

204321 - Database Management

Basics of Functional Dependencies and Normalization for Relational Databases

สอนโดย

ผู้ช่วยศาสตราจารย์ ดร.อารีรัตน์ ตรังศรีมัททอง (Section 001)

รองศาสตราจารย์ ดร.สุรี เตชะวุฒิ (Section 002)

Addison-Wesley
is an imprint of

PEARSON

Fundamentals of Database Systems

Copyright © 2011 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Chapter 15 Outline

- Informal Design Guidelines for Relation Schemas
- Functional Dependencies
- Normal Forms Based on Primary Keys
- General Definitions of Second and Third Normal Forms
- Boyce-Codd Normal Form
- Lossless Join Property

2

Introduction

- เราสามารถพิจารณาได้ว่าโครงสร้างรีเลชัน (Relation Schema) ที่ออกแบบ ดีหรือไม่ดี ได้จาก 2 ระดับ คือการออกแบบในระดับแนวคิด (Conceptual Level) และการออกแบบในระดับกายภาพ (Physical Level)
 - **Logical (or conceptual) level:** การออกแบบในระดับแนวคิด ใช้สำหรับสื่อสารกับผู้ใช้งาน ซึ่งสามารถพิจารณาจาก ความหมายของ **Attribute** หากความหมายของ **Attribute** ชัดเจนจะทำให้ผู้ใช้เข้าใจความหมายของข้อมูลได้ชัดเจนมากขึ้น และจะทำให้ผู้ใช้สามารถบอกข้อมูลที่ต้องการเพื่อการบริหารจัดการข้อมูล หรือข้อมูลที่ต้องการสืบค้นได้ชัดเจน และถูกต้อง
 - **Implementation (or physical storage) level:** การออกแบบในระดับที่จัดเก็บข้อมูลเชิงกายภาพ ใช้สำหรับเพิ่มประสิทธิภาพในการเข้าถึงข้อมูล
- **Approaches to database design:** แนวทางในการออกแบบฐานข้อมูลทำได้ 2 ลักษณะ คือ
 - **Bottom-up:** การออกแบบจากล่างขึ้นบน
 - **Top-down:** การออกแบบจากบนลงล่าง

3

Informal Design Guidelines for Relation Schemas

- **Measures of quality:** การวัดประสิทธิภาพของโครงสร้างรีเลชัน
 - ให้แน่ใจว่าความหมายของแต่ละ **Attribute** มีความชัดเจน
 - ลดความซ้ำซ้อนของข้อมูลใน Tuples
 - ลดการจัดเก็บค่า **Null** ใน Tuples
 - ไม่อนุญาตให้เกิด **Spurious Tuples** คือ Tuples ที่ผิดพลาด เช่น ได้ข้อมูลเกินจากความเป็นจริงหลังจากรวมข้อมูลจากหลายตารางข้อมูล (Table/Relation)

4

Imparting Clear Semantics to Attributes in Relations

- **Semantics of a relation:** ความหมายของรีเลชัน
 - ความหมายที่สะท้อนถึง **Relation** มาจากการแปลความหมายของค่าข้อมูล **Attribute** ต่าง ๆ ที่อยู่ใน **Tuple** ของแต่ละ **Relation**
 - จะต้องอธิบายความหมายของ **Relation** แต่ละ **Relation** ได้ง่ายและชัดเจน ไม่กำกวม

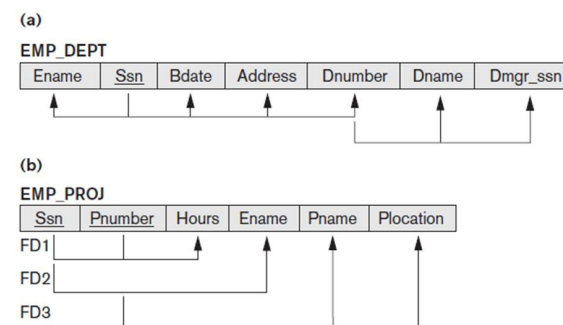
5

Guideline 1

- ต้องออกแบบให้สามารถอธิบายความหมายของแต่ละ **Relation** ได้ง่ายและชัดเจน ไม่กำกวม
- ต้องไม่รวม **Attribute** ที่มาจากข้อมูลหลายกลุ่ม หรือหลาย **Domain** ไว้ใน **Relation** เดียว
- ตัวอย่างการออกแบบที่ผิดพลาด แสดงดังรูปที่ 15.3

Figure 15.3

Two relation schemas suffering from update anomalies. (a) EMP_DEPT and (b) EMP_PROJ.



