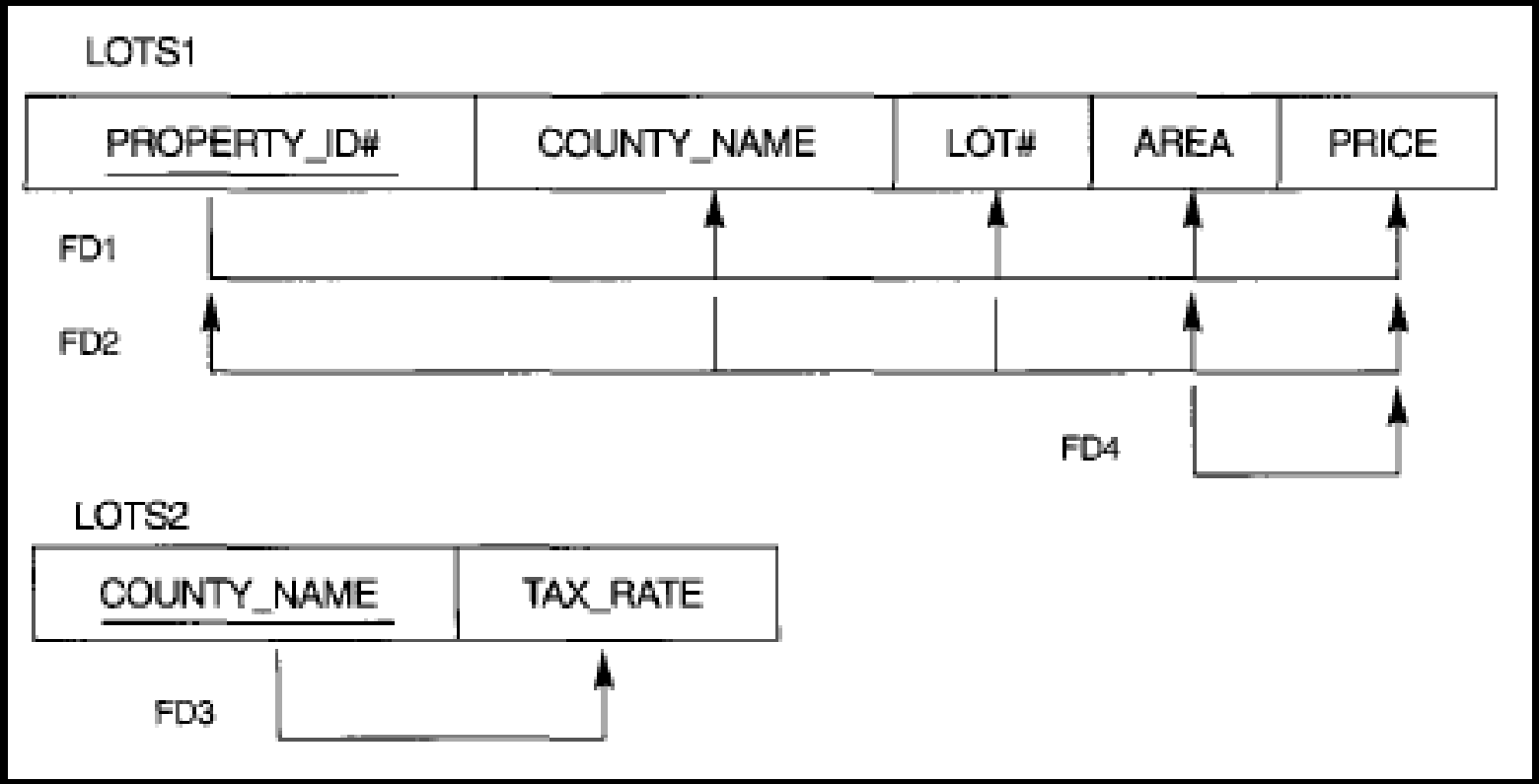
Example

Based on the two candidate keys PROPERTY_ID# and {COUNTY_NAME, LOT#}, we know that the FDs FD1 and FD2 hold.

