Functional Dependencies and Normalization for Relational Databases

Unit 4

- two approaches of database design:
  - **Bottom up**: (design by synthesis)

    considers basic **relationships among attributes** as starting point to construct relation schemas.
  - **Top down**: (design by analysis)

    starts with a number of **groups of attributes** into relations **that exist together naturally** and **further decomposing them** until certain properties are met.

Criteria for "good" base relations:

- **Informal Design Guidelines for Relational Databases**
  1. Clear Semantics of the attributes
  2. Reducing redundant information in tuples
  3. Reducing the NULL values in tuples
  4. Disallowing the possibility of generating spurious tuples
- **Formal design guidelines**:
  - Functional dependencies
  - normal forms (normalization) 1NF, 2NF,3NF,BCNF

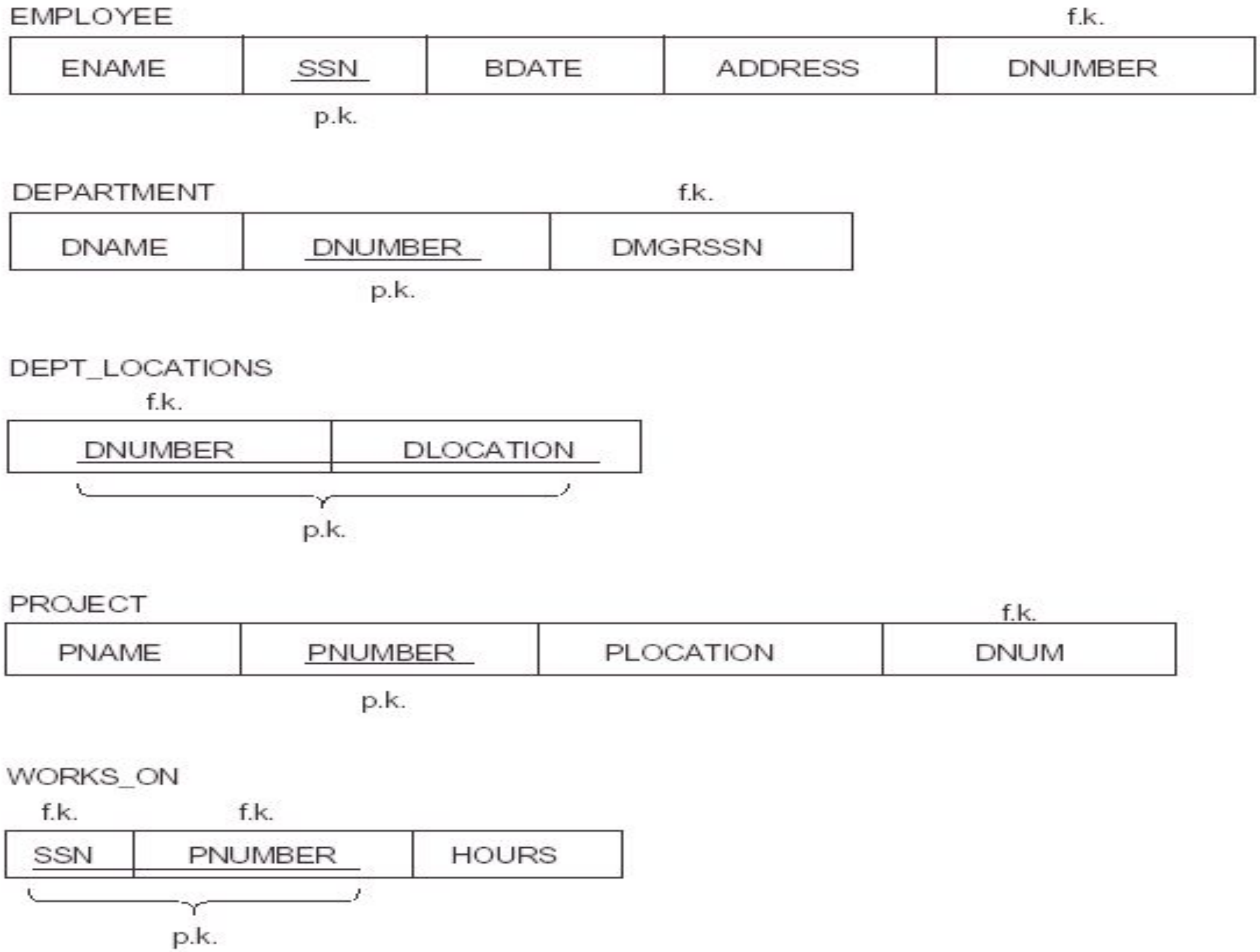
Guideline 1) Clear Semantics to the Relation Attributes

- **Each tuple in a relation should represent one entity or relationship instance**
- "Semantics" refers to interpretation of attribute values in a tuple.
  - Only foreign keys should be used to refer to other entities
  - Entity and relationship attributes should be kept apart as much as possible
- **Guideline 1: Design a relational schema so that it is easy to explain its meaning.**

  Do not combine attributes from multiple entity types and relationship types into a single relation. ex: fig; 10.1 and 10.2

FIG 10.1

Simplified version of the COMPANY relational database schema.



EMPLOYEE

f.k.

ENAME	SSN	BDATE	ADDRESS	DNUMBER

p.k.

DEPARTMENT

f.k.

DNAME	DNUMBER	DMGRSSN

p.k.

DEPT_LOCATIONS

f.k.

DNUMBER	DLOCATION

p.k.

PROJECT

f.k.

PNAME	PNUMBER	PLOCATION	DNUM

p.k.

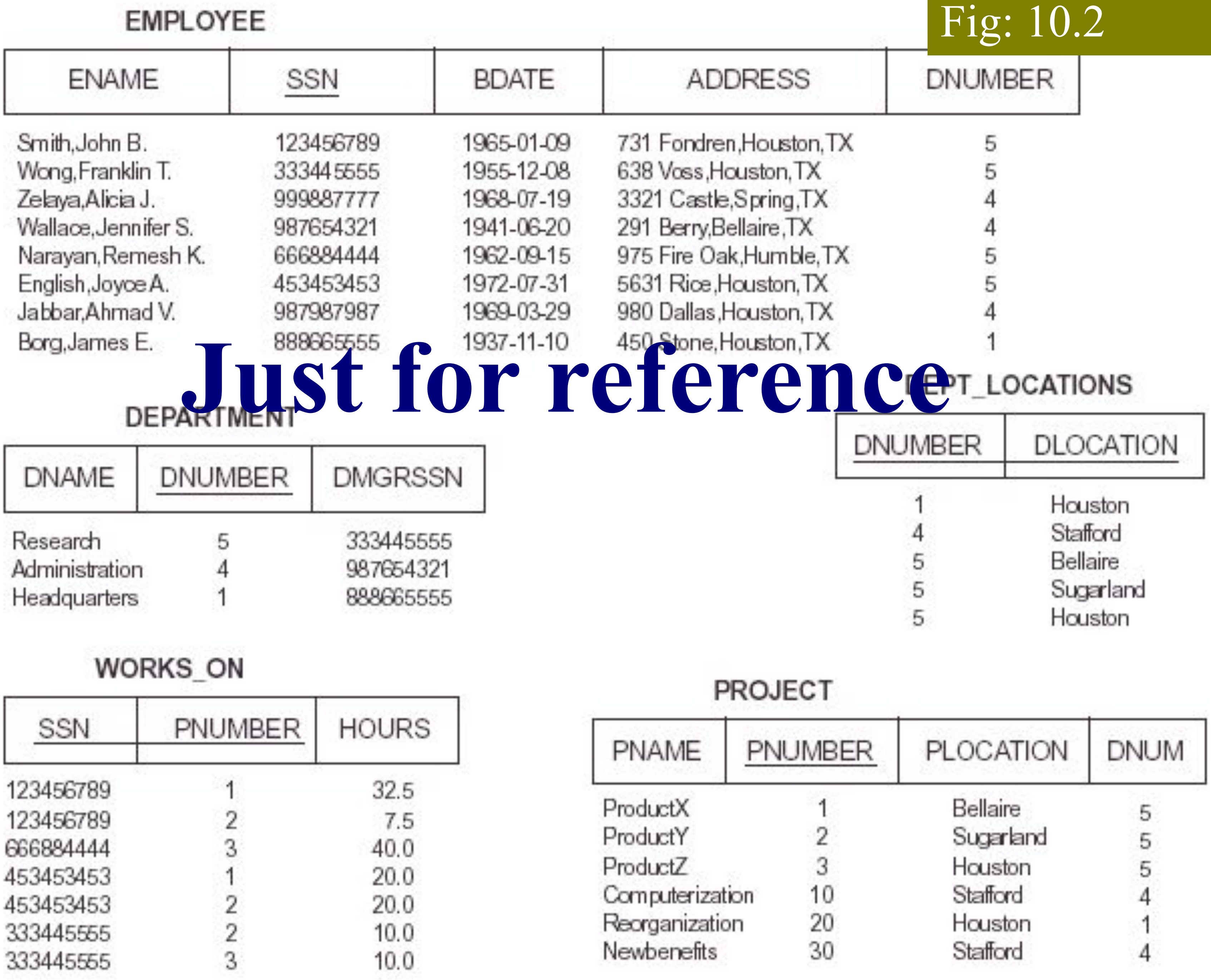
WORKS_ON

f.k. f.k.

