CMSC 127

Functional Dependencies and Normalization for Relational Databases

Reginald Neil C. Recario

Institute of Computer Science

University of the Philippines Los Baños

Relational Database Design

- What is relational database design?
  - The grouping of attributes to form "good" relation schemas
- Two levels of goodness of relation schemas
  - Logical (or conceptual) level - "user view" level
  - Implementation (or storage) level - "base relation" level

Measures of Quality

- Semantics of the Relation Attributes
- Redundant Information in Tuples and Update Anomalies
- Null Values in Tuples
- Spurious Tuples

Semantics of the Relation Attributes

- **Semantics** (of a relation) – refers to the interpretation of attribute values in a tuple

Semantics of the Relation Attributes

- Example:
  - EMPLOYEE(Ename, Ssn, Bdate, Address, Dnumber)
  - EMPPROJDEPT(Ename,Projname,Deptname)
- The ease with which the meaning of a relation's attributes can be explained is a measure of how well the relation is designed.

Guideline 1

- Each tuple in a relation should represent one entity or relationship instance only.
  - Attributes of different entities 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.

Guideline 1

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	291Berry, Bellaire, TX	4
Narayan, Ramesh 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	Dmgr_ssn
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Bottom Line

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

Redundant Information in Tuples

- Effects:
  - Wastes storage
  - Causes **update** anomalies
    - Insertion anomalies
    - Deletion anomalies
    - Modification anomalies

Insertion Anomaly

- means that some data cannot be easily inserted into the database

Insertion Anomaly

- **Example:**
  - Consider the relation: EMP_PROJ(Ssn, Pnumber, Hours, Ename, Pname, Plocation)
- Insertion Anomaly:
  - Cannot insert a project unless an employee is assigned to it
  - Cannot insert an employee unless a he/she is assigned to a project

Deletion Anomaly

- means deleting some data can cause other information to be lost

Deletion Anomaly

- Referring to the same relation…
  - EMP_PROJ(<u>Ssn</u>, Pnumber, Hours, Ename, Pname, Plocation)
  - When a project is deleted, it will result in deleting all the employees who work on that project.
  - If an employee is the sole employee on a project, deleting that employee would result in deleting the corresponding project.

Modification Anomaly

- means we have to change all copies of the redundant data or else the database will become inconsistent

Modification Anomaly

- Referring to the same relation…
  - EMP_PROJ(<u>Ssn</u>, Pnumber, Hours, Ename, Pname, Plocation)
  - 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.

Guideline 2

- Design a schema that does not suffer from the insertion, deletion and modification anomalies.
- If there are any anomalies present, then note them so that applications can be made to take them into account.

Null Values in Tuples

- Reasons for nulls:
  - Attribute not applicable or invalid
  - Attribute value unknown (may exist)
  - Value known to exist, but unavailable

Null Values in Tuples

- Can waste storage space
- Leads to difficulty of performing selections, aggregation operations and specifying joins

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

Spurious Tuples

- additional tuples that represent invalid information
- A possible result of a JOIN operation on relations that are poorly designed
- **Example:**
  - Consider the relation: EMP_PROJ(Ssn, Pnumber, Hours, Ename, Pname, Plocation)

Example:

(a)

EMP_LOCS

ENAME	PLOCATION

p.k.

EMP_PROJ1

SSN	PNUMBER	HOURS	PNAME	PLOCATION

p.k.

Example:

(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

Example:

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

Example:

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.

⋮

Guideline 4

- Relations should be designed in a way that no spurious tuples will be generated if a natural-join operation is applied to them.

Functional Dependency

- Is a constraint between two sets of attributes from the database
- Used to define normal forms for relations
- denoted by X ->Y
  - Y is functionally dependent on X
  - Attribute X functionally determines attribute Y
  - Value of X determines a unique value for Y

Functional Dependency

- **Formal Definition**: 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 holds if whenever two tuples have the same value for X, they must have the same value for Y
- If K is a key of R, then K functionally determines all attributes in R

Example:

- Consider the table EMP_PROJ(Ssn, Pnumber, Hours, Ename, Pname, Plocation)
- Functional Dependencies:
  - SSN -> ENAME
  - PNUMBER -> {PNAME, PLOCATION}
  - {SSN, PNUMBER} -> HOURS

Functional Dependencies

- A FD is a property of the meaning of data and hold at all times: certain FD's can be ruled out based on a given state of the database.

TEACH

Teacher	Course	Text
Smith	Data Structures	Bartram
Smith	Data Management	Martin
Hall	Compilers	Hoffman
Brown	Data Structures	Horowitz

Inference Rules for FDs

- **Armstrong's inference rules:**
  - IR1. (Reflexive) If Y subset-of X, then X -> Y
  - IR2. (Augmentation) If X -> Y, then XZ -> YZ
  - IR3. (Transitive) If X -> Y and Y -> Z,

    then X -> Z

Additional Inference Rules

- IR4. (Decomposition) If X -> YZ, then X -> Y and X -> Z
- IR5. (Union) If X -> Y and X -> Z, then X -> YZ
- IR6. (Pseudotransitivity) If X -> Y and WY -> Z, then WX -> Z

Additional Inference Rules

- The last three inference rules, as well as any other inference rules, can be deduced from IR1, IR2, and IR3.

