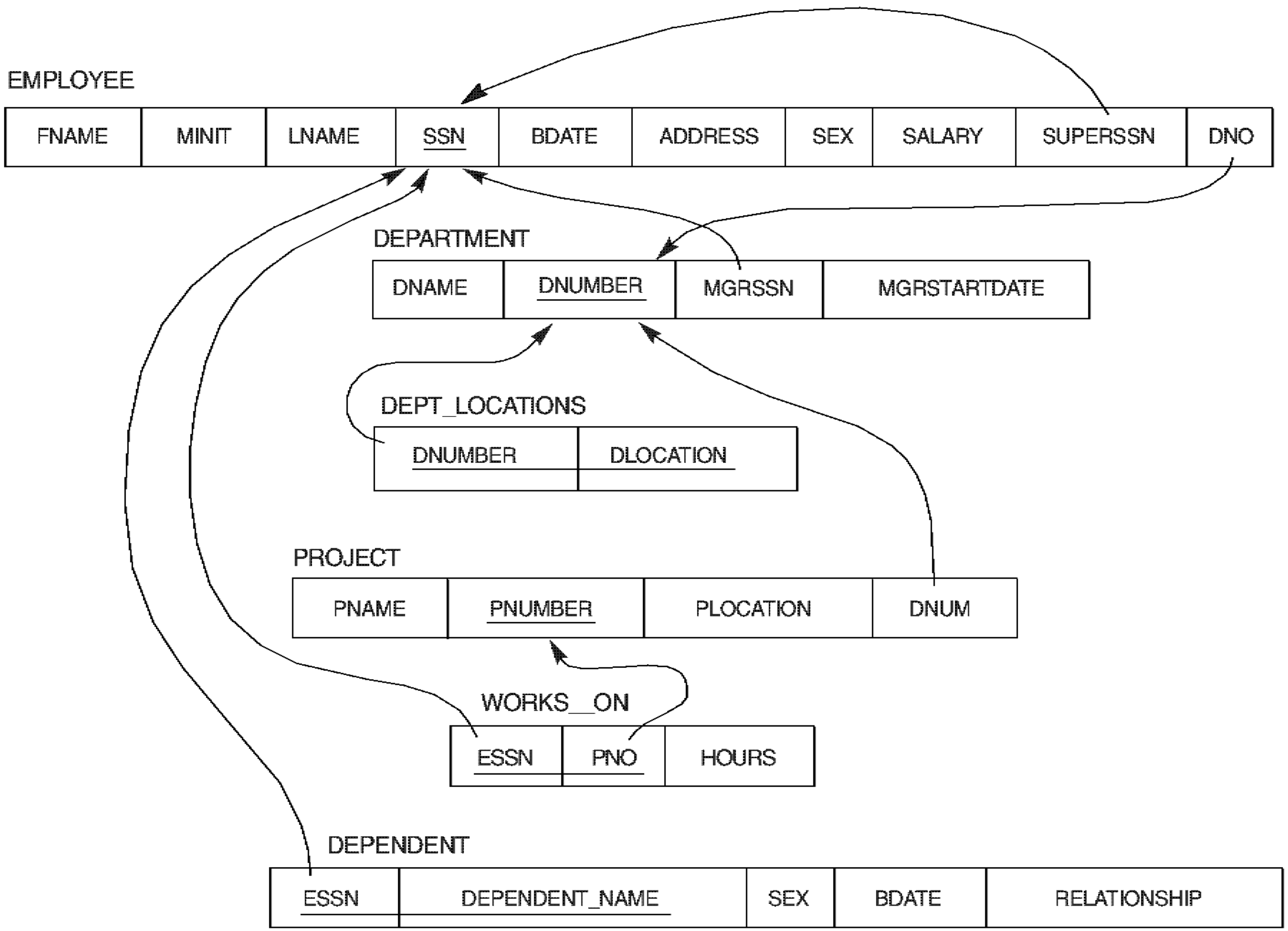
# UNIT III

**Functional Dependency**
**Normal Forms**
**Multi-valued Dependency**
**Join dependency**

**Dr. Geetha Mary A**
**Associate Professor,**
**SCSE, VITU**

**Figure 5. Referential integrity constraints displayed on the COMPANY relational database schema diagram.**



## 1.1 Semantics of the Relation Attributes

**GUIDELINE 1:** Informally, each tuple in a relation should represent one entity or relationship instance.

- 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 states for EMP_DEPT and EMP_PROJ resulting from applying NATURAL JOIN