SSN	PNUMBER	HOURS

p.k.

Fig: 10.2



Just for reference

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,Remesh 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

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

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

Guideline 2: Reducing redundant information in tuples

- **Definition:**

  Design a schema that does not suffer from the insertion, deletion and update anomalies.

  If there are anomalies, document them and check that applications that update the DB operate correctly.

Redundant Information in Tuples and Update Anomalies

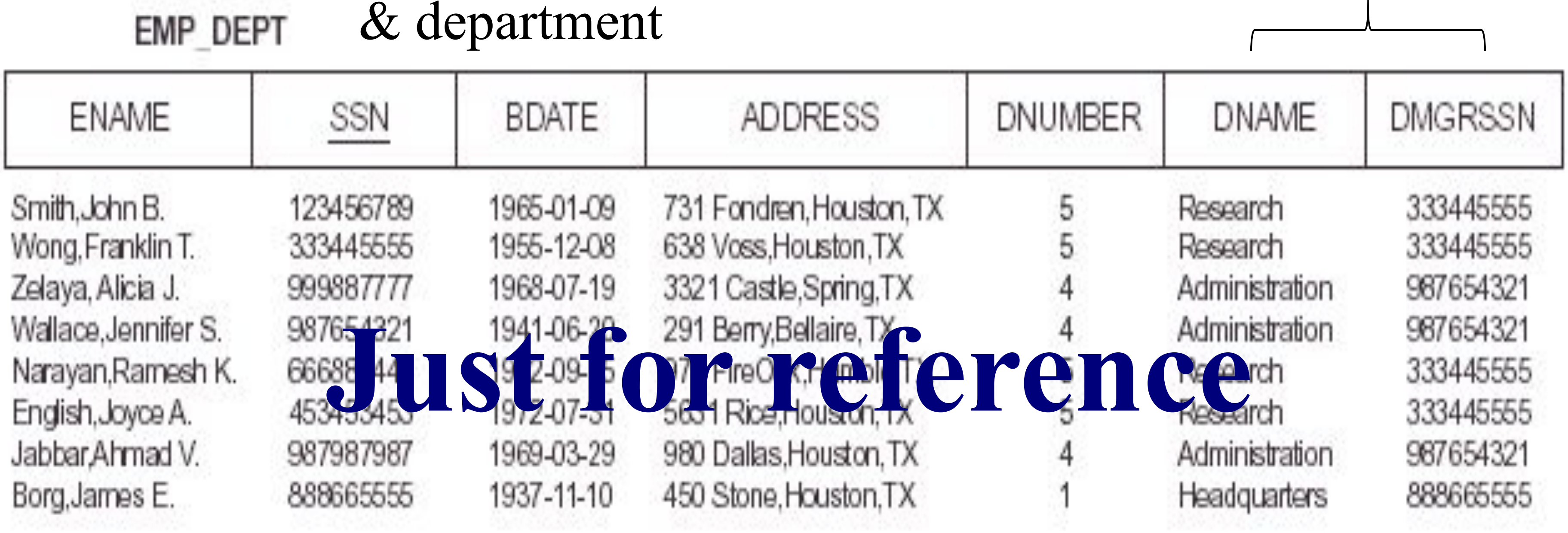
- Mixing attributes of multiple entities may cause problems
  - Information is stored redundantly wasting storage space.

    ex: compare employee and department tables

    (fig:10.2) and emp_dept schema(fig: 10.4)
  - due to this, Problems of update anomalies:
    - Insertion anomalies
    - Deletion anomalies
    - Modification anomalies

