Objectives of Normalization

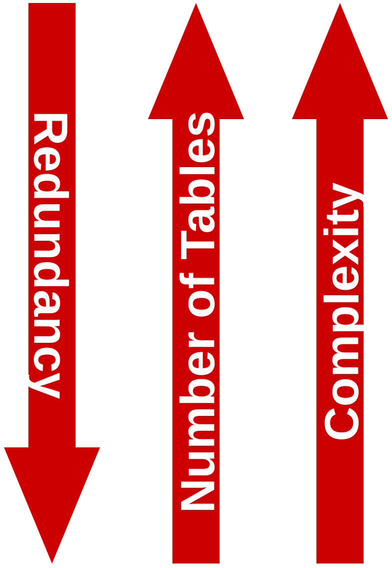
- **How tables that contain redundant data can suffer from update anomalies, which can introduce inconsistencies into a database.**

- **The rules associated with the most commonly used normal forms, namely first (1NF), second (2NF), and third (3NF).**

- **The identification of various types of update anomalies such as insertion, deletion, and modification anomalies.**

Normalization

- **Is a process of deleting different anomalies by splitting the relation into two or more classes**
  - **1NF**
  - **2NF**
  - **3NF**
  - **BCNF(Boyce Coded Normal Form)**
  - **4NF**
  - **5NF**



Data redundancy and update anomalies

- **Problems associated with data redundancy are illustrated by comparing the Staff and Branch tables with the StaffBranch table.**

  - **StaffBranch table has redundant data; the branch information are repeated for every member of staff.**

  - **In contrast, the branch information appears only once for each branch in the Branch table and only the branch number (branchNo) is repeated in the Staff table, to represent where each member of staff is located.**

Data Redundancy and Update Anomalies

staffNo	name	position	salary	branchNo	branchAddress	telNo
S1500	Tom Daniels	Manager	46000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0003	Sally Adams	Assistant	30000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0010	Mary Martinez	Manager	50000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S3250	Robert Chin	Supervisor	32000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S2250	Sally Stern	Manager	48000	B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131
S0415	Art Peters	Manager	41000	B003	14 – 8th Avenue, New York, NY 10012	212-371-3000

Staff

staffNo	name	position	salary	branchNo
S1500	Tom Daniels	Manager	46000	B001
S0003	Sally Adams	Assistant	30000	B001
S0010	Mary Martinez	Manager	50000	B002
S3250	Robert Chin	Supervisor	32000	B002
S2250	Sally Stern	Manager	48000	B004
S0415	Art Peters	Manager	41000	B003

Branch

branchNo	branchAddress	telNo
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
B002	City Center Plaza, Seattle, WA 98122	206-555-6756
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131

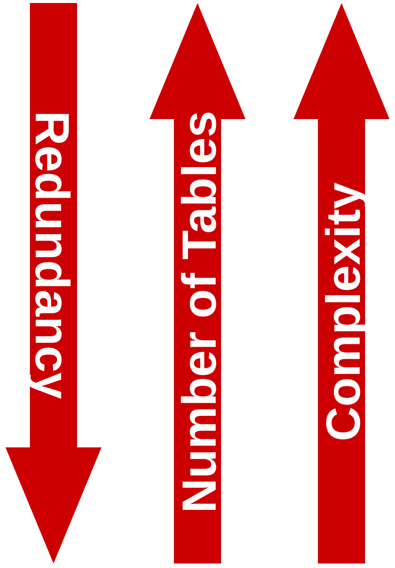
Data Redundancy and Update Anomalies

- **Tables that contain redundant information may potentially suffer from update anomalies.**

- **Types of update anomalies include:**
  - **Insertion**
  - **Deletion**
  - **Modification**

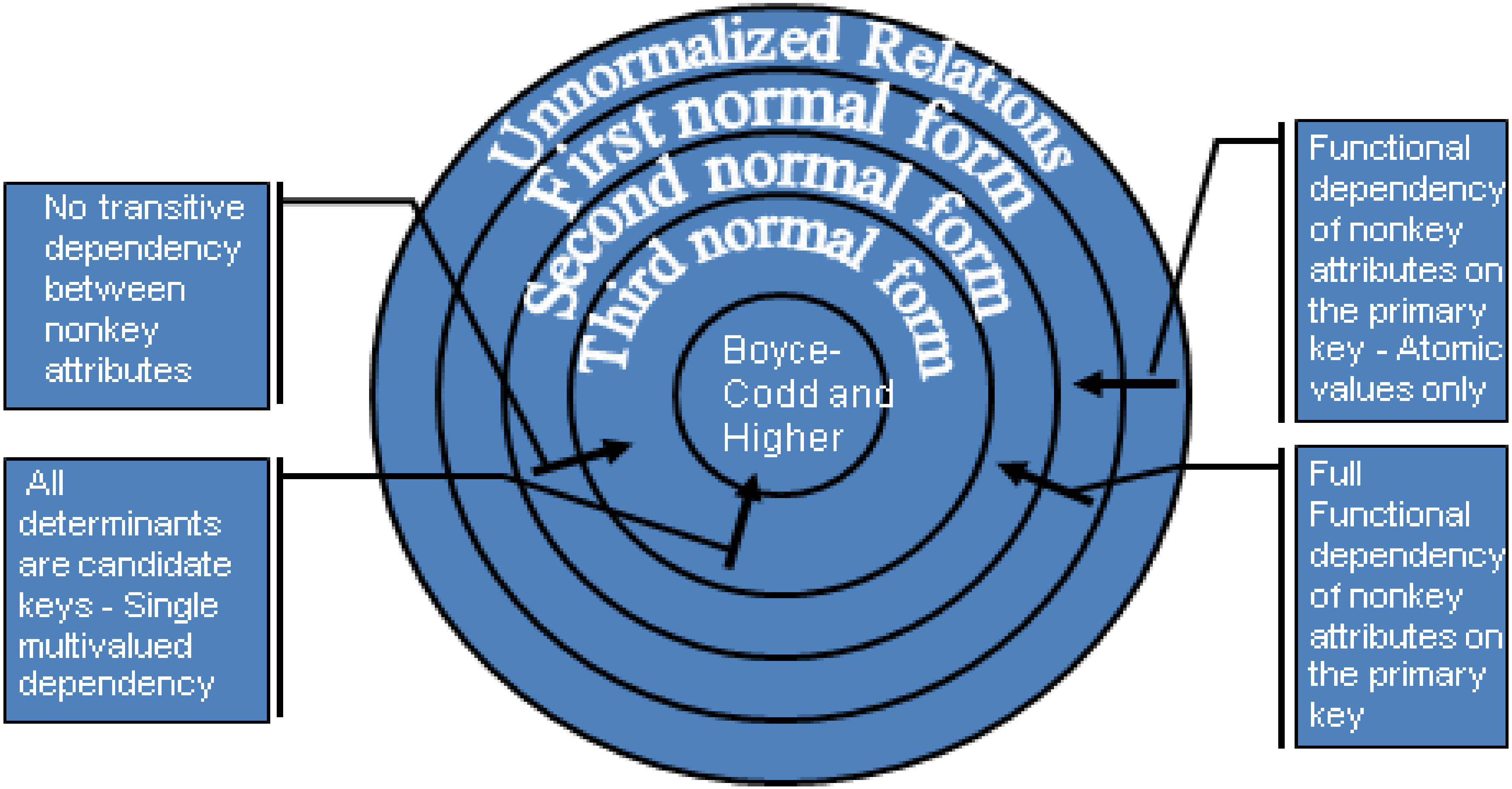
Normalization

- **Is a process of deleting different anomalies by splitting the relation into two or more classes**
  - **1NF**
  - **2NF**
  - **3NF**
  - **BCNF(Boyce Codd Normal Form)**
  - **4NF**
  - **5NF**

