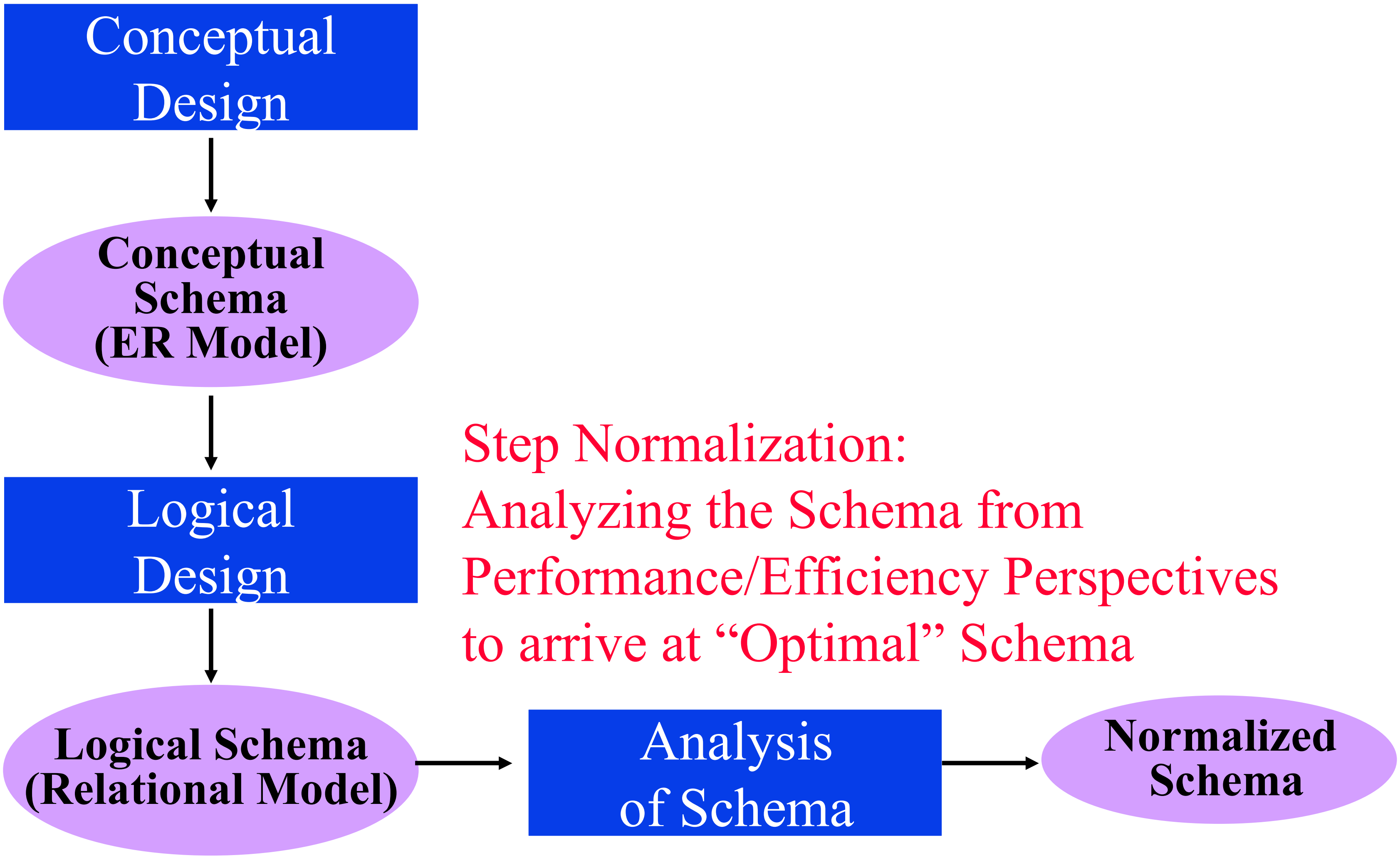
Chapter 9

Functional Dependencies and Normalization (from E&N,Silberschatz and my editing)

Chapter Outline

Design Process



Step Normalization:
Analyzing the Schema from Performance/Efficiency Perspectives to arrive at "Optimal" Schema

Guideline 1:Semantics of the Relation Attributes

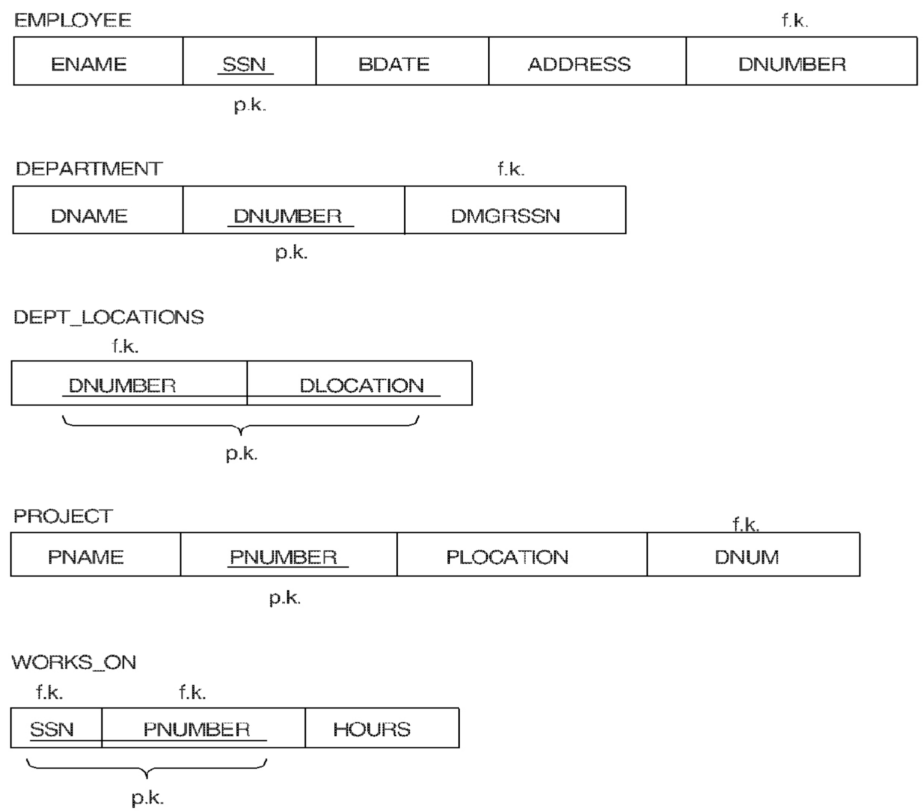
GUIDELINE 1: Informally, each tuple in a relation should represent one entity or relationship instance. (Applies to individual relations and their attributes).

- Attributes of different entities (EMPLOYEEs, DEPARTMENTs, PROJECTs) should not be mixed in the same relation
- Only foreign keys should be used to refer to other entities
- Entity and relationship attributes should be kept apart as much as possible.

Bottom Line: Design a schema that can be explained easily relation by relation. The semantics of attributes should be easy to interpret.

A simplified COMPANY relational database schema

Figure 14.1 Simplified version of the COMPANY relational database schema.



1.2 Redundant Information in Tuples and Update Anomalies

- 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 OF AN UPDATE ANOMALY (1)

Consider the relation:

EMP_PROJ (<u>Emp#, Proj#,</u> Ename, Pname, 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

Two relation schemas suffering from update anomalies

Figure 14.3 Two relation schemas and their functional dependencies. Both suffer from update anomalies. (a) The EMP_DEPT relation schema. (b) The EMP_PROJ relation schema.