6

Redundant Information in Tuples and Update Anomalies

- การรวมข้อมูลที่มาจากหลาย **Domain** ไว้ด้วยกันมีผลกระทบโดยตรงทำให้เปลืองเนื้อที่ในการจัดเก็บข้อมูล
- การจัดเก็บข้อมูลที่อยู่ต่าง **Domain** ถึงแม้จะมีความสัมพันธ์กันในลักษณะ **Natural Joins** ก็สามารถนำไปสู่ปัญหา **Update Anomalies** ได้ (ปัญหาการปรับปรุงข้อมูลที่ทำให้ข้อมูลมีลักษณะผิดปกติ)
- **Types of update anomalies:** ประเภทของการปรับปรุงข้อมูลที่อาจส่งผลให้ข้อมูลมีลักษณะผิดปกติ มีดังนี้
 - Insertion
 - Deletion
 - Modification

7

Guideline 2

- ในการออกแบบ **Relation** จะต้องออกแบบไม่ให้เกิด **Update Anomalies** ในแต่ละ **Relation**
- หากไม่สามารถหลีกเลี่ยงการเกิด **Update Anomalies** ได้
 - ให้บันทึกไว้ให้ชัดเจน
 - ให้แน่ใจว่าโปรแกรมจะปรับปรุงข้อมูลในฐานข้อมูลได้ถูกต้อง
- การรวมข้อมูลจากหลาย **Domain** มาไว้ด้วยกันจะส่งผลให้มีข้อมูลที่มีค่าเป็น **NULL** เกิดขึ้นใน **Relation**
- การจัดเก็บค่า **NULL** ทำให้เกิดปัญหาดังต่อไปนี้
 - เปลืองเนื้อที่ในการจัดเก็บข้อมูล
 - การเก็บข้อมูลที่เป็นค่า **NULL** อาจส่งผลในแง่ไม่สื่อความหมาย หรือเข้าใจความหมายได้ยาก

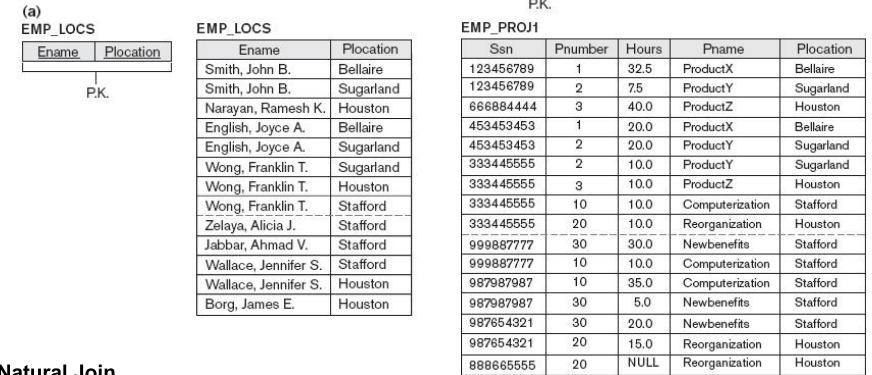
8

Guideline 3

- หลีกเลี่ยงการจัดเก็บ Attribute ที่มีโอกาสจัดเก็บค่า NULL จำนวนมาก
- แต่ถ้าหลีกเลี่ยง NULL ไม่ได้
 - ขอให้เป็นกรณียกเว้น อย่าให้เป็นกรณีส่วนใหญ่ หรือกรณีทั่วไป
- NATURAL JOIN: ในกรณีที่มีการรวมข้อมูล (Join) จากหลาย Relation โดยแต่ละ Relation ที่นำมารวมเป็น Attribute ซ้ำเดียวกันและมีชนิดข้อมูลเหมือนกัน
 - การรวมข้อมูล (Relation) ที่มีความสัมพันธ์กัน แต่เลือก Attribute ที่เป็นตัวเชื่อมความสัมพันธ์ไม่เหมาะสม จะทำให้เกิด Spurious Tuples ดังตัวอย่างแสดงในรูป 15.5 ในหน้าถัดไป
 - การที่มีข้อมูล Spurious เกิดขึ้น นั้นหมายความว่า ออกแบบไม่ถูกต้อง

9

Figure 15.5
Particularly poor design for the EMP_PROJ relation



Natural Join

Ename	Plocation	Ssn	Pnumber	Hours	Pname
Smith, John B.	Bellaire	123456789	1	32.5	ProductX
Smith, John B.	Bellaire	453453453	1	20.0	ProductX
Smith, John B.	Sugarland	123456789	2	7.5	ProductY
Smith, John B.	Sugarland	453453453	2	20.0	ProductY
Smith, John B.	Sugarland	333445555	2	10.0	ProductY
...					