Closure of F (F^+)

- The set of all dependencies that include F as well as all dependencies that can be inferred from F.
- F = {

  Ssn -> {Ename, Bdate, Address, Dnumber},

  Dnumber -> {Dname, Dmgr_ssn}

  }

Normalization of Relations

- **Normalization:**
  - the process of decomposing unsatisfactory "bad" relations by breaking up their attributes into smaller relations

Normalization of Relations

- **Normalization:**
  - takes a relation schema through a series of tests to certify whether it satisfies a certain normal form
  - is carried out in practice to minimize redundancy and insertion, deletion and modification anomalies

Normal Forms

- 1NF
  - based on atomicity of data
- 2NF, 3NF, BCNF
  - based on keys and FDs of a relation schema
- 4NF
  - based on keys, multi-valued dependencies
- 5NF
  - based on keys, join dependencies

Normal Forms

- The **normal form of a relation** refers to the highest normal form condition that it meets.
- The database designers *need not* normalize to the highest possible normal form
  - (usually up to 3NF, BCNF or 4NF)

Review

- Superkey
- Key
- Candidate Key
- Primary Key
- **Prime attribute** –an attribute that is a member of *some* candidate keys
- **Nonprime attribute** – an attribute that is not a member of any candidate key

First Normal Form (1NF)

- **Definition:** 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.
- Disallows
  - composite attributes
  - multivalued attributes
  - complex attributes

Normalizing into 1NF

- NOT IN 1NF (with one multivalued attribute)

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

- IN 1NF

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocation
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

Normalizing into 1NF

- NOT I

DEPARTMENT

Dname	Dnumber	Dmgr_ssn	Dlocations
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

- IN 1N

DEPARTMENT

Dname	Dnumber	Dmgr_ssn
Research	5	333445555
Administration	4	987654321
Headquarters	1	888665555

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Normalizing into 1NF

EMP_PROJ

		Projs	
Ssn	Ename	Pnumber	Hours

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
888665555	Borg, James E.	20	NULL

Normalizing into 1NF

- Not in 1NF (with one complex attribute)

EMP_PROJ

		Projs	
Ssn	Ename	Pnumber	Hours

- In 1NF

EMP_PROJ1

Ssn	Ename

EMP_PROJ2

Ssn	Pnumber	Hours

Normalizing into 1NF

- Not in 1NF (with two multivalued attributes)
  - PERSON (SS#, {Carlicense#}, {Phone#})

- In 1NF
  - PERSON1(SS#, Carlicense#)
  - PERSON2(SS#, Phone#)

Second Normal Form (2NF)

- **Full functional dependency:** a FD Y -> Z where removal of any attribute from Y means the FD does not hold anymore
- Examples:
  - {SSN, PNUMBER} -> HOURS
  - {SSN, PNUMBER} -> ENAME

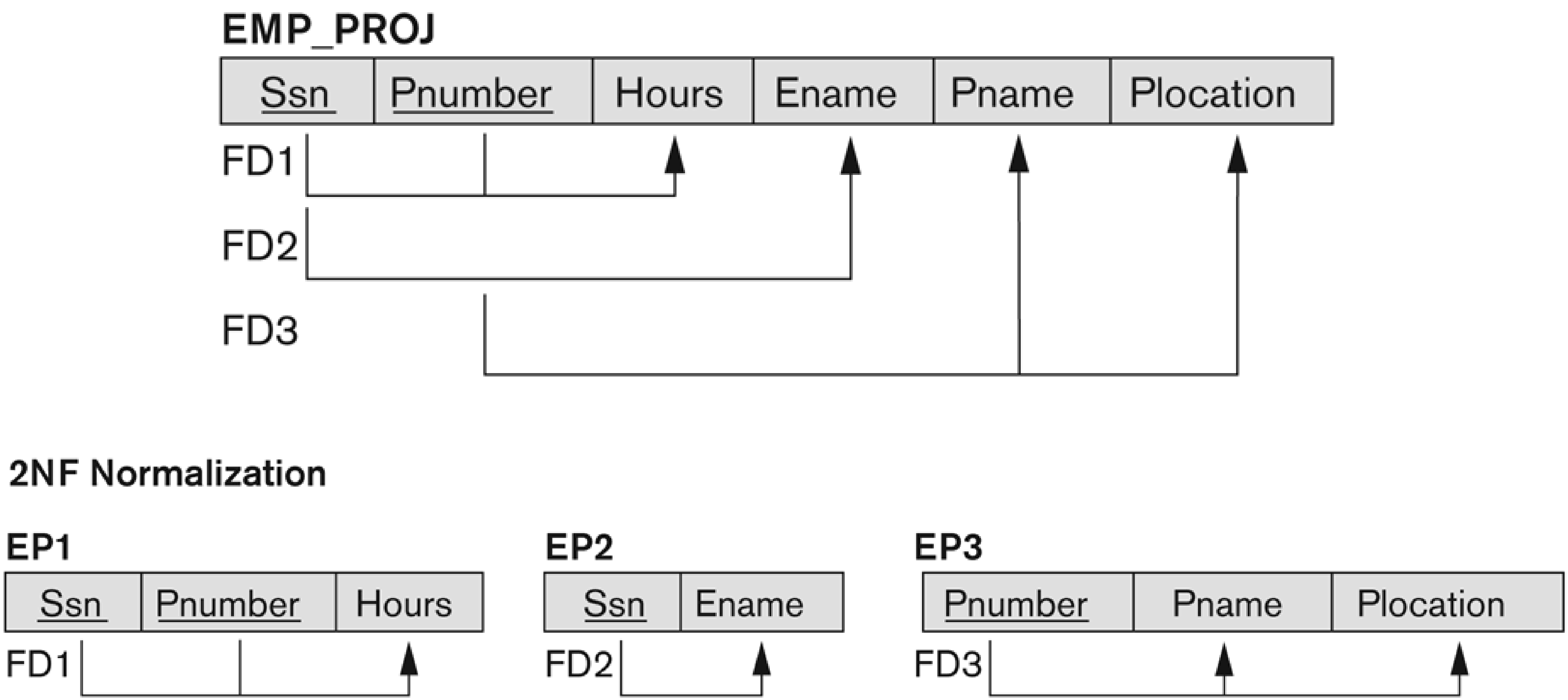
Second Normal Form (2NF)

- **Definition:** 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.