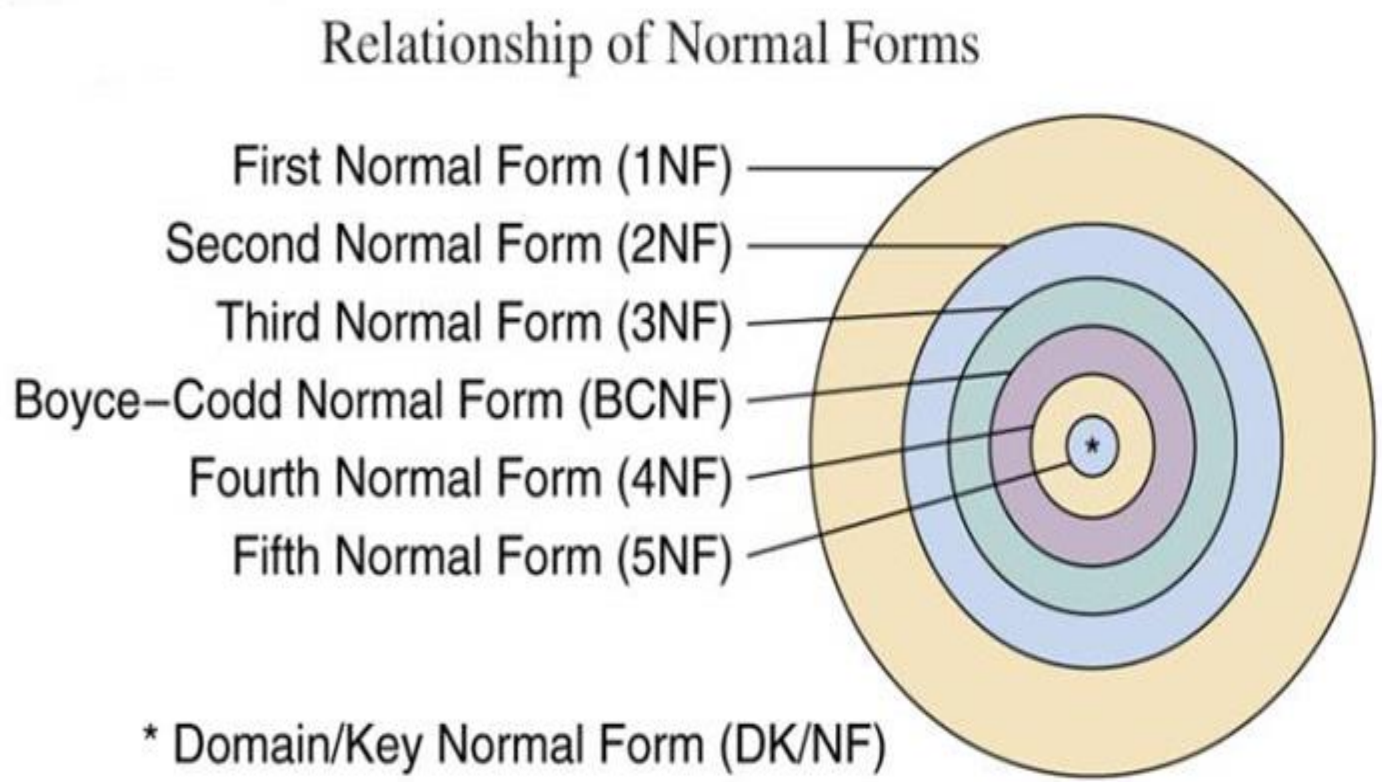


Most databases should be 3NF or BCNF in order to avoid the database anomalies.

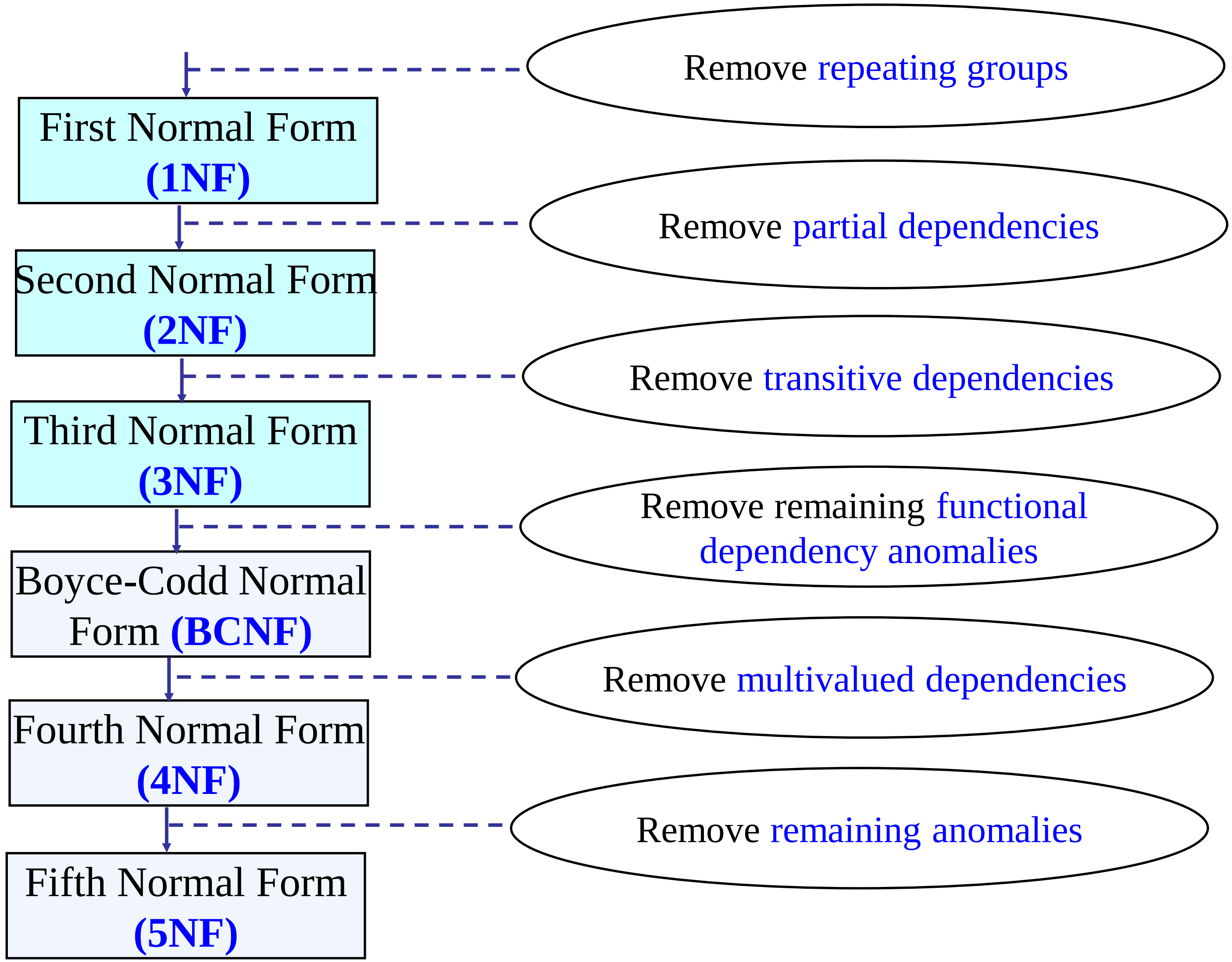
Normalization



Relationship of Normal Forms



Stages of Normalisation



First normal form (1NF)

- **A table in which the intersection of every column and record contains *only one* value.**

Branch table is not in 1NF

branchNo	branchAddress	telNos
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618, 503-555-2727, 503-555-6534
B002	City Center Plaza, Seattle, WA 98122	206-555-6756, 206-555-8836
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131, 206-555-4112

Primary key

More than
one value, so
not in 1NF

Converting Branch table to 1NF

Branch (Not 1NF)

branchNo	branchAddress	telNos
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618, 503-555-2727, 503-555-6534
B002	City Center Plaza, Seattle, WA 98122	206-555-6756, 206-555-8836
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131, 206-555-4112