Spurious Tuples

10

Guideline 4

- การออกแบบที่มีการรวมข้อมูลจาก Relation อื่น (Join) จะต้องเลือก Attribute ที่เหมาะสม ซึ่งควรอยู่ในลักษณะ Equi Join จะทำให้ไม่เกิด Spurious Tuples
- ให้หลีกเลี่ยงการนำ Attribute ที่ไม่ใช่ Foreign Key หรือ Primary key มาเป็น ตัวเชื่อมโยง ระหว่าง 2 Relation

11

Summary and Discussion of Design Guidelines

- Anomalies (ลักษณะที่ผิดปกติ) เป็นสาเหตุให้
 - ต้องทำงานซ้ำซ้อน
 - เปลืองเนื้อที่ในการจัดเก็บค่า NULLs
 - ทำให้เกิดความยุ่งยากเมื่อมีการดำเนินการใดๆกับข้อมูล และเกิดความยุ่งยากในการรวมข้อมูลที่มีค่าข้อมูลเป็น NULL
 - ทำให้เกิดข้อมูลที่ไม่พึงประสงค์จากการรวมข้อมูล

12

Functional Dependencies

- **Functional Dependency (FD)** เป็นเครื่องมือที่ใช้สำหรับวิเคราะห์โครงสร้างของ Relation
- ทำให้เราสามารถตรวจจับปัญหาที่เกิดจากข้อผิดพลาดในการออกแบบ
- **Theory of functional dependency:** คือข้อบังคับระหว่าง Attribute 2 ชุดในฐานข้อมูล โดยมีค่านิยามดังแสดงด้านล่าง
 - การขึ้นตรงต่อกันของ attribute ที่อยู่ในตารางข้อมูลเดียวกัน แสดงด้วยสัญลักษณ์ $X \rightarrow Y$ อ่านได้ 2 ลักษณะคือ
 - ค่าของ attribute X กำหนดค่าของ attribute Y (**X determine Y**)
 - attribute Y ขึ้นตรงต่อ attribute X (**Y is functionally dependent on X**)
- ข้อกำหนดของ $X \rightarrow Y$:
attribute X จะกำหนดค่าของ attribute Y ถ้ามี $t_1[X] = t_2[X] \rightarrow t_1[Y] = t_2[Y]$
- FD เป็นเครื่องมือที่เน้นความหมายของ Attribute เป็นหลัก
- **Legal relation states:** Relation ที่อยู่ในสถานะที่ถูกต้องจะต้องผ่านข้อกำหนดของ FD

13

Definition of Functional Dependency

- Given a populated relation
 - เราจะไม่มีการรู้ว่า Relation แต่ละ Relation จะผ่านข้อกำหนดของ FD หรือไม่จนกว่าเราจะรู้ความหมายของ Attribute และความสัมพันธ์ระหว่าง Attribute วิธีที่จะทำให้รู้คือ ต้องลองยกตัวอย่างข้อมูลของแต่ละ Relation
 - หลังจากที่เราได้ข้อมูลแล้ว หากพบแม้แต่ 1 Record/Tuple ที่ละเมิดข้อกำหนดของ FD จะถือว่า ไม่ผ่านข้อกำหนดของ FD ดังกล่าว

14

Normal Forms Based on Primary Keys

- **Normalization process:** คือแนวทาง หรือกระบวนการสำหรับการออกแบบโครงสร้างของ Relation
 - ออกแบบโครงสร้างระดับแนวคิด (ที่ผ่านมามีเรานำเสนอการออกแบบโครงสร้างในระดับแนวคิดในรูปแบบของแผนผังอีอาร์ (ER Diagram))
 - หลังจากนั้นทำการแปลงโครงสร้างระดับแนวคิดให้อยู่ในรูป Relation ที่คอมพิวเตอร์สามารถประมวลผลได้ (ที่ผ่านมามีเราใช้ ER/EER Mapping)
 - ขั้นตอนต่อมาคือ เราต้องรู้จักและเข้าใจความหมายของข้อมูลในแต่ละ Relation แหล่งที่มาของข้อมูลสามารถรวบรวมได้จากไฟล์ข้อมูลในระบบเก่า หรือข้อมูลที่อยู่ในแบบฟอร์มหรือรายงานต่างๆ
 - ทดสอบแต่ละ Relation ว่าอยู่ในรูปแบบที่เป็นบรรทัดฐานหรือไม่ (Normal Form) ทำในลักษณะ บนลงล่าง เริ่มจาก First Normal Form (1NF)