Second Normal Form (2NF)

- **Alternative Definition:** For relations where primary key contains multiple attributes, no nonkey attribute should be functionally dependent on a part of the primary key.

Normalizing into 2NF



Third Normal Form (3NF)

- **Transitive functional dependency:** a FD X -> Z that can be derived from two FDs X -> Y and Y -> Z
- Examples:
  - If SSN -> DNUMBER and DNUMBER -> DMGRSSN, then
  - SSN -> DMGRSSN is a **transitive** FD

Third Normal Form (3NF)

- **Definition:** 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.

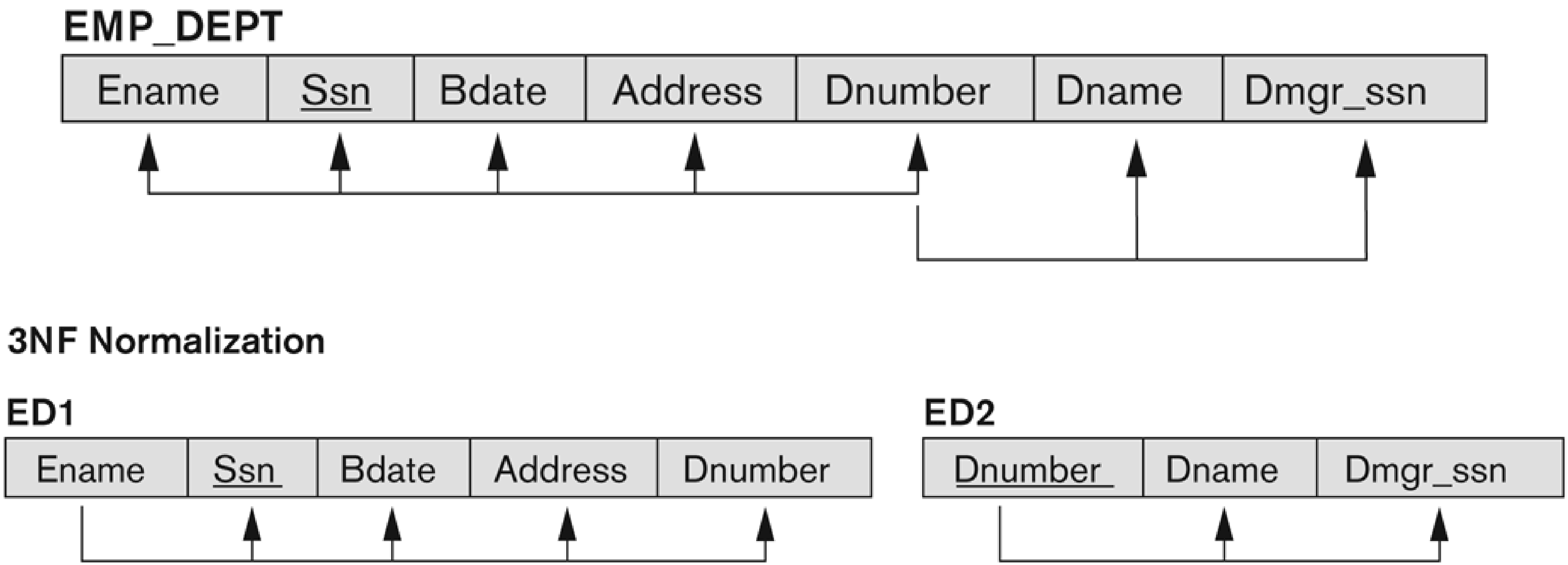
Third Normal Form (3NF)

- **Alternative Definition:** Each relation should not have a nonkey attribute that is functionally determined by another nonkey attribute (or by a set of nonkey attributes).

Note:

- Transitive dependency (X -> Y and Y -> Z) is a problem only if Y is not a candidate key.
- E.g., Consider EMP (<u>SSN</u>, Emp#, Salary)
- With FDs:
  - SSN -> Emp# and Emp# -> Salary

Normalizing into 3NF



Normal Forms Defined Informally

- 1st normal form
  - All attributes depend on **the key**
- 2nd normal form
  - All attributes depend on **the whole key**
- 3rd normal form
  - All attributes depend on **nothing but the key**