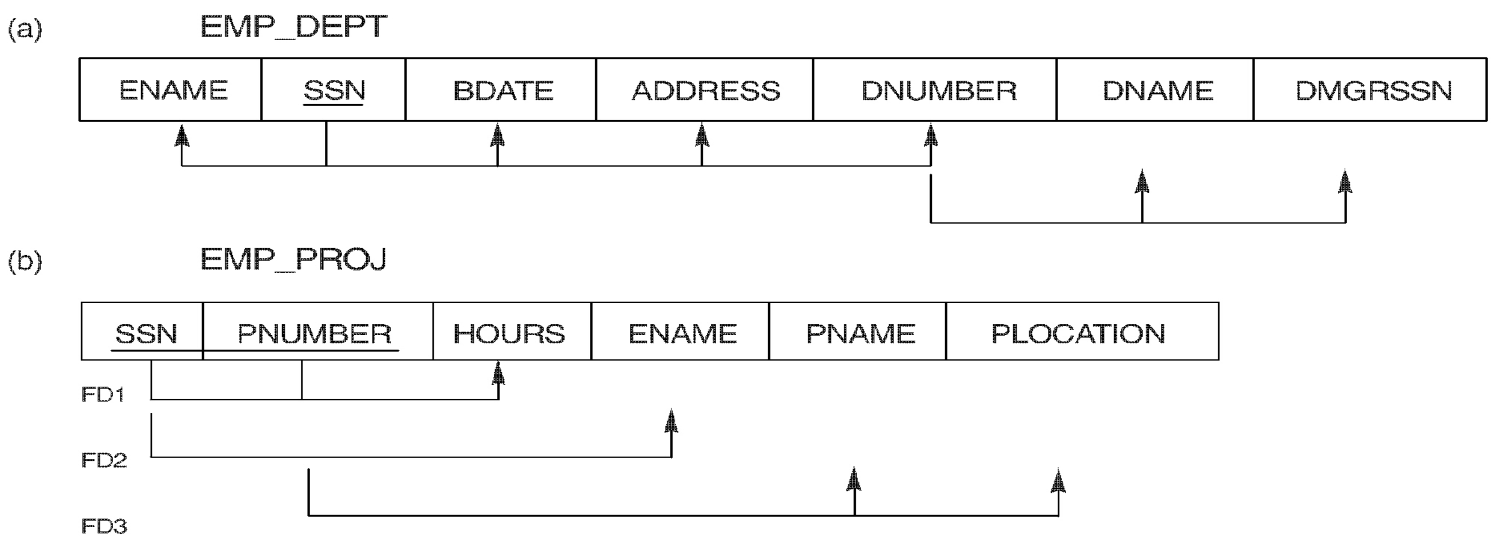


Figure 14.4 Example relations for the schemas in Figure 14.3 that result from applying NATURAL JOIN to the relations in Figure 14.2. These may be stored as base relations for performance reasons.

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

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
333445555	20	10.0	Wong,Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya,Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya,Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar,Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar,Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace,Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace,Jennifer S.	Reorganization	Houston
888665555	20	null	Borg,James E.	Reorganization	Houston

1.3 Null Values in Tuples

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

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

1.4 Spurious Tuples

GUIDELINE 4:

- 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
- There are two important properties of decompositions:
  - non-additive or losslessness of the corresponding join
  - preservation of the functional dependencies.

Note that property (a) is extremely important and *cannot* be sacrificed. Property (b) is less stringent and may be sacrificed.

- Two Other "Related" Concerns Can Arise
  - First, in Decomposing (Splitting) a Relation Apart, we May "Lose" Information
  - Second, in Attempting to Reassemble Two or More Relations into One (via a Join), Spurious Tuples may Result
- A Spurious Tuple "Wasn't" Present Originally and Makes No Sense - Didn't Exist and its Existence is Inconsistency

Suppose Split EMP_PROJ

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
333445555	20	10.0	Wong,Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya,Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya,Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar,Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar,Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace,Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace,Jennifer S.	Reorganization	Houston
888665555	20	null	Borg,James E.	Reorganization	Houston

EMP_LOCS

ENAME	PLOCATION

p.k.

EMP_PROJ1

SSN	PNUMBER	HOURS	PNAME	PLOCATION

p.k.

Semantics of Split?

- EMP_LOCS Means the Employee ENAME Works on Some Project at PLOCATION
- EMP_PROJ1 Means the Employee Identified by SSN Works HOURS per Week on Project Identified by PNAME, PNUMBER, PLOCATION

EMP_LOCS

ENAME	PLOCATION

p.k.

EMP_PROJ1

SSN	PNUMBER	HOURS	PNAME	PLOCATION

p.k.

Recall EMP_PROJ

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
333445555	20	10.0	Wong,Franklin T.	Reorganization	Houston
999887777	30	30.0	Zelaya,Alicia J.	Newbenefits	Stafford
999887777	10	10.0	Zelaya,Alicia J.	Computerization	Stafford
987987987	10	35.0	Jabbar,Ahmad V.	Computerization	Stafford
987987987	30	5.0	Jabbar,Ahmad V.	Newbenefits	Stafford
987654321	30	20.0	Wallace,Jennifer S.	Newbenefits	Stafford
987654321	20	15.0	Wallace,Jennifer S.	Reorganization	Houston
888665555	20	null	Borg,James E.	Reorganization	Houston

Tuple after Split

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

The Issues?

- Suppose EMP_PROJ1 and EMP_LOCS used in Place of EMP_PROJ
- The Split is Legitimate if we Can Recover the Information Originally in EMP_PROJ
- How could you Recover the Information?
  - Natural Join on EMP_PROJ1 and EMP_LOCS
  - What would be the Result?
- Note: "*'ed" Entries are Spurious Tuples

  We do not Obtain the "Correct" Information

  We have Conducted a "Lossy" Decomposition

When we do Join?

	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.

⋮

Lost information

- A First Example of Lost Information
  - What is Lost in the Join of R and S?

R = (A, B, C)

A	B	C
a1	b2	c1
a2	b2	c1
a3	b4	c2

S = (D, C)

D	C
d1	c1
d2	c2
d4	c2
d5	c3

RS(A, B, C, D)

A	B	C	D
a1	b2	c1	d1
a2	b2	c1	d1
a3	b4	c2	d2
a3	b4	c2	d4

lost info of (d5, c3) after join R & S

Spurious Tuple

- A Second Example of Spurious Tuples
  - What are Spurious in the Join of R1and R2?

R(A, B, C, D)

A	B	C	D
a1	b1	c1	d1
a2	b2	c2	d1
a3	b1	c1	d2
a4	b2	c2	d3

R1(B, C)

B	C
b1	d1
b2	d1
b1	d2
b2	d3

R2(A, D)

A	D
a1	d1
a2	d1
a3	d2
a4	d3

R1 and R2 Join

A	B	C	D
a1	b1	c1	d1
a2	b1	c2	d1
a1	b2	c2	d1
a2	b2	c2	d1
a3	b1	c1	d2
a4	b2	c2	d3

2.1 Functional Dependencies (1)

- Functional dependencies (FDs) are used to specify *formal measures* of the "goodness" of relational designs
- FDs and keys are used to define **normal forms** for relations
- FDs are **constraints** that are derived from the *meaning* and *interrelationships* of the data attributes
- A set of attributes X *functionally determines* a set of attributes Y if the value of X determines a unique value for Y

Functional Dependencies (2)

- X -> 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)
- Written as X -> Y; can be displayed graphically on a relation schema as in Figures. (denoted by the arrow:).
- **FDs are derived from the real-world constraints on the attributes**

Examples of FD constraints (1)

- 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 (2)

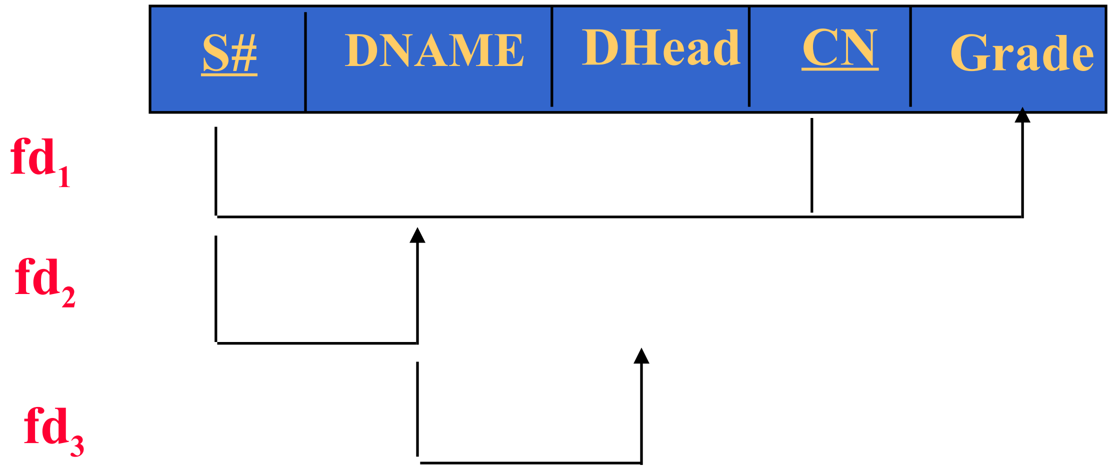
- An FD is a property of the attributes in the schema R
- The constraint must hold on *every relation instance* r(R)
- If K is a key of R, then K functionally determines all attributes in R (since we never have two distinct tuples with t1[K]=t2[K])