15

Normalization of Relations

- **Normal form tests:** การทดสอบรูปแบบบรรทัดฐานของแต่ละ Relation หมายถึง รูปแบบขั้นสูงสุดที่ Relation นั้น ๆ ผ่านเงื่อนไขการทดสอบ
- คุณสมบัติที่แต่ละ Relation ควรจะมี คือ ต้องรับประกันได้ว่าจะไม่เกิด Spurious Tuple หลังจากที่เราแยก (Decompose) Relation จากกระบวนการ Normalization คุณสมบัตินี้มีความสำคัญเป็นอย่างยิ่ง
- **Dependency preservation property:** คือ ต้องมั่นใจว่า ก่อนและหลังจากแยก Relation ที่เกิดจากกระบวนการ Normalization ข้อกำหนดของ FD ในแต่ละ Relation ยังคงอยู่
 - แต่ในบางครั้ง เราอาจจะต้องละเมิดกฎข้อบังคับของ FD เพื่อประโยชน์ในด้านอื่น เช่น ความเร็วในการรวมข้อมูล เนื่องจากการแยก Relation ที่เกิดจากกระบวนการ Normalization อาจทำให้เกิดตารางข้อมูลจำนวนมาก เมื่อต้องการรวมข้อมูลหลายๆตาราง จะทำให้ใช้เวลามากขึ้นในการรวมข้อมูล

16

Practical Use of Normal Forms

- กระบวนการ Normalization ส่งผลให้การออกแบบมีคุณภาพสูง และได้รับทั้ง 2 คุณสมบัติที่กล่าวมาก่อนหน้า
- การทำขบวนการ Normalization โดยปกติจะทำถึงขั้นที่ 3 (3rd Normal Form: 3NF) Boyce-Codd Normal Form (BCNF) หรืออย่างมากที่สุดก็ Fourth Normal Form (4NF)
- บางครั้งก็ไม่จำเป็นต้องทำกระบวนการ Normalization ถึงขั้นสูงสุดที่สามารถทำได้ ในกรณีที่ต้องการความเร็วในการประมวลผลข้อมูล ก็จำเป็นต้องทำ Denormalization คือยอมที่จะเก็บข้อมูลซ้ำซ้อนที่เกิดจากการรวม (Join) ข้อมูลจากหลายตาราง และสร้างเป็นตารางใหม่ ที่มีรูปแบบของบรรทัดฐานขั้นต่ำกว่า

17

Definitions of Keys and Attributes Participating in Keys

- **Candidate key:** Key ตัวแทนของแต่ละ Relation
 - กรณีที่ Relation มีตัวแทน Key มากกว่า 1 ตัว
 - ตัวแทน Key 1 ตัวจะถูกเลือกเป็น **Primary Key (PK)**
 - ตัวแทน Key ที่เหลือ เรียกว่า **Secondary Key(s)** หรือ **Unique Key(s)**
- **Prime Attribute:** คือ Attribute ที่เป็นสมาชิกของ Candidate Key
- **Nonprime Attribute:** คือ Attribute ที่ไม่ใช่สมาชิกของ Candidate Key

18

First Normal Form

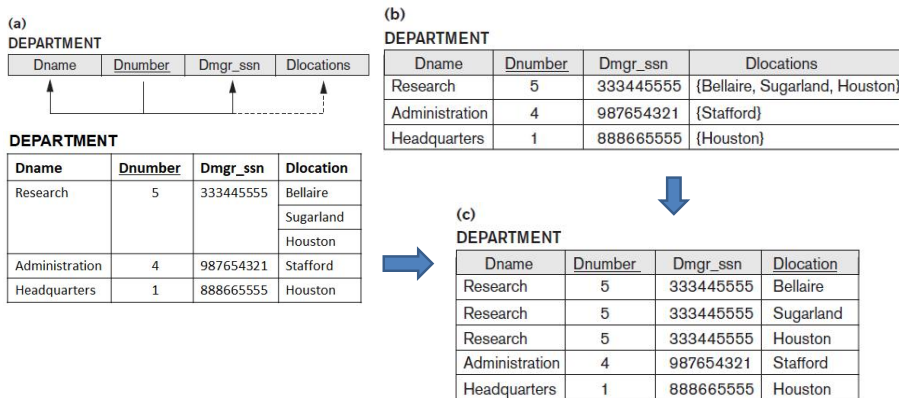
- Relation จะอยู่ในรูป **First normal Form (1NF)** ก็ต่อเมื่อ
 - ข้อมูลใน Relation ต้องอยู่ในรูปแบบระนาบเดียวกัน (flat) คือ ต้องไม่มีข้อมูลที่อยู่ในลักษณะแถวซ้อนแถว (Nested Relation)
 - แต่ละ Attribute ใน Relation ต้องไม่อยู่ในลักษณะแอตทริบิวต์องค์ประกอบ (Composite Attribute) หรือต้องอยู่ในรูปแบบที่ไม่สามารถแบ่งย่อยลงได้อีก (Indivisible or Atomic Attribute)
 - วิธีแก้ไขในกรณีนี้คือ แยก **Composite Attribute** ให้เป็น **Atomic Attribute**
 - เช่น แยก Address ออกเป็น AddressNo, Street, SubDistrict, District, Province, Zipcode เป็นต้น

