CSC10006 – Introduction to Database

Chapter 8
Normalization

Instructor: **Vũ Thị Mỹ Hằng, Ph.D.** (vtmhang@fit.hcmus.edu.vn)

Lab Assistants: **Tiết Gia Hồng, M.Sc.** (tghong@fit.hcmus.edu.vn)

Lương Hán Cơ, M.Sc. (lhco@fit.hcmus.edu.vn)

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP. HO CHI MINH

cdio 4.0

fit@hcmus

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

Outline

1. Design a relational database
2. Redundant Information in Tuples and Update Anomalies
3. Functional dependencies
4. Armstrong's inference rules
5. Normalization

1. Design a relational database

CONCEPTUAL DESIGN (THIẾT KẾ QUAN NIỆM)

↓

LOGIC DESIGN (THIẾT KẾ LÔ-GÍC)

↓

PHYSICAL DESIGN (THIẾT KẾ VẬT LÝ)

- To respond to information requests for specific users and applications.
- To provide the right storage for data, and data is easy to understand.
- To supports performance requirements: response time, processing time, storage space, ...

Informal Design Guidelines

- What is relational database design?
  - The grouping of attributes to form "good" relation schemas
- Two levels of relation schemas
  - The logical "user view" level
  - The storage "base relation" level
- Design is concerned mainly with base relations
- What are the criteria for "**good**" base relations?

Criteria for "good" base relations

- A well-designed database schema is shown by:
  - Semantics of relations and attributes are clearly defined.
  - Attributes must not overlap.
  - Data duplication (redundancy) between tuples must be reduced
  - Null value on relations is minimum.
- Duplicated data leads to:
  - Waste storage space.
  - Anomalies when performing data updates.
- How to verify a schema being "**good**" base relations:
  - By experience
  - By **normal form**

Design steps to achieve "**good**" base relations

1. Conceptual design schema
2. Transfer the schema to a logical level
3. Identify functional dependencies and normal form
4. Normalize the schema
5. Evaluate the quality of the schema

Outline

1. Design a relational database
2. **Redundant Information in Tuples and Update Anomalies**
3. Functional dependencies
4. Armstrong's inference rules
5. Normalization

2.1. Redundant Information in Tuples

- Data redundancy/duplication:
  - Waste storage space.
  - Anomalies when performing data updates.

MAGV	TENGV	NGSINH	DCHI	MABM	TENBM	TRGBM
009	Tiên	11/02/1960	119 Cống Quỳnh, Tp HCM	5	HTTT	005
005	Tùng	20/08/1962	222 Nguyễn Văn Cừ, Tp HCM	5	HTTT	[illegible]
007	Hằng	11/3/1954	332 Nguyễn Thái Học, Tp HCM	4	[illegible]	008
001	Như	01/02/1967	291 Hồ Văn Huê, Tp HCM	4	[illegible]	008
004	Hùng	04/03/1967	95 Bà Rịa, Vũng Tàu	5	HTTT	005
003	Tâm	04/05/1957	34 Mai Thị Lự, Tp HCM	5	HTTT	005
008	Quang	01/09/1967	80 Lê Hồng Phong, Tp HCM	4	MMT	008
006	Vinh	01/01/1965	45 Trưng Vương, Hà Nội	1	CNPM	006

Redundancy

Redundancy: Information about **department name** and **department head** is repeated several times.

2.1. Redundant Information in Tuples

MAGV	TENGV	NGSINH	DCHI	MABM	TENBM	TRGBM
009	Tiên	11/02/1960	119 Cống Quỳnh, Tp HCM	5	HTTT	005
005	Tùng	20/08/1962	222 Nguyễn Văn Cừ, Tp HCM	5	HTTT	005
007	Hằng	11/3/1954	332 Nguyễn Thái Học, Tp HCM	4	MMT	008
001	Như	01/02/1967	291 Hồ Văn Huê, Tp HCM	4	MMT	008
004	Hùng	04/03/1967	95 Bà Rịa, Vũng Tàu	5	HTTT	005
003	Tâm	04/05/1957	34 Mai Thị Lự, Tp HCM	5	HTTT	005
008	Quang	01/09/1967	80 Lê Hồng Phong, Tp HCM	4	MMT	008
006	Vinh	01/01/1965	45 Trưng Vương, Hà Nội	1	CNPM	006

Waste storage space.

Schema 1

Calculate the storage size of each schema when adding 10 teachers for the Dept. HTTT? (Assuming the size of each attribute is 20 bytes)

MAGV	TENGV	NGSINH	DCHI	MABM
009	Tiên	11/02/1960	119 Cống Quỳnh, Tp HCM	5
005	Tùng	20/08/1962	222 Nguyễn Văn Cừ, Tp HCM	5
007	Hằng	11/3/1954	332 Nguyễn Thái Học, Tp HCM	4
001	Như	01/02/1967	291 Hồ Văn Huê, Tp HCM	4
004	Hùng	04/03/1967	95 Bà Rịa, Vũng Tàu	5
003	Tâm	04/05/1957	34 Mai Thị Lự, Tp HCM	5
008	Quang	01/09/1967	80 Lê Hồng Phong, Tp HCM	4
006	Vinh	01/01/1965	45 Trưng Vương, Hà Nội	1

MABM	TENBM	TRGBM
5	HTTT	005
5	HTTT	005
4	MMT	008
4	MMT	008
5	HTTT	005
5	HTTT	005
4	MMT	008
1	CNPM	006

Schema 2

2.2. Update Anomalies

- Insert anomaly (Must add duplicate data)
  - Adding 1 teacher must add dept. information or must set to Null the value for dept.-related attributes. You can't add a new dept. without any teachers in that dept.
- Update anomaly (Can lead to inconsistent data)
  - When updating a dept. information, all tuples of that department must be updated or this will lead to an inconsistent data.
- Delete anomaly (Information may lost Information)
  - What happens when deleting the teacher with code 006?

MAGV	TENGV	NGSINH	DCHI	MABM	TENBM	TRGBM
009	Tiên	11/02/1960	119 Cống Quỳnh, Tp HCM	5	HTTT	005
005	Tùng	20/08/1962	222 Nguyễn Văn Cừ, Tp HCM	5	HTTT	005
007	Hằng	11/3/1954	332 Nguyễn Thái Học, Tp HCM	4	MMT	008
001	Như	01/02/1967	291 Hồ Văn Huê, Tp HCM	4	MMT	008
004	Hùng	04/03/1967	95 Bà Rịa, Vũng Tàu	5	HTTT	005
003	Tâm	04/05/1957	34 Mai Thị Lự, Tp HCM	5	HTTT	005
008	Quang	01/09/1967	80 Lê Hồng Phong, Tp HCM	4	MMT	008
006	Vinh	01/01/1965	45 Trưng Vương, Hà Nội	1	CNPM	006

2.4. Design Guidelines

Insertion Anomaly: Adding new rows forces users to create duplicate data.