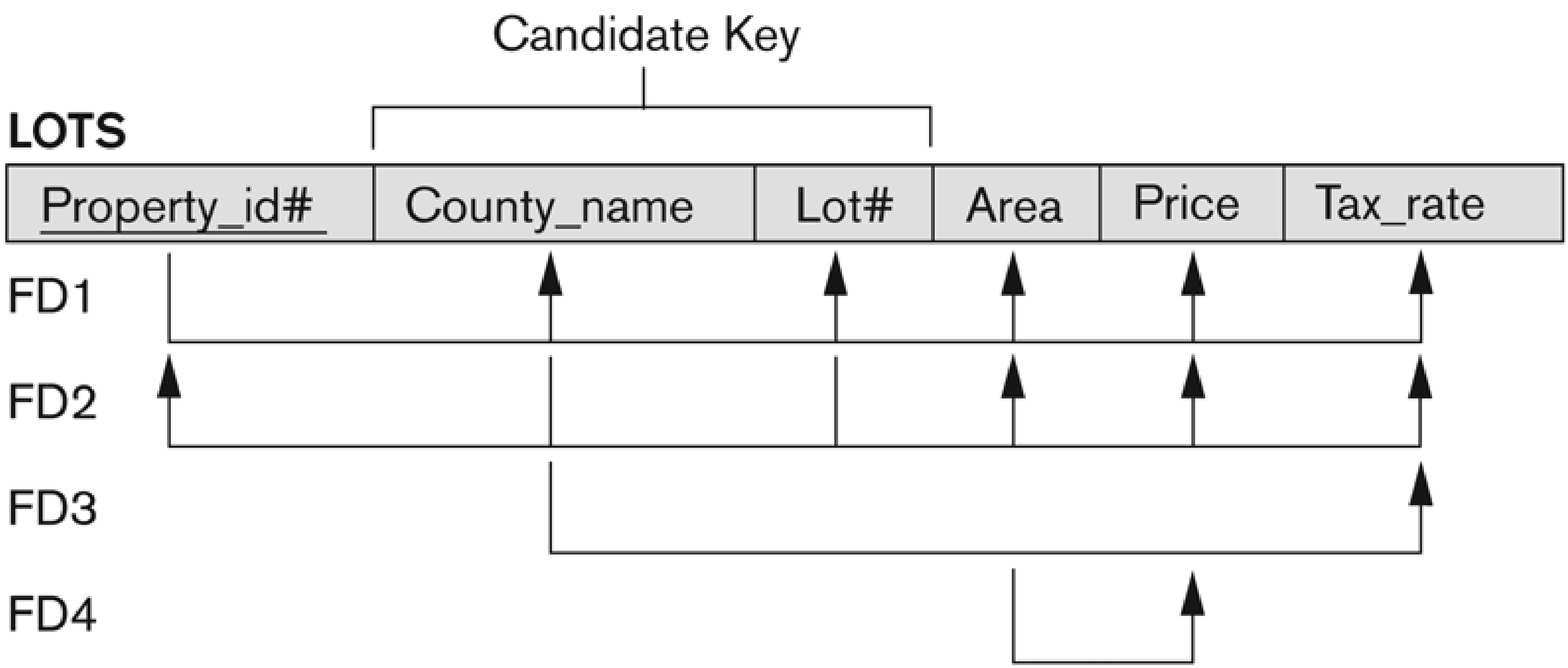
General Normal Form Definitions (For Multiple Keys)

- **In terms of primary key only:**
  - 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.

General Normal Form Definitions (For Multiple Keys)

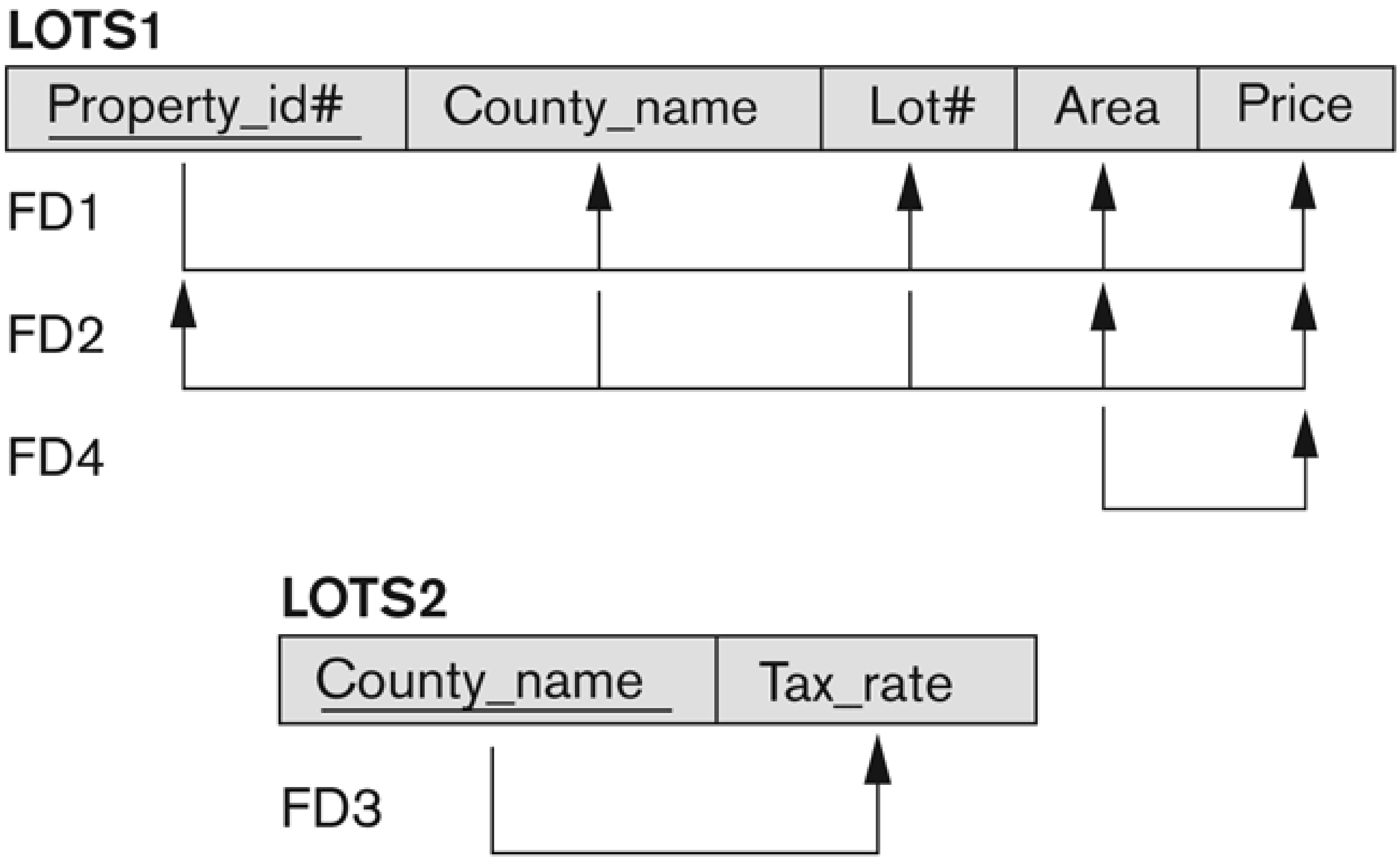
- **In general, if there are multiple keys:**
  - A relation schema R is in **second normal form (2NF)** if every non-prime attribute A in R is not partially dependent on any key of R.