Result of natural join on employee & department

redundancy



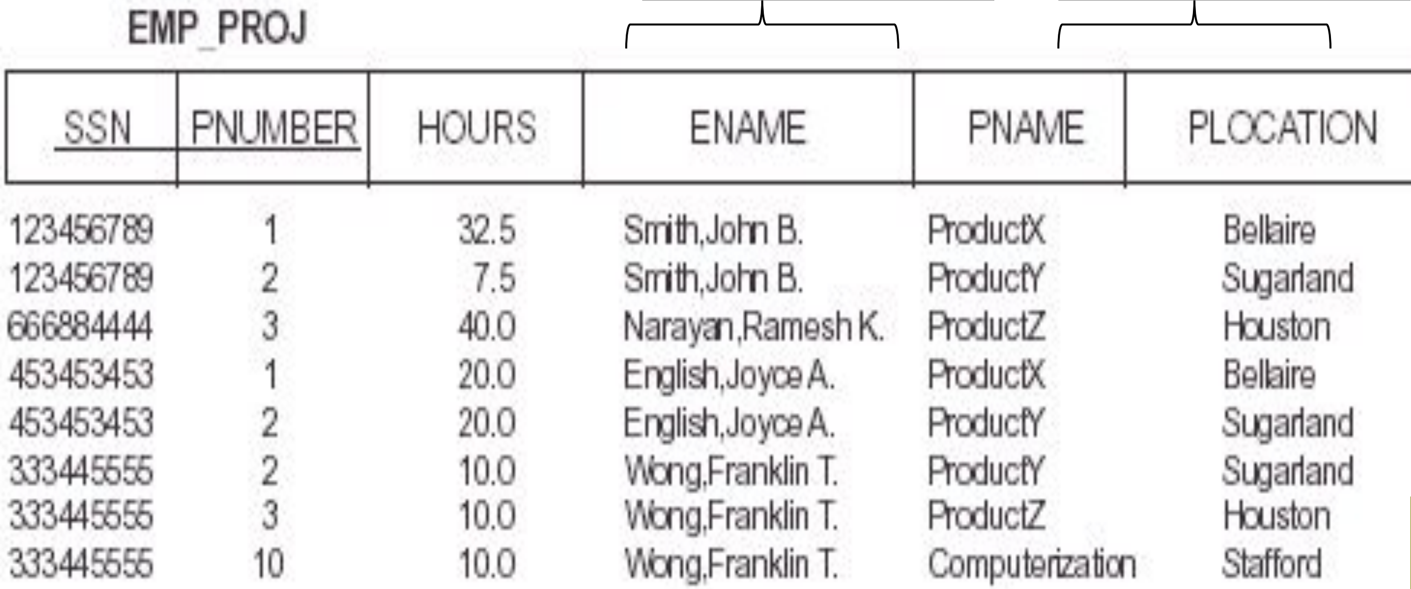
EMP_DEPT

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

Just for reference

redundancy

redundancy



EMP_PROJ

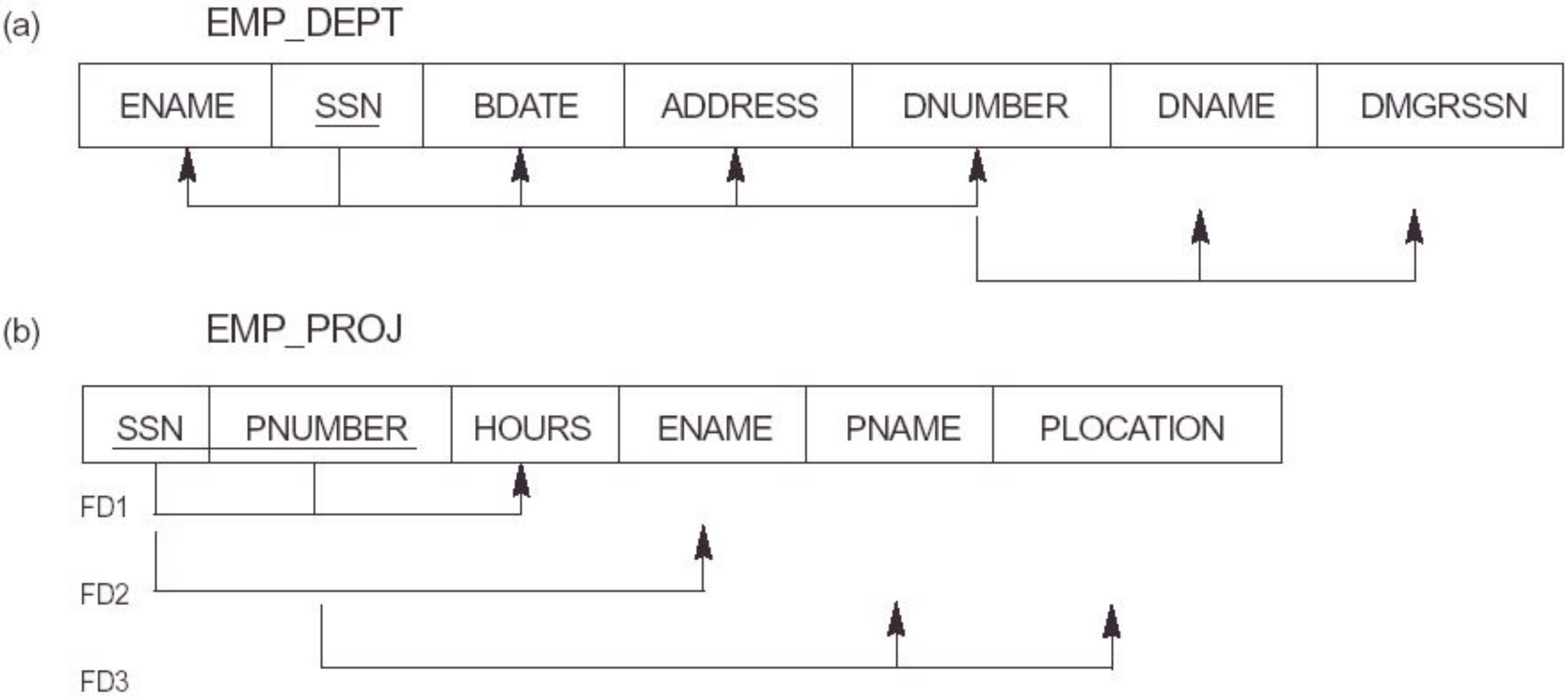
SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION
123456789	1	32.5	Smith,John B.	ProductX	Bellaire
123456789	2	7.5	Smith,John B.	ProductY	Sugarland
666884444	3	40.0	Narayan,Ramesh K.	ProductZ	Houston
453453453	1	20.0	English,Joyce A.	ProductX	Bellaire
453453453	2	20.0	English,Joyce A.	ProductY	Sugarland
333445555	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

Fig: 10.4

- **Insertion anomalies** are issues that come about when you are inserting information into the database for the first time. Exa: **Missing or incorrectly formatted entries**
- To insert the information into the table, we must **enter the correct details** so that they are consistent with the values for the other rows.

- **Insert Anomaly example**:
  - Cannot insert a new employee tuple(who has not been assigned a dept as yet) into emp_dept table unless he/she belongs to a department. Else we have to keep attribute values of dept NULL.

Fig: 10.3: tables emp_dept and emp_proj suffering from Update Anomalies



(a) EMP_DEPT

ENAME	SSN	BDATE	ADDRESS	DNUMBER	DNAME	DMGRSSN

(b) EMP_PROJ

SSN	PNUMBER	HOURS	ENAME	PNAME	PLOCATION

FD1

FD2

FD3

example

- **Deletion anomalies** are issues with data being deleted, either when attempting to delete and being stopped by an error or by the unseen drop off of data.
- If we delete a row from the table that represents the last piece of data, the details about that piece are also lost from the Database.
  - **Delete Anomaly ex:**
    - when we DELETE the last employee in a department from EMP_DEPT, we loose information on department.

- **Modification anomalies or Update** anomalies are data inconsistencies that result from data redundancy or partial update.
- The Problems resulting from data redundancy in database table are known as update anomalies.
- If a modification is not carried out on all the relevant rows, the database will become inconsistent.
  - **Update Anomaly ex:**
    - in emp_dept table, If we change an attribute of a department say manager, hence we must change in multiple places(i.e for all employees working for that dept)

Guideline 3: Reducing the NULL values in tuples

Definition:

Relations should be designed such that their tuples will have as few NULL values as possible.

In unavoidable cases see that null values apply to some tuples only.

- Null values Waste storage
- Null values are Harder to understand
- Reasons for nulls:
  - attribute not applicable or invalid
  - attribute value unkown (may exist)
  - value known to exist, but unavailable
- Hence , Attributes that are NULL frequently could be placed in separate relations (with the primary key)

Guideline 4) Disallowing the possibility of generating spurious tuples

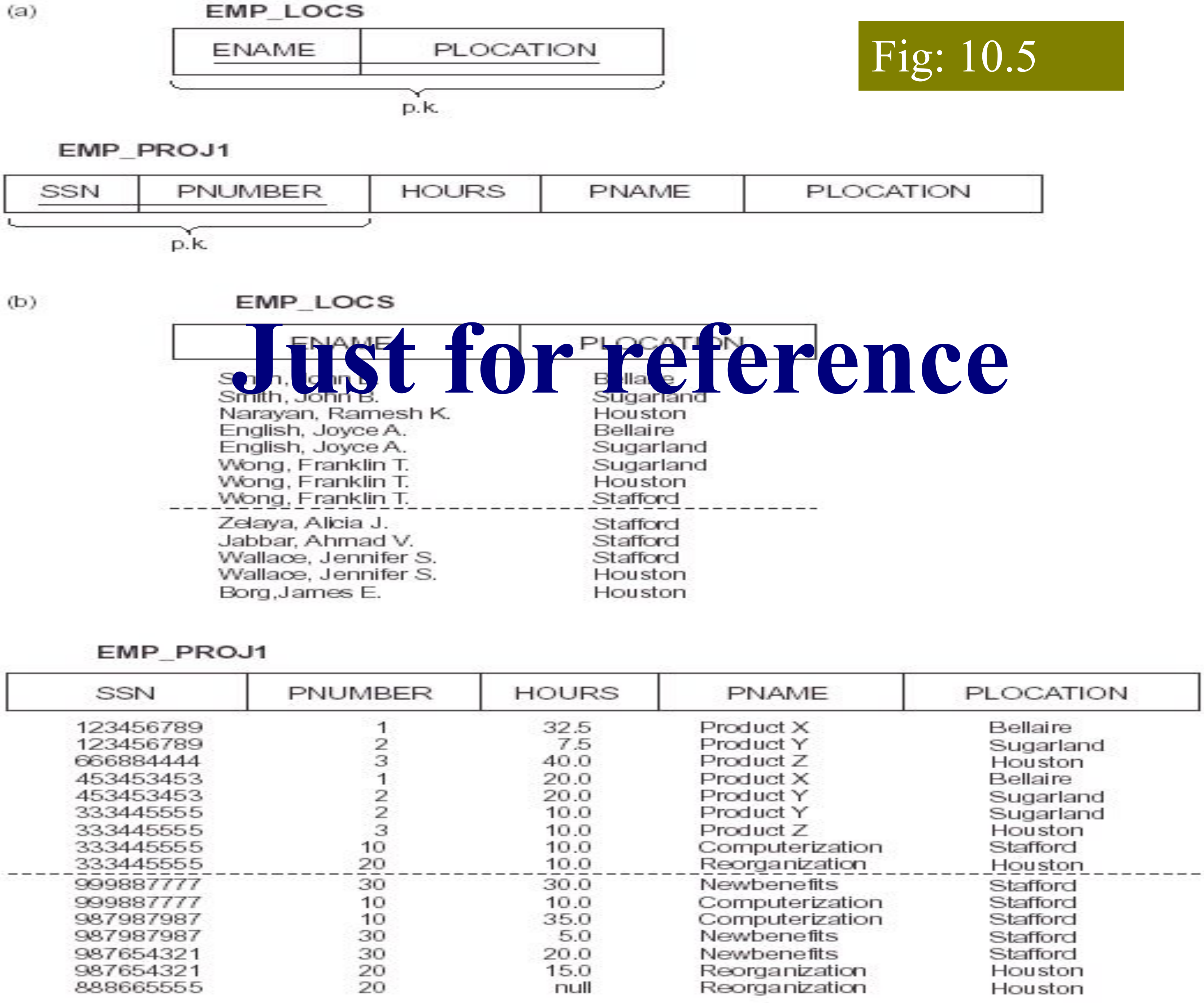
- **Definition:**

  Design relations such that they can be joined with equality conditions on attributes that are (primary key, foreign key) pairs so that no spurious tuples are generated.

  i.e. The relations should be designed to satisfy the lossless join condition.

- **Spurious tuples represent information that is not valid**
- Avoid relations that contain matching attributes that are not (primary key, foreign key) pairs because they result in spurious tuples.
- The "lossless join" property is used to guarantee meaningful results for join operations.
- ex: refer fig: 10.5 and 10.6 where decomposing a relation EMP_proj into emp_locs and emp_proj1 resulted in spurious tuples.

Fig: 10.5



(a)

EMP_LOCS

ENAME	PLOCATION

p.k.

EMP_PROJ1

SSN	PNUMBER	HOURS	PNAME	PLOCATION

p.k.