19

First Normal Form

- Relation จะอยู่ในรูป **First normal Form (1NF)** ก็ต่อเมื่อ
 - ค่าข้อมูลในแต่ละ Attribute จะต้องมามีค่าข้อมูลค่าเดียว (Atomic Values) **เทคนิคที่จะทำให้ Relation ที่มีลักษณะในข้อนี้อยู่ในรูปแบบ 1NF คือ**
 - แยก Attribute ที่มีค่าหลายค่าไปไว้ใน Relation ใหม่
 - ขยาย Key (โดยที่นำ Key ของ Relation ที่เคยอยู่ไปเป็น คีย์ประกอบ (Composite Key) ใน Relation ใหม่
- ทำเหมือน **ER Mapping** ของขั้นตอนที่ 6

20



Using ER Mapping: Step 6

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

DEPT_LOCATIONS	
Dnumber	Dlocations
5	Bellaire
5	Sugarland
5	Houston
4	Stafford
1	Houston

21

Second Normal Form

Second Normal Form (2NF)

- อยู่บนพื้นฐานของ 2 แนวคิด ดังนี้
 - Full Functional Dependency
 - Partial Dependency
- Relation จะอยู่ในรูป 2NF ก็ต่อเมื่อ
 - Relation R อยู่ใน 1NF
 - ทุก ๆ Nonprime Attribute(s) ต้องขึ้นตรงกับทั้งหมดของ Prime Attribute(s) หรือ
 - ทุก ๆ Nonprime Attribute(s) ต้องไม่ขึ้นอยู่บางส่วนใดส่วนหนึ่งของ Prime Attribute(s) หรือ
 - ต้องไม่มีส่วนใดส่วนหนึ่งของ Prime Attribute(s) ไปกำหนดค่าของ Nonprime Attribute(s)

22

Third Normal Form

Third Normal Form (3NF)

- อยู่บนพื้นฐานของแนวคิด **Transitive Dependency**
- Relation จะอยู่ในรูป 3NF ก็ต่อเมื่อ
 - Relation R อยู่ใน 2NF
 - ต้องไม่มี Nonprime Attribute(s) ไปขึ้นอยู่กับ Prime Attribute(s) ผ่านทาง Nonprime Attribute(s) ตัวอื่น หรือ
 - ต้องไม่มี Nonprime Attribute(s) ไปกำหนดค่า Nonprime Attribute(s)
 - ถ้ามี X ไปกำหนดค่า Nonprime Attribute ใดๆ X ต้องเป็น Prime Attribute เท่านั้น

Transitive โดยทั่วไป: $X \rightarrow Y \rightarrow Z \Rightarrow X \rightarrow Z$
 วิธีตรวจสอบว่า มี Transitive Dependency เกิดขึ้นในตารางข้อมูลหรือไม่
 ต้องไม่มีกรณี ดังต่อไปนี้
 Prime Attribute \rightarrow Non-Prime Attribute \rightarrow Non-Prime Attribute

23

General Definitions of Second and Third Normal Forms

Table 15.1 Summary of Normal Forms Based on Primary Keys and Corresponding Normalization

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

24

Second Normalization (2NF)