Deletion Anomaly: Deleting rows may cause a loss of data that would be needed for other future rows.

Modification Anomaly: Changing data in a row forces changes to other rows due to duplication.

General Rule:

A relation should contain minimal data redundancy and pertain one type of entity/relationship types to ensure:

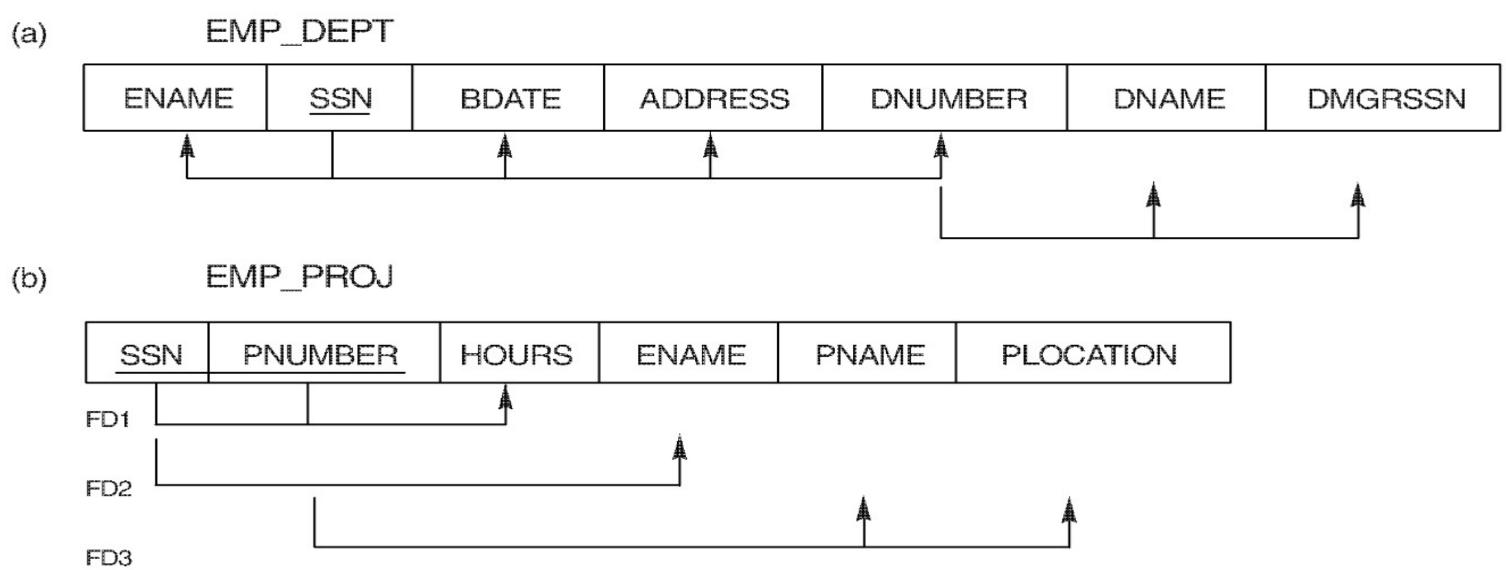
- Explainable attributes.
- Free of insertion, deletion, and update anomalies.
- Minimal numbers of null values.
- Meaningful results of jointure.

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

Quiz

- Identify update anomalies of the below database schema?

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.



Outline

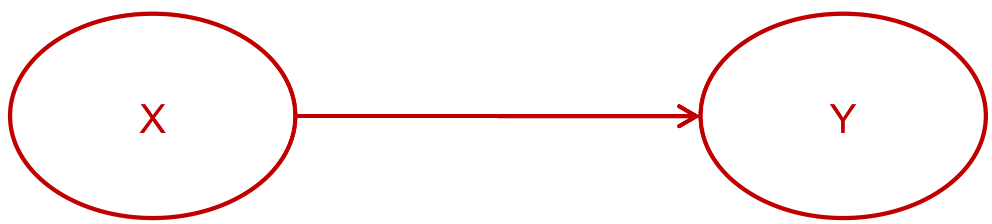
1. Design a relational database
2. Redundant Information in Tuples and Update Anomalies
3. **Functional dependencies**
4. Armstrong's inference rules
5. Normalization

3.1. Functional dependencies (phụ thuộc hàm)

- 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**.

3.1. Functional dependencies (phụ thuộc hàm)

- Specifying a **constraint** between two sets of attributes (X, Y) of a relational schema R.
- Denoted as: **X → Y**
  - **X**, **Y** are two sets of attributes of the **same schema R**
  - **Y** (functionally) **depends on X** (on R).
  - **X** (functionally) **determines Y** (on R).
  - **X holds Y** (on R).



Left-side (Determinant) (Định thuộc)

Right-side (Dependent) (Phụ thuộc)

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

fit@hcmus

3.1. Functional dependencies (phụ thuộc hàm)

Let **R(A1, A2, …, An)**, $\mathbf{R^+}$ **= {A1, …, An}**

X, Y $\subseteq$ $\mathbf{R^+}$, $X \rightarrow Y$ if and only if:

$\forall$**r** $\in$ **R, t**$_1$**, t**$_2$ $\in$ **r, if t**$_1$**[X] = t**$_2$**[X] then t**$_1$**[Y] = t**$_2$**[Y]**

Interpretation: X → Y holds on R (X determines Y on R) whenever two tuples have the same value for X, they must have the same value for Y.

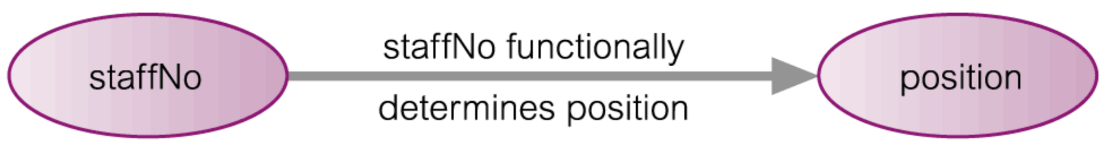
(Coronel and Morris, 2015) (Silberschatz et al. 2020)

fit@hcmus

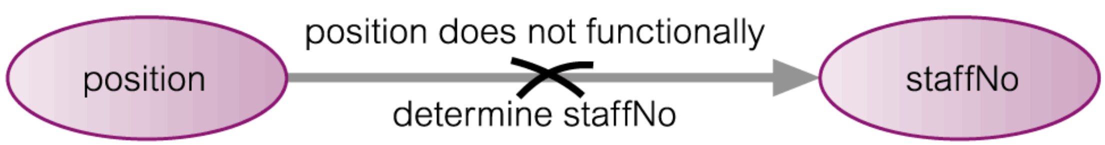
3.1. Functional dependencies (phụ thuộc hàm)

$\forall r \in R,\ t_1, t_2 \in r$, **if** $t_1[X] = t_2[X]$ **then** $t_1[Y] = t_2[Y]$

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London



SL21 → Manager



Manager → SL21
→ SG5

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

3.1. Functional dependencies (phụ thuộc hàm)

$$\forall r \in R,\ t_1, t_2 \in r,\ \text{if } t_1[X] = t_2[X] \text{ then } t_1[Y] = t_2[Y]$$

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

FD Identification based on the relation Staff:

staffNo → sName

sName → ~~staffNo~~

Business rules

The identification of FDs within a relational schema must be based on the semantics of the attributes and their interrelationships, rather than the data itself.