Not in 2NF



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

Normalizing into 2NF



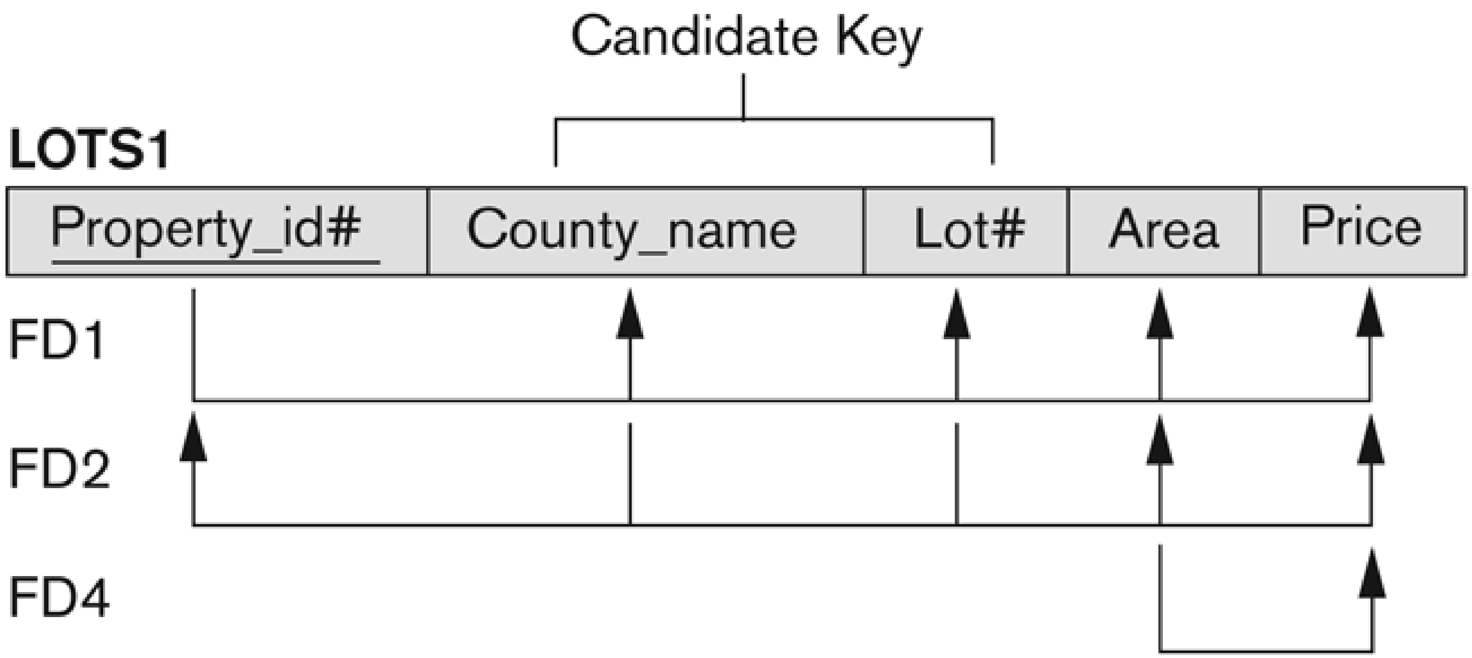
General Normal Form Definitions (For Multiple Keys)

- **In terms of primary key only:**
  - A relation schema R is in **3NF** if it is in 2NF *and* no non-prime attribute A in R is transitively dependent on the primary key.