วิธีดำเนินการให้ Relation R อยู่ในรูป 2NF

- แยก Nonprime Attribute(s) ที่ถูกกำหนดโดย Partial Key(s) ของ Prime Attribute(s) แต่ละตัวออกไปอยู่ใน Relation(s) ใหม่ R_1, R_2, \dots และ
- ให้นำ Partial Key(s) ที่เป็นตัวกำหนด Nonprime Attribute(s) ไปไว้ใน Relation(s) ใหม่ R_1, R_2, \dots ด้วย โดย Partial Key(s) จะถูกกำหนดให้เป็น PK ใน Relation R_1, R_2, \dots ใหม่ และ
- ให้คง Partial Key(s) ใน Relation เริ่มต้น R ด้วย เพื่อใช้เป็น FK เพื่อเป็นตัวเชื่อมไปยัง Relation ใหม่ R_1, R_2, \dots

25

Third Normalization (3NF)

วิธีดำเนินการให้ Relation R อยู่ในรูป 3NF

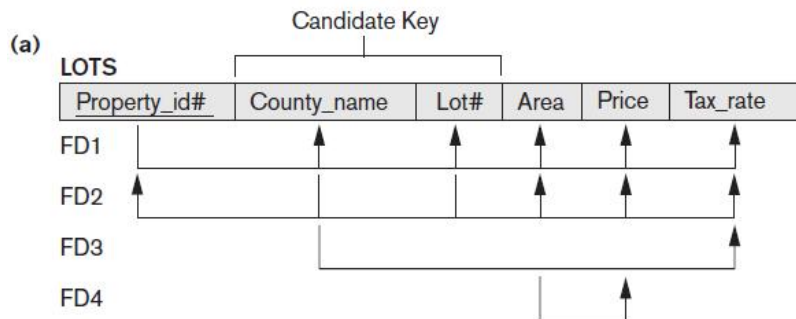
- ดำเนินการให้อยู่ในรูป 2NF ก่อน
- แยก Nonprime Attribute(s) B_1, B_2, B_3, \dots ที่ถูกกำหนดโดย Nonprime Attribute(s) A_1, A_2, A_3, \dots แต่ละตัวออกไปอยู่ใน Relation(s) ใหม่ R_1, R_2, \dots และ
- ให้นำ Nonprime Attribute(s) A_1, A_2, A_3, \dots ที่เป็นตัวกำหนด Nonprime Attribute(s) B_1, B_2, B_3, \dots ไปไว้ใน Relation(s) R_1, R_2, \dots ใหม่ด้วย โดย Nonprime Attribute(s) A_1, A_2, A_3, \dots จะถูกกำหนดให้เป็น PK ใน Relation(s) R_1, R_2, \dots ใหม่ และ
- ให้คง Nonprime Attribute(s) A_1, A_2, A_3, \dots ใน Relation เริ่มต้น R ด้วย เพื่อใช้เป็น FK เพื่อเป็นตัวเชื่อมไปยัง Relation ใหม่ R_1, R_2, \dots

26

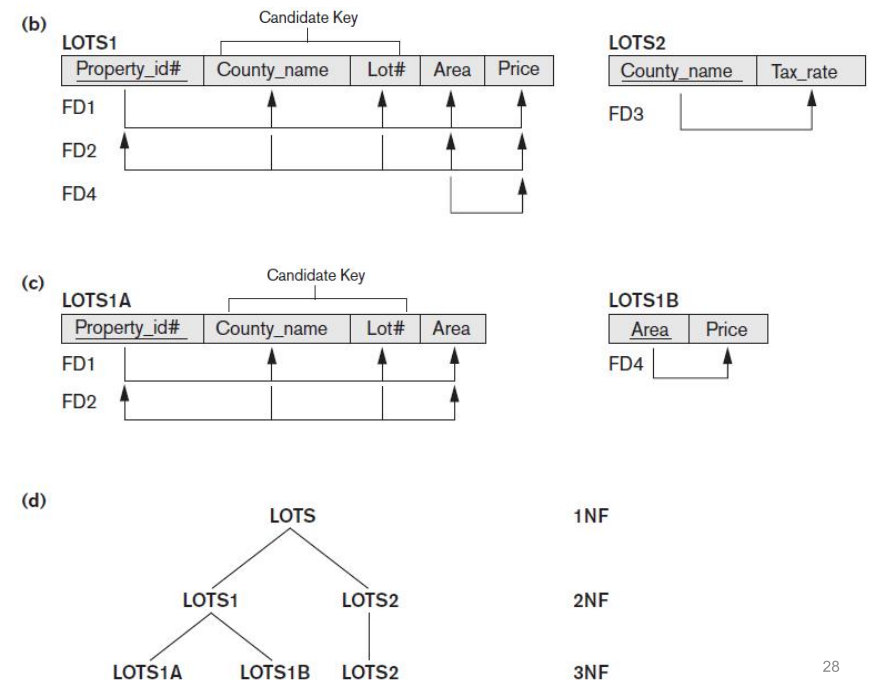
General Definition of Second Normal Form

Figure 15.12

Normalization into 2NF and 3NF. (a) The LOTS relation with its functional dependencies FD1 through FD4. (b) Decomposing into the 2NF relations LOTS1 and LOTS2. (c) Decomposing LOTS1 into the 3NF relations LOTS1A and LOTS1B. (d) Summary of the progressive normalization of LOTS.