Take copy of branchNo column to new table to become foreign key

Remove telNos column and create new column called telNo in the new table

Branch (1NF)

branchNo	branchAddress
B001	8 Jefferson Way, Portland, OR 97201
B002	City Center Plaza, Seattle, WA 98122
B003	14 – 8th Avenue, New York, NY 10012
B004	16 – 14th Avenue, Seattle, WA 98128

Primary key

BranchTelephone (1NF)

branchNo	telNo
B001	503-555-3618
B001	503-555-2727
B001	503-555-6534
B002	206-555-6756
B002	206-555-8836
B003	212-371-3000
B004	206-555-3131
B004	206-555-4112

Becomes foreign key

Becomes primary key

Second Normal Form (2NF)

- **2NF only applies to tables with composite primary keys.**

- **A table that is in 1NF and in which the values of each non-primary-key column can be worked out from the values in *all* the columns that make up the primary key.**

- **A functional dependency X $\longrightarrow$ Y is fully functional dependency if removal of any attribute A from X means that the dependency doesn't hold any more.**

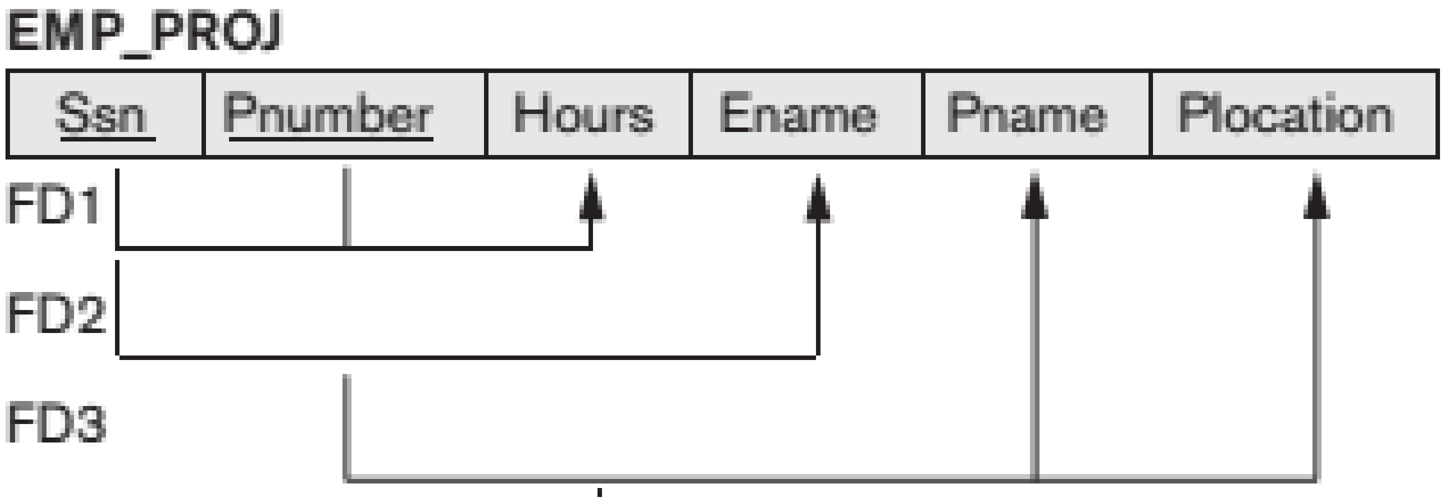
  i.e.: $A \in X, (X - \{A\})$ **does not functionally determine Y**

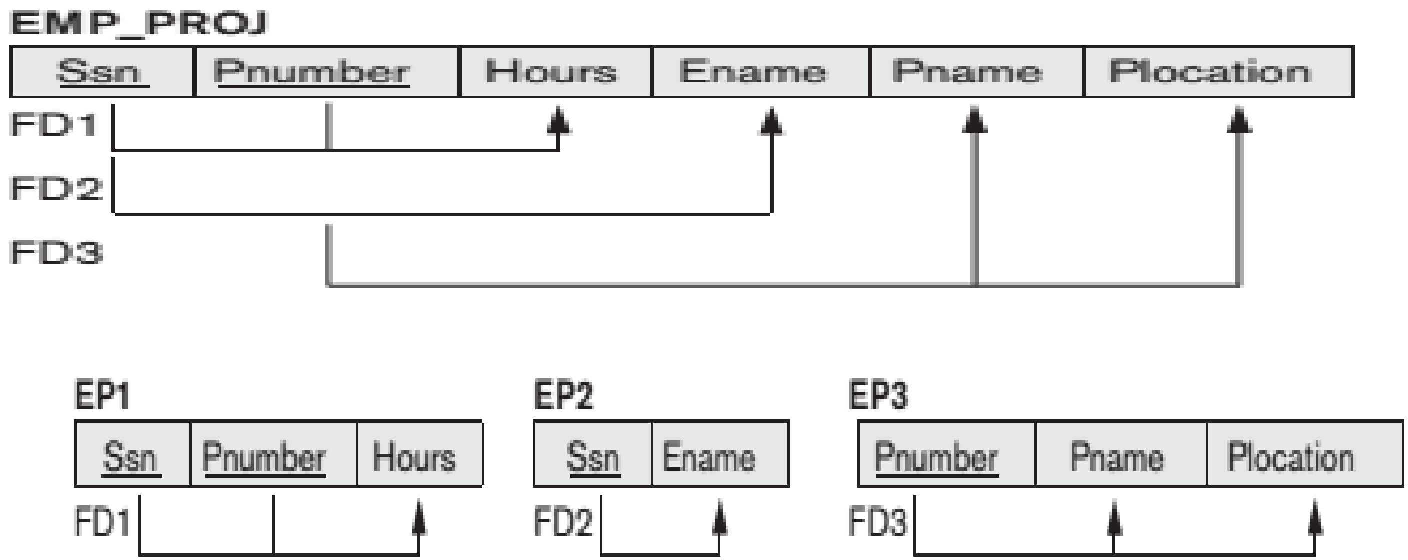
Second normal form (2NF)

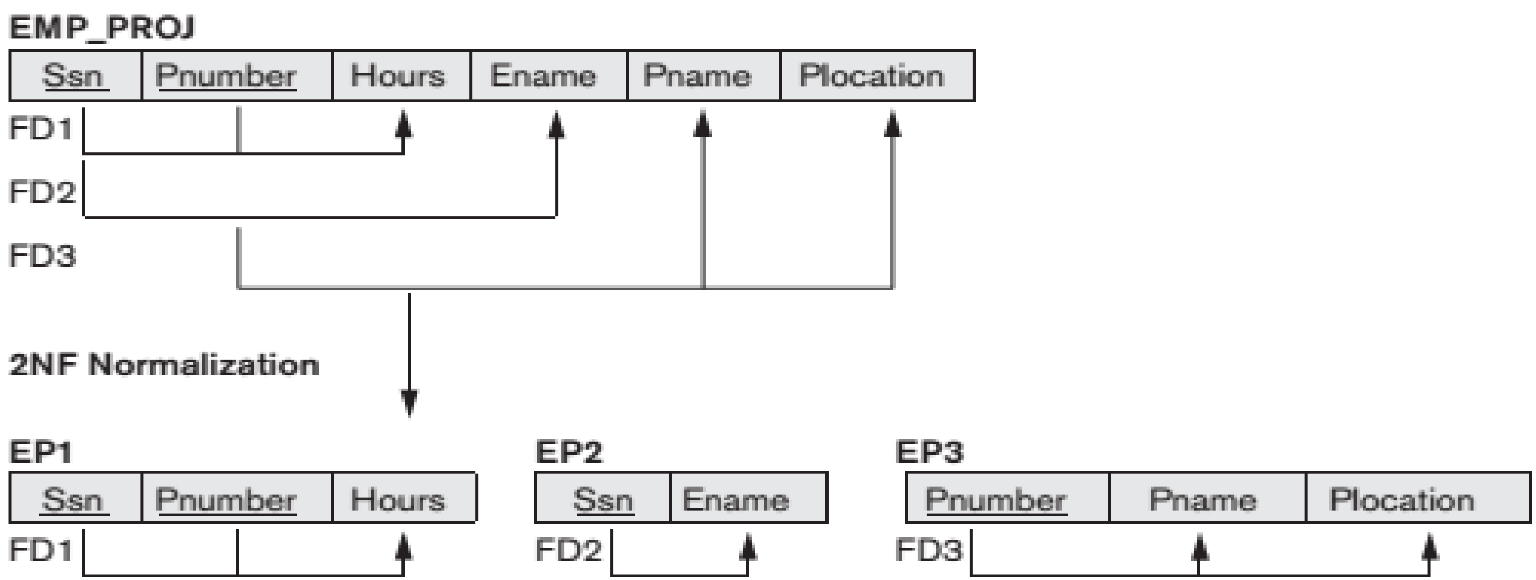
- **Formal definition of 2NF is a table that is in 1NF and every non-primary-key column is fully functional dependent on the primary key.**