Ex

STUDENT_DEPT (S#, DName, DHead, CN, Grade)

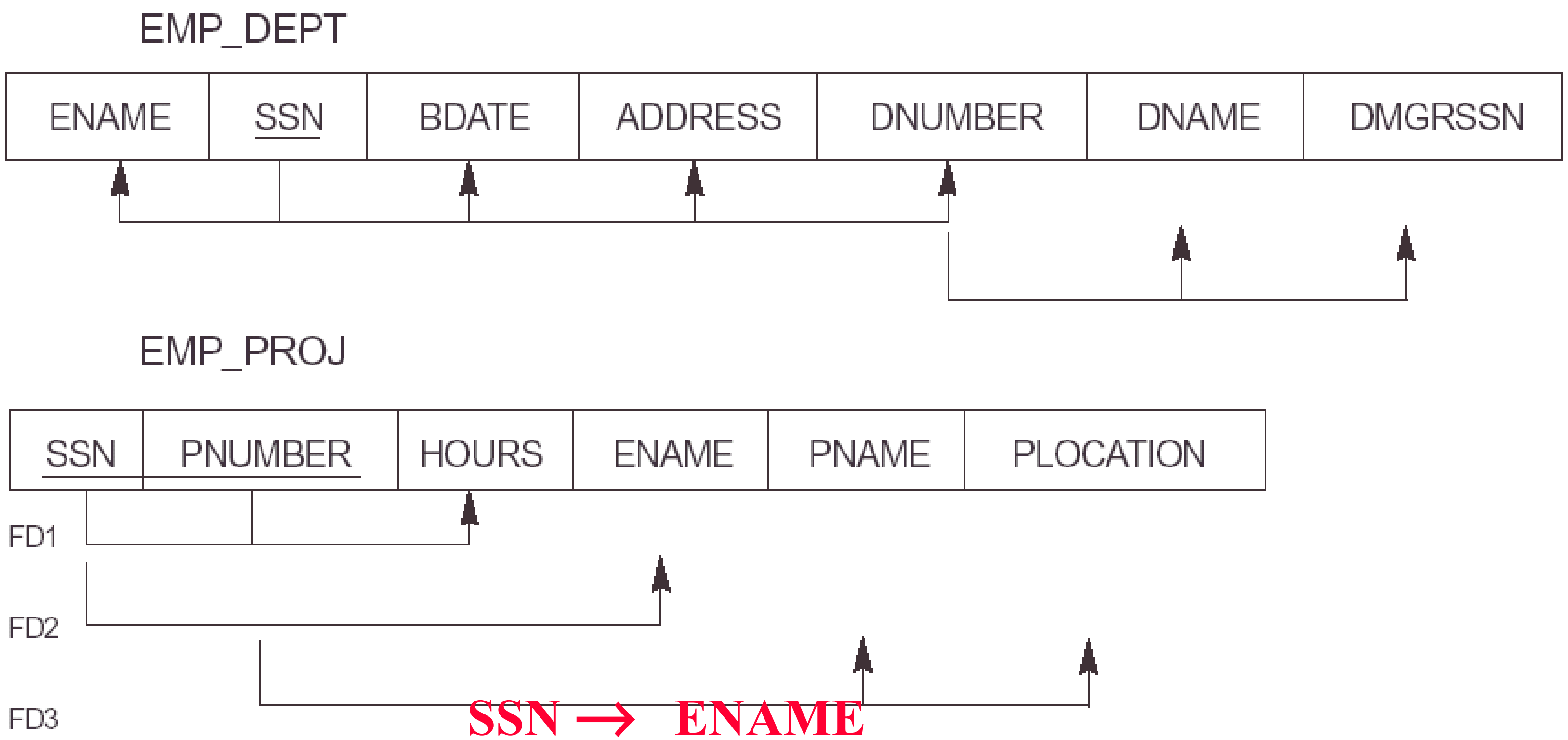
FDs over STUDENT_DEPT:

```
{S#, CN} → Grade,
      S# → DNAME,
   DNAME → DHead.
```



SSN → {ENAME, BDATE, ADDRESS, DNUMBER}

DNUMBER → {DNAME, DMGRSSN}



SSN → ENAME

PNUMBER → {PNAME, PLOCATION}

{SSN, PNUMBER} → HOURS

Determining FDs

- Must Understand the Semantics of Data Based on Schema or Current/Future Instances
- What are FDs Below?
  TEXT → COURSE?
  COURSE → TEXT?
- What if I add Row "James, Web Databases, Al-Nour"?

TEACH

TEACHER	COURSE	TEXT
Smith	Data Structures	Bartram
Smith	Data Management	Al-Nour
Hall	Compilers	Hoffman
Brown	Data Structures	Augenthaler

3 Normal Forms Based on Primary Keys

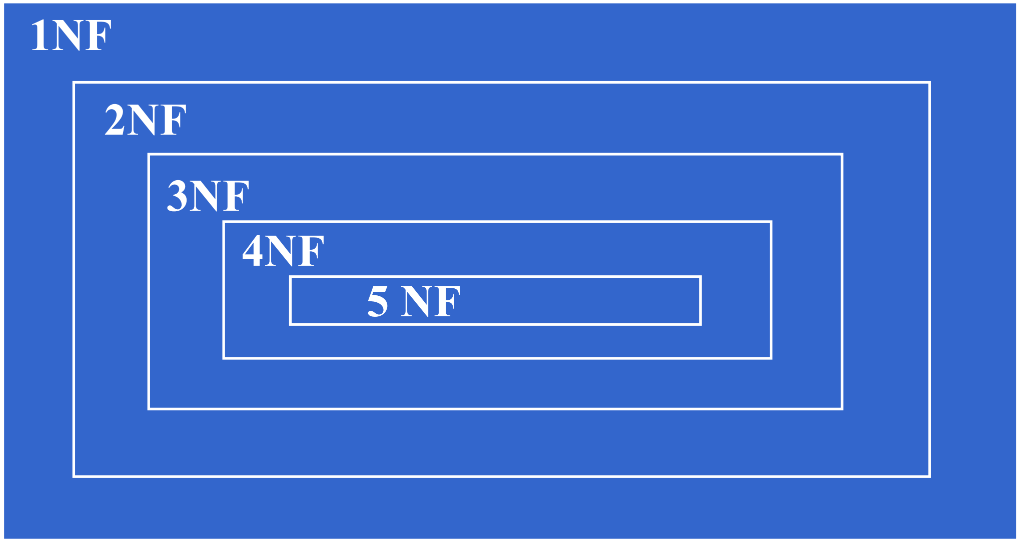
- 3.1 Normalization of Relations
- 3.2 Practical Use of Normal Forms
- 3.3 Definitions of Keys and Attributes Participating in Keys
- 3.4 First Normal Form
- 3.5 Second Normal Form
- 3.6 Third Normal Form

Normalization of Relation

- **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

Whats Normal Form

- A Normal Form is a Condition using Keys and FDs to Certify Whether a Relation Schema meets Criteria
  - Primary keys (1NF, 2NF, 3NF)
  - All Candidate Keys (2NF, 3NF, BCNF)
  - Multivalued Dependencies (4NF)
  - Join Dependencies (5NF)



- 1NF based on definition of relation
- 2NF, 3NF, BCNF based on keys and FDs of a relation schema
- 4NF based on keys, multi-valued dependencies : MVDs;
- 5NF based on keys, join dependencies : JDs
- Additional properties may be needed to ensure a good relational design (lossless join, dependency preservation)

Practical Use of Norm Form

- **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

Keys and Participating

- A **superkey** of a relation schema $R = \{A_1, A_2, \ldots., A_n\}$ is a set of attributes S *subset-of* 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.