General Normal Form Definitions (For Multiple Keys)

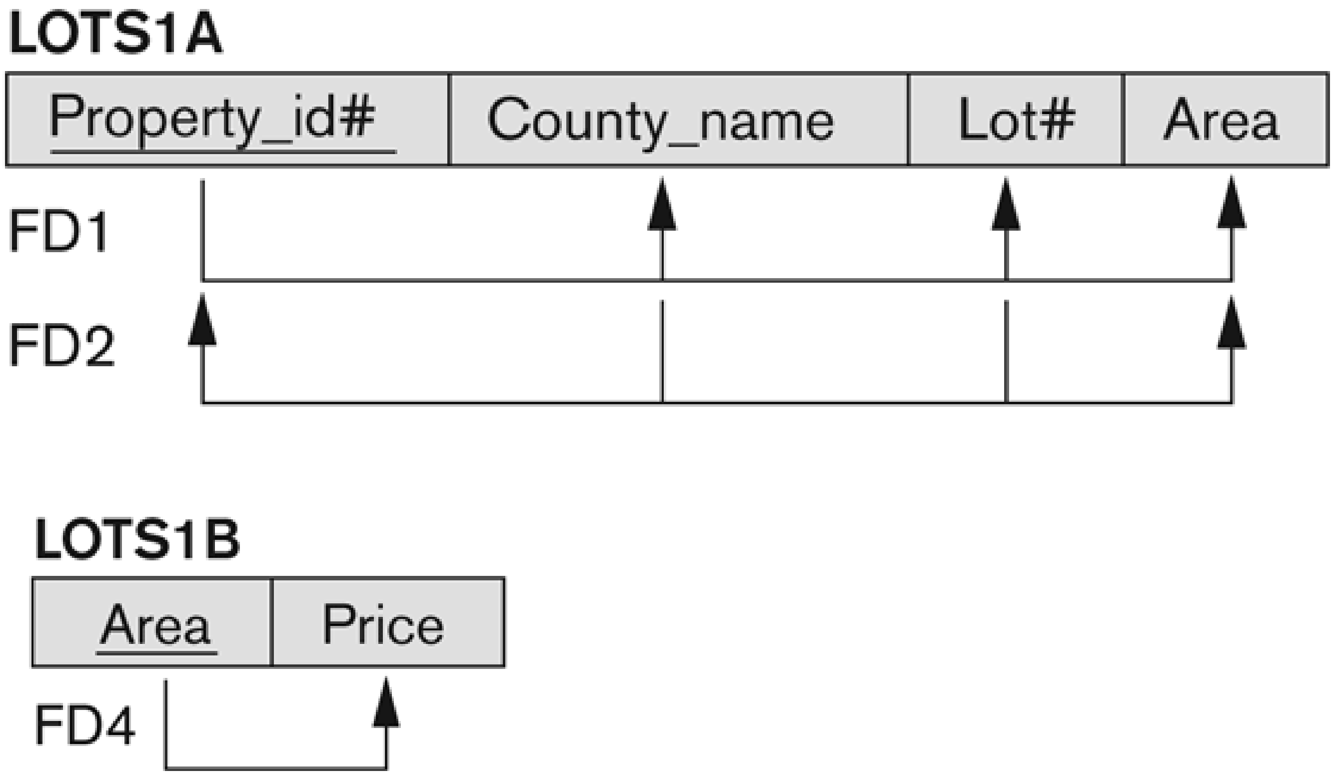
- **In general, if there are multiple keys:**
  - A relation schema R is in **third normal form (3NF)** if whenever a nontrivial FD X -> A holds in R, then either:
    - (a) X is a superkey of R, or
    - (b) A is a prime attribute of R

Not in 3NF



Candidate Key
LOTS1
Property_id#
County_name
Lot#
Area
Price
FD1
FD2
FD4

Normalizing into 3NF



Boyce-Codd Normal Form (BCNF)

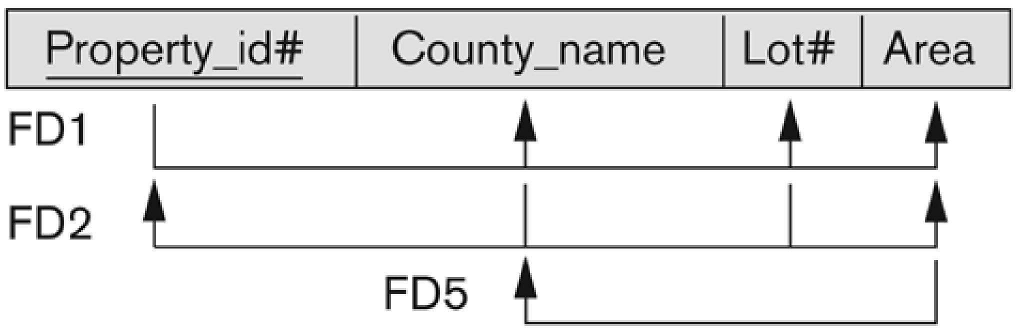
- A relation schema R is in **Boyce-Codd Normal Form (BCNF)** if whenever a nontrivial **FD X -> A** holds in R, then **X is a superkey** of R.
- Every BCNF relation is in 3NF but BCNF is stricter than 3NF.

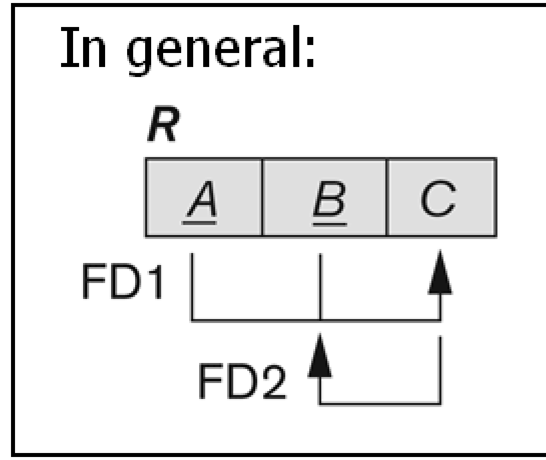
Boyce-Codd Normal Form (BCNF)

- There exist relations that are in 3NF but not in BCNF.
- The goal is to have each relation in BCNF (or 3NF).