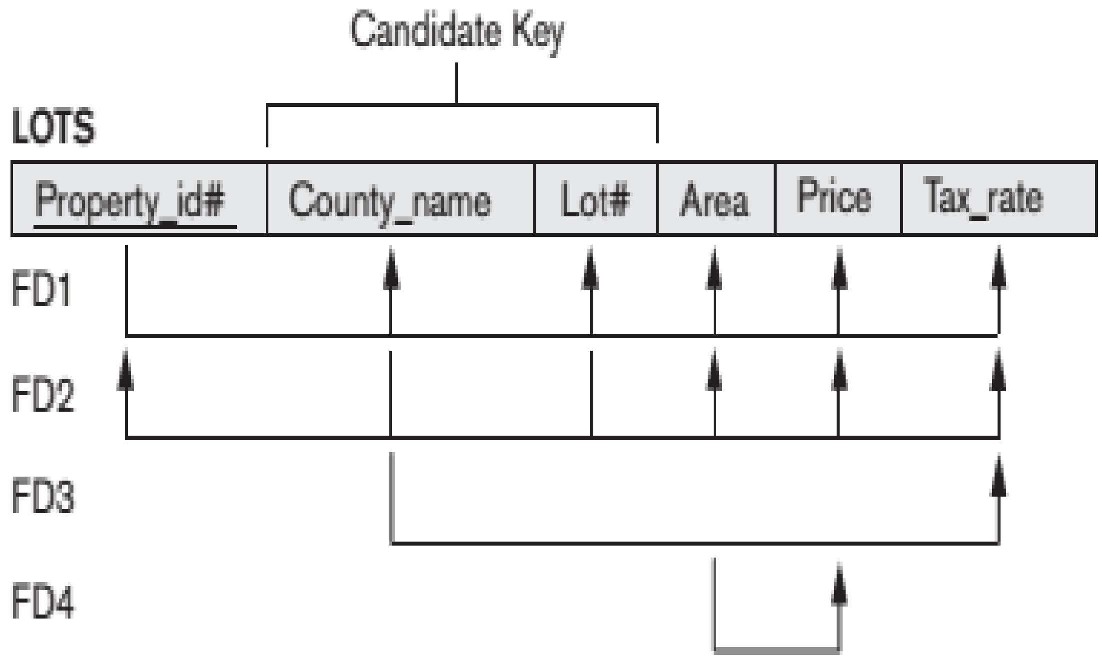
- **Full functional dependency indicates that if A and B are columns of a table, B is fully dependent on A if B is functionally dependent on A but not on any proper subset of A.**

- **A relation schema R is in second normal form if every nonprime attribute A in R is not partially depend on any key of R.**

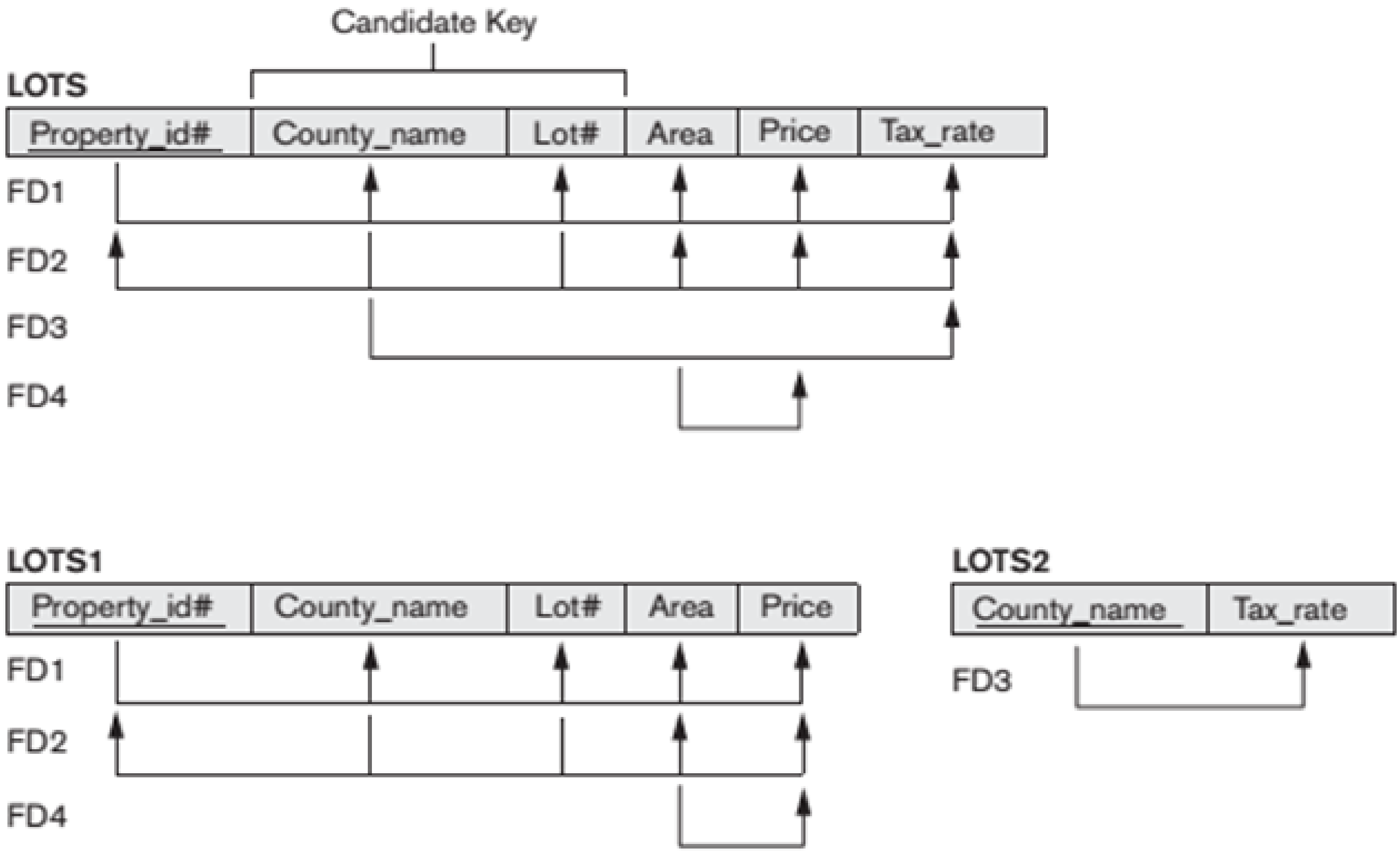

EMP_PROJ
Ssn
Pnumber
Hours
Ename
Pname
Plocation
FD1
FD2
FD3


EMP_PROJ
Ssn
Pnumber
Hours
Ename
Pname
Plocation
FD1
FD2
FD3
EP1
Ssn
Pnumber
Hours
FD1
EP2
Ssn
Ename
FD2
EP3
Pnumber
Pname
Plocation
FD3


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



Candidate Key
LOTS
Property_id#
County_name
Lot#
Area
Price
Tax_rate
FD1
FD2
FD3
FD4


Candidate Key
LOTS
Property_id#
County_name
Lot#
Area
Price
Tax_rate
FD1
FD2
FD3
FD4
LOTS1
Property_id#
County_name
Lot#
Area
Price
FD1
FD2
FD4
LOTS2
County_name
Tax_rate
FD3