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 |

# EXAMPLE OF AN UPDATE ANOMALY (1)

Consider the relation:

EMP_PROJ ( <u>Emp#, Proj#,</u> 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.

# EXAMPLE OF AN UPDATE ANOMALY (2)

- **Insert Anomaly:** Cannot insert a project unless an employee is assigned to .

  ***Inversely*** - Cannot insert an employee unless an he/she is assigned to a project.
- **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.

## Guideline to Redundant Information in Tuples and Update Anomalies

- **GUIDELINE 2:** 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

# 1.3 Null Values in Tuples

**GUIDELINE 3:** Relations should be designed such that their tuples will have as few NULL values as possible

- Reasons for nulls:
  - attribute not applicable or invalid
  - attribute value unknown (may exist)
  - value known to exist, but unavailable
- Attributes that are NULL frequently could be placed in separate relations (with the primary key)

## 1.4 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

**GUIDELINE 4:** 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.

**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 |

# Result of applying NATURAL JOIN to the tuples above the dotted lines in EMP_PROJ1 and EMP_LOCS
(2, 4, 5, 7, 8, 10, 12, 13, 14, 16, 19)

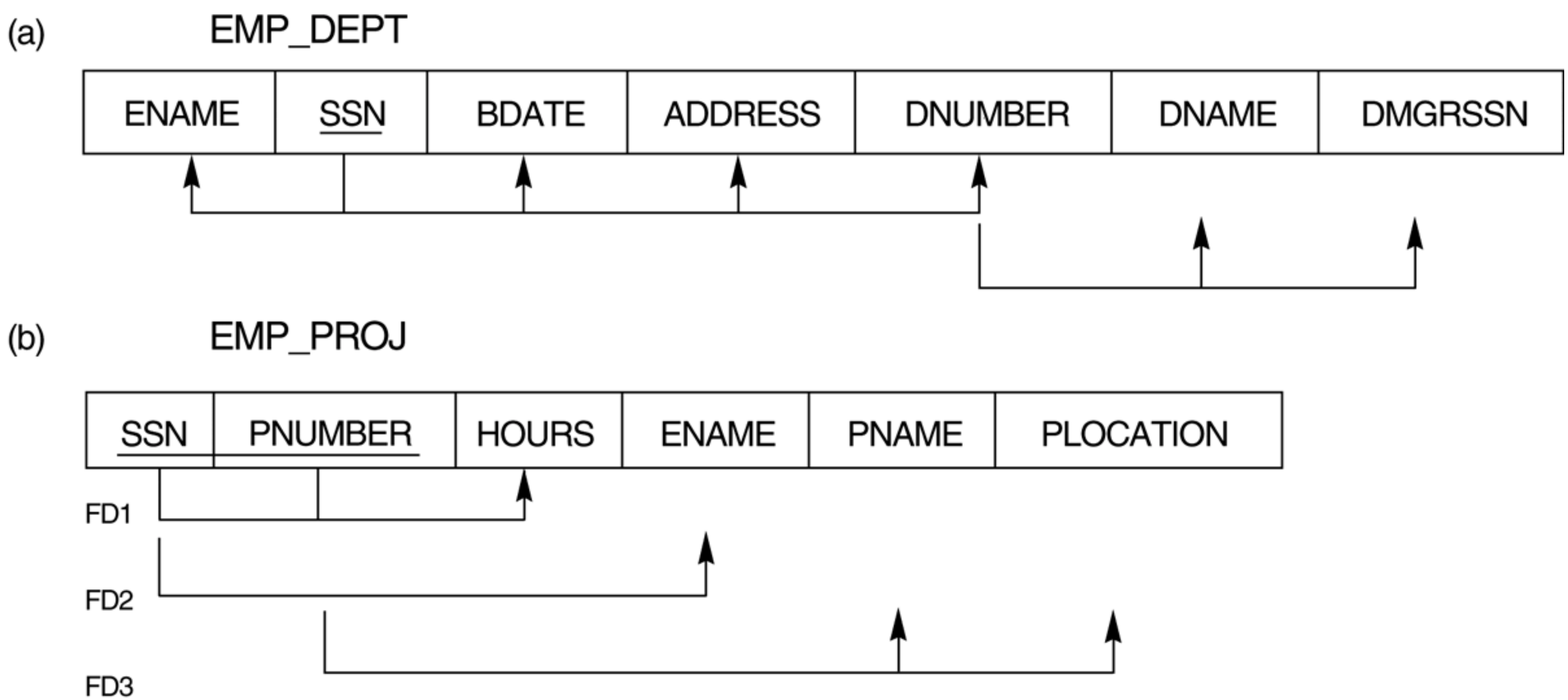
| SSN | PNUMBER | HOURS | PNAME | PLOCATION | |
|---|---|---|---|---|---|
| 123456789 | 1 | 32.5 | ProductX | Bellaire | Smith,John B. |
| 123456789 | 1 | 32.5 | ProductX | Bellaire | English,Joyce A. |
| 123456789 | 2 | 7.5 | ProductY | Sugarland | Smith,John B. |
| 123456789 | 2 | 7.5 | ProductY | Sugarland | English,Joyce A. |
| 123456789 | 2 | 7.5 | ProductY | Sugarland | Wong,Franklin T. |
| 666884444 | 3 | 40.0 | ProductZ | Houston | Narayan,Ramesh K. |
| 666884444 | 3 | 40.0 | ProductZ | Houston | Wong,Franklin T. |
| 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 T. |
| 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 T. |
