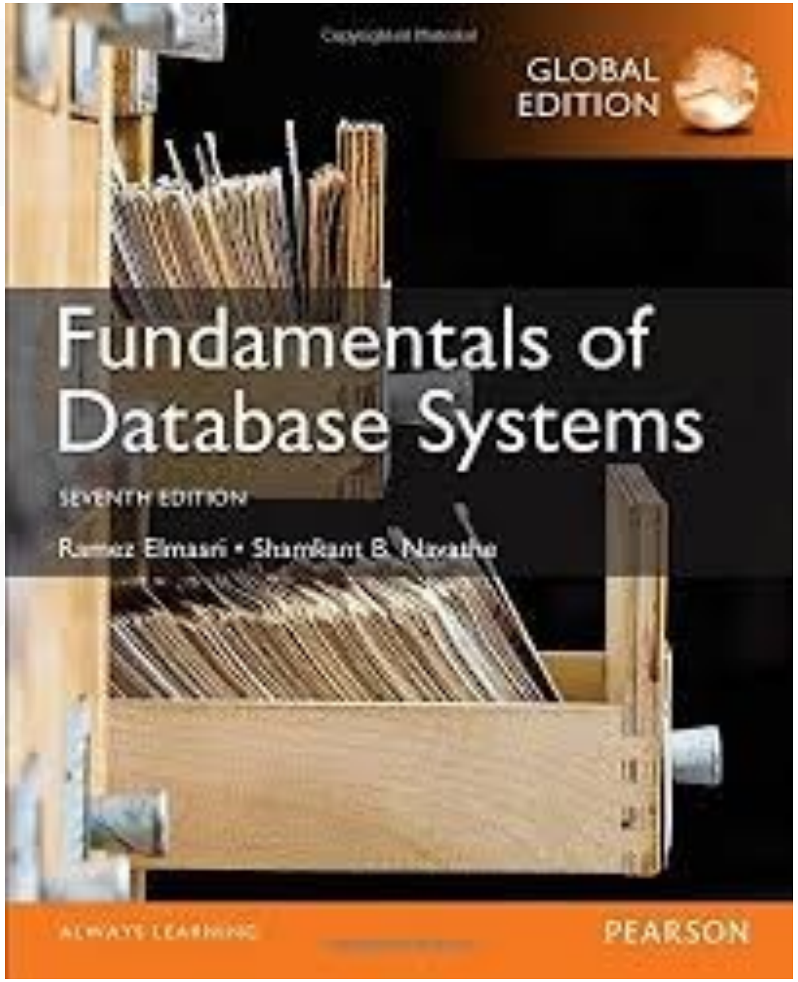
Database Systems

Program in Computer Engineering
Faculty of Engineering

King Mongkut's Institute of Technology Ladkrabang

Text

- Ramez Elmasri and Shamkant B. Navathe. "Fundamentals of Database Systems" 7th Edition., Pearson, 2017



Chapter 14

Basics of Functional Dependencies and Normalization for Relational Database

Outline

5. BCNF (Boyce-Codd Normal Form)
6. Multivalued Dependency and Fourth Normal Form
7. Join Dependencies and Fifth Normal Form

5. BCNF (Boyce-Codd Normal Form)

- A relation schema R is in **Boyce-Codd Normal Form (BCNF)** if whenever a ***nontrivial functional dependency*** **X** $\longrightarrow$ **A** holds in R, then **X is a superkey of R**

The trivial dependency is a set of attributes which are called a trivial if the set of attributes are included in that attribute.

So, X -> Y is a trivial functional dependency if Y is a subset of X.

Functional dependency which also known as a nontrivial dependency occurs when A->B holds true where B is not a subset of A.

In a relationship, if attribute B is not a subset of attribute A, then it is considered as a non-trivial dependency.