TempStaffAllocation table is not in 2NF

staffNo	branchNo	branchAddress	name	position	hoursPerWeek
S4555	B002	City Center Plaza, Seattle, WA 98122	Ellen Layman	Assistant	16
S4555	B004	16 – 14th Avenue, Seattle, WA 98128	Ellen Layman	Assistant	9
S4612	B002	City Center Plaza, Seattle, WA 98122	Dave Sinclair	Assistant	14
S4612	B004	16 – 14th Avenue, Seattle, WA 98128	Dave Sinclair	Assistant	10

Composite primary key

Values in branchAddress column can be worked out from only branchNo, so table *not* in 2NF

Values in branchNo column can be worked out from branchAddress

Values in name and position columns can be worked out from only staffNo, so table *not* in 2NF

Values in hoursPerWeek column can only be worked out from staffNo and branchNo

Converting TempStaffAllocation table to 2NF

Composite primary key

TempStaffAllocation (Not 2NF)

staffNo	branchNo	branchAddress	name	position	hoursPerWeek
S4555	B002	City Center Plaza, Seattle, WA 98122	Ellen Layman	Assistant	16
S4555	B004	16 – 14th Avenue, Seattle, WA 98128	Ellen Layman	Assistant	9
S4612	B002	City Center Plaza, Seattle, WA 98122	Dave Sinclair	Assistant	14
S4612	B004	16 – 14th Avenue, Seattle, WA 98128	Dave Sinclair	Assistant	10

Take copy of staffNo column to new table to become primary key

Take copy of branchNo column to new table to become primary key

Move branchAddress column to new table

Move name column to new table

Move position column to new table

Branch (2NF)

branchNo	branchAddress
B002	City Center Plaza, Seattle, WA 98122
B004	16 – 14th Avenue, Seattle, WA 98128

Becomes primary key

TempStaff (2NF)

staffNo	name	position
S4555	Ellen Layman	Assistant
S4612	Dave Sinclair	Assistant

Becomes primary key

TempStaffAllocation (2NF)

staffNo	branchNo	hoursPerWeek
S4555	B002	16
S4555	B004	9
S4612	B002	14
S4612	B004	10

Becomes foreign key

Composite primary key

Becomes foreign key

Third Normal Form (3NF)

- **The formal definition of 3NF is a table that is in <u>1NF and 2NF</u> and in which no non-primary-key column is *transitively dependent* on the primary key.**

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

- **If a transitive dependency exists on the primary key, the table is not in 3NF.**

- **There should be no transitive dependency of non key attribute on the primary key.**

- **Definition: According to Codd’s original definition, a relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key.**
- **A relation schema R is in 3NF if every nonprime attribute of R meets both of the following conditions:**

  - It is fully functionally dependent on every key of R.

  - It is non transitively dependent on every key of R.

Third normal form (3NF)

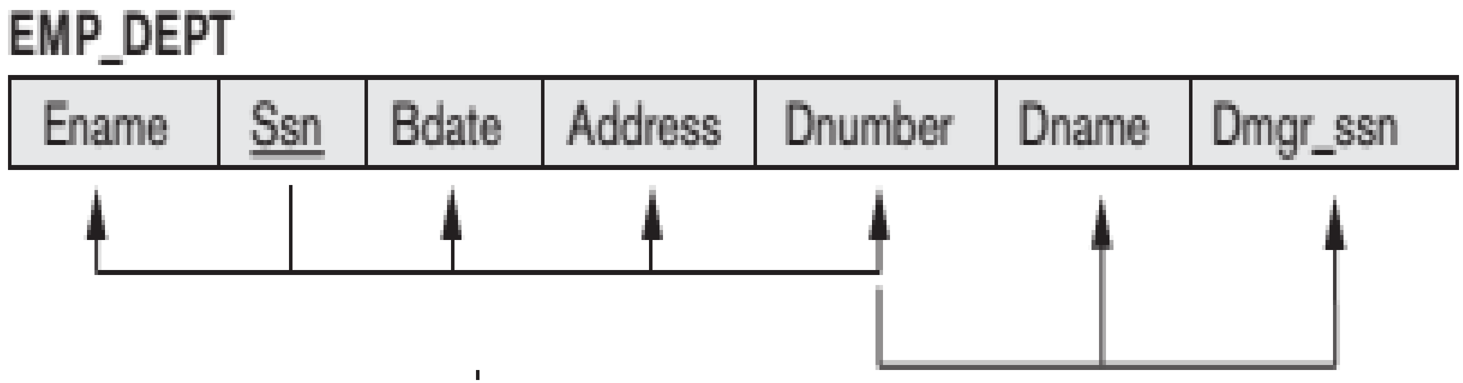
- **Definition: According to Codd's original definition, a relation schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key.**

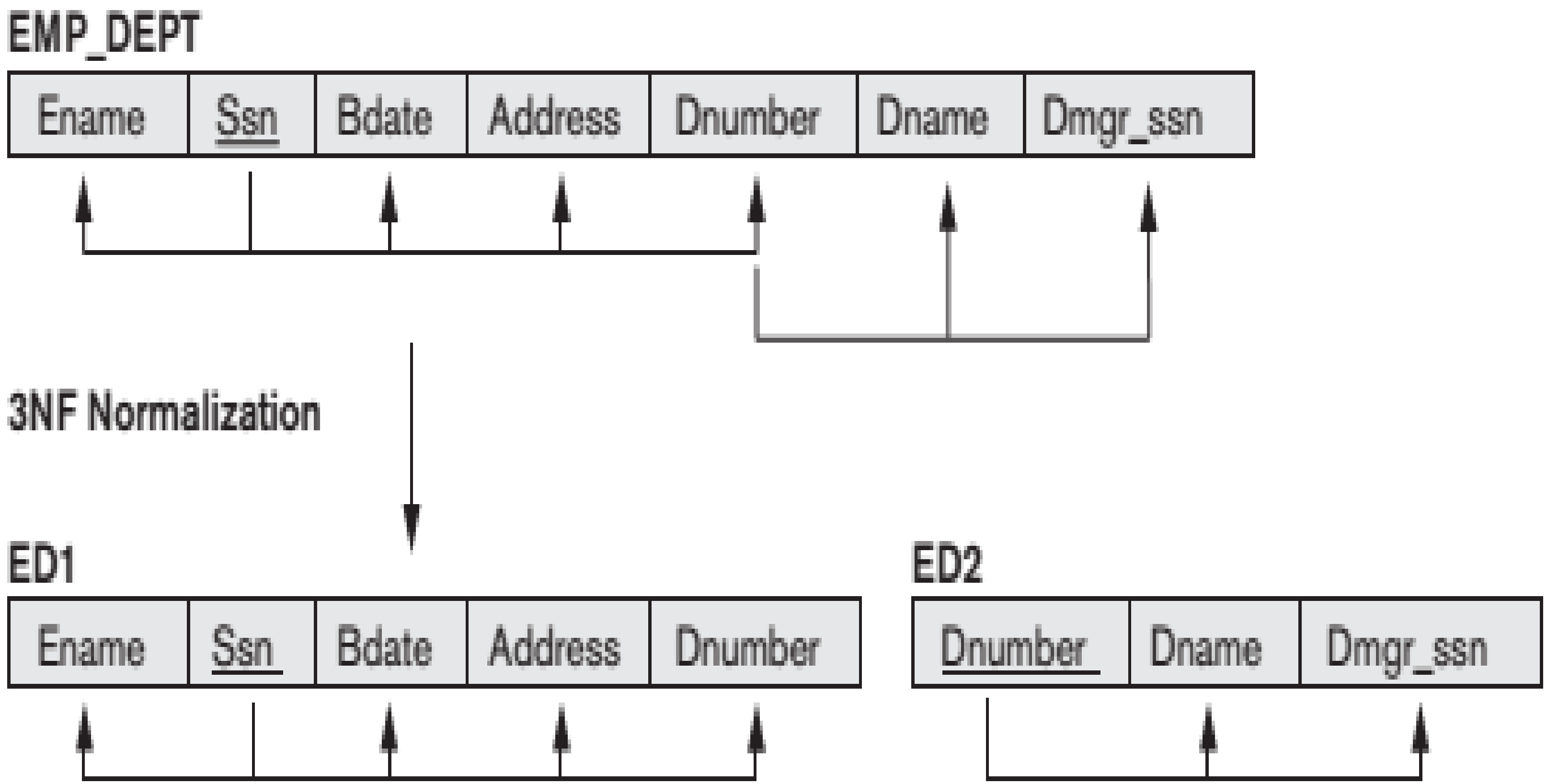
- **A relation schema R is in 3NF if every nonprime attribute of R meets both of the following conditions:**
  - **It is fully functionally dependent on every key of R.**
  - **It is non transitively dependent on every key of R.**

- **Definition: A relation schema R is in third normal form (3NF) if, whenever a non trivial functional dependency $X \rightarrow A$ holds in R, either**

  (a) X is a superkey of R, or

  (b) A is a prime attribute of R.