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

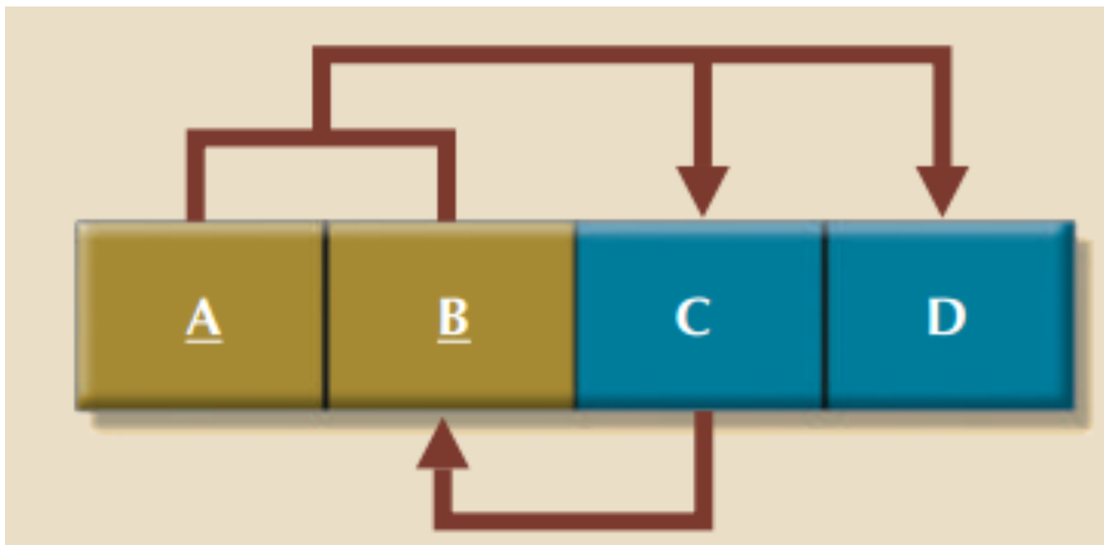
3.2. Visual Representation of FDs

Given R = <U, F> with: U = { ABCD }

F = { f1: AB →CD; f2: AC →BD; f3: C →B }

FD Diagrams

(Sơ đồ phụ thuộc hàm)



(Coronel and Morris, 2015) (Silberschatz et al. 2020)

Outline

1. Design a relational database
2. Redundant Information in Tuples and Update Anomalies
3. Functional dependencies
4. **Armstrong's inference rules**
5. Normalization

4. Armstrong's inference rules (Luật dẫn Armstrong)

- Given a set of FDs F, we can *infer* additional FDs that hold whenever the FDs in F hold
- **Armstrong's inference rules**
  - **IR1**. (*Reflexive*) If Y subset-of X, then X → Y
  - **IR2**. (*Augmentation*) If X → Y, then XZ → YZ

    (Notation: XZ stands for X ∪ Z)
  - **IR3**. (*Transitive*) If X → Y and Y → Z, then X → Z
- IR1, IR2, IR3 form a *sound* and *complete* set of inference rules.

4. Armstrong's inference rules

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

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

Outline

1. Design a relational database
2. Redundant Information in Tuples and Update Anomalies
3. Functional dependencies
4. Armstrong's inference rules
5. **Normalization**

5.1 Normalization

- **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

5.2. Complementary Concepts

Trivial vs. non-trivial functional dependencies

(Phụ thuộc hàm hiển nhiên)

- **Trivial functional dependency**

  Given $X \rightarrow Y$: **If** $\mathbf{Y} \subseteq X$ **Then** $X \rightarrow Y$ is a trivial FD (IR1).

(Phụ thuộc hàm không hiển nhiên)

- **Non-trivial functional dependency**

  Given $X \rightarrow Y$: **If** $\mathbf{Y} \not\subseteq X$ **Then** $X \rightarrow Y$ is a non-trivial FD.

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

fit@hcmus

5.2. Complementary Concepts

Full vs. partial functional dependencies

(Phụ thuộc hàm đầy đủ)

- **Full** (functional) **dependency**

  Given $X \rightarrow Y$:

  If $\neg\exists X' \subset X$ **such as** $X' \rightarrow Y$

  Then Y <u>depends fully functionnally</u> on X

(Phụ thuộc hàm riêng phần / KHÔNG đầy đủ)

- **Partial** (functional) **dependency**

  Given $X \rightarrow Y$:

  If $\exists X' \subset X$ **such as** $X' \rightarrow Y$

  Then Y <u>depends partially functionally</u> on X

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

fit@hcmus

5.2. Complementary Concepts

Example: SV (mãsv, họtên, mãlớp, tênlớp, mônhọc, điểm)
Which of the following FDs are full FDs? partial FDs?

- f1: Mãsv → Họtên, mãlớp
- f2: Mãlớp → Tênlớp
- f3: Mãsv, Mônhọc → Điểm
- f4: Mãsv, Tênsv, Mônhọc → Điểm
- f5: Mãsv, Mãlớp → Tênlớp

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

5.2. Complementary Concepts

Example: SV (mãsv, họtên, mãlớp, tênlớp, mônhọc, điểm)
Which of the following FDs are full FDs? partial FDs?

- f1: Mãsv → Họtên, mãlớp
- f2: Mãlớp → Tênlớp
- f3: Mãsv, Mônhọc → Điểm
- f4: Mãsv, Tênsv, Mônhọc → Điểm
- f5: Mãsv, Mãlớp → Tênlớp

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

5.2. Complementary Concepts

Example:

R(ABCDEI), F = { A→BCD, BCD→E, CD→EI }

Which of the following FDs is a full dependency?

- BCD→E ✗
- A→D ✓
- CD→I ✓
- AC→I ✗

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

5.2. Complementary Concepts

Transitive functional dependencies

fit@hcmus

(Phụ thuộc bắc cầu)

- **Transitive functional dependency**

  A FD $X \rightarrow A$ is a transitive dependency if it satisfies all four of the following conditions:

(1) $X \rightarrow Y$,

(2) $Y \rightarrow A$,

(3) $Y \not\rightarrow X$,

(4) $A \notin XY$.