| 333445555 | 3 | 10.0 | ProductZ | Houston | Narayan,Ramesh K. |
| 333445555 | 3 | 10.0 | ProductZ | Houston | Wong,Franklin T. |
| 333445555 | 10 | 10.0 | Computerization | Stafford | Wong,Franklin T. |
| 333445555 | 20 | 10.0 | Reorganization | Houston | Narayan,Ramesh K. |
| 333445555 | 20 | 10.0 | Reorganization | Houston | Wong,Franklin T. |

⋮

# 2.1 Functional Dependencies (1)

- 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 (X determines Y) holds if whenever two tuples have the same value for X, they *must have* the same value for Y
- For any two tuples t1 and t2 in any relation instance r(R): *If* t1[X]=t2[X], *then* t1[Y]=t2[Y]
- X -> Y in R specifies a *constraint* on all relation instances r(R)

# Two relation schemas suffering from update anomalies.



# 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

# Examples of FD constraints

- 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])

# 2.2 Inference Rules for FDs

- Given a set of FDs F, we can *infer* additional FDs that hold whenever the FDs in F hold

**<u>Armstrong's inference rules:</u>**

IR1. (**Reflexive**) If Y <u>*subset-of*</u> X, then X -> Y

IR2. (**Augmentation**) If X -> Y, then XZ -> YZ

(Notation: XZ stands for X U Z)

IR3. (**Transitive**) If X -> Y and Y -> Z, then X -> Z

- IR1, IR2, IR3 form a *sound* and *complete* set of inference rules

# Inference Rules for FDs (2)

Some **additional inference rules** that are useful:

(**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

- The last three inference rules, as well as any other inference rules, can be deduced from IR1, IR2, and IR3 (completeness property)

# Inference Rules for FDs (3)

- **Closure** of a set F of FDs is the set $F^+$ of all FDs that can be inferred from F
- **Closure** of a set of attributes X with respect to F is the set $X^+$ of all attributes that are functionally determined by X
- $X^+$ can be calculated by repeatedly applying IR1, IR2, IR3 using the FDs in F