Keys

- 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

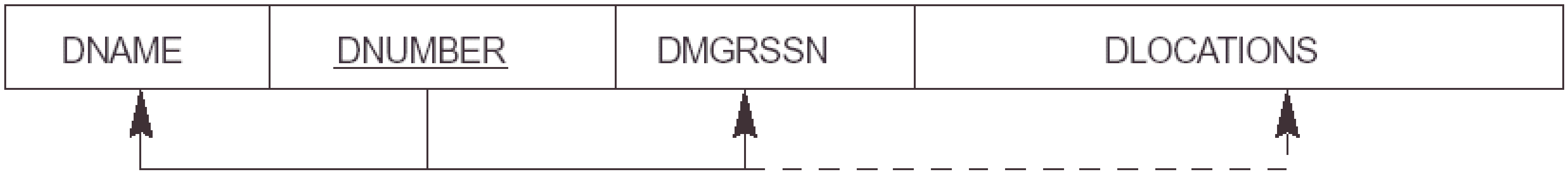
- Disallows composite attributes, multivalued attributes, and **nested relations**;
- attributes whose values *for an individual tuple* are non-atomic
- Considered to be part of the definition of relation

- All Attributes Must Be Atomic Values:
  - Only Simple and Indivisible Values in the Domain of Attributes.
  - Each Attribute in a 1NF Relation is a Single Value
  - Disallows Composite Attributes, Multivalued Attributes, and Nested Relations (Non-Atomic)
- 1NF Relation cannot have an Attribute Value :
  - A Set of Values (Set-Value)
  - A Tuple of Values (Nested Relation)
- 1NF is a Standard Assumption of Relation DBs

Normalization into 1NF

- Consider Following Department Relation
- What is the Inherent Problem?

DEPARTMENT



DEPARTMENT

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

Normalization nested relations into 1NF

SSN	ENAME	PROJS	
		PNUMBER	HOURS

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

EMP_PROJ1

SSN	ENAME

EMP_PROJ2

SSN	PNUMBER	HOURS

Possible Solutions

- **Decompose:** Move the Attribute DLOCATIONS that Violates 1NF into a Separate Relation DEPT_LOCATIONS(DNUMBER, DLOCATION)
- Expand the key to have a Separate Tuple in the DEPARTMENT relation for each location (below)
- Introduce DLOC1, DLOC2, DLOC3, if there are Three Maximum Locations
- Problems with Each? Best Solution?

DNAME	DNUMBER	DMGRSSN	DLOCATION
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

3.3 Second Normal Form (1)

- Uses the concepts of **FD**s, **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 (2)

- 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

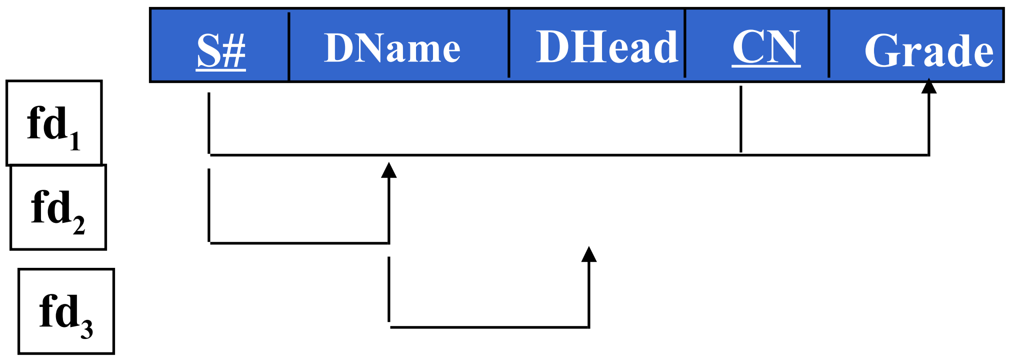
- R can be decomposed into 2NF relations via the process of 2NF normalization

- Second Normal Form Focuses on the Concepts of Primary Keys and Full Functional Dependencies
- Intuitively:
  - 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
  - Successful Process Typically Involves Decomposing R into Two or More Relations
  - Iteratively Applying to Each Relation in Schema

Ex

- Consider the Example Below

STUDENT_DEPT(S#, DName, DHead, CN, Grade)



STUDENT_DEPT $\in$ 1NF

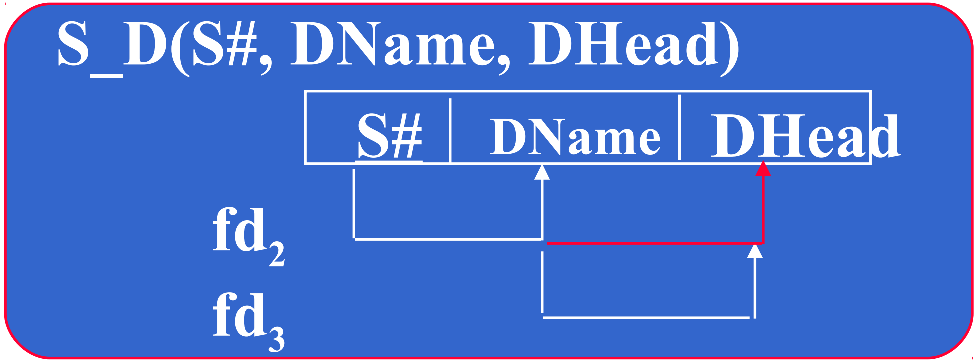
But STUDENT_DEPT $\notin$ 2NF

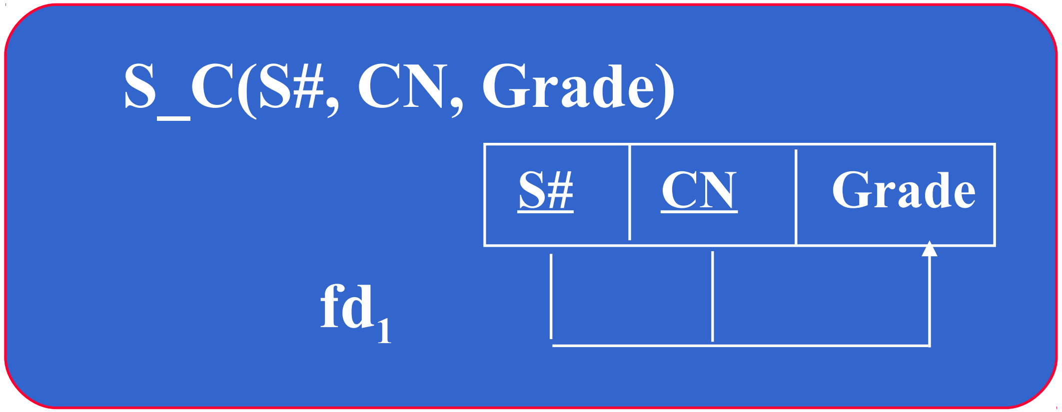
"{S#, CN} $\rightarrow$ DName, DHead" is a Partial FD which causes Update Anomalies

STUDENT_DEPT(S#, DName, DHead, CN, Grade)

- Insertion Anomalies:
  - No Department Can Be Recorded if it has No Student Who Enrolls Courses
- Deletion Anomalies:
  - Delete the Last Student in a Department will also Delete the Department
- Update Anomalies:
  - Change a Head of a Department must Modify All Students in that Department Due to Redundancies