Just for reference

(b)

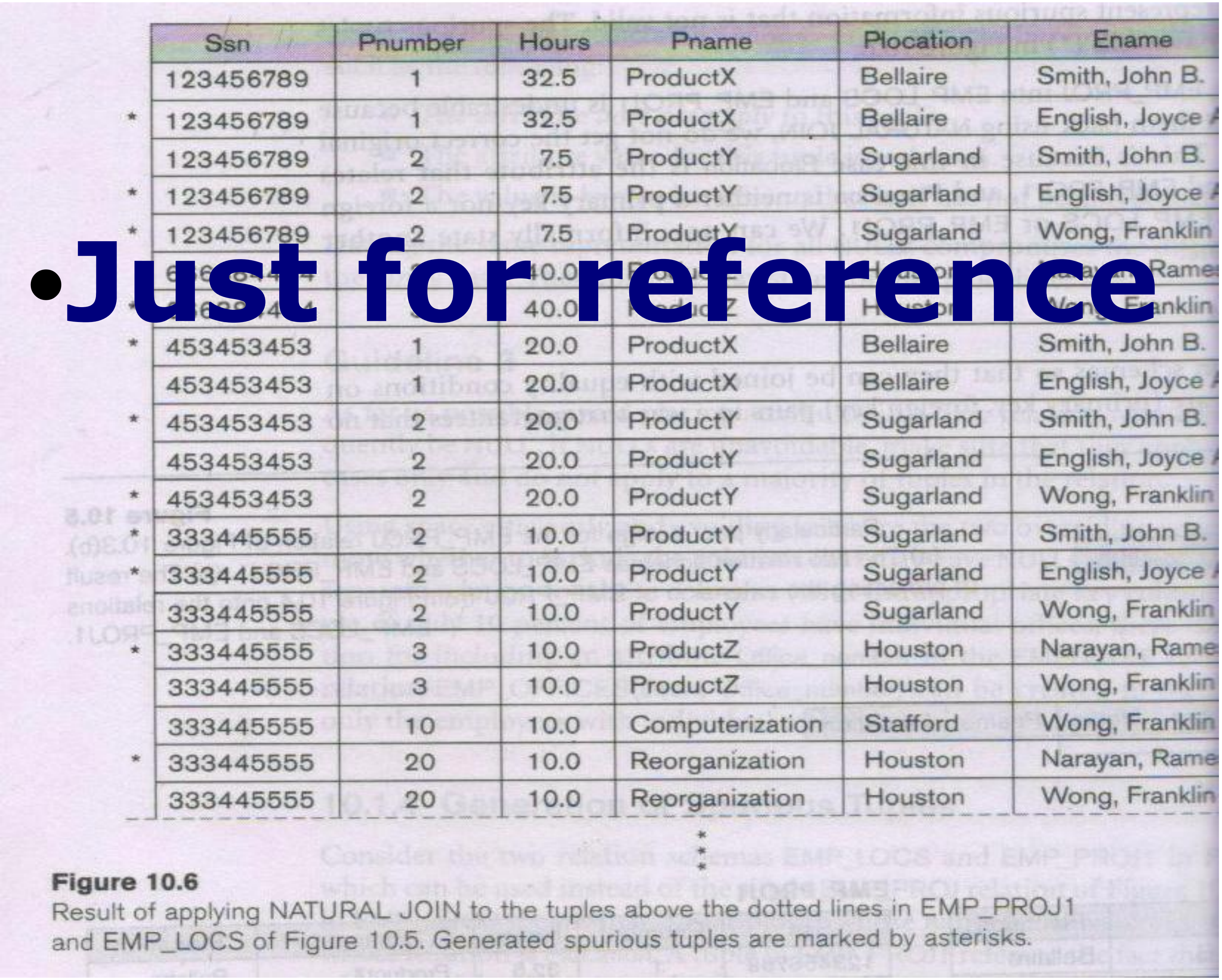
EMP_LOCS

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

- **Just for reference**



	Ssn	Pnumber	Hours	Pname	Plocation	Ename
	123456789	1	32.5	ProductX	Bellaire	Smith, John B.
*	123456789	1	32.5	ProductX	Bellaire	English, Joyce A
	123456789	2	7.5	ProductY	Sugarland	Smith, John B.
*	123456789	2	7.5	ProductY	Sugarland	English, Joyce A
*	123456789	2	7.5	ProductY	Sugarland	Wong, Franklin
	666884444	3	40.0	ProductZ	Houston	Narayan, Rames
*	666884444	3	40.0	ProductZ	Houston	Wong, Franklin
*	453453453	1	20.0	ProductX	Bellaire	Smith, John B.
	453453453	1	20.0	ProductX	Bellaire	English, Joyce A
*	453453453	2	20.0	ProductY	Sugarland	Smith, John B.
	453453453	2	20.0	ProductY	Sugarland	English, Joyce A
*	453453453	2	20.0	ProductY	Sugarland	Wong, Franklin
*	333445555	2	10.0	ProductY	Sugarland	Smith, John B.
*	333445555	2	10.0	ProductY	Sugarland	English, Joyce A
	333445555	2	10.0	ProductY	Sugarland	Wong, Franklin
*	333445555	3	10.0	ProductZ	Houston	Narayan, Rame
	333445555	3	10.0	ProductZ	Houston	Wong, Franklin
	333445555	10	10.0	Computerization	Stafford	Wong, Franklin
*	333445555	20	10.0	Reorganization	Houston	Narayan, Rame
	333445555	20	10.0	Reorganization	Houston	Wong, Franklin

*
*
*

Figure 10.6
Result of applying NATURAL JOIN to the tuples above the dotted lines in EMP_PROJ1 and EMP_LOCS of Figure 10.5. Generated spurious tuples are marked by asterisks.

Various keys

- **a superkey** is a set of attributes within a table whose values can be used to uniquely identify a tuple.

  Ex: {employeeID, Name}, {employeeID, Name, job}, and {employeeID, Name, job, departmentID}

- A **candidate key** is a minimal set of attributes necessary to identify a tuple. this is also called a minimal superkey.

Functional Dependencies

- A **functional dependency** is a **constraint between two sets of attributes** in a relation from a database**.**

- FDs are derived from the *meaning* , *interrelationships* and real world constraints on the data attributes
- It typically exists between the primary key and non-key attribute within a table.

- **FDs and keys** are used to define **normal forms** for relations.

Functional Dependencies

- **Given a relation *R*, a set of attributes *X* in *R* is said to functionally determine another attribute *Y*, also in *R*,**

  (written *X* → *Y*) if and only if, each *X* value is associated with precisely one unique *Y* value.

  Here X is called as **determinant** and Y is called as ***dependant*** and we read X□ Y as X determines 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)

Examples of FD constraints

- Social Security Number determines employee name

  SSN → ENAME

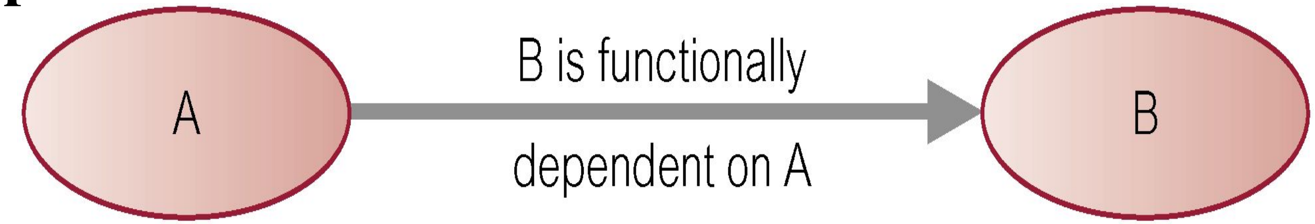
- Project Number determines project name and location