- 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
- Hence BCNF is considered a stronger form of 3NF
- The goal is to have each relation in BCNF (or 3NF)

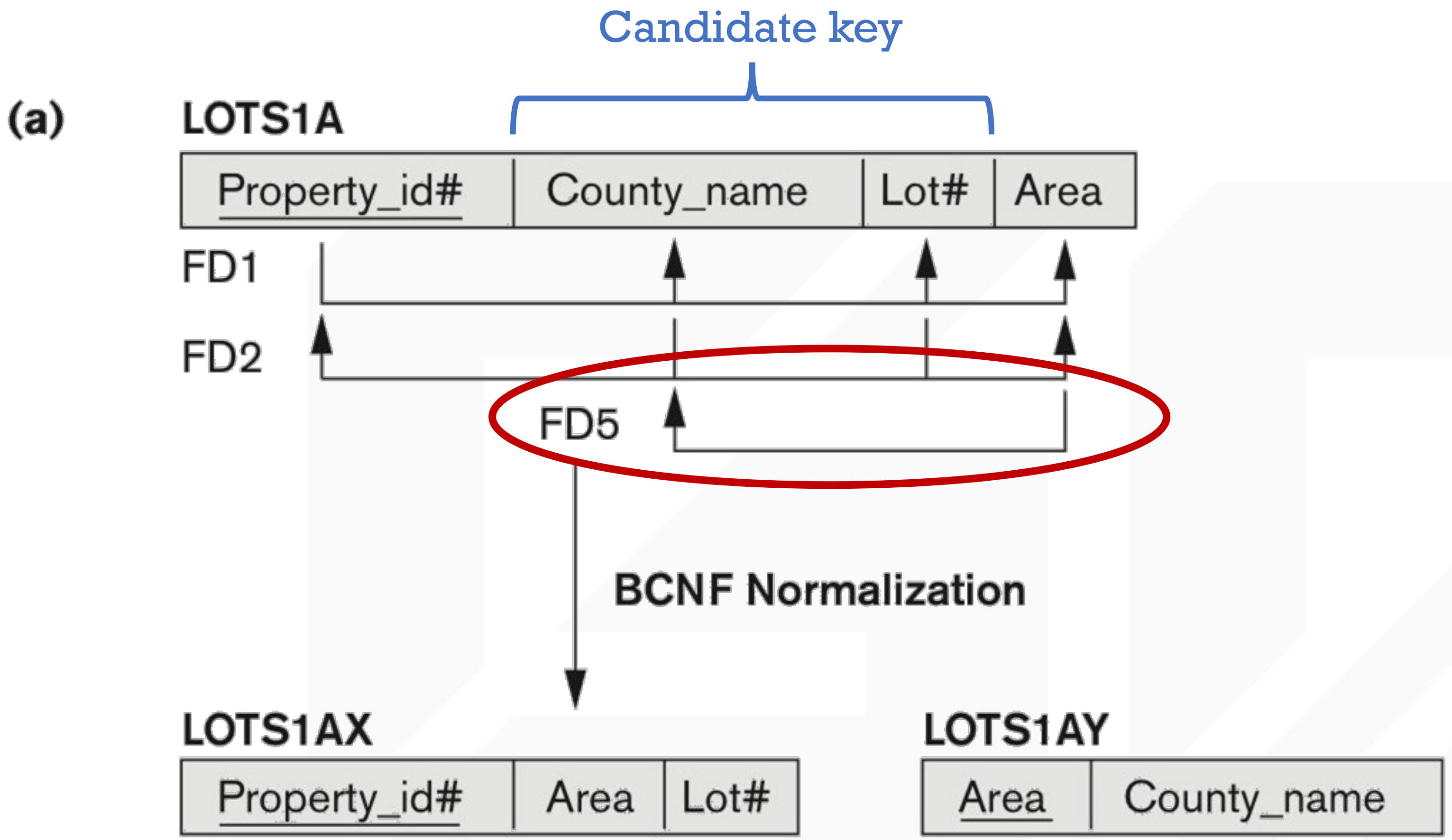
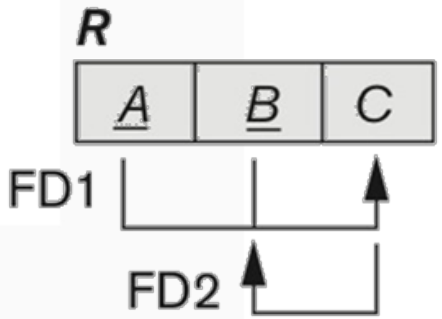
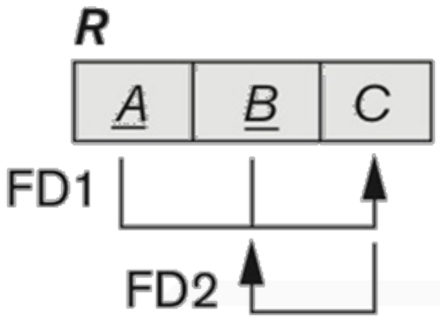


Figure 14.13
Boyce-Codd normal form. (a) BCNF normalization of LOTS1A with the functional dependency FD2 being lost in the decomposition. (b) A schematic relation with FDs; it is in 3NF, but not in BCNF due to the f.d. C → B.