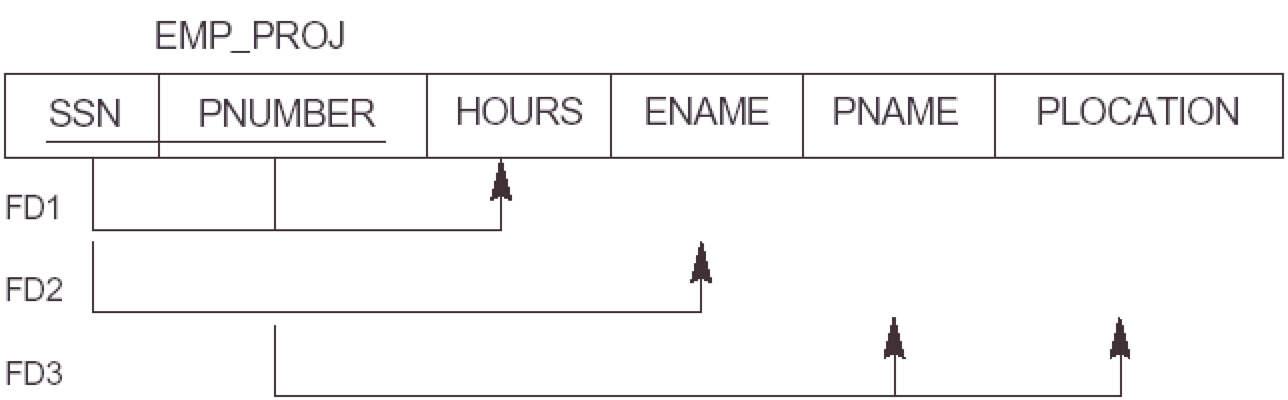
- Decomposition into 2NF by Separating Course Information from Department Information (Link S#)



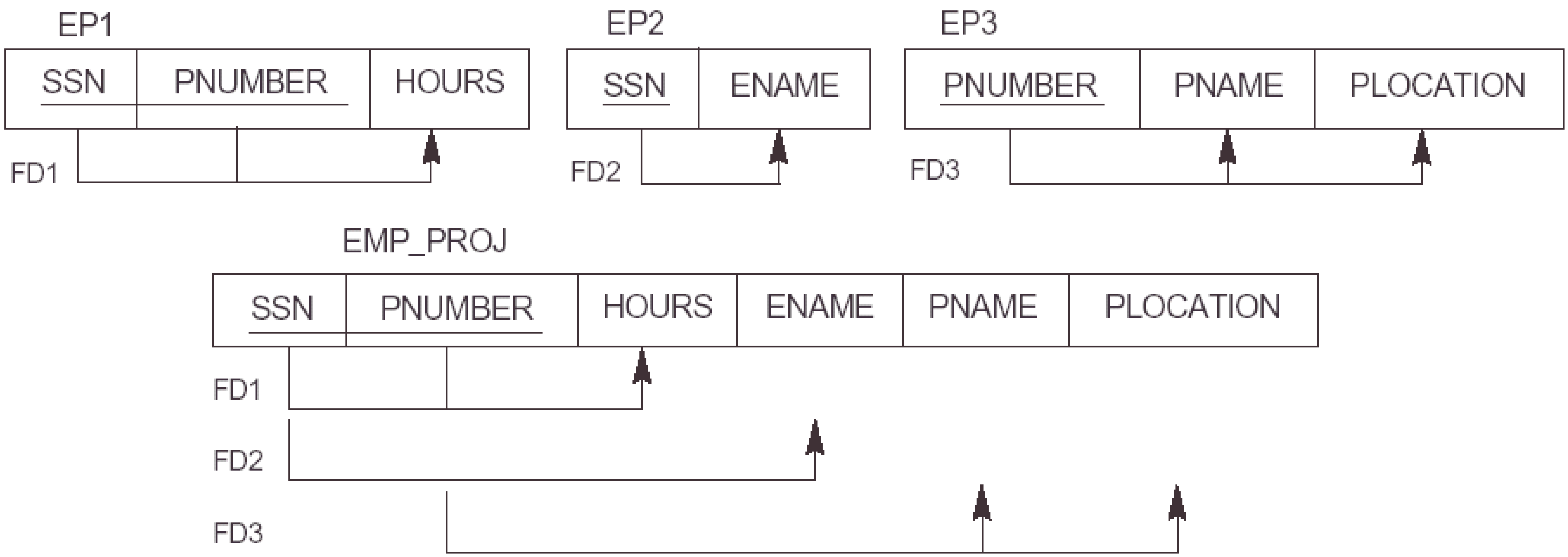


Another Ex

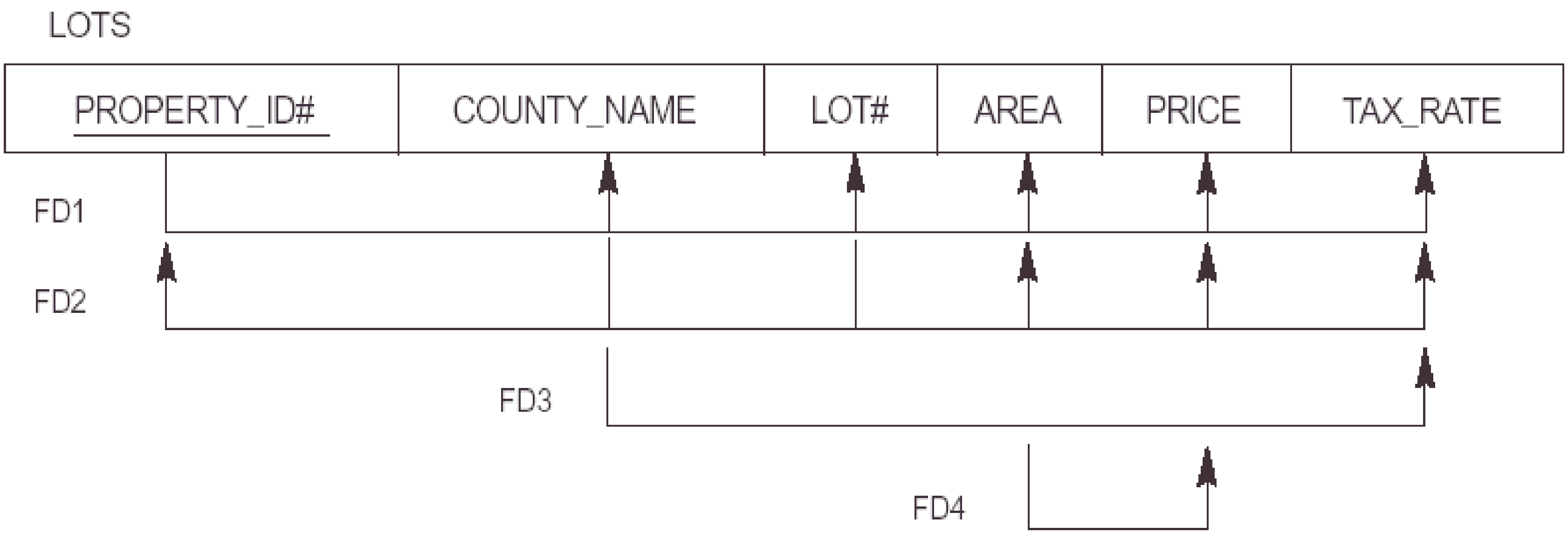
- EMP_PROJ is 1NF with Key SSN, PNUMBER but…
  - SSN $\rightarrow$ ENAME - Means ENAME, a Non-Prime Attribute, Depends Partially on SSN, PNUMBER, i.e., Depend on Only SSN and not Both
  - PNUMBER $\rightarrow$ {PNAME, PLOCATION} - Means PNAME, PLOCATION, two Non-Prime Attributes, Depends Partially on SSN, PNUMBER, i.e., Depend on Only PNUMEBER and not Both