EMP_DEPT
Ename
Ssn
Bdate
Address
Dnumber
Dname
Dmgr_ssn


EMP_DEPT
Ename
Ssn
Bdate
Address
Dnumber
Dname
Dmgr_ssn
3NF Normalization
ED1
Ename
Ssn
Bdate
Address
Dnumber
ED2
Dnumber
Dname
Dmgr_ssn

StaffBranch table is not in 3NF

StaffBranch (Not 3NF)

staffNo	name	position	salary	branchNo	branchAddress	telNo
S1500	Tom Daniels	Manager	46000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0003	Sally Adams	Assistant	30000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0010	Mary Martinez	Manager	50000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S3250	Robert Chin	Supervisor	32000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S2250	Sally Stern	Manager	48000	B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131
S0415	Art Peters	Manager	41000	B003	14 – 8th Avenue, New York, NY 10012	212-371-3000

Primary key

Values in all non-primary-key columns can be worked out from the primary key, staffNo

Values in branchAddress and telNo columns can be worked out from branchNo, so table *not* in 3NF

Values in branchNo and telNo columns can be worked out from branchAddress, so table *not* in 3NF

Values in branchNo and branchAddress columns can be worked out from telNo, so table *not* in 3NF

Converting the StaffBranch table to 3NF

StaffBranch (Not 3NF)

staffNo	name	position	salary	branchNo	branchAddress	telNo
S1500	Tom Daniels	Manager	46000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0003	Sally Adams	Assistant	30000	B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
S0010	Mary Martinez	Manager	50000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S3250	Robert Chin	Supervisor	32000	B002	City Center Plaza, Seattle, WA 98122	206-555-6756
S2250	Sally Stern	Manager	48000	B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131
S0415	Art Peters	Manager	41000	B003	14 – 8th Avenue, New York, NY 10012	212-371-3000

Take copy of branchNo column to new table to become primary key

Move branchAddress column to new table

Move telNo column to new table

Branch (3NF)

branchNo	branchAddress	telNo
B001	8 Jefferson Way, Portland, OR 97201	503-555-3618
B002	City Center Plaza, Seattle, WA 98122	206-555-6756
B003	14 – 8th Avenue, New York, NY 10012	212-371-3000
B004	16 – 14th Avenue, Seattle, WA 98128	206-555-3131

Becomes primary key

Becomes candidate key

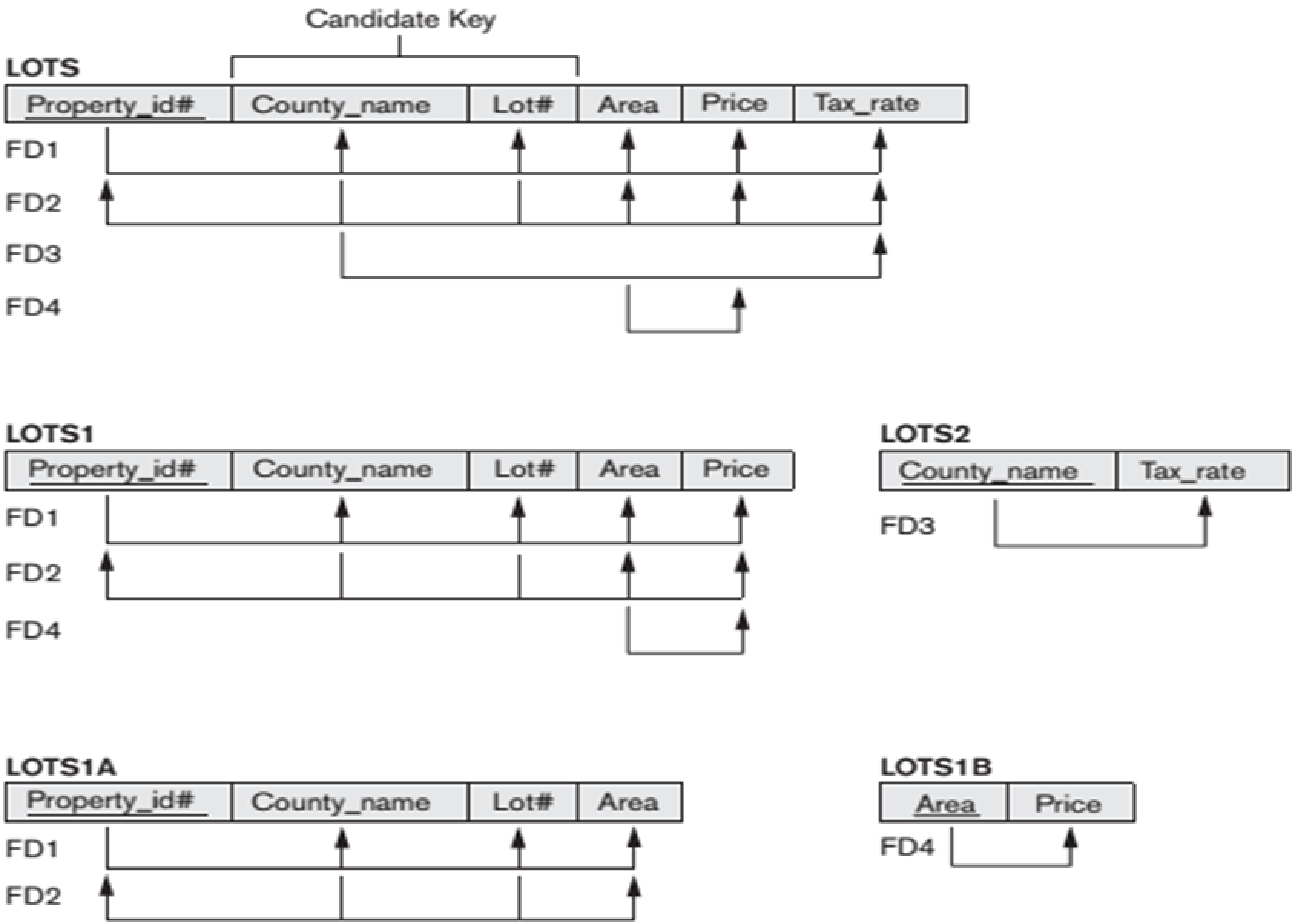
Becomes candidate key

Staff (3NF)

staffNo	name	position	salary	branchNo
S1500	Tom Daniels	Manager	46000	B001
S0003	Sally Adams	Assistant	30000	B001
S0010	Mary Martinez	Manager	50000	B002
S3250	Robert Chin	Supervisor	32000	B002
S2250	Sally Stern	Manager	48000	B004
S0415	Art Peters	Manager	41000	B003