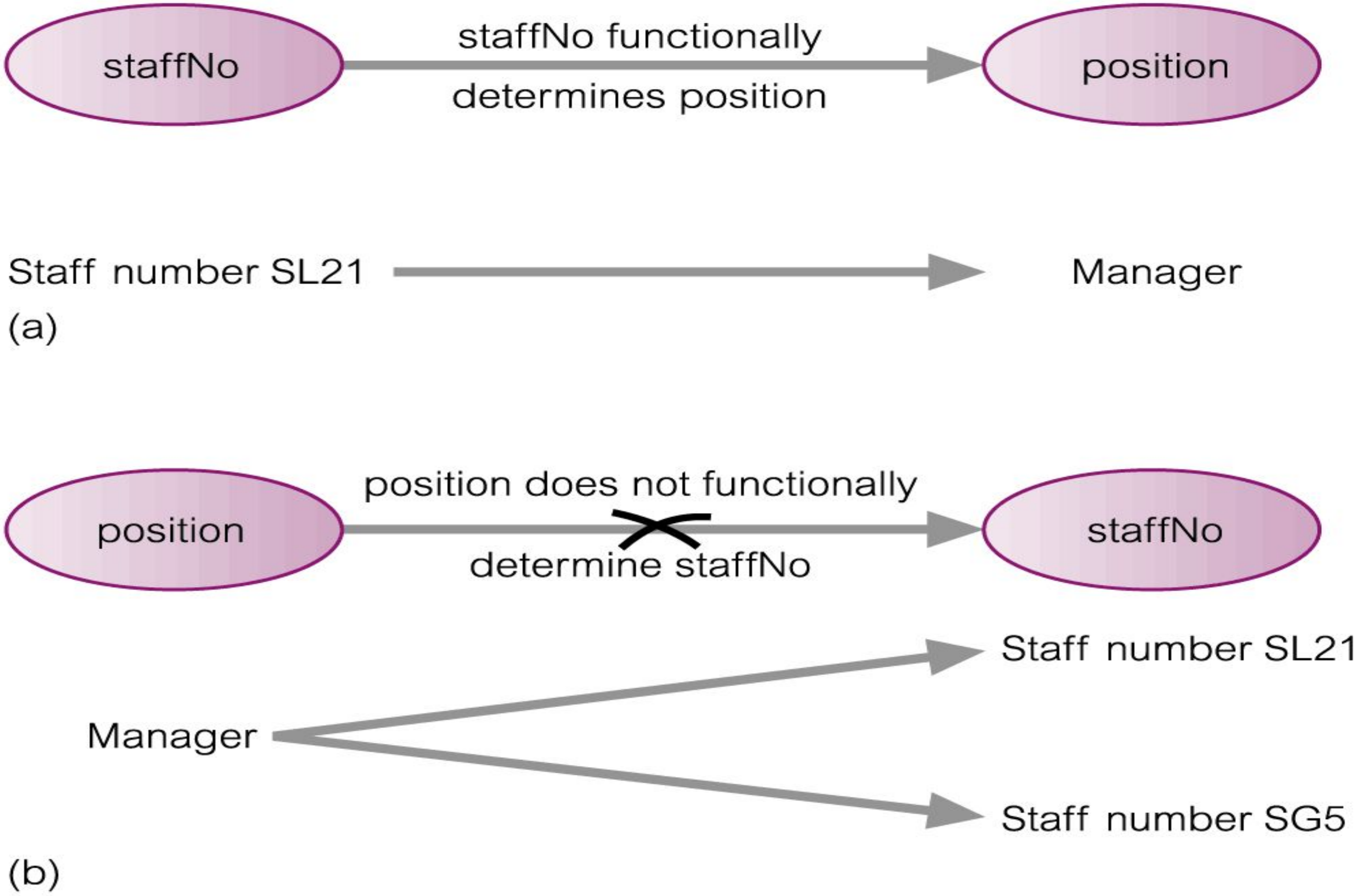
  PNUMBER → {PNAME, PLOCATION}

- Employee SSN and project number determines the hours per week that the employee works on the project

  {SSN, PNUMBER} → HOURS

Example


A
B is functionally
dependent on A
B


staffNo
staffNo functionally
determines position
position
Staff number SL21
Manager
(a)
position
position does not functionally
determine staffNo
staffNo
Manager
Staff number SL21
Staff number SG5
(b)

Types of functional dependency :

1. 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;

   Ex: {Ssn, Pnumber} → Hours is a full dependency (neither Ssn → Hours nor

   Pnumber → Hours

2. A functional dependency X → Y is a **partial functional dependency** if removal of any attribute A from X and the dependency still holds;

3. A functional dependency X→Y in a relation schema R is a **transitive dependency ,**

if there exists a set of attributes Z in R such that X→Z and Z→Y hold.

Ex: The dependency Ssn→Dmgr_ssn is transitive in EMP_DEPT, because of the dependencies

Ssn → Dnumber and Dnumber → Dmgr_ssn.

And hence Ssn→Dmgr_ssn

4. **Trivial Functional Dependency**.

If a functional dependency (FD) X → Y holds,

where Y is a subset of X, then it is called a trivial FD.

Non-trivial − If an FD X → Y holds, where Y is not a subset of X, then it is called a non-trivial FD.

Advantages of FD

- Data normalization
- Query Optimization
- Consistency of data
- Data Quality improvement

History of normalization

Edgar F. Codd first proposed the process of normalization and what came to be known as the **1st normal form** in his paper *A Relational Model of Data for Large Shared Data Banks*.

Edgar F. Codd originally established three normal forms: 1NF, 2NF and 3NF.

3NF is widely considered to be sufficient for most applications. Most tables when reaching 3NF are also in BCNF (Boyce-Codd Normal Form).

Introduction to Normalization

- **Normalization:”** Process of **analyzing and decomposing** unsatisfactory "bad" relations based on their **primary key and FD's** by breaking up their attributes into smaller relations.”

- **Normal form**: refers to the highest 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

Why normalization?

- It is carried out for **removing redundant data from tables** in order to improve storage efficiency, data integrity and scalability.
- Normalization generally involves **splitting existing tables into multiple ones**, which must be re-joined or linked each time a query is issued.

First Normal Form

- **“ It 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. “**

- Hence 1NF does not allow composite attributes, multivalued attributes, and nested relations

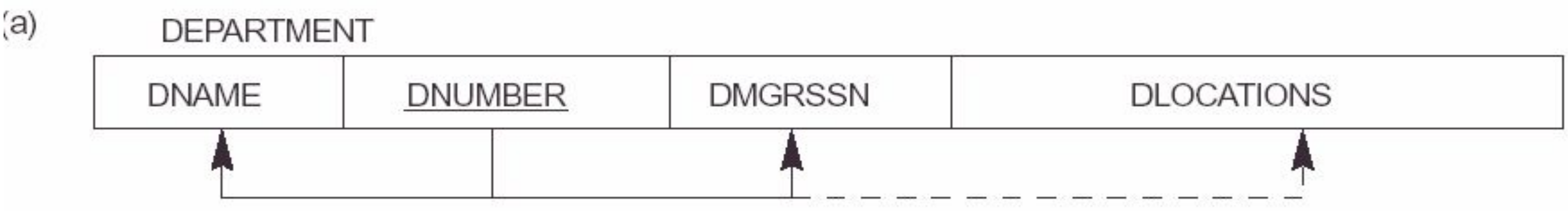
Techniques to achieve 1NF

Technique 1: remove the attribute from the relation that violates the 1NF and place it in a separate relation.

Technique 2: Expand the key so that there will be a separate tuple in the original relation for each value of multivalued attribute. This method usually introduces redundancy.

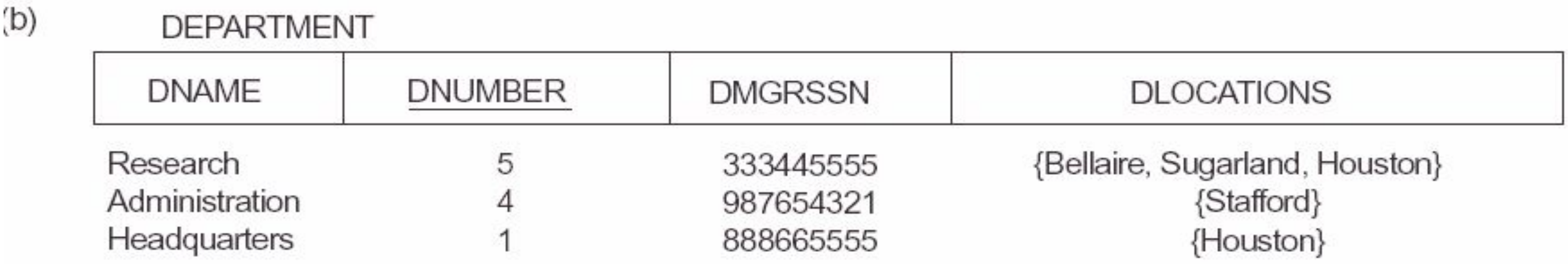
Technique 3: if a maximum number of values (domain) is known for the attribute then replace the attribute by that many number of atomic attributes. This usually introduces NULL values.

1NF Technique 1:



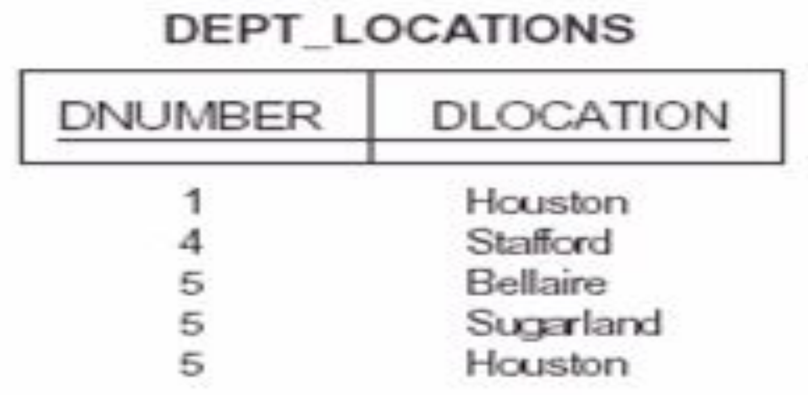
(a) DEPARTMENT

DNAME	DNUMBER	DMGRSSN	DLOCATIONS



(b) DEPARTMENT

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



DEPT_LOCATIONS

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

1NF technique 2:

Consider the diagram in next slide.