SinhVien (mãsv, họtên, mãlớp, tênlớp, mônhọc, điểm, sốcccd, ngàycấp_sốcccd)

Xác định phụ thuộc hàm bắc cầu?

f1: Mãsv → tênlớp

f2: Mãsv → ngàycấp_sốcccd

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

fit@hcmus

5.2. Complementary Concepts

Transitive functional dependencies

SinhVien (mãsv, họtên, mãlớp, tênlớp, mônhọc, điểm, sốcccd, ngàycấp_sốcccd)

Xác định phụ thuộc hàm bắc cầu?

f1: Mãsv → tênlớp

(1) Mãsv → Mãlớp ☑

(2) Mãlớp → Tênlớp ☑

(3) Mãlớp $\not\rightarrow$ Mãsv ☑

(4) Tênlớp ∉ {Mãsv, Mãlớp} ☑

→ *Phụ thuộc bắc cầu*

f2: Mãsv → ngàycấp_sốcccd

(1) Mãsv → Sốcccd ☑

(2) Sốcccd → Ngàycấp_sốcccd ☑

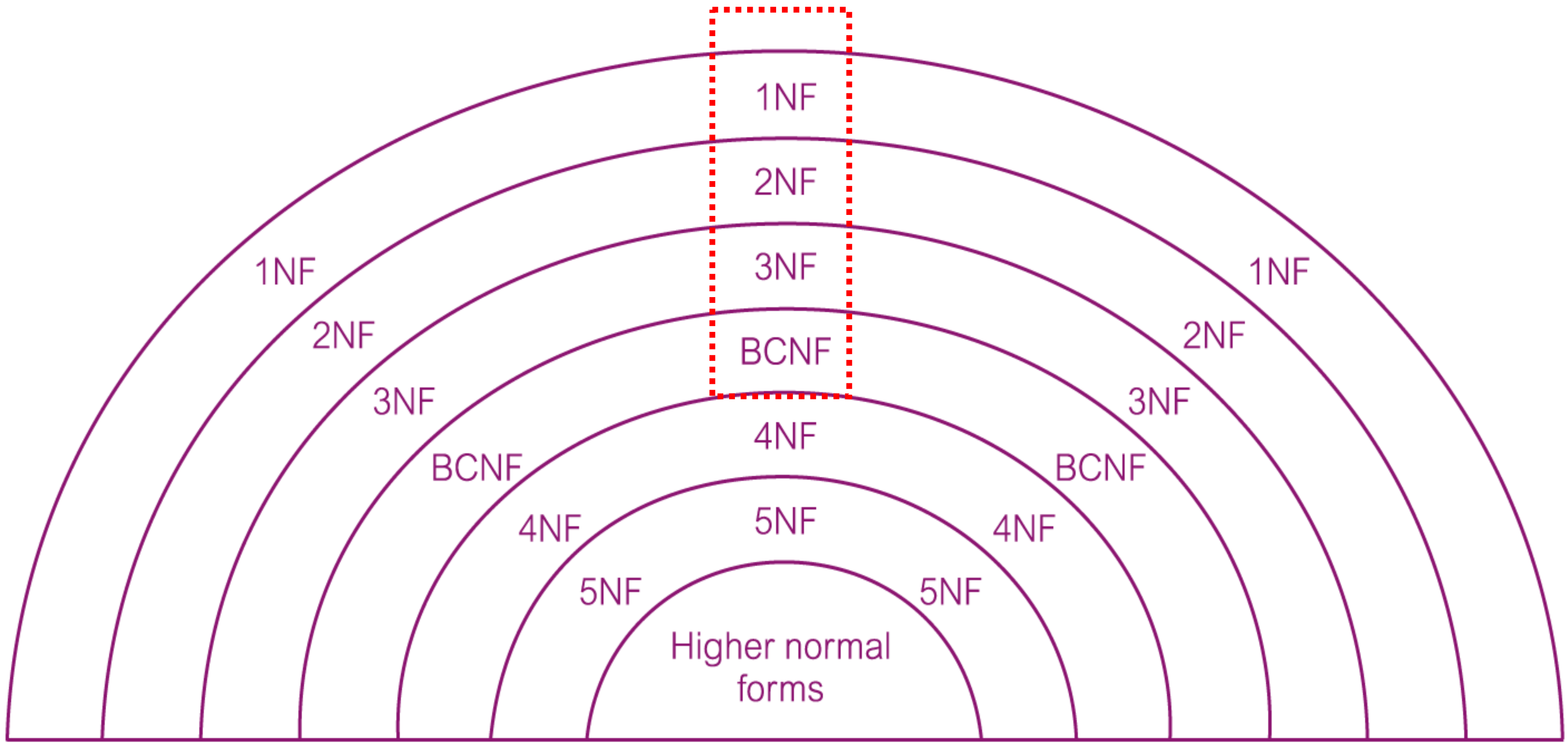
(3) Sốcccd → Mãsv ✗

→ ***KHÔNG*** *phụ thuộc bắc cầu*

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

5.3. Normal forms

- Purpose: to evaluate the level of data duplication of a database schema



5.4. 1st Normal Form (1NF)

- Disallows composite attributes, multivalued attributes, and **nested relations**; attributes whose values *for an individual tuple* are non-atomic
- Note:
  - Every relation schema belongs to 1NF
  - 1NF has high data duplication, which causes data update anomalies
- Example: given a relation THUENHA as followings

MANT	TENNT	MANHA	DCHI_NHA	NGAYTHUE_BT	NGAYTHUE_KT	GIATHUE	MACHUNHA	TENCHUNHA
CR76	L.V.Hùng	PG4	125 Lê Văn Sỹ, Q3	01/08/2012	01/09/2013	50tr	CO40	N.T Lan
		PG16	432 CMT8, QTB	01/09/2007	01/10/2011	150tr	CO72	B.T.Thanh
CR56	H.V.Gia	PG4	125 Lê Văn Sỹ, Q3	01/08/2007	01/07/2012	50tr	CO40	N.T.Lan
		PG16	432 CMT8, QTB	01//01/2012	01/01/2013	150tr	CO72	B.T.Thanh
		PG36	124 Tô Ký, Q12	01/01/2010	01/01/2014	200tr	CO20	N.T.Phuong

Nested relation

5.4. Normalization relations into 1NF

- Method 1:
  - Fill in the data in the blanks with duplicate data → leading to many duplicated data on the relation.