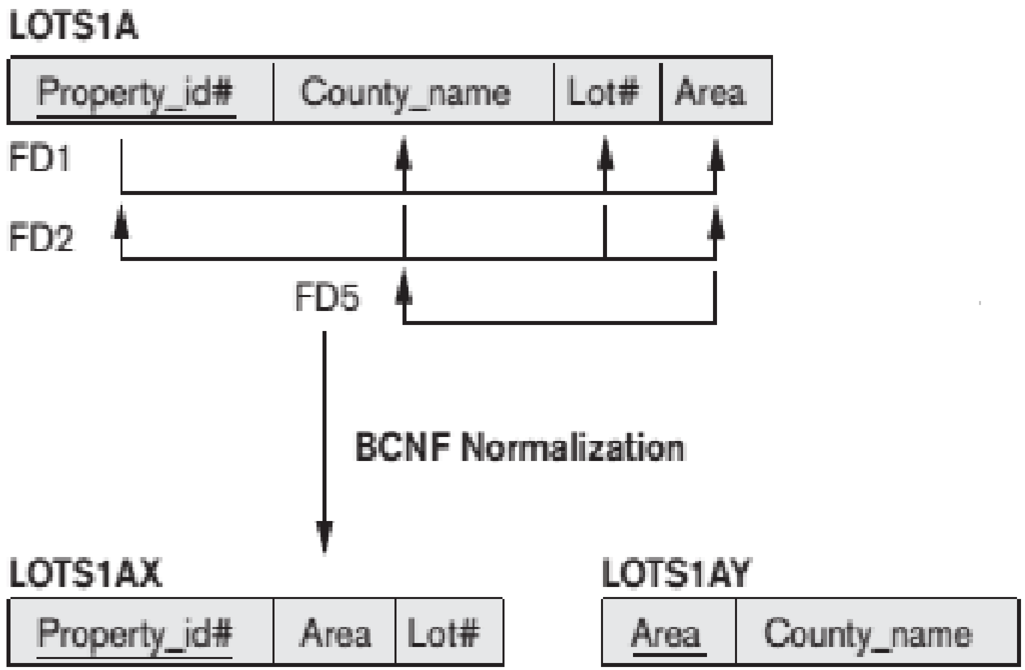
Primary key

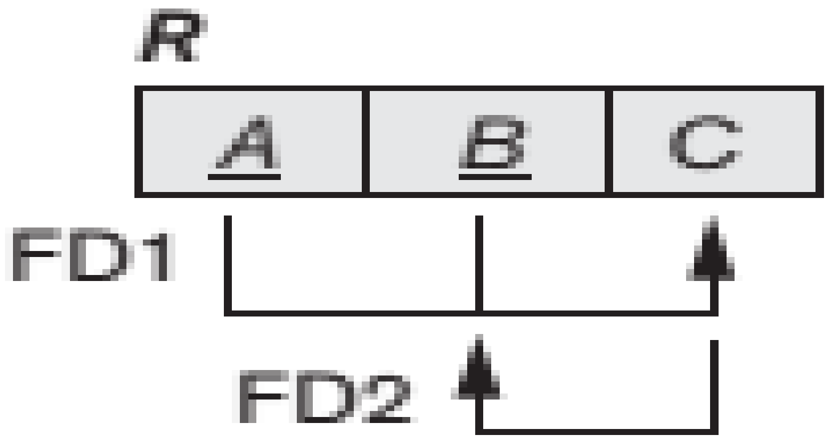
Becomes foreign key


Candidate Key
LOTS
Property_id#
County_name
Lot#
Area
Price
Tax_rate
FD1
FD2
FD3
FD4
LOTS1
Property_id#
County_name
Lot#
Area
Price
FD1
FD2
FD4
LOTS2
County_name
Tax_rate
FD3
LOTS1A
Property_id#
County_name
Lot#
Area
FD1
FD2
LOTS1B
Area
Price
FD4

Boyce-Codd Normal Form (BCNF)

- **A relation schema R is in BCNF if whenever a non trivial functional dependency X→A holds in R, then X is a superkey of R.**




R
A
B
C
FD1
FD2

R = {Student ID, Last Name, First Name, Course ID, Course Section, Course Name, Grade, Professor Last Name, Professor First Name, Bldg, Office #, Semester}

FD1 = Student ID --> Last Name, First Name

FD2 = Student ID, Semester, C_ID, C_Section → Grade

FD4 = Course ID → Course Name

FD3 = Semester, C_ID, C_Section --> Prof_LName, Prof_FName, Bldg, Office#

Given the following relational schema & constraints. Normalize relation schema to BCNF

Vendor(ID, Name, Account_No, Bank_Code_No, Bank)

1. **Name, Account_No, Bank_Code_No are functionally dependent on ID**

   ID $\rightarrow$ {Name, Account_No, Bank_Code_No}

2. **Bank is functionally dependent on Bank_Code_No**

   Bank_Code_No $\rightarrow$ Bank

Vendor(ID, Name, Account_No, Bank_Code_No, Bank)

1. Name, Account_No, Bank_Code_No are functionally dependent on ID

ID $\rightarrow$ {Name, Account_No, Bank_Code_No}

2. Bank is functionally dependent on Bank_Code_No

Bank_Code_No $\rightarrow$ Bank

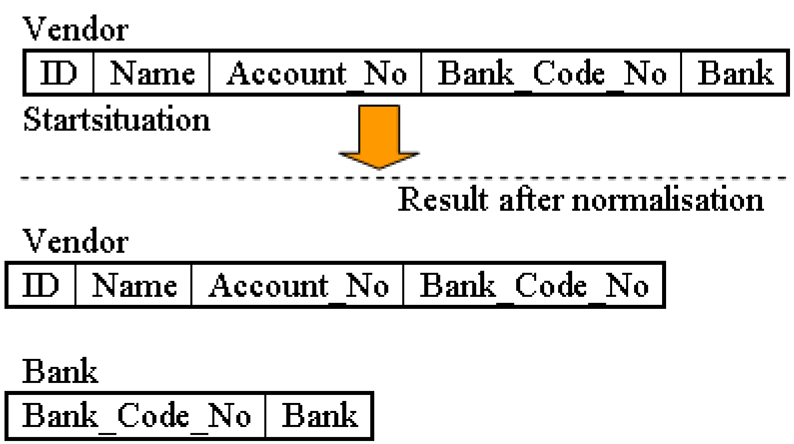
Vendor(ID, Name, Account_No, Bank_Code_No, Bank)

1. Name, Account_No, Bank_Code_No are functionally dependent on ID

$$\textbf{ID} \rightarrow \{\textbf{Name, Account\_No, Bank\_Code\_No}\}$$

2. Bank is functionally dependent on Bank_Code_No

$$\textbf{Bank\_Code\_No} \rightarrow \textbf{Bank}$$



Fourth Normal Form (4NF)

- **4NF: A relation that is in Boyce-Codd Normal Form and contains no MVDs.**

- **BCNF to 4NF involves the removal of the MVD from the relation by placing the attribute(s) in a new relation along with a copy of the determinant(s).**

Consider the following relationship: R(A,B,C,D) and following dependencies:

$A \rightarrow BCD$

$BC \rightarrow AD$

$D \rightarrow B$

Normalize the relation schema.

Consider the following relationship : **R (A,B,C,D)**

and following dependencies :

A -> BCD
BC -> AD
D -> B

Above relationship is already in 3rd NF. Keys are **A** and **BC.**

Hence, in the functional dependency, **A -> BCD**, A is the super key.
in second relation, **BC -> AD**, BC is also a key.
but in, **D -> B**, D is not a key.

Hence we can break our relationship R into two relationships **R1** and **R2**.

R(A, B, C, D)

R1 (A, D, C)   R2(D, B)

Breaking, table into two tables, one with A, D and C while the other with D and B.