# Inference Rules for FDs (4)

- **F = {SSN → ENAME,**
  **PNUMBER →{PNAME, PLOCATION}**
  **{SSN, PNUMBER} → HOURS}**
- **$\{SSN\}^+$ = {SSN, ENAME}**
  **$\{PNUMBER\}^+$ = {PNUMBER, PNAME, PLOCATION}**
  **$\{SSN, PNUMBER\}^+$ = {SSN, ENAME, PNUMBER, PNAME, PLOCATION, HOURS}**

## 2.3 Equivalence of Sets of FDs

- Two sets of FDs F and G are **equivalent** if:
  - every FD in F can be inferred from G, *and*
  - every FD in G can be inferred from F
- Hence, F and G are equivalent if $F^+ = G^+$

<u>Definition:</u> F **covers** G if every FD in G can be inferred from F (i.e., if $G^+$ is <u>*subset-of*</u> $F^+$)

- F and G are equivalent if F covers G and G covers F.

# 2.4 Minimal Sets of FDs (1)

- A set of FDs is **minimal** if it satisfies the following conditions:

(1) Every dependency in F has a single attribute for its RHS.

(2) We cannot remove any dependency from F and have a set of dependencies that is equivalent to F.

(3) We cannot replace any dependency X -> A in F with a dependency Y -> A, where Y proper-subset-of X ( Y subset-of X) and still have a set of dependencies that is equivalent to F.

# Minimal Sets of FDs (2)

- Every set of FDs has an equivalent minimal set
- There can be several equivalent minimal sets
- There is no simple algorithm for computing a minimal set of FDs that is equivalent to a set F of FDs
- To synthesize a set of relations, we assume that we start with a set of dependencies that is a minimal set

# Normalization of Relations (1)

- **Normalization**: The 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

# 3.2 Practical Use of Normal Forms

- **Normalization** is carried out in practice so that the resulting designs are of high quality and meet the desirable properties
- The practical utility of these normal forms becomes questionable when the constraints on which they are based are **hard to understand** or to **detect**
- The database designers ***need not*** normalize to the highest possible normal form. (usually up to 3NF, BCNF or 4NF)
- **Denormalization:** the process of storing the join of higher normal form relations as a base relation—which is in a lower normal form

# Definitions of Keys and Attributes Participating in Keys

- A **superkey** of a relation schema $R = \{A_1, A_2, ...., A_n\}$ is a set of attributes $S$ <u>*subset-of*</u> $R$ with the property that no two tuples $t_1$ and $t_2$ in any legal relation state $r$ of $R$ will have $t_1[S] = t_2[S]$

- 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.

# Definitions of Keys and Attributes Participating in Keys (2)

- If a relation schema has more than one key, each is called a **candidate key.** One of the candidate keys is *arbitrarily* designated to be the **primary key,** and the others are called *secondary keys*.
- A **Prime attribute** must be a member of *some candidate key*
- A **Nonprime attribute** is not a prime attribute—that is, it is not a member of any candidate key.

# 3.2 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

# Normalization into 1NF

(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 |

# 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: - {SSN, PNUMBER} -> HOURS is a full FD since neither SSN -> HOURS nor PNUMBER -> HOURS hold