- Method 2: nested relation into 1NF
  - Replace non-atomic values by specifying a attributes set as primary key, then splitting into a new relation → we can create two or more new relations, and this will reduce data duplication

5.4. Normalization relations into 1NF -Example

- Method 1: split rows

MANT	TENNT	MANHA	DCHI_NHA	NGAYTHUE_BT	NGAYTHUE_KT	GIATHUE	MACHUNHA	TENCHUNHA
CR76	L.V.Hùng	PG4 PG16	125 Lê Văn Sỹ, Q3 432 CMT8, QTB	01/08/2012 01/09/2007	01/09/2013 01/10/2011	50tr 150tr	CO40 CO72	N.T Lan B.T.Thanh
CR56	H.V.Gia	PG4 PG16 PG36	125 Lê Văn Sỹ, Q3 432 CMT8, QTB 124 Tô Ký, Q12	01/08/2007 01//01/2012 01/01/2010	01/07/2012 01/01/2013 01/01/2014	50tr 150tr 200tr	CO40 CO72 CO20	N.T.Lan B.T.Thanh N.T.Phuong

MANT	TENNT	MANHA	DCHI_NHA	NGAYTHUE_BT	NGAYTHUE_KT	GIATHUE	MACHUNHA	TENCHUNHA
CR76	L.V.Hùng	PG4	125 Lê Văn Sỹ, Q3	01/08/2012	01/09/2013	50tr	CO40	N.T Lan
CR76	L.V.Hùng	PG16	432 CMT8, QTB	01/09/2007	01/10/2011	150tr	CO72	B.T.Thanh
CR56	H.V.Gia	PG4	125 Lê Văn Sỹ, Q3	01/08/2007	01/07/2012	50tr	CO40	N.T.Lan
CR56	H.V.Gia	PG16	432 CMT8, QTB	01//01/2012	01/01/2013	150tr	CO72	B.T.Thanh
CR56	H.V.Gia	PG36	124 Tô Ký, Q12	01/01/2010	01/01/2014	200tr	CO20	N.T.Phuong

5.4. Normalization relations into 1NF -Example

- Method 2: split nested relations

NGUOI_THUE (<u>MANT</u>, TENNT)

MANT	TENNT
CR76	L.V.Hùng
CR56	H.V.Gia

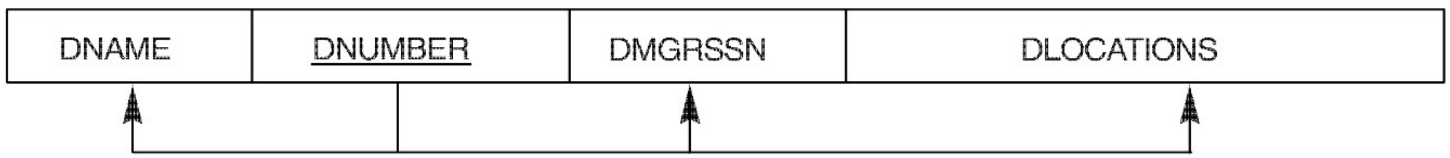
NHA_CHO_THUE (<u>MANT, MANHA</u>, DCHI_NHA, NGAYTHUE_BT, NGAYTHUE_KT, GIATHUE, MACHUNHA, TENCHUNHA)

MANT	MANHA	DCHI_NHA	NGAYTHUE_BT	NGAYTHUE_KT	GIATHUE	MACHUNHA	TENCHUNHA
CR76	PG4	125 Lê Văn Sỹ, Q3	01/08/2012	01/09/2013	50tr	CO40	N.T Lan
CR76	PG16	432 CMT8, QTB	01/09/2007	01/10/2011	150tr	CO72	B.T.Thanh
CR56	PG4	125 Lê Văn Sỹ, Q3	01/08/2007	01/07/2012	50tr	CO40	N.T.Lan
CR56	PG16	432 CMT8, QTB	01//01/2012	01/01/2013	150tr	CO72	B.T.Thanh
CR56	PG36	124 Tô Ký, Q12	01/01/2010	01/01/2014	200tr	CO20	N.T.Phuong

5.4. Normalization relations into 1NF -Example

Figure 14.8 Normalization into 1NF. (a) Relation schema that is not in 1NF. (b) Example relation instance. (c) 1NF relation with redundancy.

(a) DEPARTMENT



(b) DEPARTMENT

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

(c) DEPARTMENT

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

5.5. 2nd Normal form (2NF)

- Uses the concepts of **FD**s, **primary key**
- **(Non) Prime attribute** - attribute that is (not) member of the primary key K
- Ex: Q1(MNOPX) ; Q2(PY)
  - M,N,O are prime attributes of Q1, P is prime attribute of Q2 but is not prime attribute of Q1
- **Full dependent attribute** – Attribute A is fully dependent on X if X → A is *fully functional dependency*.

5.5. 2nd Normal form (2NF)

- A relation schema R is in **second normal form** if every *non-prime attribute* A in R is fully functionally dependent on the primary key.
- Note:
  - R can be decomposed into 2NF relations via the process of 2NF normalization
  - All relation schemas that achieve 2NF also achieve 1NF
- Ex:

NGUOI_THUE (MANT, TENNT)

MANT	TENNT
CR76	L.V.Hùng
CR56	H.V.Gia

PK = {MANT}, F = {MANT → TENNT}
NGUOI_THUE achieve 2NF ?

► **NGUOI_THUE: achieve 2NF**

5.5. 2nd Normal form (2NF) - Example

NHA_CHO_THUE (<u>MANT, MANHA</u>, DCHI_NHA, NGAYTHUE_BT, NGAYTHUE_KT, GIATHUE, MACHUNHA, TENCHUNHA)

MANT	MANHA	DCHI_NHA	NGAYTHUE_BT	NGAYTHUE_KT	GIATHUE	MACHUNHA	TENCHUNHA
CR76	PG4	125 Lê Văn Sỹ, Q3	01/08/2012	01/09/2013	50tr	CO40	N.T Lan
CR76	PG16	432 CMT8, QTB	01/09/2007	01/10/2011	150tr	CO72	B.T.Thanh
CR56	PG4	125 Lê Văn Sỹ, Q3	01/08/2007	01/07/2012	50tr	CO40	N.T.Lan
CR56	PG16	432 CMT8, QTB	01//01/2012	01/01/2013	150tr	CO72	B.T.Thanh
CR56	PG36	124 Tô Ký, Q12	01/01/2010	01/01/2014	200tr	CO20	N.T.Phuong

PK = {MANT, MANHA}
F = { MANHA → DCHI_NHA, GIATHUE, MACHUNHA, TENCHUNHA
NHA_CHO_THUE achieve 2NF?