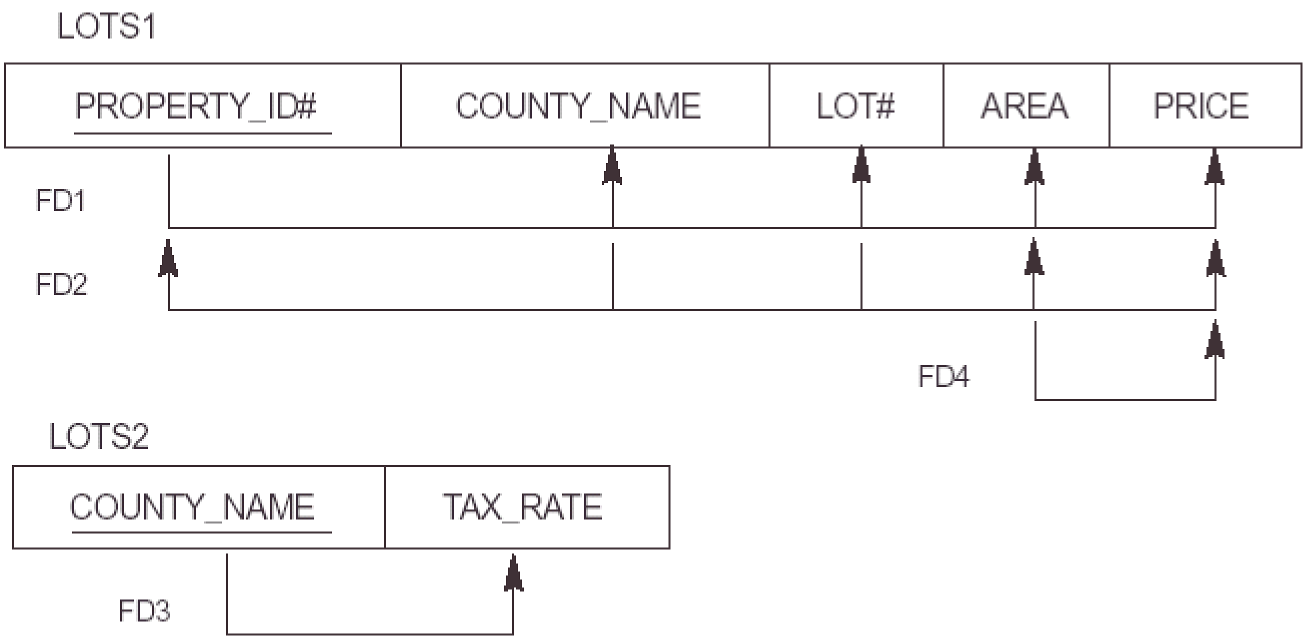
- What Does Decomposition Below Accomplish?
  - ENAME Fully Dependent on SSN
  - PNAME, PLOC Fully Dependent on PNUMBER



- Consider 1NF Lots to Track Building Lots for Towns
- What is the 2NF Problem?
  - FD3: COUNTY_NAME → TAX_RATE Means TAX_RATE Depends Partially on Candidate Key {PROPERTY_ID#,COUNTY_NAME}
  - All Other Non-Prime Attributes are Fine



- What Does Decomposition Below Accomplish?
    - TAX_RATE Fully Dependent on COUNTY_NAME
- Result: 2NF for LOTS1 and LOTS2

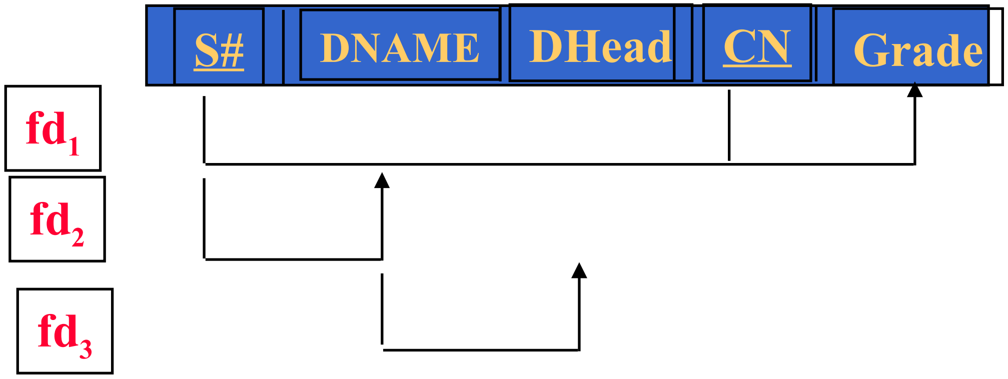


Third Normal Form (3NF)

- Third Normal Form Focuses on the Concepts of Primary Keys and Transitive Functional Dependencies
- Intuitively:
  - 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 Primary Key
  - R can be Decomposed into 3NF Relations via the Process of 3NF Normalization
- In X→Y and Y→ Z , with X as the Primary Key, there is only a problem only if Y is <u>not</u> a candidate key.
  EMP(SSN, Emp#, Salary), SSN → Emp# → Salary isn't Problem Since Emp# is a Candidate Key

Transitive FDs

- Transitive FD - Formally:
  Given R(U) and X, Y$\subseteq$U.
  If X$\rightarrow$Y, Y$\subseteq$X and Y$\rightarrow$X, Y$\rightarrow$Z, then Z is called transitively functional dependent on X.
- Transitive FD - Intuitively: a FD X$\rightarrow$ Z that can be derived from two FDs X$\rightarrow$Y and Y$\rightarrow$Z
  - SSN $\rightarrow$ ENAME is *non-transitive* Since there is no set of Attributes X where SSN $\rightarrow$ X and X $\rightarrow$ ENAME



- Formal 3NF Definition, $R \in$ 3NF iff
  - (i) $R \in$ 2NF;
  - (ii) No Non-Key Attribute of R is Transitively Dependent on Every Candidate Key.
- Alternative Definition:
  $R \in$ 3NF iff for every FD $X \rightarrow Y$, either
  - X is a superkey, or
  - Y is a key attribute.
- Reason: Transitive Functional Dependencies may cause Update Problems

Ex

STUDENT_DEPT(S#, DName, DHead, CN, Grade) $\notin$ 2NF

S_D(S#, DName, DHead) $\in$ 2NF

S_C(S#, CN, Grade) $\in$ 2NF

S_C $\in$ 3NF But S_D $\notin$ 3NF

"S# $\rightarrow$ DHead" is a **Transitive** FD in S_D and "DHead" is non-key attribute.

<u>S#</u>	DName	DHead

fd_2 S# $\rightarrow$ DName

fd_3 DName $\rightarrow$ DHead

fd S# $\rightarrow$ DHead

S_C(S#, CN, Grade) $\in$ 2NF

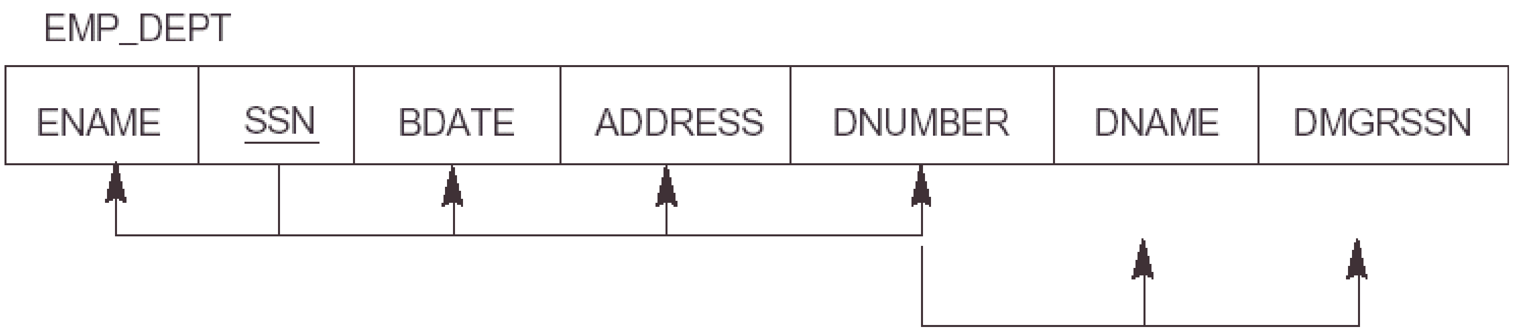
S_D(S#, DName, DHead) $\in$ 2NF

DEPT(DName, DHead)
S_D (S#, DName)