Normalizing into BCNF

LOTS1A





BCNF Normalization

LOTS1AX

Property_id#	Area	Lot#

LOTS1AY

Area	County_name

A relation TEACH that is in 3NF but not in 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
Narayan	Operating Systems	Ammar

FDs in TEACH

- Two FDs exist in the relation TEACH:
  - FD1: { Student, Course} -> Instructor
  - FD2: Instructor -> Course
- {Student, Course} is a candidate key for this relation
- So this relation is in 3NF *but not in* BCNF

Normalizing into BCNF

- Given the relation TEACH(Student, Course, Instructor) and the following FDS:
  - FD1: { Student, Course} -> Instructor
  - FD2: Instructor -> Course
- TEACH can be decomposed into:
  - TEACH1(Instructor, Course)
  - TEACH2(Student, Instructor)
- However, FD1 was lost.

Multivalued Attributes

EMP

ENAME	PNAMES	DNAMES
Smith	{ X, Y }	{John, Anna}

EMP

ENAME	PNAME	DNAME
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

Multivalued Dependencies

- **Definition:** A multivalued dependency (MVD) X ->> Y specified on relation schema R, where X and Y are both subsets of R, specifies the following constraint on any relation state r of R: If two tuples t_1 and t_2 exist in r such that $t_1[X] = t_2[X]$, then two tuples t_3 and t_4 should also exist in r with the following properties:

Multivalued Dependencies

- $t_3[X] = t_4[X] = t_1[X] = t_2[X]$
- $t_3[Y] = t_1[Y]$ and $t_4[Y] = t_2[Y]$
- $t_3[R-(X \cup Y)] = t_2[R-(X \cup Y)]$ and $t_4[R-(X \cup Y)] = t_1[R-(X \cup Y)]$

Multivalued Dependencies

- **Alternative Definition:**
  - A multivalued dependency MVD is a dependency between attributes (for example X, Y, and Z) in a relation, such that for each value of X there is a set of values for Y and a set of values for Z. However, the set of values for Y and Z are independent of each other.

Example

EMP

ENAME	PNAME	DNAME
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

- The EMP relation has MVD:
  - Ename ->> Pname

Trivial and Nontrivial MVD

- An MVD X->>Y in R is called a **trivial MVD** if
  - Y is a subset of X or
  - $X \cup Y = R$.
- An MVD that did not satisfy any of the two conditions is called a **nontrivial MVD**.

Inference Rules for FDs and MVDs

- **IR7(complementation rule for MVDs):** If X->>Y, then {X ->> (R – (X ∪ Y))}.
- **IR8(augmentation rule for MVDs):** If X->>Y and Z ⊆ W, then WX ->> YZ.
- **IR9(transitive rule for MVDs):** If X->>Y and Y->> Z, then X ->>(Z-Y).

Inference Rules for FDs and MVDs

- **IR10(replication rule for FD to MVD):**

  If X->Y, then X->>Y.

- **IR11(coalescence rule for FDs and MVDs):**

  If X->>Y and there exists W with the properties that $W \cap Y$ is empty, W->Z and $Z \subseteq Y$, then X -> Z.

Closure of F

- **Definition:** The closure of F, denoted by F^+, is the set of all dependencies that include F as well as all dependencies (functional or multivalued) that can be inferred from F.

Fourth Normal Form (4NF)

- **Definition:** A relation schema R is in 4NF with respect to a set of functional dependencies F (functional and multivalued) if, for every nontrivial multivalued dependency X ->>Y in F^+, X is a superkey for R.

Fourth Normal Form (4NF)

- **Alternative Definition:** A relation cannot contain two or more independent multivalued facts about an entity.

Example

EMP

ENAME	PNAME	DNAME
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

- The EMP relation has two MVDs:
  - Ename ->> Pname
  - Ename ->> Dname

EMP

ENAME	PNAME	DNAME
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

EMP_PROJECTS

ENAME	PNAME
Smith	X
Smith	Y

EMP_DEPENDENTS

ENAME	DNAME
Smith	John
Smith	Anna

Another Example

EMPLOYEE	SKILL	LANGUAGE
Blake	Cooking	French
Blake	Typing	French
Blake	Programming	French
Blake	Cooking	English
Blake	Typing	English
Blake	Programming	English
Ford	Cooking	Spanish
Ford	Typing	Spanish
Ford	Cooking	English
Ford	Typing	English

Normalizing into 4NF

- There are two MVDs in the sample relation:
  - Employee ->> Skill
  - Employee ->> Language

- Thus, it should be decompose into two relations:
  - R1(Employee, Skill) and
  - R2(Employee, Language).

Lossless Join Decomposition

- A decomposition property which ensures that no spurious tuples are generated when a NATURAL JOIN operation is applied to the relations in the decomposition.