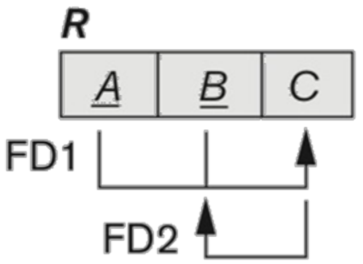


TEACH

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

Figure 14.14
A relation TEACH that is in 3NF but not BCNF.

- Two FDs exist in the relation TEACH:
  - fd1: { student, course} -> instructor
  - fd2: instructor -> course
- {student, course} is a candidate key for this relation and that the dependencies shown follow the pattern in Figure 14.13 (b).
  - So this relation is in 3NF *but not in* BCNF
- A relation **NOT** in BCNF should be decomposed so as to meet this property, while possibly forgoing the preservation of all functional dependencies in the decomposed relations.



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

Figure 14.14 A relation TEACH that is in 3NF but not BCNF.

- Three possible decompositions for relation TEACH
  - D1: {student, instructor} and {student, course}
  - D2: {course, instructor } and {course, student}
  - D3: {instructor, course } and {instructor, student} ✓
- All three decompositions will lose fd1.
  - We have to settle for sacrificing the functional dependency preservation.
    But we cannot sacrifice the non-additivity property after decomposition.
- Out of the above three, only the 3rd decomposition will not generate spurious tuples after join. (and hence has the non-additivity property).

- A test to determine whether a binary decomposition (decomposition into two relations) is non-additive (lossless) is discussed under Property NJB on the next slide.

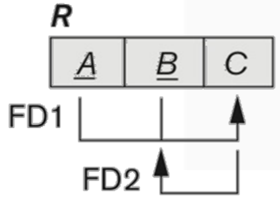
Test for checking non-additivity of Binary Relational Decompositions

- **Testing Binary Decompositions for Lossless Join (Non-additive Join) Property**
  - **Binary Decomposition:**
    Decomposition of a relation *R* into two relations.
  - **Property NJB (non-additive join test for binary decompositions):**
    A decomposition D = $\{R_1, R_2\}$ of *R* has the lossless join property with respect to a set of functional dependencies *F* on *R* *if and only if* either
    - The f.d. $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$ is in F^+, or
    - The f.d. $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$ is in F^+.

Note: F^+ is the (complete) set of all dependencies (functional or multivalued) that will hold in every relation state *r* of *R* that satisfies *F*. It is also called the **closure** of *F*.

- **PROPERTY NJB (non-additive join test for binary decompositions):** A decomposition $D = \{R_1, R_2\}$ of R has the lossless join property with respect to a set of functional dependencies F on R *if and only if* either
  - The f.d. $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$ is in F^+, or
  - The f.d. $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$ is in F^+.

- Three possible decompositions for relation TEACH
  - D1: {student, instructor} and {student, course}
  - D2: {course, instructor } and {course, student}
  - D3: {instructor, course } and {instructor, student} ✓



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

Figure 14.14 A relation TEACH that is in 3NF but not BCNF.

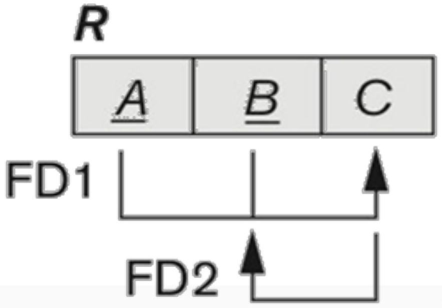
- If you apply the NJB test to the 3 decompositions of the TEACH relation:
  - D1 gives Student → Instructor or Student → Course, none of which is true.
  - D2 gives Course → Instructor or Course → Student, none of which is true.
  - However, in D3 we get Instructor → Course or Instructor → Student.
- Since Instructor → Course is indeed true, the NJB property is satisfied and D3 is determined as a non-additive (good) decomposition.