Lossless-join Decomposition

- **For the case of $R = (R_1, R_2)$, we require that for all possible relations r on schema R**

$$r = \prod_{R1}(r) \bowtie \prod_{R2}(r)$$

- **A decomposition of R into R_1 and R_2 is lossless join if and only if at least one of the following dependencies is in F^+:**
  - $R_1 \cap R_2 \rightarrow R_1$
  - $R_1 \cap R_2 \rightarrow R_2$

- **If $R_1 \cap R_2$ forms a superkey of either R_1 or R_2, the decomposition of R is lossless decomposition.**

bor_loan → (customer_id, loan_number, amount)

⇓ decomposed into

borrower → (customer_id, loan_number)

loan → (loan_number, amount)

Here **borrower ∩ *loan= loan_number***

thus it is lossless decomposition

Check the following:

- $R = (A, B, C)$
  $F = \{A \rightarrow B, B \rightarrow C)$
  - **Can be decomposed in two different ways**
    - $R_1 = (A, B), \quad R_2 = (B, C)$
      - **Lossless-join decomposition:**

        $R_1 \cap R_2 = \{B\}$ **and** $B \rightarrow BC$
      - **Dependency preserving**

    - $R_1 = (A, B), \quad R_2 = (A, C)$
      - **Lossless-join decomposition:**

        $R_1 \cap R_2 = \{A\}$ **and** $A \rightarrow AB$
      - **Not dependency preserving**

Let F_i **be the set of dependencies** F^+ **that include only attributes in** R_i**.**

- **A decomposition is dependency preserving, if**

  $(F_1 \cup F_2 \cup \ldots \cup F_n)^+ = F^+$
- **If it is not, then checking updates for violation of functional dependencies may require computing joins, which is expensive.**

BCNF Decomposition

- $R = (A, B, C)$
  $F = \{A \rightarrow B$
  $B \rightarrow C\}$
  Key = $\{A\}$
  - R is not in BCNF ($B \rightarrow C$ but B is not superkey)
  - Decomposition
    – $R_1 = (B, C)$
    – $R_2 = (A, B)$

Comparison of BCNF and 3NF

- It is always possible to decompose a relation into a set of relations that are in 3NF such that:
  – the decomposition is lossless
  – the dependencies are preserved

- It is always possible to decompose a relation into a set of relations that are in BCNF such that:
  – the decomposition is lossless
  – it may not be possible to preserve dependencies.

Comparison of BCNF and 3NF

- **It is always possible to decompose a relation into a set of relations that are in 3NF such that:**
  - **Decomposition is lossless**
  - **Dependencies are preserved**

- **It is always possible to decompose a relation into a set of relations that are in BCNF such that:**
  - **the decomposition is lossless**
  - **it may not be possible to preserve dependencies.**

Multivalued Dependencies (MVDs)

- **Let R be a relation schema and let $\alpha \subseteq$ R and $\beta \subseteq$ R. The multivalued dependency**

$$\alpha \twoheadrightarrow \beta$$

holds on R if in any legal relation r(R), for all pairs for tuples t_1 and t_2 in r such that $t_1[\alpha] = t_2[\alpha]$, there exist tuples t_3 and t_4 in r such that:

$$t_1[\alpha] = t_2[\alpha] = t_3[\alpha] = t_4[\alpha]$$
$$t_3[\beta] = t_1[\beta]$$
$$t_3[R - \beta] = t_2[R - \beta]$$
$$t_4[\beta] = t_2[\beta]$$
$$t_4[R - \beta] = t_1[R - \beta]$$

E.g.:

	α	β	$R - \alpha - \beta$
t_1	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$a_{j+1} \ldots a_n$
t_2	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$b_{j+1} \ldots b_n$
t_3	$a_1 \ldots a_i$	$a_{i+1} \ldots a_j$	$b_{j+1} \ldots b_n$
t_4	$a_1 \ldots a_i$	$b_{i+1} \ldots b_j$	$a_{j+1} \ldots a_n$

Use of Multivalued Dependencies

- **We use multivalued dependencies in two ways:**
  1. **To test relations to determine whether they are legal under a given set of functional and multivalued dependencies**

  2. **To specify constraints on the set of legal relations. We shall thus concern ourselves only with relations that satisfy a given set of functional and multivalued dependencies**

- **If a relation r fails to satisfy a given multivalued dependency, we can construct a relations r′ that does satisfy the multivalued dependency by adding tuples to r.**

- **Fourth Normal Form**
  - **A relation schema *R* is in 4NF with respect to a set *D* of functional and multivalued dependencies if for all multivalued dependencies in D^+ of the form $\alpha \rightarrow\rightarrow \beta$, where $\alpha \subseteq R$ and $\beta \subseteq R$, at least one of the following hold:**
    - **$\alpha \rightarrow\rightarrow \beta$ is trivial (i.e., $\beta \subseteq \alpha$ or $\alpha \cup \beta = R$)**
    - **α is a superkey for schema *R***

  - **If a relation is in 4NF it is in BCNF**

- # Non Additive Join Decomposition into 4 NF

Whenever we decompose a relation schema into $R_1 = (X \cup Y)$ **and** $R_2 = (R - Y)$ **based on an** $MVD: X \rightarrow\rightarrow Y$ **that holds in R, the decomposition has the non additive join property.**

$$R_1 \cap R_2 \rightarrow\rightarrow (R_1 - R_2)$$

$$OR$$

$$R_1 \cap R_2 \rightarrow\rightarrow (R_2 - R_1)$$

Algorithm: Input: A universal relation R and a set of functional & multivalued dependencies F.

1. **Set** $D := \{R\}$
2. **While there is a relational schema Q in D that is not in 4NF, do**

{ choose a relation schema Q in D that is not in 4NF;

find a nontrivial $MVD: X \rightarrow\rightarrow Y$ **in Q that violates 4NF;**

replace Q in D by two relation schemas $(Q-Y)$ *and* $(X \cup Y)$**;**

};

- **Decompose the relation schema R in to 4NF with Nonadditive join property**

R =(A, B, C, G, H, I)

F ={ $A \rightarrow\rightarrow B$

$B \rightarrow\rightarrow HI$

$CG \rightarrow\rightarrow H$ }

- $R = (A, B, C, G, H, I)$
  $F = \{ A \rightarrow\rightarrow B$
  $B \rightarrow\rightarrow HI$
  $CG \rightarrow\rightarrow H \}$
- R is not in 4NF since $A \rightarrow\rightarrow B$ and A is not a superkey for R

- Decomposition

  a) $R_1 = (A, B)$ (R_1 is in 4NF)

  b) $R_2 = (A, C, G, H, I)$ (R_2 is not in 4NF)

  c) $R_3 = (C, G, H)$ (R_3 is in 4NF)

  d) $R_4 = (A, C, G, I)$ (R_4 is not in 4NF)

- Since $A \rightarrow\rightarrow B$ and $B \rightarrow\rightarrow HI$, $A \rightarrow\rightarrow HI$, $A \rightarrow\rightarrow I$
  e) $R_5 = (A, I)$ (R_5 is in 4NF)
  f) $R_6 = (A, C, G)$ (R_6 is in 4NF)

MVD multi-valued dependency

- **Represents a dependency between attributes (for example, A, B, and C) in a relation, such that for each value of A there is a set of values for B, and a set of values for C.**

- **However, the set of values for B and C are <u>independent of each other</u>.**

Normalization BCNF to 4NF Relations

Branch_Staff_Client relation

Branch_No	*SName*	*CName*
B3	Ann Beech	Aline Stewart
B3	David Ford	Aline Stewart
B3	Ann Beech	Mike Richie
B3	David Ford	Mike Richie

↓

Branch_Staff relation

Branch_No	*SName*
B3	Ann Beech
B3	David Ford

Branch_Client relation

Branch_No	*CName*
B3	Aline Stewart
B3	Mike Richie

Computation of Super key from FD's

Given: Drinkers(name, addr, beersLiked, manf, favBeer)

Reasonable FD's to assert:

1. **name -> addr**
2. **name -> favBeer**
3. **beersLiked -> manf**

name	addr	beersLiked	manf	favBeer
Janeway	Voyager	Bud	A.B.	WickedAle
Janeway	Voyager	WickedAle	Pete's	WickedAle
Spock	Enterprise	Bud	A.B.	Bud

Because beersLiked -> manf

Because name -> addr

Because name -> favBeer

Drinkers (name, addr, beersLiked, manf, favBeer)

{name, beersLiked} is a superkey because together these attributes determine all the other attributes.

Name $\rightarrow$ {addr, favBeer}

beersLiked $\rightarrow$ manf