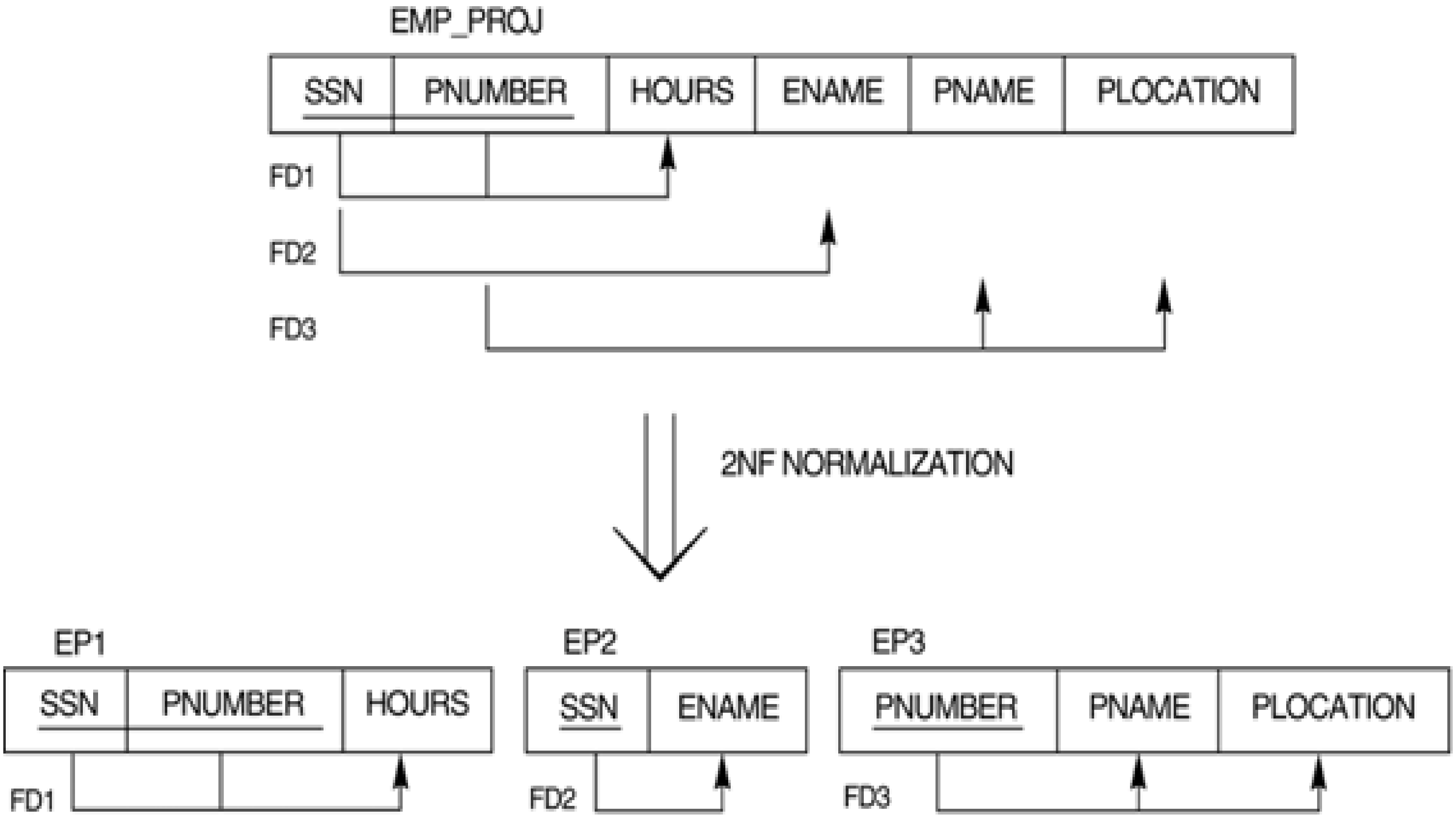
- {SSN, PNUMBER} -> ENAME is *not* a full FD (it is called a *partial dependency* ) since SSN -> ENAME also holds

# Second Normal Form

- 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

# Second Normal Form



# Third Normal Form

<u>Definition:</u>

- **Transitive functional dependency** - a FD X -> Z that can be derived from two FDs X -> Y and Y -> Z