Example:

BORROWED

Book	StudentNo	Name	Course
Communication	123456	Smith	BSCS
Communication	745896	Ford	BSAM
IT	123456	Smith	BSCS
Mathematics	745896	Ford	BSAM
Social Science	425638	Blake	BSAM

Possible Decomposition

- If we decompose BORROWED into R1(Book, Course) and R2(StudentNo, Name, Course),
- It will become:

Book	Course
Communication	BSCS
Communication	BSAM
IT	BSCS
Mathematics	BSAM
Social Science	BSAM

Student No	Name	Course
123456	Smith	BSCS
745896	Ford	BSAM
425638	Blake	BSAM

R1 * R2

Book	Course	StudentNo	Name
Communication	BSCS	123456	Smith
Communication	BSAM	745896	Ford
Communication	BSAM	425638	Blake
IT	BSCS	123456	Smith
Mathematics	BSAM	745896	Ford
Mathematics	BSAM	425638	Blake
Social Science	BSAM	745896	Ford
Social Science	BSAM	425638	Blake

- But if we decompose BORROWED into R1(Book, StudentNo) and R2(StudentNo, Name, Course),
- We will have the following:

Book	StudentNo
Communication	123456
Communication	745896
IT	123456
Mathematics	745896
Social Science	425638

StudentNo	Name	Course
123456	Smith	BSCS
745896	Ford	BSAM
425638	Blake	BSAM

Book	StudentNo	Name	Course
Communication	123456	Smith	BSCS
Communication	745896	Ford	BSAM
IT	123456	Smith	BSCS
Mathematics	745896	Ford	BSAM
Social Science	425638	Blake	BSAM

- A decomposition of BORROWED into R1(Book, StudentNo) and R2(StudentNo, Name, Course) is a lossless join decomposition.

Join Dependency

- **Definition:** A join dependency JD(R_1, R_2, …, R_n) is a constraint which states that every legal state r(R) should have a lossless join decomposition into R_1, R_2, … R_n; that is, for every such r we have

$$*(\pi_{R1}(r), \pi_{R2}(r), \ldots, \pi_{Rn}(r)) = r$$

- Note: An MVD is a special case of a JD where n = 2.

Trivial Join Dependency

- A join dependency $JD(R_1, R_2, \ldots, R_n)$, specified on relation schema R, is **trivial** if one of the relation schemas R_i in $JD(R_1, R_2, \ldots, R_n)$ is equal to R.
- If this is not the case, the JD is **nontrivial**.

Fifth Normal Form (5NF)

- **Definition:** A relation schema r is in fifth normal form (5NF) (or project-join normal form (PJNF)) with respect to a set F of functional, multivalued, and join dependencies if, for every nontrivial join dependency JD(R_1, R_2, …, R_n) in F^+, every R_i is a superkey of R.

Fifth Normal Form (5NF)

- **Alternative Definition:** A relation is in 5NF if it cannot have a lossless join decomposition into any number of smaller relations.

Illustration

SUPPLY

SNAME	PARTNAME	PROJNAME
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

- Each tuple in the SUPPLY relation represents a supplier supplying a specific part to a particular project.

Join Dependency on SUPPLY

- Suppose that the following always holds: Whenever a supplier s supplies part p, and a project j uses part p, and the supplier s supplies a part to project j, then supplier s will also be supplying part p to project j.
  - (s, p) and (j, p) and (s, j) then (s,p,j)

Join Dependency on SUPPLY

- This constraint specifies a join dependency JD(R_1, R_2, R_3) on the three projections R_1(Sname, Part_name), and R_2(Sname, Proj_name), and R_3(Part_name, Proj_name) of SUPPLY.

Normalizing into 5NF

SUPPLY

SNAME	PARTNAME	PROJNAME
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

R1

SNAME	PARTNAME
Smith	Bolt
Smith	Nut
Adamsky	Bolt
Walton	Nut
Adamsky	Nail

R2

SNAME	PROJNAME
Smith	ProjX
Smith	ProjY
Adamsky	ProjY
Walton	ProjZ
Adamsky	ProjX

R3

PARTNAME	PROJNAME
Bolt	ProjX
Nut	ProjY
Bolt	ProjY
Nut	ProjZ
Nail	ProjX

Another Example

TEACHES

INSTRUCTOR	COURSES	SEMESTER
CDDS	CMSC 11	1st
CDDS	CMSC 11	2nd
CDDS	CMSC 127	1st
RGDC	CMSC 11	2nd

- Each tuple in the TEACHES relation represents an instructor teaching a specific course during a particular semester.

Join Dependency on TEACHES

- Suppose that the following always holds: If an instructor i teaches a course c and a course c is offered during a semester s and instructor i teaches during a semester s, then instructor i teaches course c during the semester s.
  - (i, c) and (c, s) and (i, s) then (i,c,s)

Join Dependency on TEACHES

- This constraint specifies a join dependency JD(R_1, R_2, R_3) on the three projections R_1(Instructor, Course), and R_2(Course, Semester), and R_3(Instructor, Semester) of TEACHES.

Normalizing into 5NF

- Since TEACHES is not in 5NF, then it should be decomposed into:
  - R1(Instructor, Course)
  - R2(Course, Semester)
  - R3(Instructor, Semester)

Reference(s):

- **Elmasri, R. and S.B. Navathe.** 2010. Fundamentals of Database Systems. 6th Edition. Addition Wesley. ISBN-13: 978-0-136-08620-8
- **Elmasri, R. and S.B. Navathe.** 2007. Fundamentals of Database Systems. 5th Edition. Addition Wesley. ISBN: 981-06-9800-3