► **NHA_CHO_THUE: not achieve 2NF → only achieve 1NF**

5.5. Normalizing into 2NF

- Step 1: identify **primary key** of the relation 1NF

NHA_CHO_THUE with PK = {MANT, MANHA}

- Step 2: identify FDs causing **non-prime attributes** that **are not fully dependent** on the primary key

MANHA → DCHI_NHA, GIA_THUE, MACHUNHA, TENCHUNHA

- Step 3: If it exists **non-full dependent attributes** on the primary key, remove them from the old relation and add them in a new relation
  - Remove FD: MANHA → DCHI_NHA, GIA_THUE, MACHUNHA, TENCHUNHA by adding to new relation **NHA_THUE** (<u>MANHA</u>,DCHI_NHA, GIA_THUE, MACHUNHA, TENCHUNHA).
  - Rename the relation **NHA_CHO_THUE** to: **TT_THUE_NHA** (<u>MANT,MANHA</u>, NGAYTHUE_BT, NGAYTHUE_KT)

5.5. Normalizing into 2NF

1NF

NHA_CHO_THUE (<u>MANT, MANHA</u>, DCHI_NHA, NGAYTHUE_BT, NGAYTHUE_KT, GIATHUE, MACHUNHA, TENCHUNHA)

MANT	MANHA	DCHI_NHA	NGAYTHUE_BT	NGAYTHUE_KT	GIATHUE	MACHUNHA	TENCHUNHA
CR76	PG4	125 Lê Văn Sỹ, Q3	01/08/2012	01/09/2013	50tr	CO40	N.T Lan
CR76	PG16	432 CMT8, QTB	01/09/2007	01/10/2011	150tr	CO72	B.T.Thanh
CR56	PG4	125 Lê Văn Sỹ, Q3	01/08/2007	01/07/2012	50tr	CO40	N.T.Lan
CR56	PG16	432 CMT8, QTB	01//01/2012	01/01/2013	150tr	CO72	B.T.Thanh
CR56	PG36	124 Tô Ký, Q12	01/01/2010	01/01/2014	200tr	CO20	N.T.Phuong

NHA_THUE (<u>MANHA</u>, DCHI_NHA, GIATHUE, MACHUNHA, TENCHUNHA)

MANHA	DCHI_NHA	GIATHUE	MACHUNHA	TENCHUNHA
PG4	125 Lê Văn Sỹ, Q3	50tr	CO40	N.T Lan
PG16	432 CMT8, QTB	150tr	CO72	B.T.Thanh
PG36	124 Tô Ký, Q12	200tr	CO20	N.T.Phuong

2NF

2NF

TT_THUE_NHA (<u>MANT,MANHA</u>, NGAYTHUE_BT, NGAYTHUE_KT)

MANT	MANHA	NGAYTHUE_BT	NGAYTHUE_KT
CR76	PG4	01/08/2012	01/09/2013
CR76	PG16	01/09/2007	01/10/2011
CR56	PG4	01/08/2007	01/07/2012
CR56	PG16	01//01/2012	01/01/2013
CR56	PG36	01/01/2010	01/01/2014

5.6. Normalizing into 3NF

- 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
- Example:

NHA_THUE (MANHA, DCHI_NHA, GIATHUE, MACHUNHA, TENCHUNHA)

MANHA	DCHI_NHA	GIATHUE	MACHUNHA	TENCHUNHA
PG4	125 Lê Văn Sỹ, Q3	50tr	CO40	N.T Lan
PG16	432 CMT8, QTB	150tr	CO72	B.T.Thanh
PG36	124 Tô Ký, Q12	200tr	CO20	N.T.Phuong

PK = {MANHA}

F= { f1: MANHA → DCHI_NHA,GIA_THUE, MACHUNHA, TENCHUNHA

f2: MACHUNHA → TENCHUNHA }

Is NHA_THUE in 3NF?

Prove:

MANHA → MACHUNHA

MACHUNHA → TENCHUNHA

⇒ MANHA → TENCHUNHA (transitive)

► **NGUOI_THUE: not in 2NF, because TENCHUNHA is functional dependent on the key**

5.6. Normalizing into 3NF - Method

- Step 1: identifying the primary key of the relation that meets 2NF

  NHA_THUE: PK = {MANHA}

- Step 2:Identifying a FD that causes a non-prime attribute being transitive functional dependent on the primary key

  MACHUNHA → TENCHUNHA

- Step 3: Removing the FD by adding its attributes into a new relation

  Remove FD MACHUNHA → TENCHUNHA from the relation NHA_THUE by adding into a new relation **CHU_NHA**(MACHUNHA,TEN CHUNHA)

5.6. Normalizing into 3NF - Method

NHA_THUE (MANHA, DCHI_NHA, GIATHUE, MACHUNHA, TENCHUNHA)

2NF

MANHA	DCHI_NHA	GIATHUE	MACHUNHA	TENCHUNHA
PG4	125 Lê Văn Sỹ, Q3	50tr	CO40	N.T Lan
PG16	432 CMT8, QTB	150tr	CO72	B.T.Thanh
PG36	124 Tô Ký, Q12	200tr	CO20	N.T.Phuong

NHA_THUE (MANHA, DCHI_NHA, GIATHUE, MACHUNHA)

MANHA	DCHI_NHA	GIATHUE	MACHUNHA
PG4	125 Lê Văn Sỹ, Q3	50tr	CO40
PG16	432 CMT8, QTB	150tr	CO72
PG36	124 Tô Ký, Q12	200tr	CO20

3NF

CHU_NHA(MACHUNHA, TENCHUNHA)

MACHUNHA	TENCHUNHA
CO40	N.T Lan
CO72	B.T.Thanh
CO20	N.T.Phuong

3NF

5.6. Normalizing into 3NF

- Notes:
  - All relation schemas that achieve 3NF also achieve 2NF.
  - The transitive functional dependency causes data duplication and the relation schema fail to achieve 3NF.
  - 3NF is not achieved due to the presence of non-prime attributes that are mutually inferable in the relation.
  - 3NF is the minimum normal form in database design

5.7. Normalizing into Boyce-Codd Normal Form (BCNF)

- A relation schema R is in **Boyce-Codd Normal Form** (**BCNF**) if whenever an FD X → A holds in R, then X is a superkey of R
- Each normal form is strictly stronger than the previous one
  - Every 2NF relation is in 1NF
  - Every 3NF relation is in 2NF
  - Every BCNF relation is in 3NF
- There exist relations that are in 3NF but not in BCNF
- The goal is to have each relation in BCNF (or 3NF)

5.7. Normalizing into BCNF - Example

NGUOI_THUE (MANT, TENNT)

MANT	TENNT
CR76	L.V.Hùng
CR56	H.V.Gia

BCNF

BCNF

BCNF

BCNF

TT_THUE_NHA (MANT, MANHA, NGAYTHUE_BT, NGAYTHUE_KT)

MANT	MANHA	NGAYTHUE_BT	NGAYTHUE_KT
CR76	PG4	01/08/2012	01/09/2013
CR76	PG16	01/09/2007	01/10/2011
CR56	PG4	01/08/2007	01/07/2012
CR56	PG16	01//01/2012	01/01/2013
CR56	PG36	01/01/2010	01/01/2014