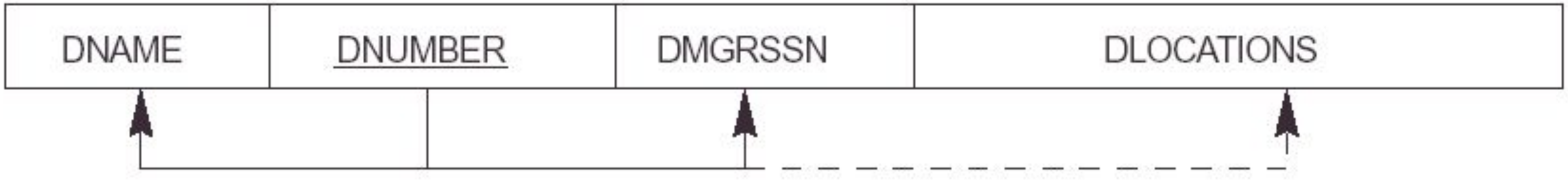
Fig (a) is not in 1NF. Hence it is normalized and converted to 1NF using the 2nd technique from previous slide.

fig (b) is the snapshot of fig (a).

However fig (c) is in 1NF.

1 NF technique 2:

(a) DEPARTMENT



DNAME	DNUMBER	DMGRSSN	DLOCATIONS

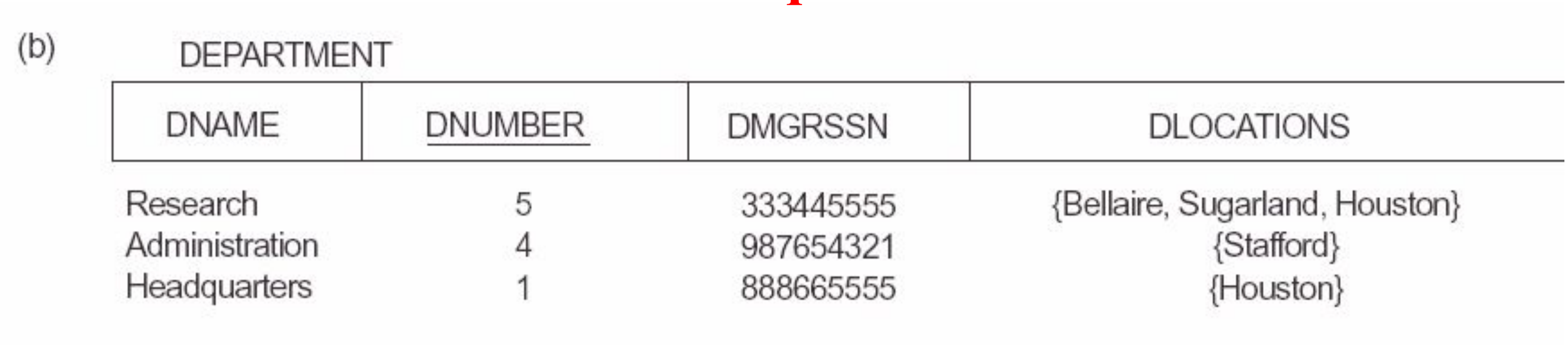
(b) DEPARTMENT

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

1 NF technique 3:



(b) DEPARTMENT

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

1NF

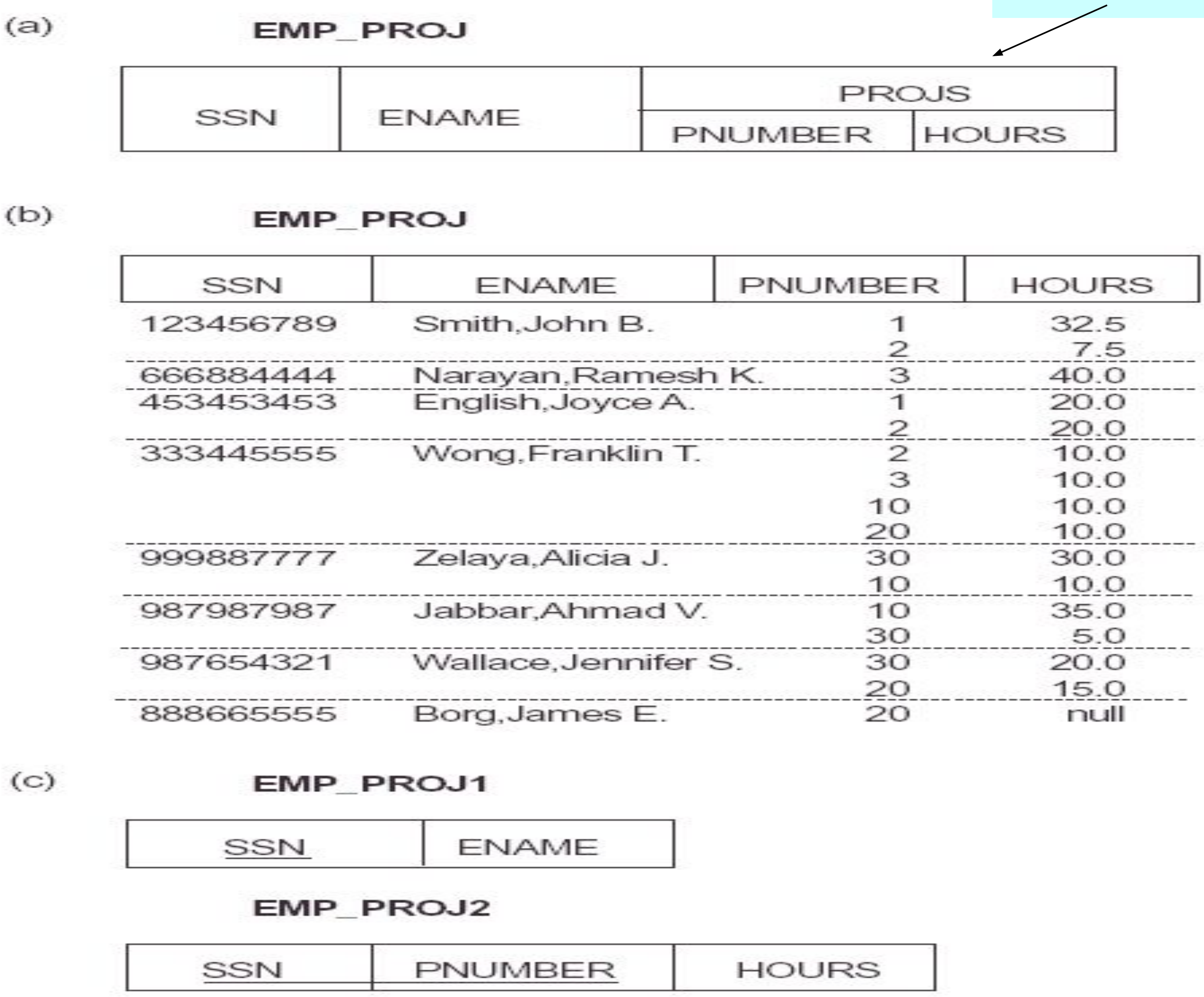
DNAME	DNUMBER	DMGRSSN	DLOC1	DLOC2	DLOC3
Research	5	333445555	Bellaire	Sugarland	houstan
Administration	4	987654321	Stafford	Null	Null
Headquaters	1	888665555	Houstan	Null	null

1NF also does not allow nested relations.

Hence the nested relation attributes are moved to new relation and the primary key of the original table is propagated into it.

hence the primary key of this new relation is the combination of the partial key of this relation and primary key of original relation.

ex: refer next slide.



Nested relation

(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 and primary key
- <u>Definitions:</u>
  - **Prime attribute** - attribute that is member of the primary key K
  - **Full functional dependency** - a FD Y Z where removal of any attribute (Y-{A}) from Y means the FD does not hold any more.
  - **Partial dependency:** a FD Y Z is a partial FD if some attribute AЄ Y can be removed from Y and the dependency still holds.