<u>Examples:</u>

- 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

# Third Normal Form

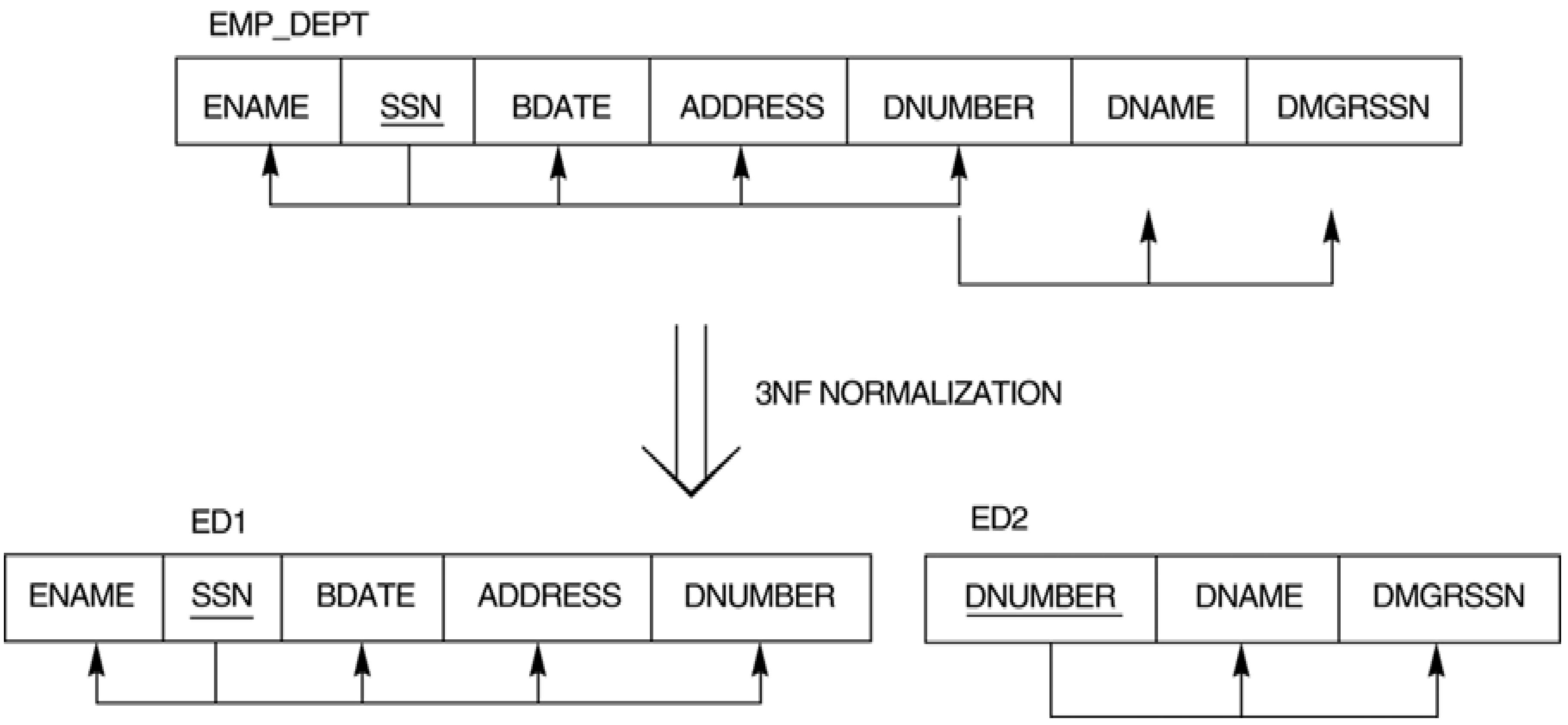
- A relation schema R is in **third normal form (3NF)** if it is in 2NF *and* no non-prime attribute A in R is transitively dependent on the primary key
- R can be decomposed into 3NF relations via the process of 3NF normalization

**NOTE:**

In X -> Y and Y -> Z, with X as the primary key, we consider this a problem only if Y is not a candidate key. When Y is a candidate key, there is no problem with the transitive dependency .