NHA_THUE (MANHA, DCHI_NHA, GIATHUE, MACHUNHA)

MANHA	DCHI_NHA	GIATHUE	MACHUNHA
PG4	125 Lê Văn Sỹ, Q3	50tr	CO40
PG16	432 CMT8, QTB	150tr	CO72
PG36	124 Tô Ký, Q12	200tr	CO20

CHU_NHA(MACHUNHA, TENCHUNHA)

MACHUNHA	TENCHUNHA
CO40 CO72	N.T Lan B.T.Thanh
CO20	N.T.Phuong

BCNF

5.7. Normalizing into BCNF - Example

PHONGVAN(MAUV, NGAYPV, GIOPV, MANV, MAPHG)

MAUV	NGAYPV	GIOPV	MANV	MAPHG
CR76	13/05/2005	10:30	SG5	G101
CR56	13/05/2005	12:00	SG5	G101
CR74	13/05/2005	12:00	SG37	G102
CR56	1/07/2005	10:30	SG5	G201

F = { FD1: MAUV, NGAYPV → GIOPV, MANV, MAPHG
FD2: MANV, NGAYPV, GIOPV → MAUV
FD3: MAPHG, NGAYPV, GIOPV → MAUV, MANV
FD4: MANV, NGAYPV → MAPHG }

Primary key = {MAUV, NGAYPV}
Candidate keys = { (MANV, NGAYPV, GIOPV); (MAPHG, NGAYPV, GIOPV)}
⇒ **PHONGVAN: not achieve BCNF,** because FD4 having the left hand side **is not** primary key/candidate key.

5.7. Normalizing into BCNF

- Step 1: Identify the FD: $A \rightarrow B$ of the relation Q, where $A \neq B$ and A is not a superkey/key.
- Step 2: Decompose the relation Q into two relations: Q1 = {A,B}, Q2 = {the set of remaining attributes of Q } – {B}
- Step 3: Repeat the above steps for Q2 until it cannot continue.
- Step 4: The relation Q1 and the {Qi} decomposed from Q2 are the relations that achieve the BCNF.

5.7. Normalizing into BCNF - Example

PHONGVAN(MAUV, NGAYPV, GIOPV, MANV, MAPHG) **3NF**

MAUV	NGAYPV	GIOPV	MANV	MAPHG
CR76	13/05/2005	10:30	SG5	G101
CR56	13/05/2005	12:00	SG5	G101
CR74	13/05/2005	12:00	SG37	G102
CR56	1/07/2005	10:30	SG5	G201

F = { f1: MAUV, NGAYPV → GIOPV, MANV, MAPHG
f2: MANV, NGAYPV, GIOPV → MAUV
f3: MAPHG, NGAYPV, GIOPV → MAUV, MANV
f4: MANV, NGAYPV → MAPHG
}

BCNF

NV_PHONG (MANV,NGAYPV, MAPHG)

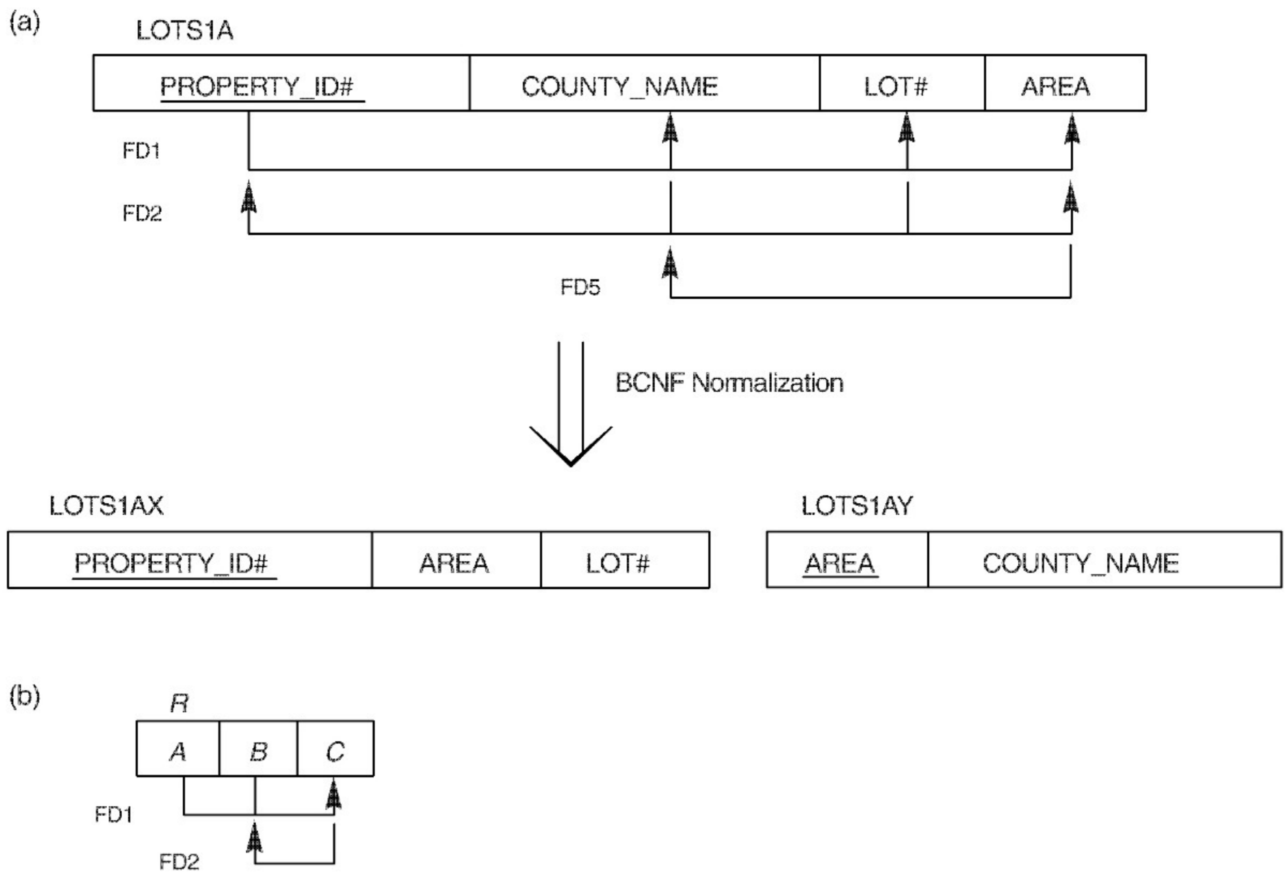
MANV	NGAYPV	MAPHG
SG5	13/05/2005	G101
~~SG5~~	~~13/05/2005~~	~~G101~~
SG37	13/05/2005	G102
SG5	1/07/2005	G201

PHONGVAN(MAUV,NGAYPV, GIOPV, MANV) **BCNF**

MAUV	NGAYPV	GIOPV	MANV
CR76	13/05/2005	10:30	SG5
CR56	13/05/2005	12:00	SG5
CR74	13/05/2005	12:00	SG37
CR56	1/07/2005	10:30	SG5

5.7. Normalizing into BCNF - Example

Figure 14.12 Boyce-Codd normal form. (a) BCNF normalization with the dependency of FD2 being "lost" in the decomposition. (b) A relation R in 3NF but not in BCNF.