27



28

Boyce-Codd Normal Form

Boyce-Codd Normal Form (BCNF)

- Relation R ที่อยู่ใน 3NF ไม่จำเป็นที่จะอยู่ใน BCNF แต่ส่วนใหญ่ของ Relation R ที่อยู่ใน 3NF จะอยู่ใน BCNF ด้วย

ได้ 3NF **ไม่ได้หมายความว่า** ได้ BCNF โดยปริยาย

Relation R จะอยู่ในรูป BCNF ก็ต่อเมื่อ

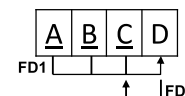
- Relation R อยู่ใน 3NF
- ต้องไม่มี Nonprime Attribute(s) ไปกำหนดค่า Prime Attribute(s)
- ****ซึ่งปกติ Attribute(s) ที่กำหนดค่าอื่น คือ Prime Attribute(s)******

29

BCNF Normalization

วิธีดำเนินการให้ Relation R อยู่ในรูป BCNF

- ดำเนินการให้อยู่ในรูป 3NF ก่อน
- แยก Prime Attribute(s) B_1, B_2, B_3, \dots ที่ถูกกำหนดโดย Nonprime Attribute(s) A_1, A_2, A_3, \dots แต่ละตัวออกไปอยู่ใน Relation(s) ใหม่ R_1, R_2, \dots และ
- ให้นำ Nonprime Attribute(s) A_1, A_2, A_3, \dots ที่เป็นตัวกำหนด Prime Attribute(s) B_1, B_2, B_3, \dots ไปไว้ใน Relation(s) R_1, R_2, \dots ใหม่ด้วย โดย Nonprime Attribute(s) A_1, A_2, A_3, \dots จะถูกกำหนดให้เป็น PK ใน Relation(s) R_1, R_2, \dots ใหม่ และ
- ให้คง Nonprime Attribute(s) A_1, A_2, A_3, \dots ใน Relation เริ่มต้น R ด้วย เพื่อใช้เป็น FK เพื่อเป็นตัวเชื่อมไปยัง Relation ใหม่ R_1, R_2, \dots



ในเมื่อ C และ D ในเชิงความหมายต่างก็เป็น Prime เราควรจะ Decompose ตัวไหน ระหว่าง C กับ D ?

$\boxed{A \ B \ C} \ \boxed{C \ D} \ ?$

$\boxed{A \ B \ D} \ \boxed{D \ C} \ ?$

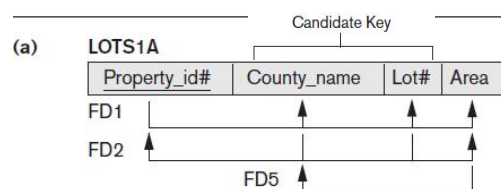


Figure 15.13

Boyce-Codd normal form. (a) BCNF normalization of LOTUS1A with the functional dependency FD2 being lost in the decomposition. (b) A schematic relation with FDs; it is in 3NF, but not in BCNF.

BCNF Normalization



ควรจะคง C หรือ B ไว้ใน Relation เริ่มต้น ?

31

Properties of Relational Decompositions (1)

- $\{R_1, R_2, \dots, R_m\}$ ที่แยกออกจาก R จะมีคุณสมบัติ **Lossless join** ก็ต่อเมื่อ

Join $\{R_1, R_2, \dots, R_m\}$ กลับมา ต้องไม่เกิด Spurious Tuples

$$* (\pi_{R_1}(r), \dots, \pi_{R_m}(r)) = r$$

- Testing Binary Decompositions for Lossless Join Property:**

- $\{R_1, R_2\}$ ที่แยกออกจาก R จะคงคุณสมบัติของ **Lossless Join** หาก $\{R_1, R_2\}$ มีคุณสมบัติ FD อย่างใดอย่างหนึ่งด้านล่าง

- FD $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$, or
- FD $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$.