Examples of full FD and partial FD

- {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

- R can be decomposed into 2NF relations via the process of 2NF normalization

Definition of 2NF:

"**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 of R.** "

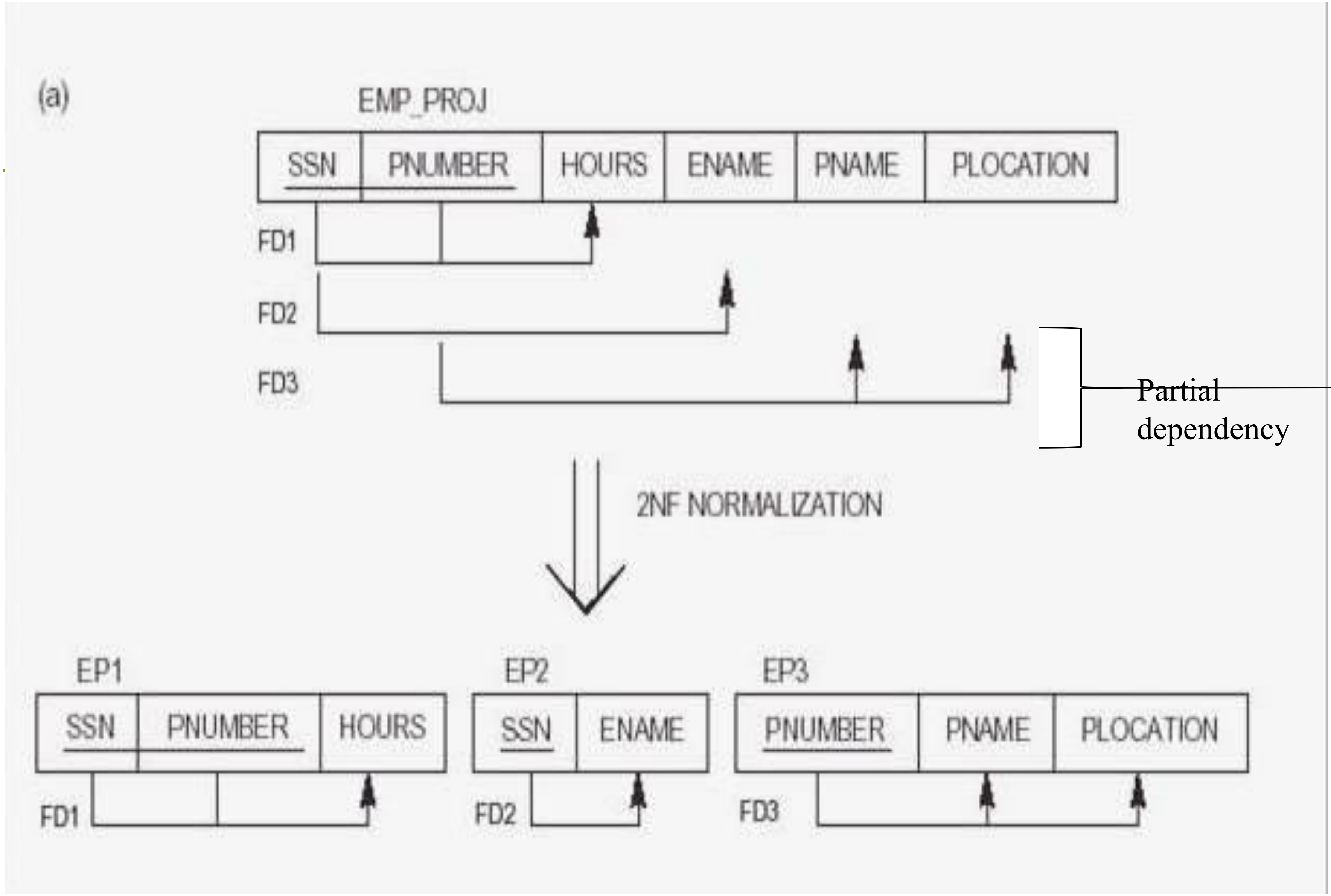
- 2NF is based on the concept of full functional dependency.
- To be in 2NF, a relation must be in first normal form and relation must not contain any partial dependency
- It applies to relations with composite keys (primary key composed of two or more attributes).
- A relation with a single-attribute primary key is automatically in at least 2NF.
- A relation that is not in 2NF may suffer from the update anomalies.
- 2NF tries to reduce redundant data stored in tables.

Steps to convert a table to 2NF

Step 1: identify the non-prime attributes that are related to only part of the key (primary key or partial key). i.e. find the partial dependencies

Step 2: decompose the relation schema into various relations that contain the above identified attributes, such that they are associated with the part of the key(primary or partial) on which they are fully functionally dependent.

Ex: consider the next slide where the **emp_proj** schema is decomposed into 3 tables EP1, EP2 and EP3. hence it is normalized to 2NF.



Partial dependency

Third Normal Form

- **Transitive functional dependency** –

a FD X → Y is transitive FD, if there is a set of attributes Z that are neither a subset of key nor candidate key and both X → Z and Z →Y holds.

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

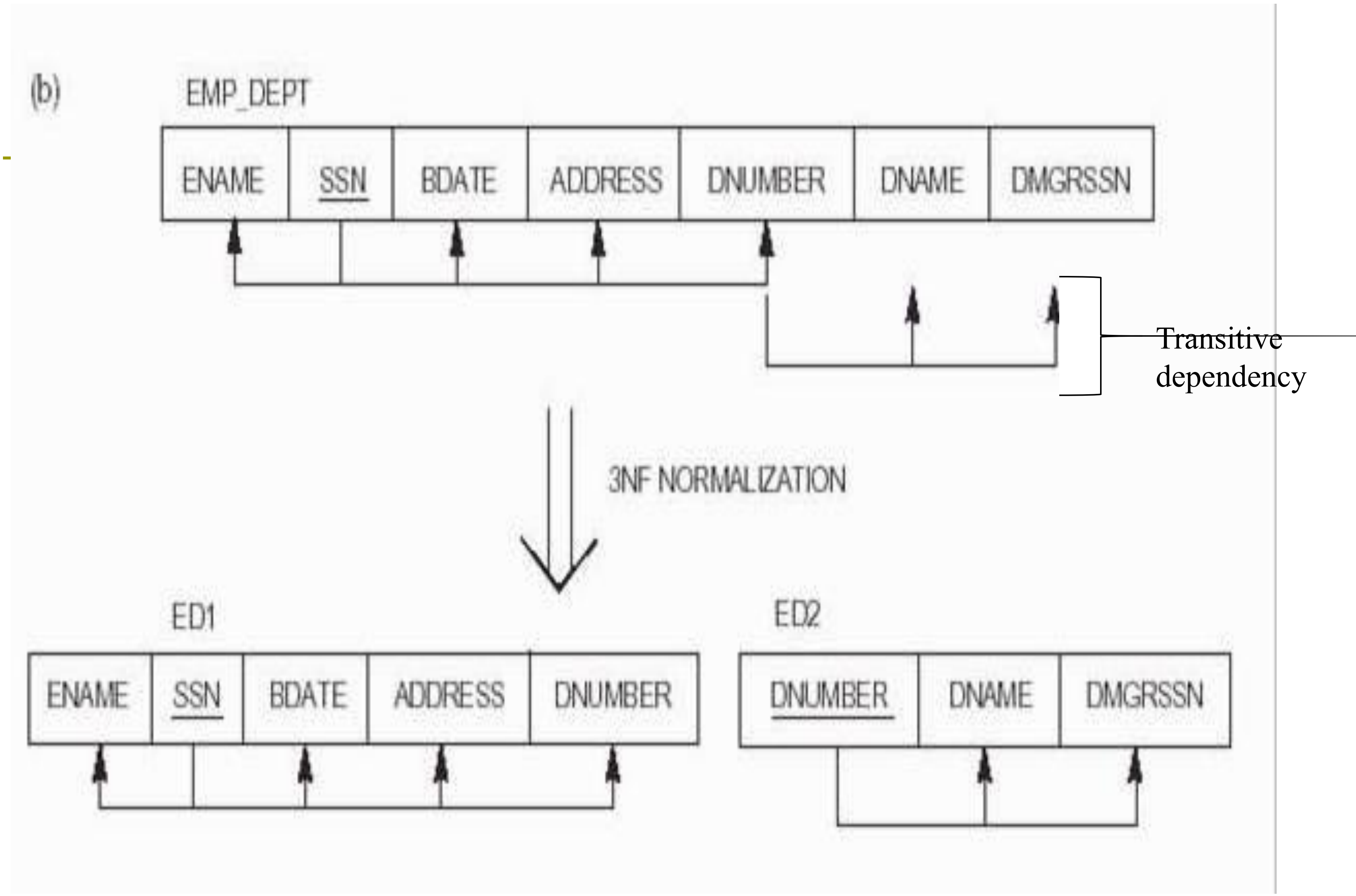
attributes X where SSN → X and X → ENAME

3rd Normal Form definiton

- " 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 "