General Procedure for achieving BCNF when a relation fails BCNF

- Here we make use the algorithm from Chapter 15 (Algorithm 15.5):
  - Let R be the relation not in BCNF,
    let X be a subset of R, and
    let $X \rightarrow A$ be the FD that causes a violation of BCNF.
    Then R may be decomposed into two relations:
    - (i) $R - A$ and (ii) $X \cup A$.
    - If either $R - A$ or $X \cup A$ is not in BCNF,
      repeat the process.

TEACH

Student	Course	Instructor
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman



- Note that the f.d. that violated BCNF in TEACH was Instructor → Course. Hence its BCNF decomposition would be :
  - (TEACH – COURSE) and (Instructor ∪ Course), which gives
  - the relations: (Instructor, Student) and (Instructor, Course) that we obtained before in decomposition D3.

6. Multivalued Dependencies and Fourth Normal Form

(a) EMP

Ename	Pname	Dname
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

- A tuple in this EMP relation represents the fact
  - An employee whose name is Ename works on the project whose name is Pname and has a dependent whose name is Dname.
  - An employee may work on several projects and may have several dependents, and the employee's projects and dependents are independent of one another.
- In the relation state shown in Figure 14.15(a), the employee with Ename Smith works on two projects 'X' and 'Y' and has two dependents 'John' and 'Anna', and therefore there are four tuples to represent these facts together.

(a) EMP

Ename	Pname	Dname
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

- The relation EMP is an all-key relation (with key made up of all attributes) and therefore has no f.d.'s and as such qualifies to be a BCNF relation.
- There is an obvious redundancy in the relation EMP—the dependent information is repeated for every project and the project information is repeated for every dependent.
- The concept of multivalued dependency (MVD) was proposed and, based on this dependency, the fourth normal form was defined.

Definition

- A **multivalued dependency (MVD)** $X \twoheadrightarrow Y$ 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, where we use Z to denote $(R - (X \cup Y))$:
    - $t_3[X] = t_4[X] = t_1[X] = t_2[X]$.
    - $t_3[Y] = t_1[Y]$ and $t_4[Y] = t_2[Y]$.
    - $t_3[Z] = t_2[Z]$ and $t_4[Z] = t_1[Z]$.

- Whenever $X \twoheadrightarrow Y$ holds, we say that X **multidetermines** Y.
- Because of the symmetry in the definition, whenever $X \twoheadrightarrow Y$ holds in R, so does $X \twoheadrightarrow Z$. Hence, $X \twoheadrightarrow Y$ implies $X \twoheadrightarrow Z$ and therefore it is sometimes written as $X \twoheadrightarrow Y | Z$.
- An MVD $X \twoheadrightarrow Y$ in R is called a **trivial MVD** if
    a) Y is a subset of X, or
    b) $X \cup Y = R$
- For example, the relation EMP_PROJECTS has the trivial MVD $Ename \twoheadrightarrow Pname$ and the relation EMP_DEPENDENTS has the trivial MVD $Ename \twoheadrightarrow Dname$.
- An MVD that satisfies neither (a) nor (b) is called a **nontrivial MVD**.

- Note that an MVD is represented by the symbol $\twoheadrightarrow$ or $\rightarrow\rightarrow$.

Definition

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

- **Note:**
  F^+ is the (complete) set of all dependencies (functional or multivalued) that will hold in every relation state r of R that satisfies F. It is also called the **closure** of F.

- We can state the following points:
    - An all-key relation is always in BCNF since it has no FDs.
    - An all-key relation such as the EMP relation in Figure 14.15(a), which has no FDs but has the MVD Ename $\twoheadrightarrow$ Pname | Dname, is not in 4NF.
    - A relation that is not in 4NF due to a nontrivial MVD must be decomposed to convert it into a set of relations in 4NF.
    - The decomposition removes the redundancy caused by the MVD.

(a) EMP

Ename	Pname	Dname
Smith	X	John
Smith	Y	Anna
Smith	X	Anna
Smith	Y	John

- The non-trivial MVD Ename ↠ Pname | Dname is decomposed to two trivial MVD Ename ↠ Pname and Ename ↠ Dname.