Lossless join: Join รีเลชันที่แยกออกไป ผลลัพธ์ที่ได้ต้องได้จำนวน Records เหมือนกับรีเลชันตั้งต้น

32

Properties of Relational Decompositions (2)

An example of testing Binary Decompositions for Lossless Join Property

Property

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

Three possible decompositions for relation TEACH

$\{\text{student, instructor}\}$ and $\{\text{student, course}\}$
 $\{\text{course, instructor}\}$ and $\{\text{course, student}\}$
 $\{\text{instructor, course}\}$ and $\{\text{instructor, student}\}$

33

Properties of Relational Decompositions (3)

An example of testing Binary Decompositions for Lossless Join Property (Cont'd.)

Property (Cont'd.)

Decomposition #1

$\{\text{student, instructor}\}$ and $\{\text{student, course}\}$

Student	Instructor	Student	Course
Narayan	Mark	Narayan	Database
Smith	Navathe	Smith	Database
Smith	Ammar	Smith	Operating Systems
Smith	Schulman	Smith	Theory
Wallace	Mark	Wallace	Database
Wallace	Ahamad	Wallace	Operating Systems
Wong	Omiecinski	Wong	Database
Zelaya	Navathe	Zelaya	Database

Join

Results are > 8 Records

Student	Instructor	Course
Narayan	Mark	Database
Smith	Navathe	Database
Smith	Navathe	Operating Systems
Smith	Navathe	Theory
Smith	Ammar	Database
Smith	Ammar	Operating Systems
Smith	Ammar	Theory
Smith	Schulman	Database
Smith	Schulman	Operating Systems
Smith	Schulman	Theory
Wallace	Mark	Database
Wallace	Mark	Operating Systems
Wallace	Ahamad	Database
Wallace	Ahamad	Operating Systems
Wong	Omiecinski	Database
Zelaya	Navathe	Database

Spurious Tuples are generated

34

student \rightarrow instructor, student \rightarrow course

Either

- The FD $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$ is in F^+ , or
- The FD $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$ is in F^+ .

Properties of Relational Decompositions (4)

An example of testing Binary Decompositions for Lossless Join Property (Cont'd.)

Decomposition #2

$\{\text{course, instructor}\}$ and $\{\text{course, student}\}$

Course	Instructor	Course	Student
Database	Mark	Database	Narayan
Database	Navathe	Database	Smith
Operating Systems	Ammar	Operating Systems	Smith
Theory	Schulman	Theory	Smith
Operating Systems	Ahamad	Database	Wallace
Database	Omiecinski	Operating Systems	Wallace
		Database	Wong
		Database	Zelaya

Join

Results are > 8 Records

Course	Instructor	Student
Database	Mark	Narayan
Database	Mark	Smith
Database	Mark	Wallace
Database	Mark	Wong
Database	Mark	Zelaya
Database	Navathe	Narayan
Database	Navathe	Smith
Database	Navathe	Wallace
Database	Navathe	Wong
Database	Navathe	Zelaya
Database	Omiecinski	Narayan
Database	Omiecinski	Smith
Database	Omiecinski	Wallace
Database	Omiecinski	Wong
Database	Omiecinski	Zelaya
...		

Spurious Tuples are generated

35

Properties of Relational Decompositions (5)

An example of testing Binary Decompositions for Lossless Join Property (Cont'd.)

Decomposition #3

$\{\text{instructor, course}\}$ and $\{\text{instructor, student}\}$

Instructor	Course	Instructor	Student
Mark	Database	Mark	Narayan
Navathe	Database	Navathe	Smith
Ammar	Operating Systems	Ammar	Smith
Schulman	Theory	Schulman	Smith
Ahamad	Operating Systems	Mark	Wallace
Omiecinski	Database	Ahamad	Wallace
		Omiecinski	Wong
		Navathe	Zelaya

Join

Results are the same as originals.

Instructor	Course	Student
Mark	Database	Narayan
Mark	Database	Wallace
Navathe	Database	Smith
Navathe	Database	Zelaya
Ammar	Operating Systems	Smith
Schulman	Theory	Smith
Ahamad	Operating Systems	Wallace
Omiecinski	Database	Wong

No Spurious Tuples

$\text{instructor} \rightarrow \text{course}$, instructor \rightarrow student

Either

- The FD $((R_1 \cap R_2) \rightarrow (R_1 - R_2))$ is in F^+ , or
- The FD $((R_1 \cap R_2) \rightarrow (R_2 - R_1))$ is in F^+ .

36

Summary

- Informal guidelines for good design
- Functional dependency
 - Basic tool for analyzing relational schemas
- Normalization:
 - 1NF, 2NF, 3NF, BCNF
- Lossless Join Property of a decomposition