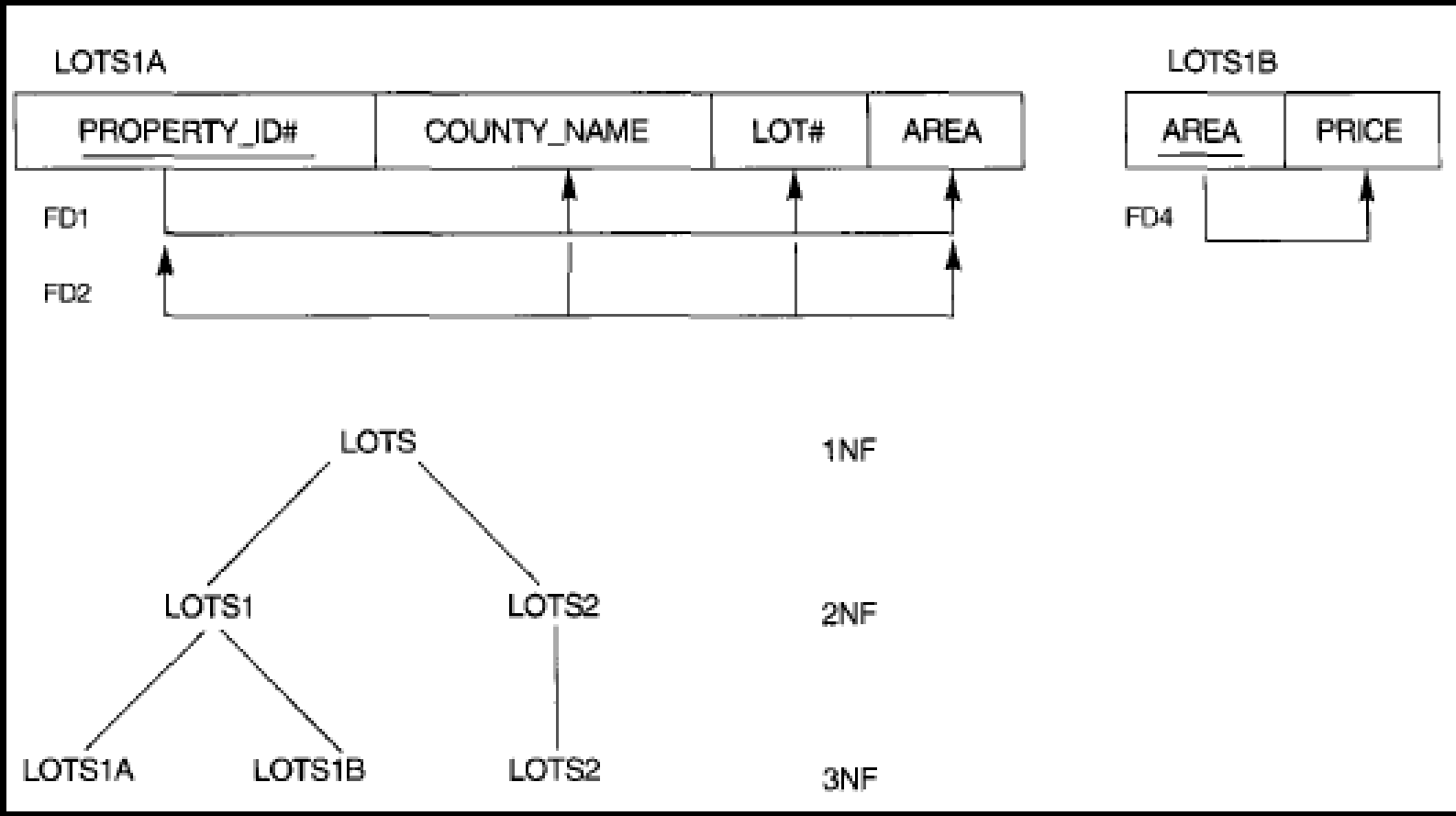
We choose PROPERTY_ID# as the primary key



EXAMPLE - CONVERT TO 2NF



EXAMPLE – CONVERT TO 3NF



Exercises

A college keeps details about a student and the various modules the student studied. These details comprise :

regno - registration number
n - student name
a - student address
tno - tutor number
tna - tutor name
dc - diploma code
dn - diploma name
mc - module code
mn - module name
res - module exam result

where

DETAILS (regno, n, a, tno, tna, dc, dn, (mc,mn,res))

dc -> dn

tno -> tna

mc,mn -> res

n -> a

mc -> mn

Reduce the relation DETAILS to third normal form.

Classify the following relations as either UNNORMALISED, 1NF, 2NF or 3NF. If the relation is not in 3NF, normalise the relation to 3NF.

EMPLOYEE(empno,empname,jobcode)

empno -> empname

empno -> jobcode

EMPLOYEE(empno,empname,(jobcode,years))

empno -> empname

empno, jobcode -> years

EMPLOYEE (empno, empname, jobcode, jobdesc)

empno -> empname, jobcode

jobcode -> jobdesc

EMPLOYEE(empno, empname, project, hoursworked)

empno -> empname

empno, project -> hoursworked