5.7. Normalizing into BCNF - Example

- A relation TEACH that is in 3NF but not in BCNF

Figure 14.13 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

Quiz

Given two statement S1 and S2 as follows:

- **S1**: All relation schemas with only 2 attributes achieve 1NF, 2NF, 3NF, BCNF
- **S2**: F= {AB→C, D→E, E→C} a **minimal** set of FDs G = {AB→C, D→E, AB→E, E→C}

Indicate which of the following statements is correct?

A. S1 correct, S2 wrong
B. S1 correct, S2 correct
C. S1 wrong, S2 correct
D. S1 wrong and S2 wrong
E. Not all above correct

Example

- R(A,B,C,D,E,F)

  $\underbrace{A,B,C,D,E,F}_{U = \{ ABCDEF \}}$

- F= { f1: D → B ,

  f2: A → C ,

  f3: AD → E ,

  f4: C → F }

Key: AD. Normal form of R?

Example

- Tìm khóa của lược đồ quan hệ :

$$R(\underbrace{A,B,C,D,E,F}_{U = \{ABCDEF\}})$$

- Có tập phụ thuộc hàm sau:

$$F = \{\ f1: A \rightarrow D,\quad f2: C \rightarrow AF,\quad f3: AB \rightarrow EC\ \}$$

Key: BA or BC. Normal form of R?

fit@hcmus

Summary of Normal Form

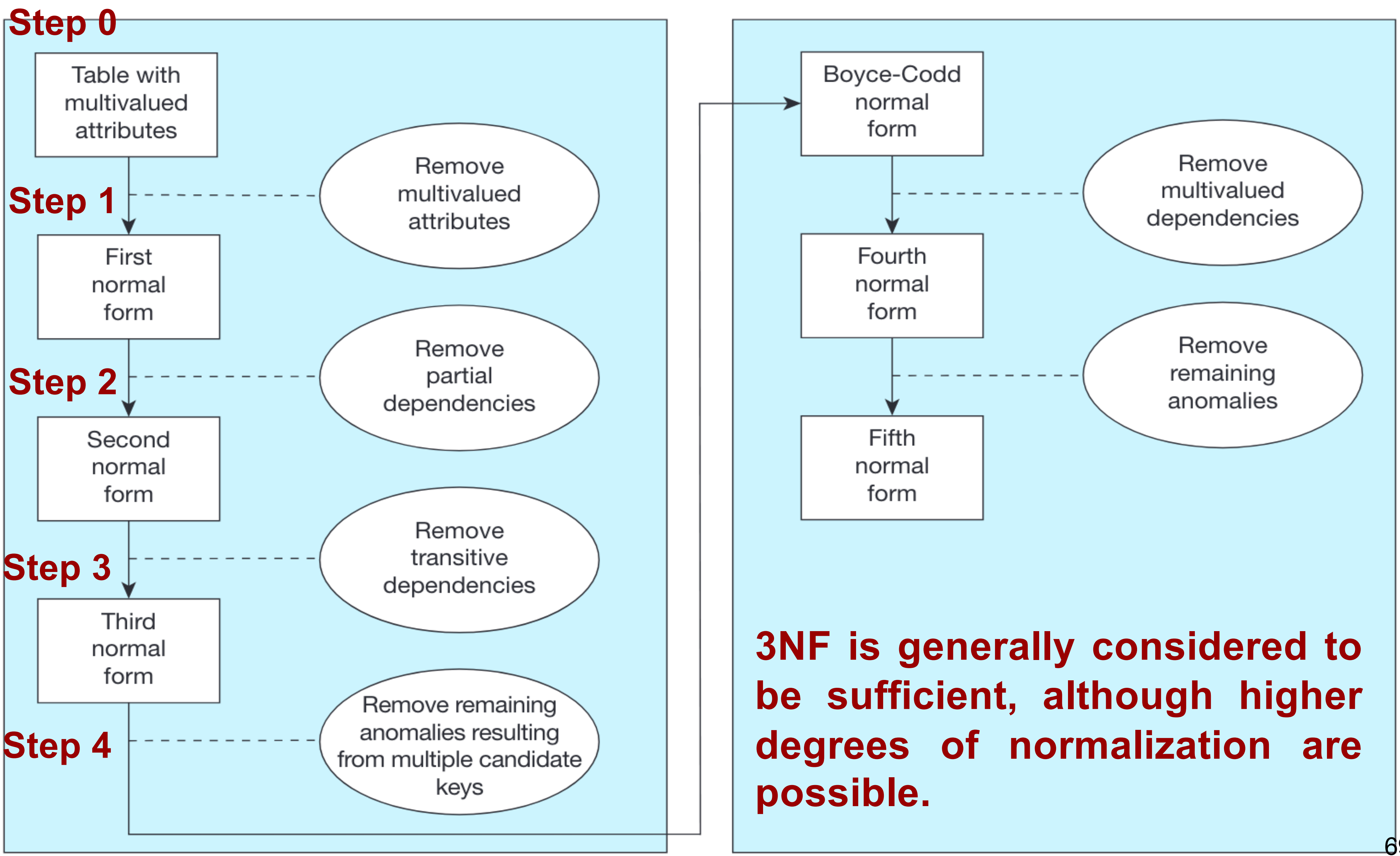
Normal Form	Characteristic
1NF	Table format, no repeating groups (or multivalued attributes), and PK identified.
2NF	1NF and no partial dependencies (between non-key attributes and any keys). **Hint**: If there is a functional dependency X -> Y, where X is part of any keys **and** Y is a non-key attribute holding in relation R, then R does not satisfy 2NF.
3NF	2NF and no transitive dependencies (no FD exists between non-key attributes). **Hint**: If there is a functional dependency X -> Y, where X is not a super key **and** Y is a non-key attribute holding in relation R, then R does not satisfy 3NF.
BCNF	Every determinant is a super key (special case of 3NF). **Hint**: If there is a functional dependency X -> Y, where X is not a super key holding in R, then R does not satisfy BCNF.

(Coronel and Morris, 2015) (Silberschatz et al. 2020)

fit@hcmus

Decomposition Process

(Based on FD Theory)



3NF is generally considered to be sufficient, although higher degrees of normalization are possible.

(Coronel and Morris, 2015) (Silberschatz et al. 2020)