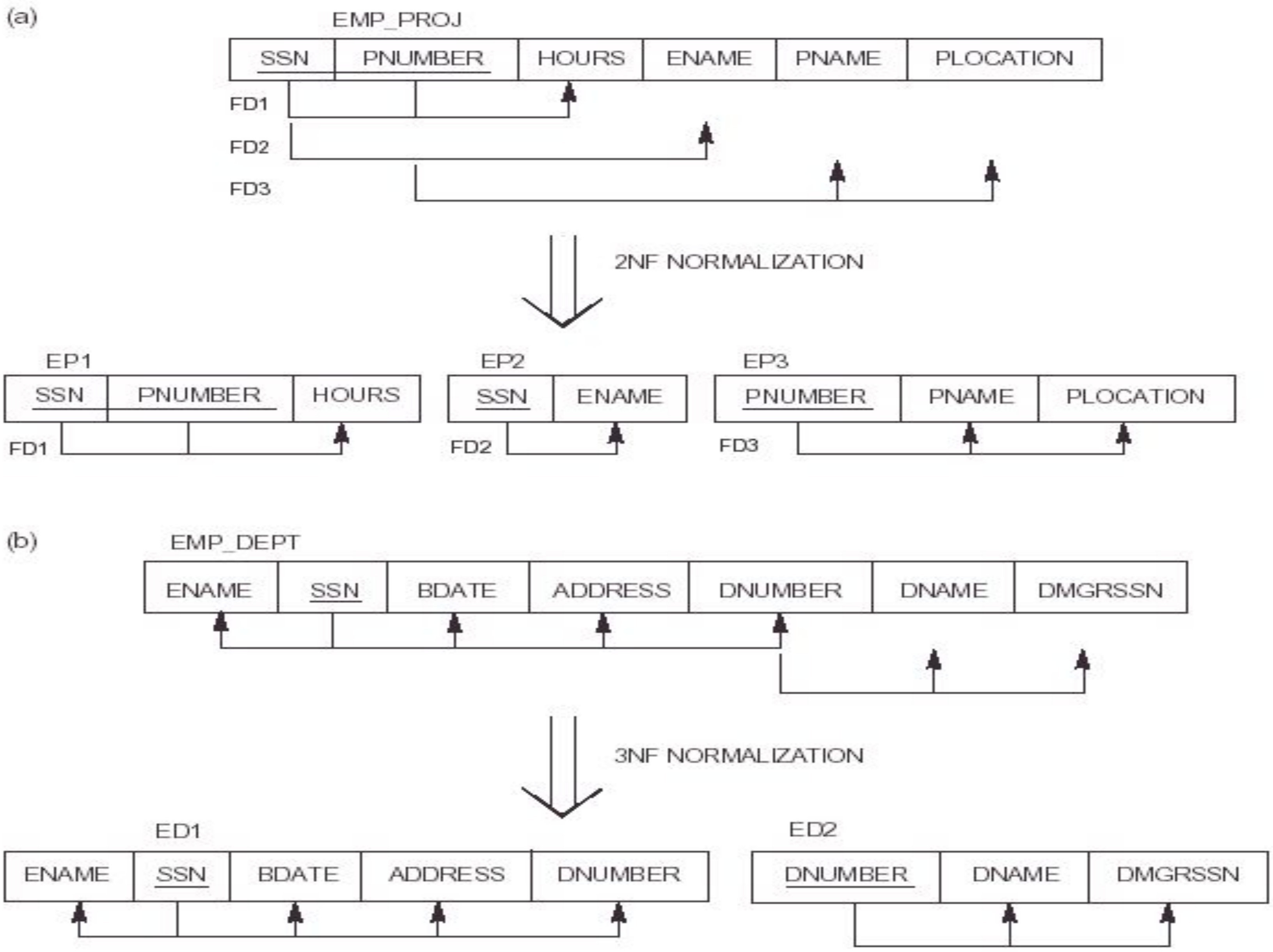
Steps to convert a table to 3NF

- Remove the transitive dependency from the table by splitting each relation into two separate relations and link them using foreign key.
- hence the two new relations should represent the independent facts about two different entities.
- also when we join the two new relations using natural join, we don’t get spurious tuples (data).



Transitive dependency

SUMMARY



(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

BCNF (Boyce-Codd Normal Form)

- Definition:

  “A relation schema R is in Boyce-Codd Normal Form (BCNF), if whenever an FD (non-trivial), X → A holds in R, then X is a superkey or candidate key of R. “

- Each normal form is strictly stronger than 3NF:
  - Every 2NF relation is in 1NF
  - Every 3NF relation is in 2NF
  - Every BCNF relation is in 3NF

- There are also relations in 3NF but not in BCNF.

Consider the lots1A table.
Suppose there are two countries Brazil and Canada.

In brazil the lot sizes allowed are 0.5, 0.6.. 1.0 and

in canada it is 1.1, 1.2… 2.0 acres.

Then we can have another FD here

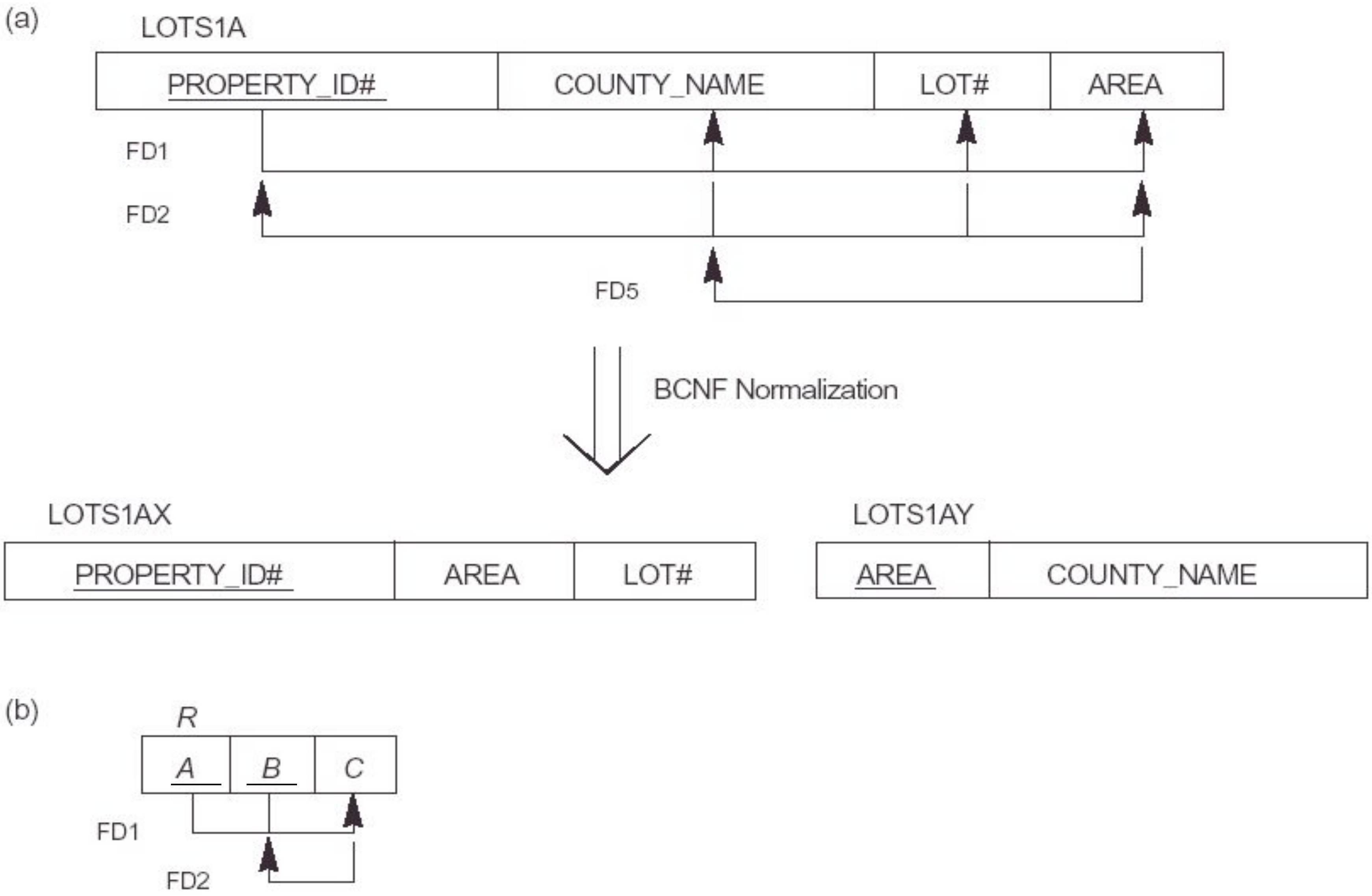
FD5 area→ Country_name

FD5 is in 3NF since country_name is prime attribute.

FD5 is not in BCNF since area is not superkey

Hence decompose lots1A into lots1AX and lots1AY.

But this will lose FD2.



(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

Example

- Consider the Given the relation

Book(Book_title, Authorname, Book_type, Listprice, Author_affil, Publisher)

The FDs are

Book_title □ Publisher, Book_type

Book_type □ Listprice

Authorname □Author_affil