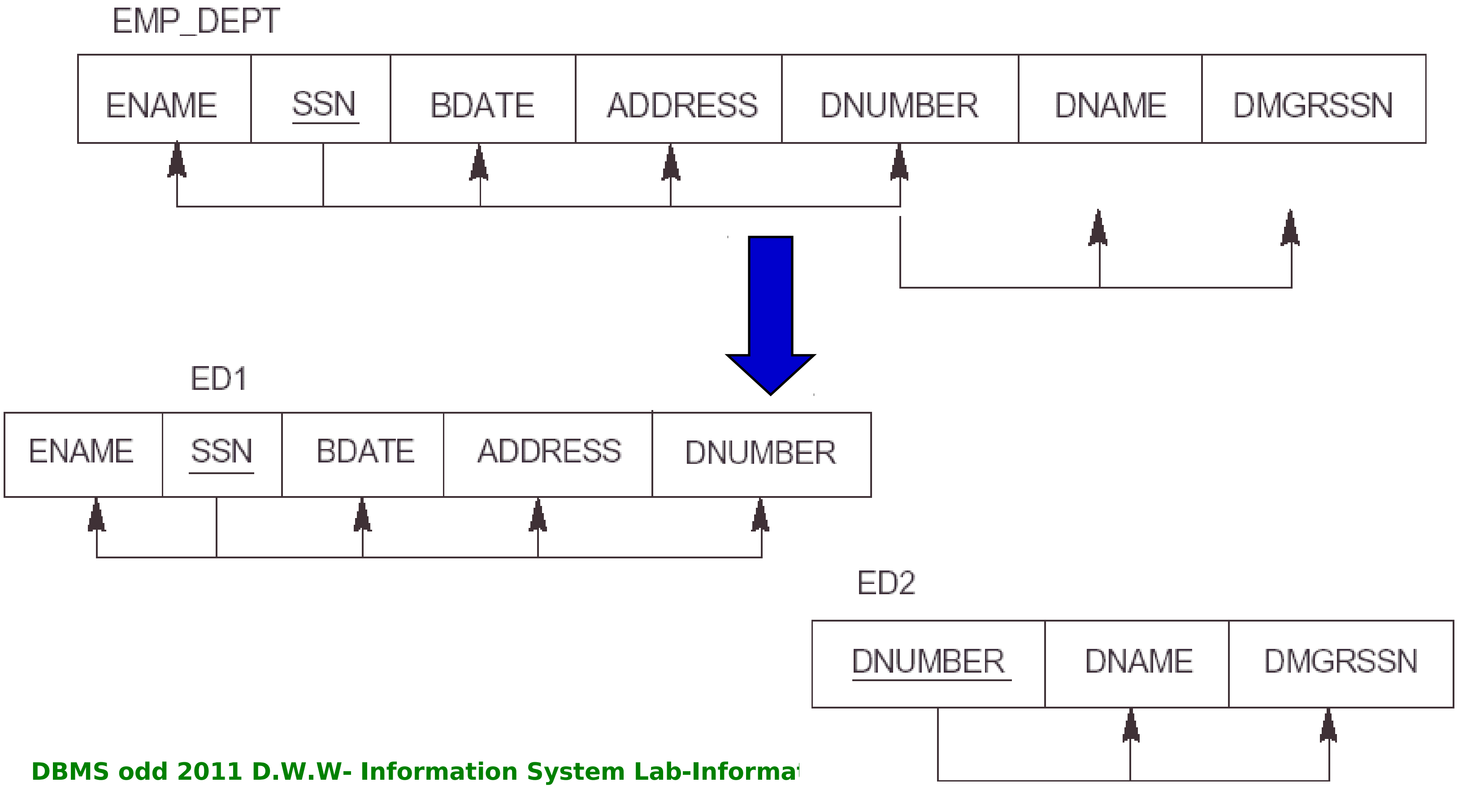
$\in$ 3NF

Decompose to Eliminate the Transitivity Within S_D

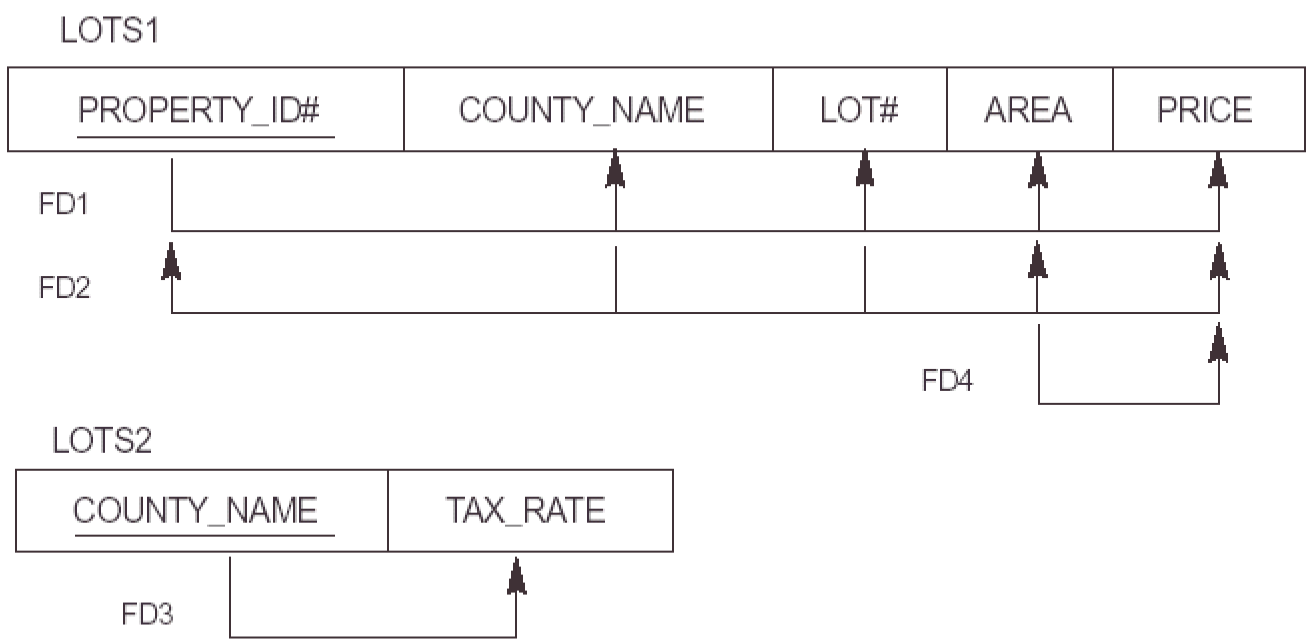
- EMP_DEPT is 2NF with Key SSN, but there are Two Transitive Dependencies (Undesirable)
  - SSN $\rightarrow$ DNUMBER and DNUMBER $\rightarrow$ DNAME Means DNAME, Neither Key Nor Subset of Key, is Transitively Dependent on SSN
  - SSN is the Only Candidate Key of EMP_DEPT!
  - Note: Also Similar Problem with SSN and DMGRSSN via DNUMBER