(b) EMP_PROJECTS

Ename	Pname
Smith	X
Smith	Y

EMP_DEPENDENTS

Ename	Dname
Smith	John
Smith	Anna

Figure 14.15
Fourth and fifth normal forms.
(a) The EMP relation with two MVDs:
Ename –>> Pname and Ename –>> Dname.
(b) Decomposing the EMP relation into two 4NF relations EMP_PROJECTS and EMP_DEPENDENTS.

7. Join Dependencies and Fifth Normal Form

- In some cases there may be no nonadditive join decomposition of *R* into two relation schemas, but there may be a nonadditive join decomposition into more than two relation schemas.
- Moreover, there may be no functional dependency in *R* that violates any normal form up to BCNF,
  and there may be no nontrivial MVD present in *R* either that violates 4NF.

Definition

- A **join dependency (JD),** denoted by JD(R_1, R_2, … , R_n), specified on relation schema R, specifies a constraint on the states r of R.

  The constraint states that every legal state r of R should have a nonadditive join decomposition into R_1 , R_2 , … , R_n .

- Hence, for every such r we have

$$*\left(\pi_{R_1}(r), \pi_{R_2}(r), \ldots, \pi_{R_n}(r)\right) = r$$

Notice

- An MVD is a special case of a JD where n = 2.
  That is, a JD denoted as JD(R_1 , R_2) implies
  an MVD $(R_1 \cap R_2) \twoheadrightarrow (R_1 - R_2)$
  (or, by symmetry, $(R_1 \cap R_2) \twoheadrightarrow (R_2 - R_1)$).
- A join dependency JD(R_1 , R_2 , … , R_n), specified on relation schema R, is a trivial JD if one of the relation schemas Ri in JD(R_1 , R_2 , … , R_n) is equal to R.

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^+ (that is, implied by F), every R_i is a superkey of R.

- Discovering join dependencies in practical databases with hundreds of relations is next to impossible. Therefore, 5NF is rarely used in practice.

(c) **SUPPLY**

Sname	Part_name	Proj_name
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

(d) R_1

Sname	Part_name
Smith	Bolt
Smith	Nut
Adamsky	Bolt
Walton	Nut
Adamsky	Nail

R_2

Sname	Proj_name
Smith	ProjX
Smith	ProjY
Adamsky	ProjY
Walton	ProjZ
Adamsky	ProjX

R_3

Part_name	Proj_name
Bolt	ProjX
Nut	ProjY
Bolt	ProjY
Nut	ProjZ
Nail	ProjX

Figure 14.15
(c) The relation SUPPLY with no MVDs is in 4NF but not in 5NF if it has the JD(R1, R2, R3).
(d) Decomposing the relation SUPPLY into the 5NF relations R1, R2, R3.

- Suppose that the following additional constraint always holds:
  - Whenever a supplier *s* supplies part *p*, and a project *j* uses part *p*, and the supplier *s* supplies at least one part to project *j*, then supplier *s* will also be supplying part *p* to project *j*.
- This constraint can be restated in other ways and specifies a join dependency JD(R_1 , R_2 , R_3) among the three projections R_1 (Sname, Part_name), R_2 (Sname, Proj_name), and R_3 (Part_name, Proj_name) of SUPPLY.

(c) SUPPLY

Sname	Part_name	Proj_name
Smith	Bolt	ProjX
Smith	Nut	ProjY
Adamsky	Bolt	ProjY
Walton	Nut	ProjZ
Adamsky	Nail	ProjX
Adamsky	Bolt	ProjX
Smith	Bolt	ProjY

(d) R_1

Sname	Part_name
Smith	Bolt
Smith	Nut
Adamsky	Bolt
Walton	Nut
Adamsky	Nail

R_2

Sname	Proj_name
Smith	ProjX
Smith	ProjY
Adamsky	ProjY
Walton	ProjZ
Adamsky	ProjX

R_3

Part_name	Proj_name
Bolt	ProjX
Nut	ProjY
Bolt	ProjY
Nut	ProjZ
Nail	ProjX

Figure 14.15
(c) The relation SUPPLY with no MVDs is in 4NF but not in 5NF if it has the JD(R1, R2, R3).
(d) Decomposing the relation SUPPLY into the 5NF relations R1, R2, R3.

- Notice that applying a natural join to any two of these relations produces spurious tuples but applying a natural join to all three together does not.

Summary

- BCNF (Boyce-Codd Normal Form)
- Fourth and Fifth Normal Forms