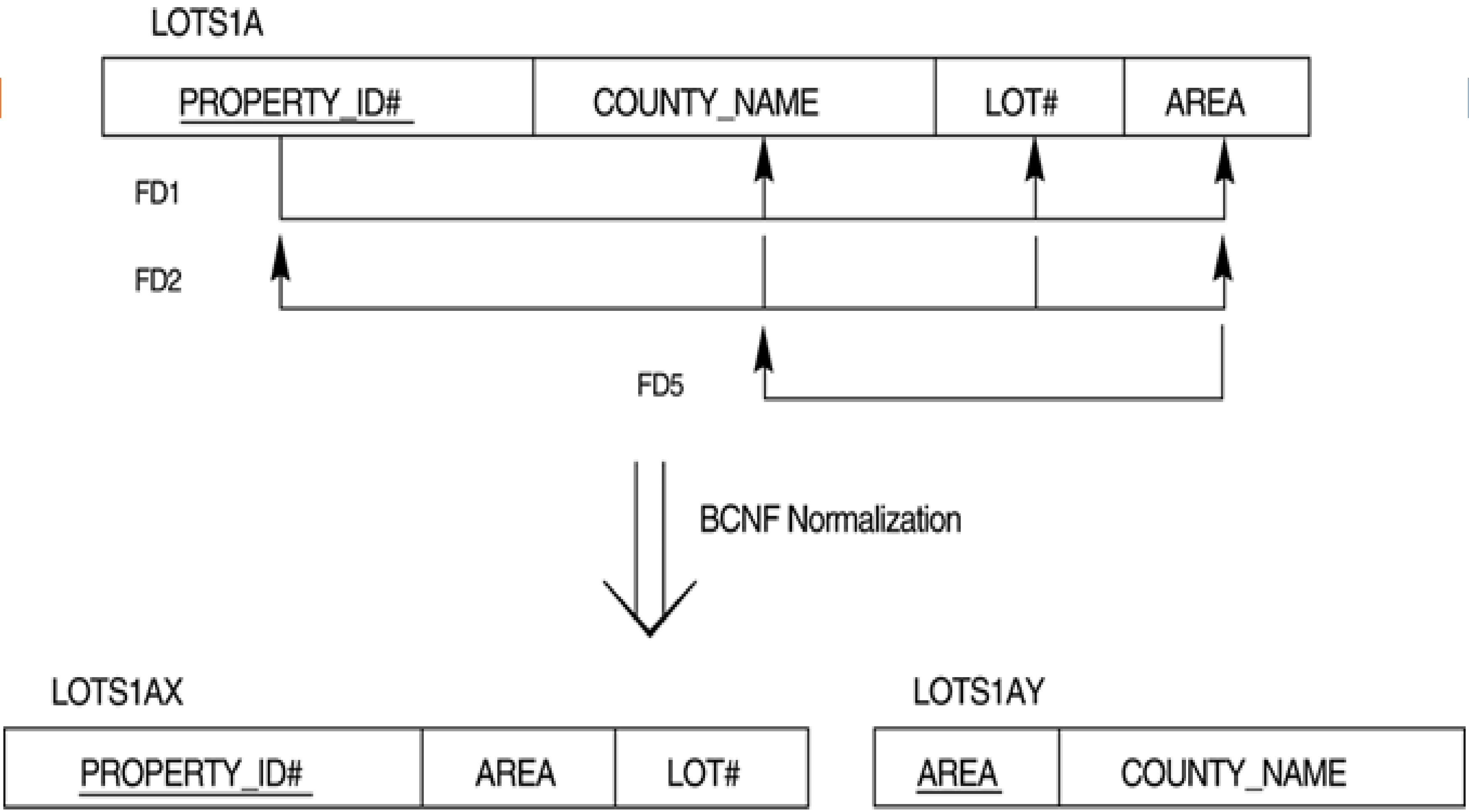
E.g., Consider EMP (SSN, Emp#, Salary ).

Here, SSN -> Emp# -> Salary and Emp# is a candidate key.


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)

- 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)


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

# FIGURE 10.13
A relation TEACH that is in 3NF but not 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 |