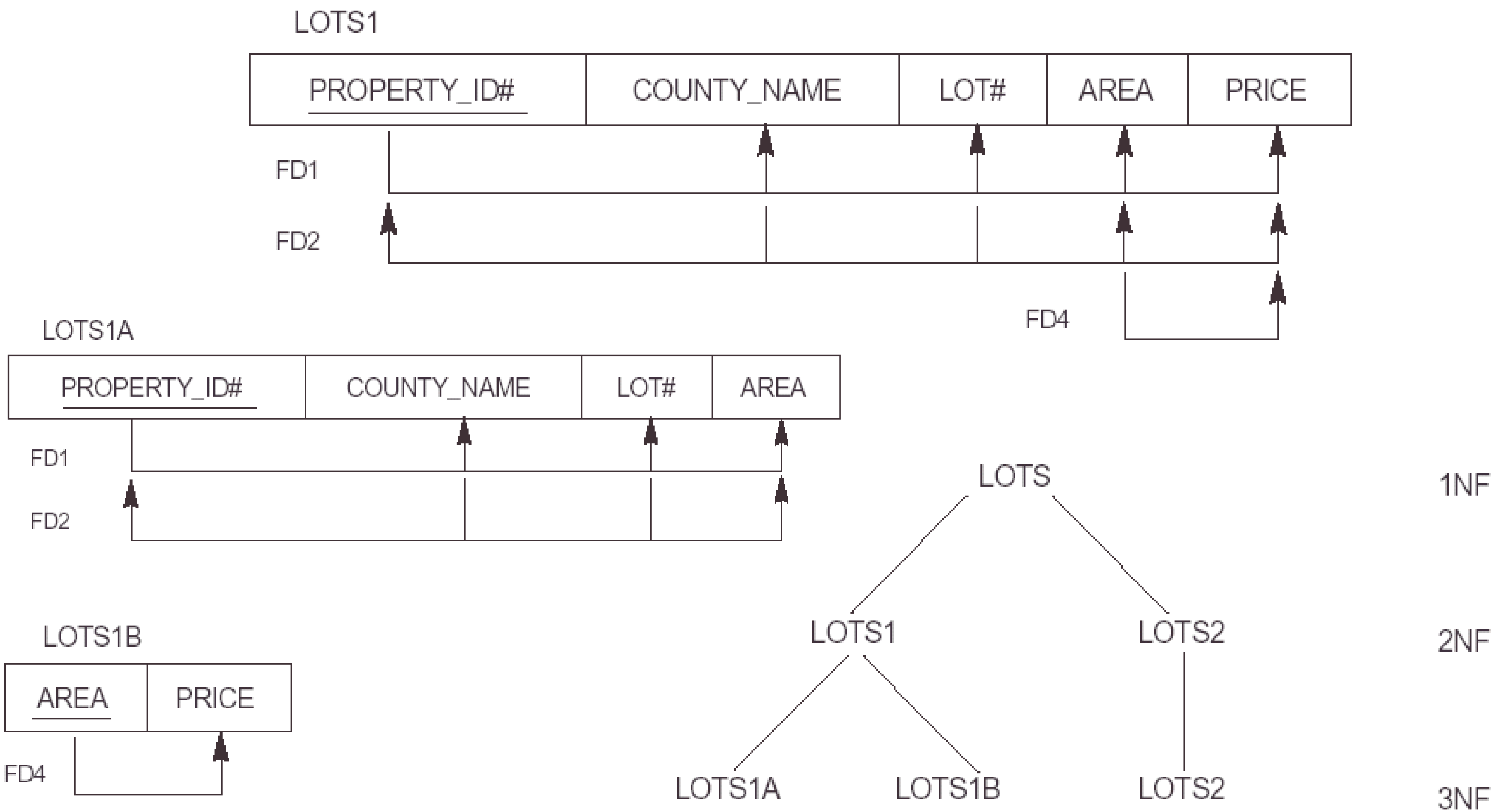
- To Attain 3NF, Decompose into ED1 and ED2
- Intuitively - we are Separating Out Employees and Departments from One Another



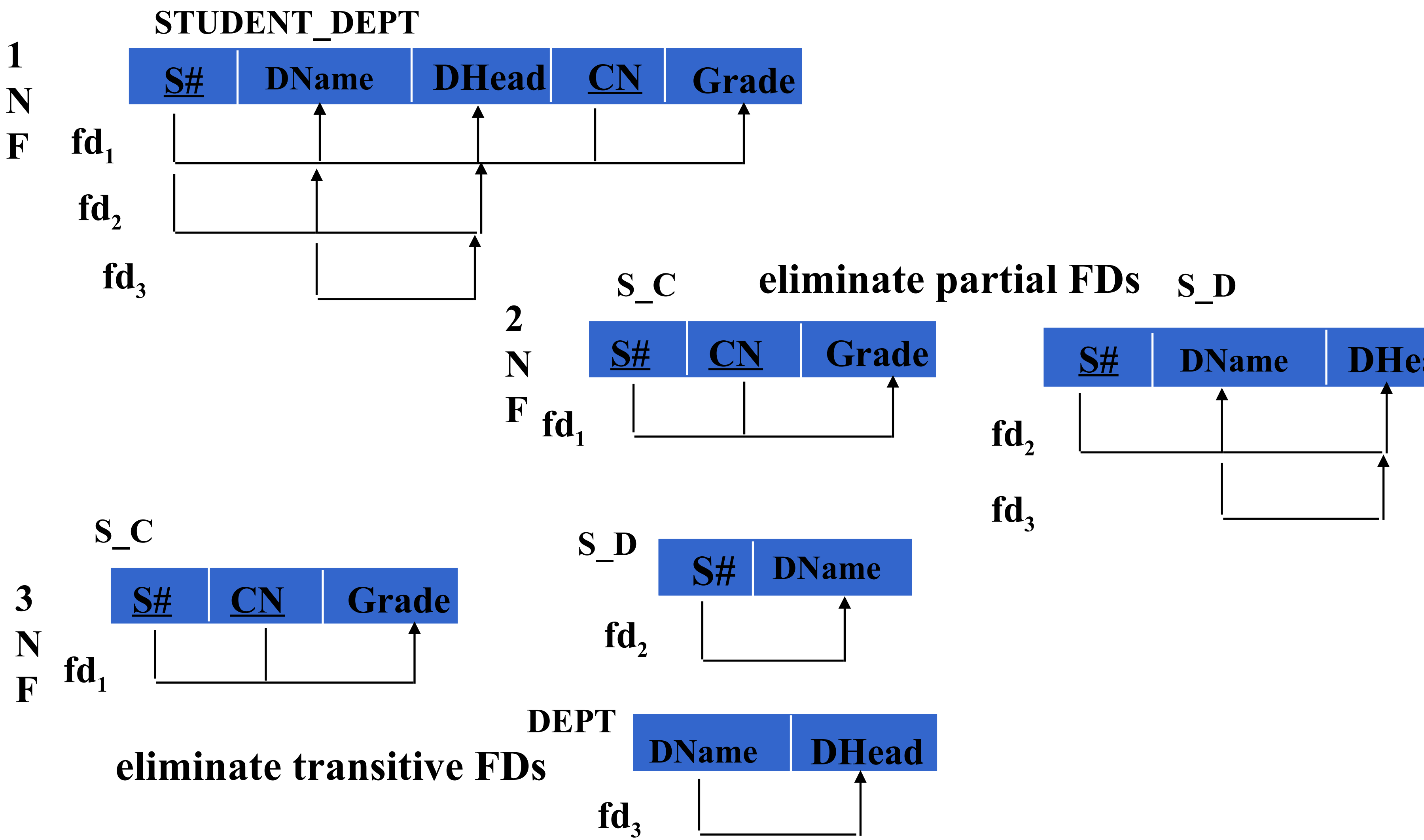
- Recall 2NF Solution for Building Lots Problem
- What is the 3NF Problem? Violate Alternative Defn.
  - In LOTS1, FD4 AREA $\rightarrow$ PRICE
    AREA is not a Superkey
    PRICE not a Prime Attribute of LOTS1



- Decompose to Introduce a Separate Key <u>AREA</u>
- Result: 3NF for LOTS1A and LOTS1B



Summary


STUDENT_DEPT
1NF
S#
DName
DHead
CN
Grade
fd1
fd2
fd3
2NF
S_C
S#
CN
Grade
fd1
eliminate partial FDs
S_D
S#
DName
DHe
fd2
fd3
3NF
S_C
S#
CN
Grade
fd1
S_D
S#
DName
fd2
DEPT
DName
DHead
fd3
eliminate transitive FDs

	Test	Remedy (Normalization)
1NF	Relation should have no nonatomic attributes or nested relations.	Form new relations for each nonatomic attribute or nested relation.
2NF	For relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.	Decompose and set up a new relation for each partial key with its dependent attribute(s). Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it.
3NF	Relation should not have a nonkey attribute functionally determined by another nonkey attribute (or by a set of nonkey attributes.) That is, there should be no transitive dependency of a nonkey attribute on the primary key.	Decompose and set up a relation that includes the nonkey attribute(s) that functionally determine(s) other nonkey attribute(s).